

Bull

AIX 4.3 - Guide de l'utilisateur Système d'exploitation et unités

AIX

Bull

AIX 4.3 - Guide de l'utilisateur Système d'exploitation et unités

AIX

Logiciel

Novembre 1999

**BULL ELECTRONICS ANGERS
CEDOC
34 Rue du Nid de Pie – BP 428
49004 ANGERS CEDEX 01
FRANCE**

86 F2 97HX 02

The following copyright notice protects this book under the Copyright laws of the United States and other countries which prohibit such actions as, but not limited to, copying, distributing, modifying, and making derivative works.

Copyright © Bull S.A. 1992, 1999

Imprimé en France

Vos suggestions sur la forme et le fond de ce manuel seront les bienvenues. Une feuille destinée à recevoir vos remarques se trouve à la fin de ce document.

Pour commander d'autres exemplaires de ce manuel ou d'autres publications techniques Bull, veuillez utiliser le bon de commande également fourni en fin de manuel.

Marques déposées

Toutes les marques déposées sont la propriété de leurs titulaires respectifs.

AIX[®] est une marque déposée d'IBM Corp. et est utilisée sous licence.

UNIX est une marque déposée licenciée exclusivement par X/Open Company Ltd.

An 2000

Le produit présenté dans ce manuel est prêt pour l'an 2000.

La loi du 11 mars 1957, complétée par la loi du 3 juillet 1985, interdit les copies ou reproductions destinées à une utilisation collective. Toute représentation ou reproduction intégrale ou partielle faite par quelque procédé que ce soit, sans consentement de l'auteur ou de ses ayants cause, est illicite et constitue une contrefaçon sanctionnée par les articles 425 et suivants du code pénal.

Ce document est fourni à titre d'information seulement. Il n'engage pas la responsabilité de Bull S.A. en cas de dommage résultant de son application. Des corrections ou modifications du contenu de ce document peuvent intervenir sans préavis ; des mises à jour ultérieures les signaleront éventuellement aux destinataires.

A propos de ce manuel

Ce manuel s'adresse aux utilisateurs novices qui souhaitent approfondir leur connaissance du système d'exploitation AIX : exécution des commandes, gestion des process, gestion des fichiers et des répertoires, impression, etc. En outre, il présente les commandes système gérant les tâches du type protection des fichiers, utilisation des supports de stockage, personnalisation des fichiers d'environnement (**.profile**, **.Xdefaults**, **.mwmrc**) et rédaction de scripts shell. Les utilisateurs DOS y trouveront les procédures permettant d'exploiter les fichiers DOS dans l'environnement AIX.

Les utilisateurs d'un environnement en réseau souhaitant en apprendre davantage sur les commandes de communication du système d'exploitation AIX peuvent consulter le *AIX 4.3 Guide de l'utilisateur : communications et réseaux*, 86 F2 98HX.

Remarque : La version en ligne de cet ouvrage, conçue pour être visualisée à l'aide d'un navigateur web compatible HTML version 3.2 est disponible sur le "Hypertext Library for AIX 4.3" CD-ROM.

A qui s'adresse ce manuel ?

Ce manuel s'adresse à tous les utilisateurs du système.

Mode d'emploi du manuel

Vous trouverez ci-après une présentation du contenu du manuel, suivie des conventions typographiques adoptées.

Contenu

Ce manuel comporte les chapitres suivants.

- Chapitre 1 – Noms de connexion, ID système et mots de passe : commandes d'identification des utilisateurs et des systèmes, et procédures de connexion et de déconnexion du système.
- Chapitre 2 – Système et environnement utilisateur : commandes d'identification de l'environnement système et utilisateur.
- Chapitre 3 – Commandes et process : procédures de base relatives au système, aux commandes et aux process.
- Chapitre 4 – Réacheminement des entrées/sorties : procédures de gestion des entrées/sorties.
- Chapitre 5 – Systèmes de fichiers et répertoires : commandes relatives aux systèmes de fichiers et aux répertoires.
- Chapitre 6 – Fichiers : commandes de traitement des fichiers.
- Chapitre 7 – Imprimantes, travaux d'impression et files d'attente : procédures d'impression des fichiers.
- Chapitre 8 – Fichiers de sauvegarde et supports de stockage : commandes de sauvegarde des fichiers et mode d'emploi des disquettes et des bandes.
- Chapitre 9 – Sécurité du système et des fichiers : commandes de protection des fichiers.
- Chapitre 10 – Personnalisation de l'environnement utilisateur : commandes de personnalisation et fichiers d'initialisation.
- Chapitre 11 – Shells : shells Korn (également appelé shell POSIX), Bourne et C.

- Chapitre 12 – Outils et utilitaires : autres commandes utiles.
- Chapitre 13 – Service de recherche documentaire : présente le service et explique comment procéder à la recherche de documents HTML en ligne.
- Annexe A – Accès aux informations avec InfoExplorer : décrit les différents types de documentation disponibles dans la *bibliothèque de base de données en hypertexte* et explique comment y accéder.

Conventions typographiques

Les conventions typographiques suivantes sont utilisées dans ce manuel :

Gras	Commandes, mots-clés, fichiers, répertoires et autres éléments dont le nom est prédéfini par le système.
<i>Italique</i>	Paramètres dont le nom ou la valeur est fourni par l'utilisateur.
Espacement fixe	Exemples (valeurs spécifiques, texte affiché, code programme), messages système ou données entrées par l'utilisateur.

ISO 9000

Ce produit répond aux normes qualité ISO 9000.

Support AIX pour les spécifications X/Open UNIX95 (Spec 1170)

A partir d'AIX Version 4.2.2, le système d'exploitation est conçu pour prendre la charge la spécification X/Open UNIX95 relative à la portabilité des systèmes d'exploitation basés sur UNIX95. De nombreuses interfaces ont été ajoutées, d'autres ont été améliorées pour répondre à ces spécifications. A partir de la version 4.2, AIX est encore plus ouvert et portable pour les applications.

En outre, la compatibilité avec les versions antérieures d'AIX est maintenue grâce à la création d'une nouvelle variable d'environnement, qui sert à configurer l'environnement sur la base du système, de l'utilisateur ou du process.

Pour déterminer la meilleure façon de développer une application portable UNIX95, reportez-vous à la spécification X/Open UNIX95, disponible sur CD-ROM lorsque vous commandez la copie papier du document *AIX Commands Reference*, CEDOC 86 A2 38JX à 86 A2 43JX, ou lorsque vous commandez *Go Solo: How to Implement and Go Solo with the Single Unix Specification*, qui inclut cette spécification X/Open UNIX95 sur CD-ROM.

Bibliographie

Les manuels suivants complètent la documentation AIX.

CEDOC	Titre
86 F2 71WE	<i>AIX - Bibliographie</i>
86 F2 75HX	<i>AIX 4.3 Initiation</i>
86 F2 98HX	<i>AIX 4.3 Guide de l'utilisateur : communications et réseaux</i>
86 F2 99HX	<i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i>
86 A2 37JX	<i>AIX Guide to Printers and Printing</i>
86 A2 38JX à 86 A2 43JX	<i>AIX Commands Reference</i>
86 A2 79AP	<i>AIX Files Reference</i>
86 A2 84AT	<i>Common Desktop Environment 1.0: User's Guide</i>

Documentation non Bull

- Quercia, Valerie, and Tim O'Reilly. *The Definitive Guides to the X Window System: Volume 3, X Window System User's Guide, Motif Edition*. O'Reilly & Associates, 1993.
- Garfinkel, Simson, and Gene Spafford. *Practical UNIX and Internet Security*. Sebastopol, CA: O'Reilly & Associates, 1996.
- Todino, Grace, and John Strang. *Learning the UNIX Operating System*. Sebastopol, CA: O'Reilly & Associates, 1993.
- Gilly, Daniel. *UNIX in a Nutshell: A Desktop Quick Reference for System V and Solaris 2.0*. Sebastopol, CA: O'Reilly & Associates, 1992.
- Cutler, Ellie, Daniel Gilly, and Tim O'Reilly, ed. *The Definitive Guides to the X Window System: Desktop Quick Reference, The X Window System in a Nutshell for Version 11 Release 4 and Release 5 of the X Window System*. O'Reilly & Associates, 1992.
- Peek, Jerry, Mike Loukides, and Tim O'Reilly. *UNIX Power Tools*. O'Reilly & Associates, 1993.
- Open Software Foundation. *Introduction to OSF*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
- Rosenblatt, Bill. *Learning the Korn Shell*. Sebastopol, CA: O'Reilly & Associates, 1993.
- Hunter, Bruce H., and Karen Bradford Hunter. *UNIX Systems Advanced Administration and Management Handbook*. New York: Macmillan, 1991.
- Rosenberg, Barry. *KornShell Programming Tutorial*. Reading, MA: Addison-Wesley Publishing Company, 1991.
- X/Open Company, Ltd. and Prentice Hall, Inc. *Go Solo: How to Implement and Go Solo with the Single Unix Specification*. United Kingdom: X/Open Company, Ltd., 1995.

Commande de manuels

Pour toute commande de manuel, adressez-vous à votre représentant ou à votre distributeur attitré. Pour commander d'autres exemplaires de ce manuel, précisez la référence 86 F2 97HX. Si vous disposez de la brochure *AIX - Bibliographie*, consultez-la pour savoir comment vous procurer les publications qui vous intéressent.

Table des matières

A propos de ce manuel	iii
A qui s'adresse ce manuel ?	iii
Mode d'emploi du manuel	iii
ISO 9000	iv
Support AIX pour les spécifications X/Open UNIX95 (Spec 1170)	iv
Bibliographie	iv
Documentation non Bull	v
Commande de manuels	v
Chapitre 1. Noms de connexion, ID système et mots de passe	1-1
Voir aussi	1-2
Connexion et déconnexion : généralités	1-3
Connexion au système d'exploitation	1-3
Connexions multiples (commande login)	1-4
Passage à un autre nom d'utilisateur (commande su)	1-4
Suppression des messages de connexion	1-4
Déconnexion du système d'exploitation (commandes exit et logout)	1-5
Arrêt du système d'exploitation (commande shutdown)	1-5
Voir aussi	1-5
Système et environnement utilisateur	1-6
Affichage du nom de connexion (commandes whoami et logname)	1-6
Affichage du nom du système d'exploitation (commande uname)	1-7
Affichage du nom de votre système (commande uname)	1-7
Affichage des utilisateurs connectés (commande who)	1-7
Affichage de l'ID d'un utilisateur (commande id)	1-7
Mots de passe	1-8
Mot de passe : règles	1-8
Modification du mot de passe (commande passwd)	1-9
Définition d'un mot de passe Null (commande passwd)	1-10
Récapitulatif des commandes relatives à la connexion	1-11
Connexion et déconnexion	1-11
Identification utilisateur et système	1-11
Mot de passe	1-11
Chapitre 2. Système et environnement utilisateur	2-1
Liste des unités du système (commande lscfg)	2-2
Affichage du nom de votre console (commande lscons)	2-3
Affichage du nom de votre terminal (commande tty)	2-3
Liste des écrans disponibles (commande lsdisp)	2-4
Liste des polices disponibles (commande lsfont)	2-4
Liste des mappes clavier (commande lskbd)	2-4
Liste des logiciels (commande lspp)	2-5
Liste des affectations de touches du terminal (commande stty)	2-6
Liste de toutes les variables d'environnement (commande env)	2-7
Affichage de la valeur d'une variable d'environnement (commande printenv)	2-8
Langues particulières (commande aixterm)	2-8
Récapitulatif des commandes relatives à l'environnement utilisateur et aux informations système	2-9

Chapitre 3. Commandes et process	3-1
Commandes : généralités	3-2
Syntaxe des commandes	3-2
Diagrammes de syntaxe	3-4
Instructions d'usage	3-6
Utilisation de Web-based System Manager	3-6
Commande smit	3-7
Localisation d'une commande ou d'un programme (commande whereis)	3-7
Affichage d'informations sur une commande (commande man)	3-7
Affichage de la fonction d'une commande (commande whatis)	3-8
Affichage des commandes précédentes (commande shell history)	3-8
Répétition de commandes via la commande shell history	3-9
Substitution de chaînes via la commande shell history	3-9
Modification de l'historique des commandes	3-10
Création d'un alias (commande shell alias)	3-10
Commandes de formatage de texte	3-11
Process : généralités	3-13
Process d'avant et d'arrière-plan	3-13
Démons	3-13
Process zombie	3-14
Lancement d'un process	3-14
Contrôle des process (commande ps)	3-14
Définition de la priorité initiale d'un process (commande nice)	3-16
Changement de la priorité d'un process en cours (commande renice)	3-17
Interruption d'un process d'avant-plan	3-17
Arrêt d'un process d'avant-plan	3-18
Relance d'un process	3-18
Planification d'un process en différé (commande at)	3-18
Liste des process planifiés	3-20
Déprogrammation d'un process	3-20
Suppression d'un process d'arrière-plan (commande kill)	3-21
Récapitulatif des commandes relatives aux commandes et aux process	3-22
Commandes	3-22
Process	3-22
Chapitre 4. Réacheminement des entrées/sorties	4-1
Entrées, sorties et erreurs standard	4-2
Réacheminement des sorties standard	4-2
Réacheminement des sorties vers un fichier	4-2
Réacheminement des sorties avec ajout à un fichier	4-3
Création d'un fichier texte avec réacheminement à partir du clavier	4-3
Concaténation de fichiers texte	4-3
Réacheminement des entrées standard	4-3
Élimination des sorties via le fichier /dev/null	4-4
Réacheminement des erreurs standard et autres sorties	4-4
Documents entrée en ligne (Here)	4-5
Pipes et filtres	4-5
Affichage de la sortie d'un programme avec copie dans un fichier (commande tee)	4-6
Effacement de l'écran (commande clear)	4-6
Envoi d'un message vers la sortie standard (commande echo)	4-6
Ajout d'une ligne de texte à un fichier (commande echo)	4-7
Copie de l'écran dans un fichier (commandes capture et script)	4-7
Affichage de texte en gros caractères (commande banner)	4-8
Récapitulatif des commandes relatives au réacheminement des entrées/sorties ..	4-9

Chapitre 5. Systèmes de fichiers et répertoires	5-1
Systèmes de fichiers	5-2
Types de systèmes de fichiers	5-2
Structure des systèmes de fichiers	5-2
Espace disponible sur un système de fichiers (commande df)	5-4
Répertoires : généralités	5-5
Types de répertoires	5-5
Organisation des répertoires	5-6
Conventions d'appellation des répertoires	5-6
Chemins d'accès aux répertoires	5-6
Abréviations propres aux répertoires	5-7
Procédures de gestion des répertoires	5-8
Création d'un répertoire (commande mkdir)	5-8
Déplacement ou changement de nom d'un répertoire (commande mvdir)	5-8
Affichage du répertoire courant (commande pwd)	5-9
Passage à un autre répertoire (commande cd)	5-9
Copie d'un répertoire (commande cp)	5-9
Affichage du contenu d'un répertoire (commandes li et ls)	5-10
Suppression ou retrait d'un répertoire (commande rmdir)	5-12
Comparaison entre répertoires (commande dircmp)	5-13
Récapitulatif des commandes relatives aux systèmes de fichiers et aux répertoires	5-14
Systèmes de fichiers	5-14
Abréviations des répertoires	5-14
Procédures de gestion des répertoires	5-14
 Chapitre 6. Fichiers	 6-1
Types de fichiers	6-2
Fichiers standard	6-2
Fichiers répertoire	6-2
Fichiers spéciaux	6-2
Conventions d'appellation des fichiers	6-3
Chemins d'accès aux fichiers	6-3
Caractères joker et métacaractères	6-3
Recherche générique et expressions régulières	6-4
Procédures de gestion de fichier	6-5
Suppression de fichiers (commandes del et rm)	6-5
Déplacement et changement de nom d'un fichier (commande mv)	6-6
Copie de fichiers (commande cp)	6-7
Recherche de fichiers (commande find)	6-8
Affichage du type d'un fichier (commande file)	6-9
Affichage du contenu d'un fichier (commandes pg, more, page et cat)	6-9
Recherche de chaînes dans un fichier texte (commande grep)	6-10
Tri des fichiers texte (commande sort)	6-10
Comparaison de fichiers (commande diff)	6-11
Décompte des mots, des lignes et des octets d'un fichier (commande wc)	6-12
Affichage des premières lignes d'un fichier (commande head)	6-12
Affichage des dernières lignes d'un fichier (commande tail)	6-12
Coupe de sections de fichiers texte (commande cut)	6-13
Collage de sections de fichiers texte (commande paste)	6-13
Numérotation des lignes d'un fichier texte (commande nl)	6-14
Suppression de colonnes dans un fichier texte (commande colrm)	6-15
Liaison entre fichiers et répertoires	6-16
Types de liens	6-16
Liaison de fichiers (commande ln)	6-17

Suppression de fichiers liés	6-18
Fichiers DOS	6-19
Copie de fichiers DOS dans des fichiers AIX	6-19
Copie de fichiers AIX dans des fichiers DOS	6-19
Suppression de fichiers DOS	6-20
Contenu d'un répertoire DOS	6-20
Récapitulatif des commandes relatives aux fichiers	6-21
Procédures de gestion de fichier	6-21
Liaison de fichiers et de répertoires	6-21
Fichiers DOS	6-21
Chapitre 7. Imprimantes, travaux d'impression et files d'attente	7-1
Terminologie	7-2
Travail d'impression	7-2
File d'attente	7-2
Unité de file d'attente	7-2
qdaemon	7-2
Spouleur d'impression	7-2
Imprimante réelle	7-3
Imprimante virtuelle	7-3
Imprimantes locales et distantes	7-3
Programme expéditeur de l'imprimante	7-3
Lancement d'un travail d'impression (commande qprt)	7-4
Prérequis	7-4
Commande qprt	7-4
Commande smit	7-6
Annulation d'un travail d'impression (commande qcan)	7-7
Prérequis	7-7
Web-based System Manager Raccourci	7-7
Commande qcan	7-7
Commande smit	7-7
Vérification de l'état d'un travail (commande qchk)	7-8
Prérequis	7-8
Raccourci Web-based System Manager	7-8
Commande qchk	7-8
Commande smit	7-8
Etat de l'imprimante	7-9
Définition de la priorité d'un travail (commande qpri)	7-10
Prérequis	7-10
Raccourci Web-based System Manager	7-10
Commande qpri	7-10
Commande smit	7-10
Blocage et libération d'un travail (commande qhld)	7-11
Prérequis	7-11
Web-based System Manager Raccourci	7-11
Commande qhld	7-11
Commande smit	7-11
Déplacement d'un travail vers une autre file d'attente (commande qmov)	7-12
Prérequis	7-12
Raccourci Web-based System Manager	7-12
Commande qmov	7-12
Commande smit	7-12
Formatage des fichiers à imprimer (commande pr)	7-13
Impression de fichiers ASCII sur une imprimante PostScript	7-14
Prérequis	7-14

Automatisation de la conversion ASCII–PostScript	7-16
Annulation de la détermination automatique du type de fichier	7-16
Récapitulatif des commandes relatives à l'impression	7-17
Chapitre 8. Fichiers de sauvegarde et supports de stockage	8-1
Stratégie de sauvegarde	8-2
Supports de stockage	8-3
Formatage de disquettes (commande format ou fdformat)	8-5
Vérification de l'intégrité du système de fichiers (commande fsck)	8-6
Copie de ou sur des disquettes (commande fcopy)	8-6
Copie de fichiers sur bande ou disque (commande cpio –o)	8-7
Copie de fichiers à partir de bande ou de disque (commande cpio–i)	8-7
Copie de et sur des bandes (commande tcopy)	8-9
Vérification de l'intégrité d'une bande (commande tapechk)	8-9
Compression de fichiers (commandes compress et pack)	8-9
Commande compress	8-10
Commande pack	8-10
Décompression de fichiers (commandes uncompress et unpack)	8-11
Commande uncompress	8-11
Commande unpack	8-11
Sauvegarde de fichiers (commande backup)	8-12
Commande backup	8-12
Commande smit	8-13
Restauration de fichiers (commande restore)	8-14
Commande restore	8-14
Commande smit	8-15
Archivage de fichiers (commande tar)	8-16
Récapitulatif des commandes relatives à la sauvegarde	8-17
Voir aussi	8-18
Chapitre 9. Sécurité du système et des fichiers	9-1
Menaces sur la sécurité	9-2
Sécurité de base	9-2
Propriété des fichiers et groupes d'utilisateurs	9-4
Changement du propriétaire d'un fichier ou d'un répertoire (commande chown)	9-4
Modes d'accès aux fichiers et aux répertoires	9-4
Affichage d'informations sur le groupe (commande lsgroup)	9-6
Modification des droits d'accès aux fichiers et aux répertoires (commande chmod)	9-8
Listes de contrôle des accès (ACL)	9-9
Droits d'accès de base	9-9
Droits d'accès étendus	9-10
Exemple de liste de contrôle des accès	9-10
Autorisations d'accès	9-11
Affichage des informations de contrôle des accès (commande aclget)	9-12
Définition du contrôle des accès (commande aclput)	9-12
Modification des informations de contrôle des accès (commande acledit)	9-13
Verrouillage du terminal (commande lock ou xlock)	9-14
Contrôle antivirus (commande virscan)	9-15
Format des fichiers signature	9-15
Ajout de signatures de virus	9-16
Récapitulatif des commandes relatives à la sécurité	9-17

Chapitre 10. Personnalisation de l'environnement utilisateur	10-1
Voir aussi	10-1
Support AIX pour la X/Open UNIX95 Specification	10-2
Fichiers de lancement du système : généralités	10-3
Fichier /etc/profile	10-3
Fichier /etc/environment	10-4
Fichier .profile	10-4
Fichier .env	10-5
Fichiers de lancement AIXwindows : généralités	10-6
Fichier .xinitrc	10-6
Fichier .Xdefaults	10-7
Fichier .mwmrc	10-8
Procédures de personnalisation	10-11
Exportation de variables shell (commande shell export)	10-11
Changement de la police d'affichage (commande chfont)	10-11
Changement de l'affectation des touches de contrôle (commande stty)	10-12
Changement de l'invite système	10-12
Personnalisation d'InfoExplorer (programme d'aide)	10-13
Modification des paramètres par défaut d'InfoExplorer (Window)	10-13
Choix des préférences dans InfoExplorer (Window)	10-14
Récapitulatif des commandes de personnalisation	10-16
Fichiers de lancement du système	10-16
AIXwindows Fichiers de lancement	10-16
Procédures de personnalisation	10-16
Chapitre 11. Shells	11-1
Caractéristiques des shells	11-3
Shells disponibles	11-4
Terminologie	11-5
Création et exécution d'un script shell	11-7
Spécification d'un shell pour un fichier script	11-8
Voir aussi	11-8
Shell Korn ou POSIX	11-9
Environnement shell Korn	11-10
Lancement du shell	11-10
Historique des commandes (shell Korn ou POSIX)	11-11
Déclaration de caractères	11-12
Mots réservés	11-14
Alias de commandes (shell Korn ou POSIX)	11-15
Alias de trace	11-16
Substitution de tilde	11-16
Substitution de paramètres (shell Korn ou POSIX)	11-17
Paramètres (shell Korn)	11-17
Substitution de paramètres	11-18
Paramètres spéciaux prédéfinis	11-19
Variables définies par le shell Korn ou POSIX	11-20
Variables utilisées par le shell Korn ou POSIX	11-20
Substitution de commandes (shell Korn ou POSIX)	11-23
Evaluation arithmétique (shell Korn ou POSIX)	11-24
Séparation de zones	11-25
Substitution de noms de fichiers (shell Korn ou POSIX)	11-26
Suppression des caractères de déclaration	11-27
Réacheminement des entrées/sorties (shell Korn ou POSIX)	11-28
Fonction coprocess	11-29
Etats de sortie	11-31

Commandes du shell Korn ou POSIX	11-32
Commandes composées du shell Korn	11-33
Fonctions	11-35
Liste des commandes intégrées du shell Korn ou POSIX	11-37
Description des commandes spéciales	11-37
Description des commandes intégrées standard	11-44
Expressions conditionnelles	11-49
Contrôle des travaux (shell Korn ou POSIX)	11-51
Traitement des signaux	11-51
Edition en ligne (shell Korn ou POSIX)	11-53
Mode d'édition emacs	11-53
Mode d'édition vi	11-57
Liste des commandes intégrées du shell Korn ou POSIX	11-61
Commandes intégrées spéciales	11-61
Commandes intégrées standard	11-61
Liste des commandes intégrées du shell Bourne	11-63
Liste des commandes intégrées du shell C	11-64
Voir aussi Shell Korn	11-67
Voir aussi Shell Bourne	11-68
Voir aussi Shell C	11-69
Shell Bourne	11-70
Environnement du shell Bourne	11-70
Shell restreint	11-72
Commandes du shell Bourne	11-73
Déclaration de caractères	11-74
Traitement des signaux	11-74
Commandes composées (shell Bourne)	11-75
Mots réservés	11-75
Commandes intégrées (shell Bourne)	11-76
Description des commandes spéciales	11-77
Substitution de commandes (shell Bourne)	11-82
Substitution de variables et de noms de fichiers (shell Bourne)	11-83
Substitution de variables (shell Bourne)	11-83
Variables définies par l'utilisateur	11-83
Substitution conditionnelle	11-88
Paramètres positionnels	11-89
Substitution de noms de fichiers (shell Bourne)	11-89
Classes de caractères	11-90
Réacheminement des entrées/sorties (shell Bourne)	11-91
Shell C	11-92
Règles d'utilisation	11-93
Traitement des signaux	11-93
Commandes du shell C	11-94
Commandes intégrées (shell C)	11-94
Description des commandes	11-95
Expressions et opérateurs du shell C	11-102
Substitution de commandes (shell C)	11-103
Exécution des commandes Shell C non intégrées	11-104
Substitution d'historique (shell C)	11-105
Liste d'historique	11-105
Spécification d'événement	11-106
Guillemets et apostrophes	11-107
Substitution d'alias (shell C)	11-108
Substitution de variables et de noms de fichiers (shell C)	11-109
Substitution de variable (shell C)	11-109

Substitution de nom de fichier (shell C)	11-111
Développement de nom de fichier	11-111
Abréviation de nom de fichier	11-111
Classes de caractères	11-112
Variables d'environnement (shell C)	11-113
Réacheminement des entrées/sorties (shell C)	11-116
Flux de contrôle	11-117
Contrôle des travaux (shell C)	11-118
Chapitre 12. Outils et utilitaires	12-1
Recherche d'une commande par mot-clé (commande apropos)	12-1
Affichage d'un calendrier (commande cal)	12-2
Affichage des messages de l'agenda (commande calendar)	12-2
Affichage de l'aide (commande help)	12-3
Appel du didacticiel (commande learn)	12-3
Heure de sortie (commande leave)	12-4
Décomposition d'un nombre en facteurs premiers (commande factor)	12-4
Conversion des unités de mesure (commande units)	12-5
Envoi de messages à un autre utilisateur (commande write)	12-7
Récapitulatif des commandes	12-9
Chapitre 13. Service de bibliothèque documentaire	13-1
Utilisation de la documentation AIX en ligne	13-1
Annexe A. Accès aux informations avec InfoExplorer	A-1
A l'aide de l'interface InfoExplorer (ASCII)	A-1
A l'aide des écrans InfoExplorer	A-1
A l'aide des menus	A-2
Obtention de l'aide	A-2
Initiation à InfoExplorer (ASCII)	A-3
Lancement d'InfoExplorer (ASCII)	A-3
Reconnaissance des types d'écran (InfoExplorer ASCII)	A-3
Sélection d'un lien hypertexte (InfoExplorer ASCII)	A-5
Défilement des informations (InfoExplorer ASCII)	A-5
Sélection d'une option de menu (InfoExplorer ASCII)	A-6
Retour à un emplacement précédent (InfoExplorer ASCII)	A-6
Recherche d'informations (InfoExplorer ASCII)	A-7
Impression d'informations (InfoExplorer ASCII)	A-10
Accès à l'aide (InfoExplorer ASCII)	A-11
Touches et combinaisons de touches spéciales (InfoExplorer ASCII)	A-12
Lancement d'une bibliothèque InfoExplorer secondaire (InfoExplorer ASCII) ..	A-14
Arrêt d'InfoExplorer (ASCII)	A-14
Personnalisation d'InfoExplorer (ASCII)	A-15
Modification des paramètres par défaut d'InfoExplorer (ASCII)	A-15
Choix des préférences dans InfoExplorer (ASCII)	A-16
Ressources X disponibles sous InfoExplorer	A-18
Index	X-1

Chapitre 1. Noms de connexion, ID système et mots de passe

Le système d'exploitation ne peut vous fournir un environnement correct que s'il a le moyen de vous identifier. C'est pourquoi vous êtes invité, pour vous connecter, à préciser votre *nom de connexion* (également appelé ID utilisateur ou nom utilisateur) et un *mot de passe*. Les mots de passe constituent une première sécurité : personne ne peut accéder à votre système sans connaître à la fois votre nom de connexion et votre mot de passe.

Si le système est multi-utilisateur, chaque utilisateur autorisé se voit attribuer un compte, un mot de passe et un nom de connexion. Vous pouvez les utiliser simultanément dans des shells distincts, ou successivement dans un même shell. Il s'agit de la *comptabilité système*. Chaque utilisateur est également doté d'un espace de stockage sur le système, appelé *système de fichiers*. Lorsque vous vous connectez, ce système de fichiers semble ne contenir que vos propres fichiers, même si des milliers d'autres fichiers se trouvent sur le système.

Vous pouvez, sur un même système, disposer de plusieurs noms de connexion. Pour passer de l'un à l'autre, il est inutile de vous déconnecter. Vous pouvez les utiliser simultanément dans des shells distincts ou successivement dans un même shell. En outre, si votre système fait partie d'un réseau connecté à d'autres systèmes, vous avez accès à tous les systèmes sur lesquels un nom de connexion vous a été attribué. On parle alors de *téléconnexion*.

A la fin de votre travail, déconnectez-vous : vous serez ainsi assuré que vos fichiers et vos données sont en sécurité.

Ce chapitre traite des points suivants :

- Connexion et déconnexion : généralités, page 1-3
 - Connexion au système d'exploitation
 - Connexions multiples (commande **login**)
 - Passage à un autre nom d'utilisateur (commande **su**)
 - Suppression des messages de connexion
 - Déconnexion du système d'exploitation
 - Arrêt du système d'exploitation (commande **shutdown**)
- Système et environnement utilisateur, page 1-6.
 - Affichage du nom de connexion (commandes **whoami** et **logname**)
 - Affichage du nom du système d'exploitation (commande **name**)
 - Affichage du nom de votre système (commande **uname**)
 - Affichage des utilisateurs connectés (commande **who**)
 - Affichage de l'ID d'un utilisateur (commande **id**)
- Mots de passe, page 1-8.
 - Modification du mot de passe (commande **passwd**)
 - Définition d'un mot de passe Null (commande **passwd**)
- Récapitulatif des commandes relatives à la connexion, page 1-11.

Voir aussi

Commandes et process

Sécurité du système et des fichiers

Système et environnement utilisateur

Personnalisation de l'environnement utilisateur

Shells

Shell Korn ou POSIX

Shell Bourne

Shell C

Connexion et déconnexion : généralités

Pour accéder au système d'exploitation, votre système doit être lancé et vous devez y être connecté. Lorsque vous ouvrez une session, vous êtes invité à décliner votre identité : le système peut alors configurer votre environnement.

Pour en savoir plus, reportez-vous à "Démarrage du système" dans le manuel *AIX 4.3 Initiation*.

Cette section traite des points suivants :

- Connexion au système d'exploitation, page 1-3.
- Connexions multiples (commande **login**), page 1-4.
- Passage à un autre nom utilisateur (commande **su**), page 1-4.
- Suppression des messages de connexion, page 1-4.
- Déconnexion du système d'exploitation (commandes **exit** et **logout**), page 1-5.
- Arrêt du système d'exploitation (commande **shutdown**), page 1-5.

Connexion au système d'exploitation

Pour travailler sur le système, il vous faut ouvrir une session, c'est-à-dire, après mise sous tension du système, vous y connecter.

Le système peut être configuré pour que vous ne puissiez y accéder qu'à certaines heures de la journée ou certains jours de la semaine : l'accès vous sera refusé si vous tentez une connexion en dehors des plages autorisées. Renseignez-vous auprès de votre administrateur système.

Vous vous connectez à l'affichage de l'invite de connexion. La connexion établie, vous êtes automatiquement placé dans votre répertoire personnel (également appelé *répertoire de connexion*).

Si le système n'est pas sous tension

1. Mettez sous tension toutes les unités raccordées.
2. Amenez le sélecteur de mode sur normal.
3. Mettez l'unité centrale sous tension (interrupteur sur **I**).
4. Consultez l'afficheur : si les autotests ont abouti, il doit être vide.

Si une erreur s'est produite, un code (de 3 chiffres) s'affiche et le système s'arrête. Prévenez votre administrateur système ou consultez le manuel *AIX Version 4.3 - Guide de résolution des incidents*.

Les autotests terminés, l'invite de connexion s'affiche :

```
login:
```

Si elle ne s'affiche pas, prévenez votre administrateur système ou consultez la section "Identification des incidents" du manuel *AIX Version 4.3 - Guide de résolution des incidents*

Si le système est sous tension

1. A l'invite `login:`, tapez votre nom de connexion :

```
login: NomConnexion
```

Par exemple, si votre nom utilisateur est `denise` :

```
login: denise
```

2. Si l'invite `password:` s'affiche, tapez votre mot de passe. (Ce dernier ne s'affiche pas.)

```
password: [mot de passe]
```

Si l'invite password ne s'affiche pas, c'est que vous n'avez pas de mot de passe défini. Vous pouvez commencer à travailler.

Une fois la connexion établie, le système appelle, selon la configuration définie, soit une interface ligne de commande (shell), soit une interface graphique (par exemple, AIXwindows ou CDE Desktop).

Connexions multiples (commande login)

Vous pouvez ouvrir simultanément plusieurs sessions. Pour cela, spécifiez le même nom ou des noms de connexion différents. Ceci vous permet, si vous travaillez sur plusieurs projets, d'affecter les ressources consommées à des comptes distincts.

Remarque : A chaque système est associé un nombre maximal de noms de connexion pouvant être simultanément actifs. Ce nombre dépend de votre licence et varie selon les installations.

Par exemple, si votre deuxième nom de connexion est denise2, tapez, à l'invite :

```
login denise2
```

Si l'invite password: s'affiche, tapez votre mot de passe. (Ce dernier ne s'affiche pas.)

Vous disposez à présent de deux sessions.

Reportez-vous à la commande **login** dans le manuel *AIX Commands Reference*.

Passage à un autre nom d'utilisateur (commande su)

La commande *su* (switch user) permet de modifier l'ID utilisateur associé à une session, sous réserve de connaître le nom utilisateur initial.

Si, par exemple, vous voulez devenir l'utilisateur joyce, tapez, à l'invite :

```
su joyce
```

Si l'invite password: s'affiche, tapez le mot de passe associé à joyce. Si vous ne le connaissez pas, l'accès vous est refusé.

Votre ID utilisateur est désormais joyce.

Pour le vérifier, entrez la commande **id**. Pour en savoir plus, reportez-vous à "Affichage de l'ID d'un utilisateur", page 1-7.

Reportez-vous à la commande **su** dans le manuel *AIX Commands Reference*.

Suppression des messages de connexion

A l'ouverture d'une session, la commande **login** affiche la date du jour, la date et l'heure de la dernière tentative de connexion (aboutie ou non), ainsi que le nombre total de connexions inabouties, depuis la dernière mise à jour des données d'authentification (mot de passe, en général). Ce message est supprimé si votre répertoire personnel contient un fichier **.hushlogin**.

A l'invite, dans votre répertoire utilisateur, tapez :

```
touch .hushlogin
```

La commande **touch** crée, s'il n'existe pas, le fichier **.hushlogin**.

A l'ouverture de session suivante, les messages de connexion seront éliminés. Vous pouvez aussi ne conserver que l'affichage de la date du jour à l'exclusion des autres messages.

Reportez-vous à la commande **touch** dans le manuel *AIX Commands Reference*.

Déconnexion du système d'exploitation (commandes exit et logout)

A l'invite :

Appuyez sur la combinaison de touches de contrôle fin de fichier (Ctrl-D).

OU

Tapez `exit` et appuyez sur Entrée.

OU

Tapez `logout` et appuyez sur Entrée.

Une fois la session fermée, le système affiche l'invite `login:`.

Reportez-vous à la commande **logout** dans le manuel *AIX Commands Reference*.

Avertissement : N'arrêtez pas le système. En effet, vous interrompriez tous les process en cours. Si d'autres utilisateurs sont connectés ou que des travaux sont traités en arrière-plan, des données peuvent être perdues. Exécutez la procédure d'arrêt appropriée avant d'arrêter le système.

Arrêt du système d'exploitation (commande shutdown)

Si vous détenez les droits root, vous avez accès à la commande **shutdown** pour arrêter le système : ne mettez pas le système hors tension avant d'avoir lancé cette commande, si vous ne voulez pas risquer de perdre des données. Si vous n'avez pas accès à la commande **shutdown**, déconnectez-vous simplement et laissez le système en marche. Pour en savoir plus, reportez-vous à "Fermeture du système" dans le manuel *AIX 4.3 Initiation*.

A l'invite, entrez :

```
shutdown
```

La commande **shutdown** exécutée, le système s'arrête avec le message :

```
....Shutdown completed....
```

Reportez-vous à la commande **shutdown** dans le manuel *AIX Commands Reference*.

Voir aussi

Commandes et process

Sécurité du système et des fichiers

Système et environnement utilisateur

Personnalisation de l'environnement utilisateur

Personnalisation des shells

Shell Korn ou POSIX

Shell Bourne

Shell C

Systeme et environnement utilisateur

Cette section traite des commandes permettant d'afficher des informations identifiant les utilisateurs du système et le système utilisé.

- Affichage du nom de connexion (commandes **whoami** et **logname**), page 1-6.
- Affichage du nom du système d'exploitation (commande **uname**), page 1-7.
- Affichage du nom de votre système (commande **uname**), page 1-7.
- Affichage des utilisateurs connectés (commande **who**), page 1-7.
- Affichage de l'ID d'un utilisateur (commande **id**), page 1-7.

Affichage du nom de connexion (commandes **whoami** et **logname**)

Si vous ouvrez plusieurs sessions, il n'est pas toujours évident de se souvenir des noms de connexion utilisés, notamment pour la session active.

Commande **whoami**

Pour afficher le nom de connexion utilisé, entrez, à l'invite :

```
whoami
```

Le système affiche un écran semblable à celui-ci :

```
denise
```

Ici, le nom de connexion est `denise`.

Reportez-vous à la commande **whoami** dans le manuel *AIX Commands Reference*.

Commande **whoami**

Une variante de la commande **who**, la commande **who am i**, affiche le nom de connexion, le nom du terminal et l'heure de la connexion.

A l'invite, entrez :

```
who am i
```

Le système affiche un écran semblable à celui-ci :

```
denise pts/0 Jun 21 07:53
```

Ici, le nom de connexion est `denise`, le nom du terminal, `pts/0`, et l'utilisateur s'est connecté à 7:53 (l'horloge AIX fonctionne sur 24 heures), le 21 juin.

Reportez-vous à la commande **who** dans le manuel *AIX Commands Reference*.

Commande **logname**

Une autre variante de la commande **who**, la commande **logname**, affiche les mêmes informations que la commande **who**.

A l'invite, entrez :

```
logname
```

Le système affiche un écran semblable à celui-ci :

```
denise
```

Ici, le nom de connexion est `denise`.

Reportez-vous à la commande **logname** dans le manuel *AIX Commands Reference*.

Affichage du nom du système d'exploitation (commande `uname`)

La commande `uname` affiche le nom du système d'exploitation utilisé.

Par exemple, entrez, à l'invite :

```
uname
```

Le système affiche un écran semblable à celui-ci :

```
AIX
```

Ici, le nom du système d'exploitation est AIX.

Reportez-vous à la commande `uname` dans le manuel *AIX Commands Reference*.

Affichage du nom de votre système (commande `uname`)

La commande `uname` assortie du paramètre `-n` affiche le nom de votre système (si vous êtes sur un réseau). Ce nom, qui identifie le système vis-à-vis du réseau, est distinct de votre ID de connexion.

Par exemple, entrez, à l'invite :

```
uname -n
```

Le système affiche un écran semblable à celui-ci :

```
barnard
```

Ici, le système s'appelle `barnard`.

Reportez-vous à la commande `uname` dans le manuel *AIX Commands Reference*.

Affichage des utilisateurs connectés (commande `who`)

La commande `who` affiche des informations sur les utilisateurs connectés au système local : Les informations suivantes s'affichent : nom de connexion, station de travail et date/heure de la connexion.

Remarque : Cette commande n'identifie que les utilisateurs se trouvant le noeud local.

Pour obtenir des informations sur les utilisateurs connectés au noeud local, entrez :

```
who
```

Le système affiche un écran semblable à celui-ci :

```
joe    lft/0 Jun 8 08:34  
denise pts/1 Jun 8 07:07
```

Ici, l'utilisateur `joe`, sur le terminal `lft/0`, s'est connecté à 8:34, le 8 juin.

Reportez-vous à la commande `who` dans le manuel *AIX Commands Reference*.

Affichage de l'ID d'un utilisateur (commande `id`)

La commande `id` affiche les ID d'un utilisateur donné. Les ID système sont des chiffres identifiant utilisateurs et groupes d'utilisateurs vis-à-vis du système. La commande `id` affiche les informations suivantes, le cas échéant :

- le nom et l'ID utilisateur réel ;
- le nom du groupe de l'utilisateur et l'ID groupe réel ;
- le nom et l'ID des groupes complémentaires, le cas échéant.

Par exemple, entrez, à l'invite :

```
id
```

Le système affiche un écran semblable à celui-ci :

```
uid=1544(sah) gid=300(build) euid=0(root) egid=9(printq)
groups=0(system),10(audit)
```

Dans cet exemple, l'utilisateur porte le nom `sah` avec un numéro d'ID 1544; un nom de groupe principal `build` avec un numéro d'ID 300; un nom d'utilisateur réel `root` avec un numéro d'ID 0; un nom de groupe réel `printq` avec un numéro d'ID 9; et deux noms de groupes supplémentaires `system` et `audit`, avec les numéros d'ID 0 et 10, respectivement.

Par exemple, entrez, à l'invite :

```
id denise
```

Le système affiche un écran semblable à celui-ci :

```
uid=2988(denise) gid=1(staff)
```

Ici, l'utilisateur `denise` (ID 2988) n'appartient qu'au groupe principal `staff` (ID 1).

Reportez-vous à la commande **id** dans le manuel *AIX Commands Reference*.

Mots de passe

Le système associe à chaque compte un mot de passe, qui assure une sécurité minimale aux fichiers. Ressources système et données sont des éléments précieux, qu'il convient de protéger. Il s'agit d'interdire aux utilisateurs non autorisés l'accès au système et de prévenir tout risque d'altération accidentelle ou volontaire des fichiers. Il s'agit également de réserver à certains l'usage de certaines commandes et l'accès à certains fichiers – droits et privilèges accordés exclusivement par les administrateurs système.

Cette section traite des points suivants :

- Mots de passe : règles, page 1-8
- Modification du mot de passe (commande **passwd**), page 1-9.
- Définition d'un mot de passe Null (commande **passwd**), page 1-10.

Mot de passe : règles

Un mot de passe doit être unique. *Il ne peut en aucun cas être partagé.* Gardez au contraire le secret, comme vous le faites pour les données sensibles de l'entreprise. Choisissez des mots de passe difficiles à deviner, mais faciles à retenir (que vous ne soyez pas obligé de noter).

Évitez les classiques : noms de vos enfants, dates de naissance, etc. Les noms communs sont également peu sûrs.

Un "bon" mot de passe comporte au moins 6 caractères dont des caractères non alphabétiques. Optez pour des combinaisons de mots bizarres ou des mots volontairement mal orthographiés.

Remarque : Si vous devez noter votre mot de passe pour vous en souvenir, choisissez—en un autre...

Voici quelques règles à respecter :

- Ne notez pas vos mots de passe sur un papier ou, si vous y êtes contraints, rangez le papier en lieu sûr (armoire fermée à clé, par exemple).
- Choisissez un mot de passe autre que votre ID utilisateur (même inversé, doublé ou quelque peu modifié).

- Ne réutilisez pas les mots de passe : le système peut être configuré pour les refuser.
- Ne choisissez pas un nom ou un prénom.
- N'utilisez pas de mots se trouvant dans le dictionnaire servant à la vérification orthographique.
- N'utilisez pas de mots de moins de 6 caractères.
- Proscrivez les mots obscènes : ce sont les premiers essayés par les éventuels intrus.
- Choisissez des mots de passe suffisamment faciles à retenir pour ne pas avoir à les écrire.
- Panachez lettres et chiffres, ainsi que majuscules et minuscules.
- Inventez un mot composé, séparé par un chiffre.
- Optez pour un mot prononçable, plus facile à retenir.

Modification du mot de passe (commande passwd)

Pour modifier votre mot de passe :

1. A l'invite, entrez :

```
passwdSi aucun mot de passe ne vous a été attribué, sautez
l'étape 2.
```

2. A l'invite :

```
Modification du mot de passe pour IDutil
Ancien mot de passe IDutil :
```

Cette précaution permet d'éviter qu'un utilisateur non autorisé change votre mot de passe en votre absence. Entrez votre mot de passe actuel.

3. A l'invite :

```
Nouveau mot de passe IDutil :
entrez le nouveau mot de passe souhaité.
```

4. A l'invite :

```
Faites une nouvelle tentative.
```

Cette confirmation évite d'enregistrer un mot de passe mal orthographié, que vous auriez du mal à retrouver.

Exemples

1. Pour modifier le mot de passe de l'utilisateur denise, entrez :

```
passwd
```

Le système affiche un écran semblable à celui-ci :

```
Modification du mot de passe pour "denise"
Ancien mot de passe :
Nouveau mot de passe :
Nouveau mot de passe.
$
```

2. Si vous indiquez un mot de passe (actuel) erroné, le système affiche :

```
$ passwd
Modification du mot de passe pour "denise"
Ancien mot de passe :
Les données entrées ne correspondent pas à votre ancien mot de
passe.
Vous n'êtes pas autorisé à modifier le mot de passe de "denise".
$
```

3. Si vous ne retapez pas exactement le même mot de passe, le système affiche :

```
$ passwd
Modification du mot de passe pour "denise"
Ancien mot de passe :
Nouveau mot de passe :
Nouveau mot de passe.
Entrez une seconde fois le nouveau mot de passe.
Nouveau mot de passe :
Nouveau mot de passe.
$
```

Reportez-vous à la commande **passwd** dans le manuel *AIX Commands Reference*.

Définition d'un mot de passe Null (commande passwd)

Si vous ne voulez pas avoir à entrer un mot de passe à chaque connexion, définissez un mot de passe "null" (blanc).

Pour ce faire, entrez :

```
passwd
```

Lorsque vous êtes invité à préciser le mot de passe, appuyez sur Entrée ou sur Ctrl-D.

Aucune confirmation ne vous est demandée. Seul un message rappelle que le mot de passe est défini comme NULL.

Reportez-vous à la commande **passwd** dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives à la connexion

Connexion et déconnexion

login	Ouvre la session.
logout	Arrête tous les process.
shutdown	Met fin aux opérations du système.
su	Modifie l'ID utilisateur associé à une session.
touch	Met à jour l'heure d'accès et de modification d'un fichier, ou crée un fichier vide.

Identification utilisateur et système

id	Affiche les identifications système d'un utilisateur.
logname	Affiche le nom de connexion.
uname	Affiche le nom du système d'exploitation actif.
who	Identifie les utilisateurs connectés.
whoami	Affiche votre nom de connexion.

Mot de passe

passwd	Modifie le mot de passe.
---------------	--------------------------

Chapitre 2. Système et environnement utilisateur

A chaque nom de connexion est associé un environnement système, à savoir une zone où sont enregistrées les informations communes à tous les processus en cours dans une session. Il existe également des commandes qui affichent des informations sur le système.

Ce chapitre traite des points suivants :

- Liste des unités du système (commande **lscfg**), page 2-2.
- Affichage du nom de votre console (commande **lscons**), page 2-3.
- Affichage du nom de votre terminal (commande **tty**), page 2-3.
- Liste des écrans disponibles (commande **lsdisp**), page 2-4.
- Liste des polices disponibles (commande **lsfont**), page 2-4.
- Liste des mappes clavier (commande **lskbd**), page 2-4.
- Liste des logiciels (commande **lsipp**), page 2-5.
- Liste des affectations de touches du terminal (commande **stty**), page 2-6.
- Liste de toutes les variables d'environnement (commande **env**), page 2-7.
- Affichage de la valeur d'une variable d'environnement (commande **printenv**), page 2-8.
- Langues particulières (commande **aixterm**), page 2-8.
- Récapitulatif des commandes relatives à l'environnement utilisateur et aux informations système, page 2-9.

Liste des unités du système (commande lscfg)

La commande **lscfg** affiche le nom, la position et la description de chaque unité de la configuration courante. La liste est triée sur la position des unités.

Pour lister les unités configurées de votre système, entrez, à l'invite :

```
lscfg
```

Un message semblable au suivant s'affiche :

```
LISTE DES RESSOURCES INSTALLEES
```

```
Les ressources suivantes sont installées sur la machine.
```

```
+/- = Ajoutée ou supprimée de la Liste des ressources testées.
```

```
*   = NON prise en charge par le programme de diagnostics.
```

```
+ sysplanar0    00-00          Carte CPU
+ fpa0          00-00          Processeur calcul virgule flottante
+ mem0         00-0A          Carte mémoire
+ mem1         00-0B          Carte mémoire
+ ioplanar0    00-00          Carte d'E/S
* f2bus0       00-00          Bus Micro Channel
+ rs2320       00-01          Carte RS232
+ tty0         00-01-0-01     Port carte RS232
- tty1         00-01-0-02     Port carte RS232
..
..
..
```

Pour afficher des informations sur une unité particulière, vous disposez de l'indicateur **-l**.

Ainsi, pour afficher les informations sur l'unité **sysplanar0**, entrez, à l'invite :

```
lscfg -l sysplanar0
```

Un message semblable au suivant s'affiche :

```
DEVICE          LOCATION      DESCRIPTION
sysplanar0      00-00        CPU Planar
```

Lancez également la commande **lscfg** pour afficher les données techniques essentielles (VPD), telles que références des pièces, numéros de série et niveaux d'EC. Pour certaines unités, ces données sont automatiquement collectées et ajoutées à la configuration du système. Pour d'autres, il faut les entrer : les données sont alors précédées de l'abréviation **ME** (manually entered).

Ainsi, pour lister les unités configurées du système, avec leurs données techniques essentielles, entrez :

```
lscfg -v
```

Un message semblable au suivant s'affiche :

```
LISTE DES RESSOURCES INSTALLEES AVEC VPD
```

Les unités suivantes sont installées sur le système.

```
sysplanar0  00-00  Carte CPU
```

```
  Référence.....342522
```

```
  Niveau modif technique.....254921
```

```
  Numéro de série.....353535
```

```
fpa0          00-00          Processeur calcul virgule flottante
```

```
mem0          00-0A          Carte mémoire
```

```
  Niveau modif technique.....990221
```

```
  .  
  .  
  .
```

Reportez-vous à la commande **lscfg** dans le manuel *AIX Commands Reference*.

Affichage du nom de votre console (commande **lscons**)

La commande **lscons** inscrit le nom de la console courante sur l'unité de sortie standard, généralement votre écran.

Par exemple, entrez, à l'invite :

```
lscons
```

Un message semblable au suivant s'affiche :

```
/dev/lft0
```

Reportez-vous à la commande **lscons** dans le manuel *AIX Commands Reference*.

Affichage du nom de votre terminal (commande **tty**)

La commande **tty** affiche le nom de votre terminal.

Par exemple, entrez, à l'invite :

```
tty
```

Le système affiche un écran semblable à celui-ci :

```
/dev/tty06
```

Dans cet exemple, `tty06` est le nom du terminal, et `/dev/tty06`, le fichier unité contenant l'interface avec ce terminal.

Reportez-vous à la commande **tty** dans le manuel *AIX Commands Reference*.

Liste des écrans disponibles (commande `lsdisp`)

La commande **lsdisp** affiche la liste des écrans disponibles sur le système, avec leur nom (identificateur et nom complet), le numéro de leur emplacement et une description.

Par exemple, pour afficher la liste de tous les écrans disponibles, entrez :

```
lsdisp
```

Voici un exemple de liste obtenue, affichant le nom de l'écran, son numéro d'emplacement et sa description. La liste est triée par ordre croissant de numéro d'emplacement.

Unité	Emplacement	Nom	Description
ppr0	00-01	POWER_G4	Midrange Graphics Adapter
gda0	00-03	colorgda	Color Graphics Display Adapter
ppr1	00-04	POWER_Gt3	Midrange Entry Graphics Adapter

Reportez-vous à la commande **lsdisp** dans le manuel *AIX Commands Reference*.

Liste des polices disponibles (commande `lsfont`)

La commande **lsfont** affiche la liste des polices disponibles sur votre écran.

Pour afficher la liste de toutes les polices disponibles, entrez :

```
lsfont -l
```

Voici un exemple de liste obtenue, affichant l'identificateur de la police, son type et le nombre de caractères par écran :

ID	Nom	Style	Graisse	Codage	Col X Lignes
0	Normal-R-N	Roman	Normal	PC850	80 X 251
1	Normal-I-N	Italic	Normal	PC850	120 X 35
2	Bold-R-Bol	Roman	Bold	PC850	120 X 35

Reportez-vous à la commande **lsfont** dans le manuel *AIX Commands Reference*.

Liste des mappes clavier (commande `lskbd`)

La commande **lskbd** affiche la liste des mappes clavier disponibles, avec leur nom et numéro prédéfinis.

Par exemple, pour afficher toutes les mappes clavier, entrez :

```
lskbd
```

Voici une exemple de liste obtenue, affichant le nom et le numéro prédéfinis du clavier :

```
0 USA
```

Reportez-vous à la commande **lskbd** dans le manuel *AIX Commands Reference*.

Liste des logiciels (commande **lslpp**)

La commande **lslpp** affiche des informations sur les logiciels disponibles sur le système.

Pour afficher la liste de tous les logiciels disponibles, entrez :

```
lslpp -l -a
```

Voici un exemple de liste obtenue :

Name	Fix Id	State	Description

Path: /usr/lib/objrepos			
INed.obj		APPLIED	INed Editor
X11_3d.gl.dev.obj		APPLIED	AIXwindows/3D GL Development Utilities
Fonts			
X11fnt.oldX.fnt		APPLIED	AIXwindows Miscellaneous X Fonts
X11mEn_US.msg		APPLIED	AIXWindows NL Message files
.			
.			
.			

Si la liste est très longue, le début défile et n'est plus visible. Pour éviter cet inconvénient, associez à la commande **lslpp** la commande **pg**. A l'invite, entrez :

```
lslpp | pg
```

Reportez-vous à la commande **lslpp** dans le manuel *AIX Commands Reference*.

Liste des affectations de touches du terminal (commande stty)

La commande **stty** affiche les paramètres du terminal. Vous pouvez en ignorer la plupart, mais vous devez connaître les touches de contrôle de votre terminal.

Par exemple, entrez, à l'invite :

```
stty -a
```

Le système affiche un écran semblable à celui-ci :

```
.  
. .  
intr = ^C; quit = ^\; erase = ^H; kill = ^U; eof = ^D;  
eol = ^@ start = ^Q; stop = ^S; susp = ^Z; dsusp = ^Y;  
reprint = ^R discard = ^O; werase = ^W; lnext = ^V  
. . .
```

Dans cet exemple, les lignes telles que `intr = ^C; quit = ^\; erase = ^H;` définissent l'affectation des touches de contrôle. La touche `^H` est la touche Retour arrière, qui exécute la fonction de suppression. Pour en savoir plus, reportez-vous à la section "Touches de contrôle" dans le manuel *AIX 4.3 Initiation*.

Si la liste est très longue, le début défile et n'est plus visible. Pour éviter cet inconvénient, associez à la commande **stty** la commande **pg**. A l'invite, entrez :

```
stty -a | pg
```

Reportez-vous à la commande **stty** dans le manuel *AIX Commands Reference*.

Liste de toutes les variables d'environnement (commande env)

La commande **env** affiche les variables d'environnement courantes. Une variable accessible à tous les process est dite *variable globale*.

Toutes les variables (avec leurs valeurs) connues d'une commande lors de son lancement constituent son *environnement* : variables héritées du process parent et variables spécifiées comme paramètres-clés sur la ligne de commande. Les interactions entre le shell et l'environnement sont multiples. A son lancement, le shell balaye l'environnement et, chaque fois qu'il trouve un nom, crée un paramètre auquel il affecte la valeur correspondante et qu'il marque pour exportation. Les commandes exécutées héritent de l'environnement.

Pour en savoir plus, reportez-vous à :

- Environnement shell Korn, page 11-10.
- Environnement shell Bourne, page 11-70.

Par exemple, pour afficher la liste de toutes les variables d'environnement, entrez :

```
env
```

Voici un exemple de liste obtenue :

```
TMPDIR=/usr/tmp
myid=denise
LANG=En_US
USERNAME=barnard
PAGER=/bin/pg
VISUAL=vi
PATH=/usr/ucb:/usr/lpp/X11/bin:/bin:/usr/bin:/etc:/u/denise:/u/denise/bin:/u/binl
MAILPATH=/usr/mail/denise?denise has mail !!!
MAILRECORD=/u/denise/.Outmail
EXINIT=set beautify noflash nomesg report=1 showmode showmatch
EDITOR=vi
PSCH=>
HISTFILE=/u/denise/.history
LOGNAME=denise
MAIL=/usr/mail/denise
PS1=denise@barnard:${PWD}>
PS3=#
PS2=>
epath=/usr/bin
USER=denise
SHELL=/bin/ksh
HISTSIZ=500
HOME=/u/denise
FCEDIT=vi
TERM=lft
MAILMSG=**YOU HAVE NEW MAIL. USE THE mail COMMAND TO SEE YOUR
PWD=/u/denise
ENV=/u/denise/.env
```

Si la liste est très longue, le début défile et n'est plus visible. Pour éviter cet inconvénient, associez à la commande **env** la commande **pg**. A l'invite, entrez :

```
env | pg
```

Reportez-vous à la commande **env** dans le manuel *AIX Commands Reference*.

Affichage de la valeur d'une variable d'environnement (commande `printenv`)

La commande `printenv` affiche la valeur des variables d'environnement. Si vous précisez le paramètre *Nom*, le système affiche la valeur de la variable concernée. Sinon, il affiche la valeur de toutes les variables courantes, sous la forme *Nom*, la commande `printenv` affiche toutes les variables d'environnement courantes, présentant une *Valeur*.

Ainsi, pour afficher la valeur de la variable `MAILMSG`, entrez :

```
printenv MAILMSG
```

La commande renvoie la valeur de la variable `MAILMSG`. Par exemple :

```
VOUS AVEZ DU COURRIER
```

Reportez-vous à la commande `printenv` dans le manuel *AIX Commands Reference*.

Langues particulières (commande `aixterm`)

La commande `aixterm` prend en charge les langues, tels l'arabe et l'hébreu, qui s'écrivent de droite à gauche. Ces langues sont particulières puisqu'on peut les lire et les écrire dans les deux directions. Pour travailler avec des applications dans ces langues, ouvrez une fenêtre en spécifiant l'arabe ou l'hébreu comme langue locale.

Reportez-vous à la commande `aixterm` dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives à l'environnement utilisateur et aux informations système

aixterm	Permet de travailler dans des langues bidirectionnelles.
env	Affiche l'environnement courant ou définit l'environnement d'exécution d'une commande.
lscfg	Affiche des informations de diagnostic sur une unité.
lscons	Affiche le nom de la console courante.
lsdisp	Liste les écrans disponibles sur le système.
lsfont	Liste les polices disponibles sur l'écran.
lskbd	Liste les mappes clavier chargées sur le système.
lslpp	Liste les logiciels disponibles.
printenv	Affiche la valeur des variables d'environnement.
stty	Affiche les paramètres du système.
tty	Affiche le chemin d'accès complet à votre terminal.

Chapitre 3. Commandes et process

Une *commande* est une demande d'exécution d'une opération ou d'un programme. C'est par le biais de commandes que vous indiquez au système d'exploitation les tâches à effectuer. Les commandes entrées sont décodées par un interpréteur de commandes (également appelé *shell*) et les tâches exécutées.

Un programme ou une commande en cours d'exécution est appelé un *process*. Le système peut exécuter simultanément plusieurs process.

Le système d'exploitation permet de gérer les entrées et les sorties (E/S) par le biais de commandes et de symboles spécifiques. Vous pouvez contrôler les entrées en indiquant où lire les données. Vous pouvez ainsi définir l'origine des entrées : clavier (entrée standard) ou fichier. Vous pouvez aussi contrôler la sortie en spécifiant l'emplacement d'affichage ou de stockage des données : par exemple, l'écran (sortie standard) ou un fichier.

Ce chapitre traite des points suivants :

- Commandes : généralités, page 3-2.
 - Syntaxe des commandes, page 3-2.
 - Diagrammes de syntaxe, page 3-4.
 - Instructions d'usage, page 3-6.
 - Utilisation de Web-based System Manager
 - Commande **smit**, page 3-7.
 - Localisation d'une commande ou d'un programme (**whereis**), page 3-7.
 - Affichage d'informations sur une commande (commande **whatis**), page 3-8.
 - Affichage des commandes précédentes (commande **history**), page 3-8.
 - Répétition de commandes via la commande shell History, page 3-9.
 - Substitution de chaîne via la commande shell history, page 3-9.
 - Modification de l'historique des commandes, page 3-10.
 - Création d'un alias (commande **alias**), page 3-10.
 - Commandes de formatage de texte, page 3-11
- Process : généralités, page 3-13.
 - Process d'avant et d'arrière-plan, page 3-13.
 - Démons, page 3-13.
 - Process zombie, page 3-14.
 - Lancement d'un process, page 3-14.
 - Contrôle des process (commande **ps**), page 3-14.
 - Définition de la priorité initiale d'un process (commande **nice**), page 3-16
 - Changement de la priorité d'un process en cours (commande **renice**), page 3-17
 - Interruption d'un process d'avant-plan, page 3-17.
 - Arrêt d'un process d'avant-plan, page 3-18.
 - Relance d'un process, page 3-18.
 - Planification d'un process en différé (commande **at**), page 3-18.
 - Liste des process planifiés (commande **at** ou **atq**), page 3-20
 - Déprogrammation d'un process (commande **at**), page 3-20
 - Suppression d'un process d'arrière-plan (commande **kill**), page 3-21.
- Récapitulatif des commandes relatives aux commandes et aux process, page 3-22.

Commandes : généralités

Certaines commandes sont composées d'un seul mot. D'autres peuvent être combinées de sorte que le résultat de la première devienne l'entrée de la suivante. Cette méthode est appelée *chaînage*.

Les indicateurs permettent de préciser l'action d'une commande. Un *indicateur* est un modificateur spécifié sur la ligne de commande, généralement précédé d'un tiret.

Les commandes peuvent être regroupées dans un fichier. Ces fichiers de commandes sont appelés *procédures shell* ou *scripts shell*. Exécuter un script revient à exécuter en séquence les commandes qu'il contient, au lieu de les lancer une par une.

Pour lancer une commande, tapez son nom en regard de l'invite et appuyez sur Entrée.

```
$ NomCommande
```

Ce chapitre traite des points suivants :

- Syntaxe des commandes, page 3-2.
- Diagrammes de syntaxe, page 3-4.
- Instructions d'usage, page 3-6.
- Utilisation de Web-based System Manager
- Commande **smit**, page 3-7.
- Localisation d'une commande ou d'un programme (**whereis**), page 3-7.
- Affichage d'informations sur une commande (commande **whatis**), page 3-8.
- Affichage des commandes précédentes (commande **history**), page 3-8.
- Répétition de commandes via la commande shell History, page 3-9.
- Substitution de chaînes via la commande shell History, page 3-9.
- Modification de l'historique des commandes, page 3-10.
- Création d'un alias (commande **alias**), page 3-10.
- Commandes de formatage de texte, page 3-11

Syntaxe des commandes

Si certaines commandes sont composées d'un seul mot, d'autres requièrent paramètres et indicateurs. A chaque commande est associée une syntaxe précisant les paramètres et les indicateurs, obligatoires ou facultatifs. La syntaxe générale d'une commande est la suivante :

```
NomCommande indicateur(s) paramètre(s)
```

Quelques règles :

- Les espaces entre commandes, indicateurs et paramètres sont significatifs.
- Deux commandes peuvent figurer sur une même ligne, sous réserve d'être séparées par un point-virgule (;). Par exemple :

```
$ commande1;commande2
```

Le shell exécute les commandes en séquence.

- Les commandes différencient majuscules et minuscules. Le shell fait la distinction entre les majuscules et les minuscules. Vis-à-vis du shell, les commandes `print`, `PRINT` et `Print` sont distinctes.

- Une commande très longue peut être spécifiée sur plusieurs lignes, sous réserve de terminer chaque ligne incomplète par une barre oblique inversée (\). Le shell est ainsi prévenu que la ligne continue. Par exemple :

```
$ ls Mail info temp \    (appuyez sur Entrée)
> diary                  (l'invite > s'affiche)
```

Le caractère > est l'invite secondaire (\$ étant l'invite principale par défaut de l'utilisateur non racine), indiquant que la ligne courante est la suite de la précédente. Notez que **cs** ne génère pas d'invite secondaire, que la coupure doit intervenir à la fin d'un mot, et que l'invite principale est %.

- Pour lancer une commande, tapez son nom en regard de l'invite et appuyez sur Entrée.

Nom d'une commande

Le premier mot d'une commande est obligatoirement son nom. Pour certaines commandes, ce nom est unique.

Indicateurs

A la suite du nom de la commande, peuvent se trouver un ou plusieurs indicateurs. Ils sont quelquefois appelés *options*. Un indicateur est encadré d'espaces ou de tabulations, et il est généralement précédé d'un tiret (-). Exceptions : **ps**, **tar** et **ar** (pas de tiret). Un indicateur modifie la fonction d'une commande. Par exemple, dans la commande :

```
ls -a -F
```

`ls` est le nom de la commande, `-a -F`, des indicateurs.

Lorsque des indicateurs sont associés à une commande, ils doivent suivre immédiatement son nom. Les indicateurs avec et sans tiret peuvent être combinés au sein d'une même commande. Ainsi, la commande précédente peut également s'écrire :

```
ls -aF
```

Certains paramètres doivent parfois être également précédés d'un tiret (-). Dans ce cas, spécifiez le délimiteur -- (tiret tiret) avant le paramètre. Le signe -- indique à la commande qu'il s'agit d'un paramètre et non d'un indicateur.

Si, par exemple, vous avez lancé la commande suivante pour créer le répertoire `-tmp` :

```
mkdir -tmp
```

Un message d'erreur semblable au suivant s'affiche :

```
mkdir : indicateur non reconnu : t
Syntaxe : mkdir [-p] [-m mode] Directory ...
```

Il convient en effet de spécifier :

```
mkdir -- -tmp
```

Le répertoire `-tmp` est créé.

Paramètres des commandes

A la suite du nom de la commande, peuvent se trouver un ou plusieurs indicateurs, suivis de paramètres. Ces paramètres sont parfois appelés *arguments* ou *opérandes*. Les paramètres précisent les informations requises pour l'exécution de la commande. Certains paramètres sont dotés d'une valeur par défaut, adoptée si vous n'en précisez pas d'autre. Par exemple, dans la commande :

```
ls -a tmp
```

`ls` est le nom de la commande, `-a`, un indicateur, et `temp`, un paramètre. Cette commande affiche tous (`-a`) les fichiers du répertoire `temp`. Dans l'exemple :

```
ls -a
```

en l'absence de paramètres, ce sont les fichiers du répertoire courant qui sont affichés par défaut. Dans l'exemple :

```
ls temp mail
```

aucun indicateur n'est fourni, et `temp` et `mail` sont des paramètres. `temp` et `mail` sont des paramètres (noms de répertoires). La commande affiche tous les fichiers des deux répertoires.

Lorsqu'un paramètre est constitué ou contient une valeur numérique, le nombre est interprété comme un entier décimal – sauf spécification contraire. Les nombres compris entre 0 et **INT_MAX** (défini dans **/usr/include/sys/limits.h**) sont syntaxiquement reconnus comme des valeurs numériques.

Si une commande accepte des valeurs négatives, vous pouvez y spécifier une valeur comprise entre **INT_MIN** et **INT_MAX**, tous deux définis dans **/usr/include/sys/limits.h**. Ce qui ne signifie pas que tous les nombres de cet intervalle sont sémantiquement corrects. Certaines commandes (commandes d'impression notamment) sont limitées à un intervalle plus restreint. Le message d'erreur généré dans ce cas précise qu'il s'agit d'un problème de dépassement des limites de l'intervalle, et non de syntaxe incorrecte.

Diagrammes de syntaxe

Les diagrammes de syntaxe schématisent l'écriture des commandes (sur la ligne de commande). Un diagramme précise :

- les indicateurs possibles,
- les indicateurs obligatoirement suivis d'un paramètre (option - argument),
- les indicateurs éventuellement suivis d'un paramètre,
- les valeurs par défaut éventuelles des indicateurs et des paramètres,
- la compatibilité des indicateurs entre eux,
- les indicateurs et paramètres optionnels,
- les possibilités de répétition de séquences d'indicateurs et de paramètres.

Les conventions adoptées dans les diagrammes de syntaxe sont les suivantes :

- les éléments à entrer littéralement sont en gras: nom de la commande, indicateurs, caractères littéraux, etc.
- les variables (à remplacer par leur valeur réelle) sont en *italique* : paramètres suivant les indicateurs et paramètres lus par la commande (*fichiers*, *répertoires*, etc.)
- les valeurs par défaut sont en caractères normaux, mais sur un trait plus épais.

0 Ligne de chemin d'accès Le diagramme de syntaxe commence par la ligne de chemin d'accès.

1 Nom de commande Dans le diagramme, cet élément est le nom de la commande que vous voulez exécuter. Il est en gras pour indiquer qu'il doit être saisi tel qu'il apparaît dans le diagramme.

Dans le diagramme donné en exemple, le chemin d'accès se divise en deux chemins à la suite du nom de la commande. Vous pouvez suivre le chemin inférieur (voir élément 2) ou le chemin supérieur (voir élément 3).

2 Boîte à choix unique	Si vous suivez le chemin inférieur, vous tombez sur une boîte surmontée des mots <i>l'un des</i> . Vous ne pouvez choisir qu'un seul élément dans cette boîte.
3 Ligne par défaut	Si vous suivez le chemin supérieur, vous contournez la boîte à choix unique et ne spécifiez rien. La ligne en gras autour de la boîte est une ligne par défaut qui signifie que vous n'avez rien à saisir dans cette partie du diagramme. Les exceptions sont généralement mentionnées dans la partie "Description". La ligne vierge par défaut autour des fichiers d'entrée et de sortie constitue une exception importante qui est expliquée dans l'élément 10.
4 Flèche d'itération	Lorsque vous suivez un chemin d'accès qui vous mène à une boîte entourée d'une flèche, vous devez choisir au moins un élément dans cette boîte. Vous pouvez ensuite soit retourner vers la boîte et continuer à choisir des éléments, soit poursuivre le chemin d'accès. Si vous suivez la flèche autour de la boîte (plutôt qu'une flèche qui comprend plusieurs branches sur le diagramme), vous ne pouvez pas choisir plusieurs fois le même élément.
5 Élément requis	La branche comportant la flèche d'itération est une branche à trois choix sans ligne par défaut. Vous devez donc faire un choix entre A, B ou C.
6 Aller à la ligne suivante	Si un diagramme est trop long pour tenir sur une ligne, ce caractère vous indique d'aller à la ligne suivante du diagramme pour continuer à taper votre commande. Rappelez-vous que le diagramme n'est pas terminé tant que vous n'avez pas atteint la marque verticale (voir élément 12).
7 Continuer le diagramme	Ce caractère vous indique comment continuer le diagramme après être arrivé à la fin d'une ligne.
8 Paramètre facultatif	Si un indicateur peut (mais n'est pas obligé de) prendre en compte un paramètre, le chemin d'accès se divise en branches après l'indicateur. Si vous ne pouvez pas taper d'espace entre l'indicateur et le paramètre, une note de bas de page vous en informe (voir élément 11).
9 Valeur par défaut	Une commande comporte souvent des valeurs ou des actions par défaut qui seront appliquées en l'absence de saisie d'un élément spécifique. Ces valeurs par défaut sont indiquées en caractères normaux dans la ligne par défaut si elles sont équivalentes à un élément que vous pouvez taper dans la ligne de commande (par exemple, un indicateur avec une valeur). Si la valeur par défaut n'est pas un élément que vous pouvez saisir sur la ligne de commande, elle n'est pas indiquée dans le diagramme. Nota: Les valeurs par défaut sont données dans le diagramme pour information. Il n'est pas nécessaire de les faire figurer dans la ligne de commande.
10 Entrée ou Sortie	Une commande qui peut lire des fichiers d'entrée ou des entrées classiques comporte une ligne par défaut vide au-dessus du paramètre du fichier. Si la commande peut écrire sa sortie vers un fichier de sortie ou vers une sortie classique, elle est également montrée avec une ligne par défaut vide au-dessus du paramètre du fichier de sortie.

Si une commande peut lire uniquement des entrées classiques, il n'y a pas de fichier d'entrée dans le diagramme et l'entrée classique est implicite. Si une commande écrit uniquement à destination d'une sortie classique, il n'y a pas de fichier de sortie dans le diagramme et la sortie classique est implicite.

Lorsque vous devez donner un nom de fichier pour l'entrée ou la sortie, le paramètre du fichier est compris dans le diagramme et aucune ligne par défaut vide ne figure au-dessus.

11 Note de bas de page

Si une commande obéit à des exigences ou des restrictions particulières, une note de bas de page le signale.

12 Marque verticale

Elle termine le diagramme de syntaxe.

Instructions d'usage

Les instructions d'usage sont un autre moyen de représenter la syntaxe d'une commande. Les instructions d'usage sont un autre moyen de représenter la syntaxe d'une commande. Comme les diagrammes, elles indiquent comment entrer des commandes sur la ligne de commande, mais sous une forme différente. Elles utilisent notamment des symboles tels que [] (crochets), { } (accolades) et | (barres verticales). Voici un exemple concernant la commande **unget** :

unget [*-rSID*] [*-s*] [*-n*] *Fichier ...*

Les conventions quant à la mise en **gras** et en *italique* sont les mêmes que pour les diagrammes de syntaxe. En outre :

- les paramètres entre crochets sont facultatifs ;
- les paramètres entre accolades sont obligatoires ;
- les paramètres isolés sont obligatoires ;
- une barre verticale signifie que vous ne pouvez sélectionner plus d'un élément : de même, [a | b] indique vous *pouvez* choisir a ou b, ou rien ; De même, { a | b } indique que vous *devez* choisir a ou b.
- des points de suspension (. . .) signifient que le paramètre peut être répété sur la ligne de commande.
- le tiret (-) représente l'entrée standard.

Utilisation de Web-based System Manager

Web-based System Manager est une interface GUI de gestion du système, soit à partir d'un écran rattaché en local ou à distance, depuis un autre système AIX ou depuis un ordinateur personnel équipé d'un navigateur Web. Le lancement de Web-based System Manager peut s'effectuer de multiples façons :

- à partir d'un terminal à ligne de commande dans l'environnement CDE (Common Desktop Environment) à l'aide d'un raccourci.
- à partir d'un terminal à ligne de commande dans l'environnement AIXwindows à l'aide d'un raccourci.
- à partir du gestionnaire d'applications environnement CDE (Common Desktop Environment) en cliquant sur son icône dans le dossier System_Admin.
- à partir d'un navigateur web compatible HTML version 3.2 sur un ordinateur personnel.

Commande **smit**

La commande **smit** est un outil permettant de lancer d'autres commandes. Les noms des commandes sont dans ce cas spécifiés comme paramètres de la commande **smit**, laquelle vous amène alors souvent à un sous-menu ou à la boîte de dialogue correspondant à la commande spécifiée. Par exemple, `smit lsuser` appelle directement la boîte de dialogue **List All Users**, qui permet de répertorier les attributs utilisateurs sur votre système.

Reportez-vous à la commande **smit** dans le manuel *AIX Commands Reference*.

Localisation d'une commande ou d'un programme (commande **whereis**)

La commande **whereis** localise la commande ou le programme spécifié dans les répertoires source et binaires, et les sections de manuels. Elle procède à la recherche à partir d'une liste d'emplacements standard.

Par exemple, pour rechercher les fichiers du répertoire courant non documentés, entrez :

```
whereis -m -u *
```

Pour rechercher tous les fichiers contenant le mot `Mail`, entrez :

```
whereis Mail
```

Le système affiche un écran semblable à celui-ci :

```
Mail: /usr/bin/Mail /usr/lib/Mail.rc
```

Reportez-vous à la commande **whereis** dans le manuel *AIX Commands Reference*.

Affichage d'informations sur une commande (commande **man**)

La commande **man** affiche des informations sur les commandes, les routines et les fichiers. Elle respecte la syntaxe :

```
man NomCommande
```

Par exemple, pour obtenir des informations sur la commande **pg**, entrez :

```
man pg
```

Le système affiche un écran semblable à celui-ci :

```
Commande pg
```

```
Purpose
```

```
Formats files to the display.
```

```
Syntaxe
```

```
pg [ - Numéro_ligne ] [ -c ] [ -e ] [ -f ] [ -n ] [ -pChaîne ]  
[ -s ] [ +Ligne | +/Forme/ ] [ Fichier ... ]
```

```
Description
```

```
The pg command reads a file name from the File parameter and writes the file to standard output one screen at a time. If you specify a - (dash) as the File parameter, or run the pg command without options, the pg command reads standard input. Each screen is followed by a prompt. If you press the Enter key, another page is displayed. Subcommands used with the pg command let you review or search in the file.
```

Les informations affichées par **man** sont également accessibles via InfoExplorer.

Reportez-vous à la commande **man** dans le manuel *AIX Commands Reference*.

Affichage de la fonction d'une commande (commande **whatis**)

La commande **whatis** recherche la commande, l'appel système, la fonction de bibliothèque et le nom de fichier spécial, spécifié par le paramètre *commande*, dans la base de données que vous avez créée via la commande **catman -w**. Elle affiche l'en-tête de section correspondant. Vous pouvez alors, en lançant la commande **man**, obtenir des informations complémentaires.

La commande **whatis** équivaut à la commande **man -f**.

Par exemple, pour connaître l'objet de la commande **ls**, entrez :

```
whatis ls
```

Le système affiche un écran semblable à celui-ci :

```
ls(1)  -Displays the contents of a directory.
```

Reportez-vous à la commande **whatis** dans le manuel *AIX Commands Reference*.

Affichage des commandes précédentes (commande **shell history**)

La commande **history** est une commande intégrée du shell Korn, qui répertorie les 16 dernières commandes émises. Le shell enregistre ces commandes dans un fichier d'historique, généralement appelé **\$HOME/.sh_history**. Cette fonction permet de gagner du temps lorsque vous devez répéter une commande.

Par défaut, le shell Korn sauvegarde le texte des 128 dernières commandes. La taille du fichier d'historique (spécifiée par la variable **HISTSIZE**) n'est pas limitée ; un fichier trop volumineux risque toutefois de ralentir le shell.

Remarque : La commande shell **history** est distincte de la commande **history** utilisée avec l'éditeur **INed**. Notez également que le shell Bourne ne prend pas en charge d'historique de commandes.

Pour en savoir plus, voir "Shells", page 11-1.

Par exemple, pour afficher la liste des commandes précédentes, entrez, à l'invite :

```
history
```

Cette commande liste les 16 dernières commandes. Le système affiche un écran semblable à celui-ci :

```
928  ls
929  mail
930  printenv MAILMSG
931  whereis Mail
932  whatis ls
933  cd /usr/include/sys
934  ls
935  man pg
936  cd
937  ls | pg
938  lscons
939  tty
940  li *.txt
941  printenv MAILMSG
942  pwd
943  history
```

La liste indique d'abord la position de la commande dans **\$HOME/.sh_history**, puis la commande.

Par exemple, pour afficher les 5 dernières commandes, entrez, à l'invite :

```
history -5
```

Une liste semblable à la suivante s'affiche :

```
939 tty
940 li *.txt
941 printenv MAILMSG
942 pwd
943 history
944 history -5
```

La commande **history** suivie d'un numéro affiche les commandes émises, à partir de ce numéro.

Par exemple, pour afficher les commandes à partir de 938, entrez, à l'invite :

```
history 938
```

Une liste semblable à la suivante s'affiche :

```
938 lscons
939 tty
940 li *.txt
941 printenv MAILMSG
942 pwd
943 history
944 history -5
945 history -938
```

Répétition de commandes via la commande shell history

Pour répéter une commande antérieure, lancez l'alias du shell Korn **r**. Tapez **r**, suivi éventuellement du numéro ou du (des) premier(s) caractère(s) de la commande.

Par exemple, pour afficher la liste des écrans disponibles, vous entrez la commande **lsdisp** à l'invite. Le système vous renvoie les informations demandées. Si vous souhaitez qu'il les réaffiche, entrez, à l'invite :

```
r
```

Le système réexécute la dernière commande. Il s'agit de **lsdisp**, dans l'exemple.

Pour répéter la commande **li *.txt**, entrez, à l'invite :

```
r li
```

L'alias **r** du shell Korn localise la dernière commande commençant par les caractères spécifiés.

Substitution de chaînes via la commande shell history

Vous pouvez également utiliser l'alias **r** du shell Korn pour modifier une commande avant de la lancer. Vous pouvez, dans ce cas, passer par un paramètre de substitution de la forme *Ancien=Nouveau* pour modifier la commande avant de l'exécuter.

Par exemple, si la ligne de commande 940 spécifie **li *.txt**, et que vous souhaitez exécuter **li *.exe**, entrez, à l'invite :

```
r txt=exe 940
```

La commande 940 est exécutée, avec substitution de **txt** par **exe**.

Si la commande 940 est la dernière des commandes commençant par la lettre **l** vous pouvez également entrer :

```
r txt=exe l
```

Remarque : Seule la première occurrence de l'*ancienne* chaîne est remplacée par la *nouvelle*. Entrer l'alias shell Korn **r** sans numéro de commande ou lettre d'identification effectue la substitution dans la dernière commande spécifiée.

Modification de l'historique des commandes

La commande intégrée du shell Korn **fc** permet d'afficher et de modifier tout ou partie du fichier d'historique. Le cas échéant, précisez le numéro ou l'intervalle de numéros (ou encore les premiers caractères) des commandes qui vous intéressent. Vous pouvez spécifier une ou plusieurs commandes.

Si vous ne précisez pas de programme d'édition en argument de la commande **fc**, c'est l'éditeur spécifié par la variable **FCEDIT** qui est utilisé. En cas de non définition de la variable **FCEDIT**, l'éditeur **/usr/bin/ed** est utilisé. Les commandes modifiées sont affichées et exécutées dès que vous quittez l'éditeur. Pour afficher la valeur de la variable **FCEDIT**, lancez la commande **printenv**.

Par exemple, pour lancer la commande :

```
cd /usr/tmp
```

très proche de la ligne de commande 933, entrez, à l'invite :

```
fc 933
```

L'éditeur par défaut est appelé, affichant la ligne de commande 933. Modifiez `include/sys` en `tmp`, et la commande modifiée est exécutée lorsque vous quittez l'éditeur.

Vous pouvez également choisir l'éditeur.

Par exemple, pour modifier une commande avec l'éditeur `/usr/bin/vi`, entrez, à l'invite :

```
fc -e vi 933
```

L'éditeur **vi** est chargé, affichant la commande 933.

Vous pouvez également indiquer un intervalle de commandes à modifier.

Par exemple, pour modifier les commandes 930 à 940, entrez, à l'invite :

```
fc 930 940
```

L'éditeur par défaut est appelé, affichant les lignes de commande 930 à 940. Lorsque vous quittez l'éditeur, toutes les commandes affichées sont exécutées en séquence.

Création d'un alias (commande shell alias)

Créer un *alias* permet d'attribuer un mnémonique à une commande, à un nom de fichier ou à tout texte shell. Vous pouvez ainsi accélérer le lancement des tâches que vous exécutez fréquemment. La commande intégrée du shell Korn **alias** permet de définir un mot comme alias de certaines commandes. Vous avez le droit de redéfinir par un alias les commandes intégrées, mais non les mots réservés.

L'initiale d'un alias peut être n'importe quel caractère imprimable (non spécial). Les autres caractères doivent respecter les règles applicables aux noms de fichiers.

Format de création d'un alias :

```
alias Nom=Chaîne
```

où *nom* indique le nom de l'alias et *chaîne*, une chaîne de caractères. Si *chaîne* comporte des espaces, mettez-la entre guillemets.

Par exemple, pour créer un alias de la commande **rm -i** (qui vous demande confirmation avant de supprimer des fichiers), entrez, à l'invite :

```
alias rm="/usr/bin/rm -i"
```

A partir de là, chaque fois que vous entrez la commande **rm**, la commande **/usr/bin/rm -i** est exécutée.

Pour créer un alias de la commande **ls -aLF | pg** (qui affiche des informations complètes sur tous les fichiers du répertoire courant, fichiers invisibles compris ; marque l'un des fichiers exécutables d'un * et les répertoires d'un /; défile écran par écran), entrez, à l'invite :

```
alias dir="/usr/bin/ls -aLF | pg"
```

A partir de là, chaque fois que vous entrez la commande alias **dir**, la commande **/usr/bin ls -aLF | pg** est exécutée.

Pour afficher tous les alias définis, entrez, à l'invite :

```
alias
```

Le système affiche un écran semblable à celui-ci :

```
rm="/usr/bin/rm -i"  
dir="/usr/bin/ls -aLF | pg"
```

Commandes de formatage de texte

Vous disposez de commandes de formatage pour les textes composés avec le jeu de caractères internationaux d'AIX, utilisé pour les langues européennes.

Formatage de texte en caractères internationaux

Le jeu étendu de caractères internationaux d'AIX comporte les caractères et les symboles utilisés par nombre de langues européennes, avec un sous-ensemble ASCII composé de caractères, de chiffres et de signes de ponctuation anglais.

Tous les caractères du jeu étendu d'AIX ont une forme ASCII, qui permet de représenter le caractères en entrée. Les caractères peuvent également être entrés directement au clavier (sous réserve que celui-ci prenne en charge le jeu étendu de caractères européens).

Les commandes suivantes permettent de traiter des textes dans différentes langues, si tous les caractères sont mono-octets. Ces commandes se trouvent dans `/usr/bin`. (Celles suivies d'un astérisque (*) permettent de traiter les textes composés dans les langues multi-octets.)

adbbib*	hyphen	pic*	pstext
checkmm	ibm3812	ps4014	refer*
checknr*	ibm3816	ps630	roffbib*
col*	ibm5587G*	psbanne	soelim*
colcrt	ibm5585H-T*	psdit	sortbib*
deroff*	indxbib*	psplot	tbl*
enscript	lookbib*	psrev	troff*
eqn*	makedev*	psroff	vgrind
grap*	neqn*	psrv	xpreview*
hplj	nroff*		

Les commandes de formatage et les macros non citées ne s'appliquent pas aux caractères internationaux d'AIX.

Entrée de caractères étendus mono-octets

Si votre unité d'entrée prend en charge les caractères du jeu étendu de caractères européens, vous pouvez les entrer directement. Sinon, passez par la séquence d'échappement ASCII :

`\[N]`, *N* étant le code hexadécimal (2 ou 4 chiffres) du caractère.

Remarque : La forme NCesc `\<xx>` n'est plus prise en charge.

Les textes comportant des caractères étendus sont émis en sortie selon les conventions de formatage de la langue utilisée. Les caractères non définis pour l'interface d'une unité ne génèrent ni sortie ni notification d'erreur.

Bien que la plupart des demandes, des macros et des commandes soient basées sur l'anglais, elles acceptent presque toutes les entrées (noms de fichiers, paramètres, etc.) comportant des caractères du jeu européen étendu.

Pour les commandes **nroff** et **troff** et leurs préprocesseurs, l'entrée doit être en ASCII, faute de quoi une erreur de syntaxe irrémédiable est générée. Les caractères internationaux, mono ou multi-octets, peuvent être entrés entre guillemets – s'il n'y a pas d'autre texte à formater. Par exemple, en utilisant des macros de la commande **pic** :

```
define foobar % texte %
```

Après la directive `define`, le premier nom, `foobar`, doit être en ASCII. Mais le `texte` de remplacement peut comporter des caractères non ASCII.

Formatage de texte en caractères multi-octets

Certaines commandes permettent de traiter des textes dans des langues multi-octets. Elles sont repérées par un astérisque (*) dans la liste "Formatage de texte en caractères internationaux", page 3-11. Les commandes de formatage de texte qui ne sont pas répertoriées dans cette liste ne peuvent pas traiter les caractères internationaux sous AIX.

Entrée de caractères multi-octets

Si votre unité d'entrée prend en charge les caractères multi-octets, vous pouvez les entrer directement. Sinon, passez par le format ASCII `\[N]`, où N est le code hexadécimal (2, 4, 6, 7 ou 8 chiffres) du caractère.

Bien que la plupart des demandes, des macros et des commandes soient basées sur l'anglais, elles acceptent presque toutes des entrées (noms de fichiers, paramètres, etc.) comportant des caractères multi-octets.

Pour les familiers du formatage de texte en caractères mono-octets, la liste suivante récapitule les caractéristiques remarquables ou propres aux environnements multi-octets :

- il n'y a pas de césure du texte ;
- les sorties numériques requièrent des types de format spéciaux ; les types japonais sont disponibles ;
- la sortie est effectuée en lignes horizontales, remplies de gauche à droite ;
- les caractères sont automatiquement alignés en colonnes ;
- les caractères non définis pour l'interface d'une unité ne génèrent ni sortie ni notification d'erreur.

Comme pour les commandes **nroff** et **troff** et leurs préprocesseurs, l'entrée doit être en ASCII, faute de quoi une erreur de syntaxe irrémédiable est générée. Les caractères internationaux, mono ou multi-octets, peuvent être entrés entre guillemets, éventuellement à l'intérieur de texte à formater. Par exemple, en utilisant des macros de la commande **pic** :

```
define foobar % texte %
```

Après la directive `define`, le premier nom, `foobar`, doit être en ASCII. Mais le `texte` de remplacement peut comporter des caractères non ASCII.

Process : généralités

Un programme ou une commande en cours d'exécution est appelé un *process*. Les process sont structurés selon une hiérarchie parent-enfant. Un process lancé par un programme ou une commande est un *process parent*; le process résultant étant un *process enfant*. Un process parent peut engendrer plusieurs process enfant, tandis qu'un process enfant ne peut avoir qu'un seul parent.

Le système attribue un numéro d'identification (PID) à chaque process, au moment de son lancement : si vous lancez plusieurs fois le même process, un PID différent lui est associé à chaque fois.

Un process lancé mobilise une partie des ressources système. Lorsque plusieurs process sont simultanément en cours, un répartiteur (partie intégrante du système d'exploitation) assure le partage du temps machine, en fonction des priorités définies. Pour modifier ces priorités, vous disposez des commandes **nice** et **renice**.

Remarque : Seuls les détenteurs des droits root sont habilités à attribuer à un process une priorité plus élevée. Tout utilisateur peut, en revanche, attribuer à un process une priorité moindre, via la commande **nice** (ou **renice** pour un process déjà en cours).

Ce chapitre traite des points suivants :

- Process d'avant et d'arrière-plan
- Démons
- Process zombie
- Lancement d'un process
- Planification d'un process en différé (commande **at**)
- Liste des process planifiés (commande **at** ou **atq**)
- Contrôle des process (commande **ps**)
- Définition de la priorité initiale d'un process (commande **nice**)
- Changement de la priorité d'un process en cours (commande **renice**)
- Interruption d'un process d'avant-plan
- Arrêt d'un process d'avant-plan
- Relance d'un process
- Déprogrammation d'un process (commande **at**)
- Suppression d'un process d'arrière-plan (commande **kill**)

Process d'avant et d'arrière-plan

Les process lancés par un utilisateur et qui requièrent une interaction avec celui-ci sont dits *d'avant-plan*. Les process exécutés indépendamment de l'utilisateur sont dits *d'arrière-plan*. Par défaut, programmes et commandes sont exécutés en avant-plan : pour exécuter un process en arrière-plan, placez une perluète (&) à la fin du nom de la commande de lancement du process.

Démons

Les *démons* sont des process exécutés sans contrôle. Ils se trouvent en permanence à l'arrière-plan et sont toujours disponibles. Ils sont généralement lancés et arrêtés en même temps que le système. Un démon, qui exécute des tâches système, peut être appelé par plusieurs tâches ou utilisateurs. Les démons sont démarrés par l'utilisateur ou le shell root et arrêtés exclusivement par l'utilisateur racine. Ainsi, le démon **qdaemon** donne accès à des ressources système, telles que les imprimantes. Le démon **sendmail** est également très utilisé.

Process zombie

Un *process zombie* est un process arrêté, mais toujours reconnu dans la table des process (autrement dit, il possède un PID) sans avoir d'autre espace système attribué. Ces process se sont ou ont été arrêtés, mais continuent d'exister jusqu'à l'arrêt de leur process parent ou jusqu'à l'arrêt puis la relance du système. Les process zombie sont marqués `<defunct>` lorsqu'ils sont répertoriés via la commande **ps**.

Lancement d'un process

Pour lancer un process d'avant-plan à partir d'une station de travail, entrez le nom du programme ou de la commande à l'invite du système. Une fois démarré, le process entre en interaction avec vous. Vous ne pouvez envisager d'autres interactions (entrer une autre commande, par exemple) tant que le process n'a pas pris fin.

Un utilisateur peut lancer simultanément plusieurs process (par défaut, 40 au maximum).

Lancement en avant-plan

Pour lancer un process d'avant-plan, entrez le nom du programme ou de la commande, assorti des paramètres et indicateurs appropriés :

```
$ NomCommande
```

Lancement en arrière-plan

Pour lancer un process d'arrière-plan, entrez le nom du programme ou de la commande, assorti des paramètres et indicateurs appropriés, et suivi d'une perluète (&) :

```
$ NomCommande&
```

Pendant que le process s'exécute en arrière-plan, vous avez toute latitude pour lancer d'autres commandes ou programmes à partir de votre station de travail.

Les process en arrière-plan sont fort utiles pour les commandes longues à exécuter. Toutefois, dans la mesure où ils accroissent la charge du processeur, l'ensemble des opérations du système est ralenti.

La plupart des process, même exécutés en arrière-plan, dirigent leurs sorties vers la sortie standard. Sauf s'ils ont été explicitement réacheminés, ils se dirigent vers l'écran. Pour éviter toute interférence entre les résultats d'un process en arrière-plan et votre travail en cours, nous vous conseillons de réacheminer les sorties des process d'arrière-plan vers un fichier ou une imprimante. Vous pourrez les consulter à votre guise quand vous serez disponible.

Remarque : Dans certains cas, la séquence des sorties d'un process peut différer selon que le process est exécuté en avant ou en arrière-plan. Les programmeurs disposent de la sous-routine **fflush** pour garantir l'ordre des sorties, indépendamment du mode d'exécution du process.

Tant qu'un process d'arrière-plan est en cours, vous pouvez vérifier son état via la commande **ps**.

Contrôle des process (commande ps)

Dès que le système est actif, plusieurs process le sont également. Pour afficher la liste de ces process et des informations les concernant, vous disposez de la commande **ps** ou d'un raccourci Web-based System Manager.

Commande ps

A la commande **ps** sont associés divers indicateurs, permettant de préciser les process à lister et les informations à afficher pour chaque process.

Ainsi, pour afficher tous les process en cours, entrez, à l'invite :

```
ps -ef
```

Le système affiche un écran semblable à celui-ci :

```
USER  PID  PPID  C   STIME      TTY  TIME  CMD
root   1    0    0   Jun 28     -   3:23  /etc/init
root  1588  6963  0   Jun 28     -   0:02  /usr/etc/biod 6
root  2280   1    0   Jun 28     -   1:39  /etc/syncd 60
mary  2413 16998  2 07:57:30   -   0:05  aixterm
mary 11632 16998  0 07:57:31  lft/1 0:01  xbiff
mary 16260 2413  1 07:57:35  pts/1 0:00  /bin/ksh
mary 16469   1    0 07:57:12  lft/1 0:00  ksh /usr/lpp/X11/bin/xinit
mary 19402 16260 20 09:37:21  pts/1 0:00  ps -ef
```

Voici l'explication des titres des colonnes :

USER	nom de connexion de l'utilisateur
PID	ID process parent
PPID	ID process parent
C	CPU
STIME	heure de début du process
TTY	contrôle de la station de travail
TIME	temps CPU utilisé
CMD	commande

Dans l'exemple précédent, l'ID process de la commande `ps -ef` est 19402. L'ID de son process parent est 16260, correspondant à la commande `/bin/ksh`.

Si la liste est très longue, le début défile et n'est plus visible. Pour obtenir un affichage page par page, chaînez les commandes **ps** et **pg**. A l'invite, entrez :

```
ps -ef | pg
```

Ainsi, pour afficher des informations sur les process en cours, entrez, à l'invite :

```
ps gv
```

Cette commande affiche des statistiques sur chaque process actif. La sortie obtenue est semblable à :

PID	TTYSTAT	TIME	PGIN	SIZE	RSS	LIM	TSIZ	TRS	%CPU	%MEM	COMMAND
0	- A	0:44	7	8	8	xx	0	0	0.0	0.0	swapper
1	- A	1:29	518	244	140	xx	21	24	0.1	1.0	/etc/init
771	- A	1:22	0	16	16	xx	0	0	0.0	0.0	kproc
1028	- A	0:00	10	16	8	xx	0	0	0.0	0.0	kproc
1503	- A	0:33	127	16	8	xx	0	0	0.0	0.0	kproc
1679	- A	1:03	282	192	12	32768	130	0	0.7	0.0	pcidossvr
2089	- A	0:22	918	72	28	xx	1	4	0.0	0.0	/etc/sync
2784	- A	0:00	9	16	8	xx	0	0	0.0	0.0	kproc
2816	- A	5:59	6436	2664	616	8	852	156	0.4	4.0	/usr/lpp/
3115	- A	0:27	955	264	128	xx	39	36	0.0	1.0	/usr/lib/
3451	- A	0:00	0	16	8	xx	0	0	0.0	0.0	kproc
3812	- A	0:00	21	128	12	32768	34	0	0.0	0.0	usr/lib/lpd/
3970	- A	0:00	0	16	8	xx	0	0	0.0	0.0	kproc
4267	- A	0:01	169	132	72	32768	16	16	0.0	0.0	/etc/sysl
4514	lft/0A	0:00	60	200	72	xx	39	60	0.0	0.0	/etc/gett
4776	pts/3A	0:02	250	108	280	8	303	268	0.0	2.0	-ksh
5050	- A	0:09	1200	424	132	32768	243	56	0.0	1.0	/usr/sbin
5322	- A	0:27	1299	156	192	xx	24	24	0.0	1.0	/etc/cron
5590	- A	0:00	2	100	12	32768	11	0	0.0	0.0	/etc/writ
5749	- A	0:00	0	208	12	xx	13	0	0.0	0.0	/usr/lpp/
6111	- T	0:00	66	108	12	32768	47	0	0.0	0.0	/usr/lpp/

Reportez-vous à la commande **ps** dans le manuel *AIX Commands Reference*.

Web-based System Manager Raccourci

Vous pouvez vous servir du raccourci Web-based System Manager pour découvrir les process actifs et pour afficher des informations sur ces process. A l'invite, tapez :

```
wsm processes
```

Dans le conteneur Processes, sélectionnez le process, puis vérifiez son état à l'aide des menus.

Vous pouvez également utiliser les commandes **smit process** et **smit monitors**.

Définition de la priorité initiale d'un process (commande nice)

Pour attribuer à un process une priorité inférieure, lancez-le via la commande **nice** :

Remarque : Pour attribuer à un process une priorité supérieure, vous devez détenir les droits de l'utilisateur racine.

Commande nice

Pour définir la priorité initiale d'un process, entrez :

```
nice -n Numéro ChaîneCommande
```

où *Numéro* est un nombre compris entre 0 et 39, 39 étant la priorité la plus basse (plus le nombre est élevé, plus la priorité est basse – zéro conservant la priorité de planification initiale). *ChaîneCommande* est la chaîne de commande du process à exécuter.

Reportez-vous à la commande **nice** dans le manuel *AIX Commands Reference*.

Vous pouvez également exécuter cette tâche à l'aide de la commande **smit nice**.

Changement de la priorité d'un process en cours (commande renice)

Pour modifier la priorité d'un process en cours, lancez la commande **renice** : Cette commande modifie la valeur nice d'un process. Cette commande se lance de deux façons :

- à partir de la ligne de commande
- à partir du raccourci Web-based System Manager

Remarque : Pour attribuer à un process (que vous n'avez pas lancé) une priorité supérieure, vous devez détenir les droits de l'utilisateur racine.

A partir de la ligne de commande

Pour modifier la priorité initiale d'un process en cours, entrez :

```
renice Priorité -p IDProcess
```

où *Priorité* est un nombre compris entre -20 et 20 (plus le nombre est élevé, plus la priorité est basse). La valeur zéro conserve la priorité de planification initiale. *IDProcess* est le PID dont vous souhaitez modifier la priorité.

Web-based System Manager Raccourci

Vous pouvez utiliser le raccourci Web-based System Manager pour modifier la priorité d'un process actif. A l'invite, tapez :

```
wsm processes
```

Dans le conteneur Processes, sélectionnez le process, puis vérifiez son état à l'aide des menus.

Vous pouvez également exécuter cette tâche à l'aide de la commande **smit renice**.

Interruption d'un process d'avant-plan

Pour interrompre un process d'avant-plan avant la fin, appuyez sur la touche d'interruption, généralement Ctrl-C ou Ctrl-Arrière. Pour plus de précisions, reportez-vous à la section "Liste des affectations de touches du terminal (commande stty), page 2-6.

Remarque : La touche d'INTERRUPTION (Ctrl-C) n'arrête pas les process d'arrière-plan. Pour annuler un process d'arrière-plan, vous devez utiliser la commande **kill**.

Les commandes très simples ne peuvent servir d'exemple de cette procédure : leur exécution est si rapide que vous n'auriez pas le temps de les interrompre. C'est pourquoi les exemples qui suivent portent sur la commande `find / -type f`, qui s'exécute un peu plus longtemps. Cette commande affiche le nom des chemins d'accès de tous les fichiers du système. Vous n'avez pas besoin d'approfondir la commande **find** pour poursuivre : elle n'est citée qu'à titre d'illustration du fonctionnement des process.

Dans l'exemple suivant, la commande **find** lance un process que vous pouvez interrompre au bout de quelques secondes :

```
$ find / -type f
/usr/sbin/acct/lastlogin
/usr/sbin/acct/prctmp
/usr/sbin/acct/prdaily
/usr/sbin/acct/runacct
/usr/sbin/acct/sdisk
/usr/sbin/acct/shutacct INTERRUPT (Ctrl-C)
$ _
```

L'invite système est réaffichée : vous pouvez lancer d'autres commandes.

Arrêt d'un process d'avant-plan

Il est possible d'arrêter un process sans que son PID soit supprimé de la table des process. Le process peut être arrêté par Ctrl-Z, par exemple.

Remarque : Ctrl-Z fonctionne avec les shells Korn (**ksh**) et C (**cs**h), mais pas avec le shell Bourne (**bs**h).

Relance d'un process

Cette procédure explique comment relancer un process arrêté par Ctrl-Z.

Remarque : Ctrl-Z fonctionne avec les shells Korn (**ksh**) et C (**cs**h), mais pas avec le shell Bourne (**bs**h). Pour relancer un process arrêté, vous devez soit l'avoir lancé, soit être un utilisateur racine.

1. Pour afficher tous les process en cours ou arrêtés, mais non tués (commande kill), entrez :

```
ps -ef
```

Vous pouvez chaîner cette commande avec une commande **grep** pour mieux cibler les process qui vous intéressent. Ainsi, pour relancer une session vi, entrez :

```
ps -ef | grep vi
```

Cette commande restreint l'affichage généré par la commande **ps** aux lignes contenant le mot vi. Une sortie semblable à la suivante est affichée :

UID	PID	PPID	C	STIME	TTY	TIME	COMMAND
root	1234	13682	0	00:59:53	-	0:01	vi test
root	14277	13682	1	01:00:34	-	0:00	grep vi

2. Repérez le process que vous souhaitez relancer et notez son PID (1234 dans l'exemple).
3. Pour transmettre le signal de continuation au process arrêté, entrez :

```
kill -19 1234
```

N'oubliez pas de remplacer 1234 par le PID de votre process. -19 est le signal de CONTINUATION, qui relance le process en arrière-plan. Si ce mode d'exécution vous convient, la procédure est terminée. Si vous devez l'exécuter en avant-plan (comme dans une session vi), poursuivez la procédure.

4. Pour amener le process à l'avant-plan, entrez :

```
fg 1234
```

Là aussi, n'oubliez pas de remplacer 1234 par le PID de votre process. Celui-ci est maintenant exécuté en avant-plan. La session vi est ouverte.

Planification d'un process en différé (commande at)

Pour exécuter un process d'arrière-plan en différé, à une heure programmée, vous pouvez le définir comme *process par lots*. Cela peut s'effectuer par le biais des commandes **at** et **smit**.

Remarque : L'utilisation de la commande **at** dépend des fichiers **/var/adm/cron/at.allow** et **/var/adm/cron/at.deny**. Un utilisateur racine peut créer, éditer ou modifier ces fichiers. Leurs entrées comprennent un nom de connexion utilisateur par ligne. Voici un exemple de fichier **at.allow** :

```
racine
nick
dee
sarah
```

Si le fichier **at.allow** existe, seuls les utilisateurs dont le nom de connexion y est répertorié peuvent lancer la commande **at**. Un administrateur système peut explicitement limiter l'utilisation de la commande **at** en établissant la liste des noms de connexion utilisateur dans le fichier **at.deny**. Si le fichier **at.deny** existe, les utilisateurs dont le nom n'apparaît pas dans le fichier peuvent lancer la commande **at**.

Vous ne pouvez pas utiliser **at** dans les cas suivants :

- Les fichiers **at.allow** et **at.deny** n'existent pas (seuls les utilisateurs racine sont autorisés).
- **at.allow** existe mais le nom de connexion utilisateur n'y figure pas.
- **at.deny** existe et le nom de connexion utilisateur y figure.

Si **at.allow** et **at.deny** n'existent pas, ou que ce dernier est vide, seuls les utilisateurs racine peuvent soumettre un travail via **at**.

La commande **at** permet de spécifier une chaîne date, une chaîne date et heure, ou encore une chaîne incrément, pour définir le moment d'exécution du process. Elle vous autorise également à préciser le shell ou la file d'attente à utiliser. Voici quelques exemples.

Commande at

Si votre nom de connexion est joyce et que vous disposez d'un script appelé WorkReport que vous voulez exécuter à minuit, procédez comme suit.

1. Entrez l'heure à laquelle démarrer le programme :

```
at midnight
```

2. Entrez le nom des programmes à exécuter, en appuyant sur Entrée après chaque nom. Appuyez ensuite sur Ctrl-D (fin de fichier) pour terminer.

```
WorkReport^D
```

Un message semblable au suivant s'affiche :

```
job joyce.741502800.a at Fri Jul 6 00:00:00 CDT 1994.
```

Le programme WorkReport s'est vu attribuer le numéro de travail `joyce.741502800.a` et sera exécuté le 6 juillet à minuit.

Pour afficher la liste des programmes qui seront exécutés en différé, entrez :

```
at -l
```

Le système affiche un écran semblable à celui-ci :

```
joyce.741502800.a      Fri Jul 6 00:00:00 CDT 1994
```

Pour annuler l'exécution différée d'un programme, affichez la liste des numéros attribués à vos travaux, par la commande **at -l**. Notez le nom du programme à annuler, puis entrez :

```
at -r joyce.741502800.a
```

Le travail `joyce.741502800.a` est annulé.

Reportez-vous à la commande **at** dans le manuel *AIX Commands Reference*.

Les commandes **smit at** et **smit sjat** peuvent également exécuter cette tâche.

Liste des process planifiés

Pour afficher la liste des process planifiés, vous pouvez lancer la commande **at** assortie de l'indicateur **-l** ou la commande **atq**. Les deux génèrent la même sortie, la commande **atq** pouvant en outre trier les process selon l'heure d'émission de la commande **at** et n'afficher que les seuls process en file d'attente.

Pour afficher tous les process planifiés, vous lancez :

- la commande **at** à partir de la ligne de commande,
- la commande **atq**.

Pour plus de détails sur les limites d'utilisation de la commande **at**, reportez-vous aux Remarques, page 3-19.

Commande **at**

Pour afficher la liste des process planifiés, entrez :

```
at -l
```

Cette commande affiche la liste des process planifiés dans votre file d'attente. Si vous êtes utilisateur racine, elle affiche tous les process planifiés de tous les utilisateurs. Pour en savoir plus sur la syntaxe, reportez-vous à la commande **at**.

Commande **atq**

Pour afficher la liste des process planifiés de la file d'attente, entrez :

```
atq
```

Si vous êtes utilisateur racine, vous pouvez afficher les process planifiés d'un utilisateur en entrant :

```
atq NomUtilisateur
```

Pour afficher le nombre de process de la file d'attente, entrez :

```
atq -n
```

Déprogrammation d'un process

Pour déprogrammer un process, lancez la commande **at** assortie de l'indicateur **-r** ou le raccourci Web-based System Manager. Pour plus de détails sur les limites d'utilisation de la commande **at**, reportez-vous aux remarques, page 3-19.

A partir de la ligne de commande

1. Pour déprogrammer un process, vous devez connaître son numéro. Lancez pour ce faire la commande **at -l** ou **atq**. Reportez-vous à la section "Liste de tous les process planifiés".
2. Lorsque vous connaissez le numéro du process à déprogrammer, entrez :

```
at -r NuméroProcess
```

Web-based System Manager Raccourci

Vous pouvez utiliser le raccourci Web-based System Manager pour déprogrammer un process. A l'invite, tapez :

```
wsm processes
```

Dans le conteneur Processes, sélectionnez le process, puis vérifiez son état à l'aide des menus.

Vous pouvez également exécuter cette tâche à l'aide de la commande **smit rmat**.

Suppression d'un process d'arrière-plan (commande kill)

Si la touche d'INTERRUPTION n'a pas eu l'effet escompté sur le process d'avant-plan ou si vous souhaitez arrêter un process d'arrière-plan, vous pouvez recourir à la commande **kill**. Pour ce faire, il vous faut connaître le PID du process concerné. La syntaxe générale de la commande **kill** est la suivante :

```
kill IDProcess
```

Remarque : Pour supprimer un process, vous devez soit l'avoir lancé, soit être un utilisateur racine. Le signal par défaut lancé sur un process par la commande **kill** est **-15 (SIGTERM)**.

Commande kill

1. Pour déterminer le PID du process à supprimer, lancez la commande **ps**. Vous pouvez chaîner cette commande avec une commande **grep** pour mieux cibler les process qui vous intéressent. Ainsi, pour obtenir le PID d'une session vi, entrez :

```
ps -l | grep vi
```

2. Dans l'exemple suivant, la commande **find** a été émise en arrière-plan. Si vous décidez d'annuler le process, lancez la commande **ps** pour afficher la liste des PID.

```
$ find / -type f > dir.paths &
[1] 21593
$ ps
  PID  TTY  TIME  COMMAND
 1627 pts3  0:00  ps
 5461 pts3  0:00  ksh
17565 pts3  0:00  -ksh
21593 pts3  0:00  find / -type f
$ kill 21593
$ ps
  PID  TTY  TIME  COMMAND
 1627 pts3  0:00  ps
 5461 pts3  0:00  ksh
17565 pts3  0:00  -ksh
[1] + Terminated 21593      find / -type f > dir.paths &
```

La commande **kill 21593** arrête le process d'arrière-plan **find**, la seconde commande **ps** en renvoie aucune information d'état sur le PID 21593. Le système n'affiche le message de fin qu'à l'entrée de la commande suivante, sauf s'il s'agit de la commande **cd**.

La commande **kill** permet de supprimer un process d'arrière-plan : exécutez-la si vous avez, par erreur, lancé un process en arrière-plan ou que vous vous apercevez qu'il est trop long, par exemple.

Remarques :

- a. Supprimer un process parent supprime automatiquement tous ses process enfants.
- b. Pour supprimer un process zombie, vous devez supprimer son parent.

Reportez-vous à la commande **kill** dans le manuel *AIX Commands Reference*.

Web-based System Manager Raccourci

Vous pouvez utiliser le raccourci Web-based System Manager pour annuler un process actif. A l'invite, tapez :

```
wsm processes
```

Dans le conteneur Processes, sélectionnez le process, puis vérifiez son état à l'aide des menus.

Vous pouvez également exécuter cette tâche à l'aide de la commande **smit kill**.

Récapitulatif des commandes relatives aux commandes et aux process

Commandes

alias	Commande shell imprimant la liste des alias sur la sortie standard.
AUCUN LIEN history	Commande shell affichant l'historique des événements.
man	Affiche des informations sur les commandes, les sous-routines et les fichiers en ligne.
Web-based System Manager	
	Gère le système à partir d'un navigateur web.
whatis	Décrit l'objet d'une commande.
whereis	Localise les fichiers source, binaire ou les références de manuels des programmes installés.

Process

at	Diffère l'exécution d'une commande, répertorie les process programmés, ou déprogramme un process.
atq	Affiche la file d'attente des travaux en attente d'exécution.
kill	Envoie un signal aux process en cours.
nice	Modifie la priorité d'exécution d'une commande.
ps	Affiche l'état des process.
renice	Modifie la priorité d'un process en cours.

Chapitre 4. Réacheminement des entrées/sorties

Le système d'exploitation permet de gérer les entrées et les sorties (E/S) par le biais de commandes et de symboles spécifiques. Vous pouvez contrôler les entrées en indiquant où lire les données. Vous pouvez ainsi définir l'origine des entrées : clavier (entrée standard) ou fichier. Vous pouvez aussi contrôler la sortie en spécifiant l'emplacement d'affichage ou de stockage des données : par exemple, l'écran (sortie standard) ou un fichier.

Le système est multitâche, ce qui signifie qu'il traite les process en combinaison les uns avec les autres. Ce chapitre explicite les avantages du réacheminement des entrées et des sorties, et du regroupement des process.

Cette section traite des points suivants :

- Entrées, sorties et erreurs standard, page 4-2.
- Réacheminement des sorties standard, page 4-2.
- Réacheminement des sorties vers un fichier, page 4-2.
- Réacheminement des sorties avec ajout à un fichier, page 4-3.
- Création d'un fichier texte avec réacheminement à partir du clavier, page 4-3.
- Concaténation de fichiers texte, page 4-3.
- Réacheminement des entrées standard, page 4-3.
- Elimination des sorties via le fichier `/dev/null`, page 4-4.
- Réacheminement des erreurs standard et autres sorties, page 4-4.
- Documents entrée en ligne (Here), page 4-5.
- Pipes et filtres, page 4-5.
- Affichage de la sortie d'un programme avec copie dans un fichier (commande **tee**), page 4-6.
- Effacement de l'écran (commande **clear**), page 4-6.
- Envoi d'un message vers la sortie standard (commande **echo**), page 4-6.
- Ajout d'une ligne de texte à un fichier (commande **echo**), page 4-7.
- Copie de l'écran dans un fichier (commandes **capture** et **script**), page 4-7.
- Affichage de texte en gros caractères (commande **banner**), page 4-8.
- Récapitulatif des commandes relatives au réacheminement des entrées/sorties, page 4-9.

Entrées, sorties et erreurs standard

Lorsque qu'une commande est lancée, trois fichiers sont supposés ouverts : celui des entrées, celui des sorties et celui des erreurs standard (parfois appelé sortie erreurs ou sortie diagnostic). Un numéro, appelé *descripteur de fichier*, est associé à chaque fichier, comme suit :

Descripteur de fichier 0	Entrée standard.
Descripteur de fichier 1	Sortie standard.
Descripteur de fichier 2	Erreur (diagnostic) standard

Un process fils hérite normalement ces trois fichiers de son process parent. Ces fichiers sont, à l'origine, affectés à la station de travail (0 au clavier, 1 et 2 à l'écran). Le shell permet de les réacheminer avant de passer le contrôle à une commande.

Lorsque vous lancez une commande, le clavier est, par défaut, *l'entrée standard*, parfois référencée *stdin*. Le résultat de la commande est, par défaut, affiché à l'écran.

L'écran est en effet la *sortie standard*, parfois référencée *stdout*. . Par défaut, les commandes lisent les entrées à partir de l'entrée standard et envoient le résultat vers la sortie standard.

La sortie des erreurs standard, parfois référencée *stderr*, reçoit les messages d'erreur. L'écran constitue la sortie d'erreurs standard par défaut.

Ces options sont modifiables : vous pouvez désigner un fichier comme entrée et un autre comme destinataire des résultats. C'est l'objet de la fonction de *réacheminement des entrées/sorties*, une des caractéristiques les plus intéressantes d'un système UNIX.

Le résultat d'une commande, normalement dirigé vers le terminal, peut être facilement dévié vers un fichier : il s'agit du *réacheminement des sorties*. Cette fonction est très utile lorsque le résultat d'une commande est trop long pour être aisément lu à l'écran ou que vous souhaitez regrouper plusieurs petits fichiers dans un seul.

Moins utilisé, le réacheminement des entrées (normalement lues à partir du clavier) permet de créer un fichier à l'avance et d'y lire les données. il s'agit du *réacheminement des entrées*. Le réacheminement des entrées vous permet de préparer un fichier à l'avance puis que la commande lise le fichier.

Réacheminement des sorties standard

Lorsque la mention *nomfichier* est ajoutée à la fin d'une commande, le résultat de la commande est inscrit dans le fichier spécifié. Le symbole > est appelé *opérateur de réacheminement des sorties*.

Toute commande dont les résultats sont normalement affichés à l'écran peut être réacheminée de la sorte.

Réacheminement des sorties vers un fichier

Par exemple, pour envoyer les résultats de la commande **who** dans le fichier `users`, entrez :

```
who > users
```

Remarque : Si le fichier `users` existe déjà, il est écrasé, sauf si vous avez spécifié l'option **noclobber** de la commande intégrée **set** du ksh (shell Korn) ou csh (shell C).

Pour afficher le contenu du fichier `users`, entrez :

```
cat users
```

Un écran semblable au suivant s'affiche :

```
denise    lft/0 May 13 08:05
marta    pts/1 May 13 08:10
endrica   pts/2 May 13 09:33
```

Pour envoyer la liste du répertoire courant vers un fichier, entrez :

```
ls > dirlist
```

Réacheminement des sorties avec ajout à un fichier

Lorsque la notation `> > nomfichier` est ajoutée à la fin d'une commande, ses résultats sont ajoutés au contenu d'un fichier au lieu d'en écraser les données. Le symbole `> >` est appelé *opérateur de réacheminement avec ajout*.

Par exemple, pour ajouter `fichier2` à `fichier1`, entrez :

```
cat fichier2 > > fichier1
```

Remarque : Si le fichier `fichier1` n'existe pas, il est créé, sauf si vous avez spécifié l'option **noclobber** de la commande intégrée **set** du ksh (shell Korn) ou csh (shell C).

Création d'un fichier texte avec réacheminement à partir du clavier

La commande **cat** isolée récupère en entrée tout ce que vous tapez au clavier. Vous pouvez rediriger ces entrées vers un fichier. Tapez Ctrl-D (sur une nouvelle ligne) pour signaler la fin du texte.

A l'invite, entrez :

```
cat > nomfichier
Ceci est un test.
^D
```

Concaténation de fichiers texte

La *concaténation* consiste à regrouper des fichiers dans un seul.

Par exemple, à l'invite système, entrez :

```
cat fichier1 fichier2 fichier3 > fichier4
```

Dans cet exemple, `fichier4` est créé, constitué de `fichier1`, `fichier2` et `fichier3`.

Voici une erreur à ne pas commettre :

```
cat fichier1 fichier2 fichier3 > fichier1
```

Avertissement : Dans cet exemple, on pourrait croire que la commande **cat** regroupe le contenu de `fichier1`, `fichier2` et `fichier3` dans `fichier1`. En fait, cette commande commence par créer le fichier de sortie, `fichier1`, en écrasant son contenu, puis y insère le contenu de `fichier2` et `fichier3`.

Réacheminement des entrées standard

Lorsque la mention `< nomfichier` est ajoutée à la fin d'une commande, l'entrée de la commande est lue à partir du fichier spécifié. Le symbole `<` est appelé *opérateur de réacheminement des entrées*.

Remarque : Seules les commandes lisant normalement leurs entrées à partir du clavier peuvent faire l'objet d'un réacheminement de ce type.

Par exemple, pour envoyer le fichier `lettre1` comme message à l'utilisateur `denise`, via la commande **mail**, entrez :

```
mail denise < lettre1
```

Élimination des sorties via le fichier `/dev/null`

Le fichier `/dev/null` est un fichier spécial. Le fichier a une particularité puisqu'il est déjà vide. Toutes les données que vous y envoyez sont supprimées. Ceci est une fonction utile lorsque vous exécutez un programme dont les sorties n'ont aucun intérêt pour vous.

Soit, par exemple, le programme `myprog`, qui lit les données à l'écran et renvoie des messages que vous souhaitez ignorer. Pour qu'il lise les entrées dans le fichier `monscript` et rejette les messages de sortie standard, entrez :

```
monprog < monscript >/dev/null
```

`monprog` lit désormais les entrées dans `myscript` et rejette toutes les sorties standard.

Réacheminement des erreurs standard et autres sorties

Outre les entrées et sorties standard, les commandes génèrent souvent d'autres sorties, telles que des messages d'erreur ou d'état (appelés diagnostics). Ces sorties sont, comme les sorties standard, envoyées à l'écran, sauf si vous les réacheminez.

Généralement, au lancement d'une commande, trois fichiers sont ouverts : **stdin** (entrée standard), **stdout** (sortie standard) et **stderr** (erreurs standard). Pour réacheminer les entrées ou les sorties standard, vous disposez des symboles `<`, `>` ou `>>`. Pour réacheminer les erreurs standard ou d'autres sorties, vous devez spécifier un descripteur de fichier. Vous pouvez également spécifier des descripteurs de fichiers pour réacheminer les entrées et les sorties standard, mais ils sont déjà attribués par défaut.

Un *descripteur de fichier* est un numéro associé à chaque fichier d'E/S utilisé habituellement par une commande. Les descripteurs associés aux entrées, aux sorties et aux erreurs standard sont les suivants :

0	Entrée standard (clavier)
1	Sortie standard (écran)
2	Erreur standard (écran)

Pour réacheminer les erreurs standard, spécifiez le descripteur de fichier 2 devant le symbole de réacheminement (`>` ou `>>`) et un nom de fichier après le symbole. Par exemple, pour ajouter la sortie d'erreurs standard de la commande **cc** utilisée pour compiler `fichier.c` à la fin du fichier `ERRORS`, entrez :

```
cc fichier.c 2> > ERRORS
```

Les autres types de sorties peuvent également être réacheminées, par le biais des descripteurs de fichier 0 à 9. Par exemple, si la commande **cmd** envoie ses sorties vers le descripteur de fichier 9, vous pouvez les réacheminer vers le fichier `sauvedonnées` via la commande :

```
cmd 9> sauvedonnées
```

Si la commande génère des sorties envoyées vers plusieurs destinations, vous pouvez les réacheminer indépendamment les unes des autres. Soit une commande envoyant ses sorties standard vers le descripteur de fichier 1, ses erreurs standard vers le descripteur de fichier 2 et qui construit un fichier de données sur le descripteur de fichiers 9. Pour réacheminer chaque sortie vers un fichier distinct, entrez :

Documents entrée en ligne (Here)

Avec une commande de la forme :

```
commande << chaîneeof
```

où *chaîneeof* est une chaîne quelconque ne contenant pas de caractères joker, le shell considère les lignes qui suivent comme entrée standard de *commande* jusqu'à ce qu'il rencontre une ligne ne contenant que *chaîneeof* (éventuellement précédé d'une ou de plusieurs tabulations). Les lignes entre la première occurrence de *chaîneeof* et la seconde sont souvent appelées document *d'entrée en ligne* ou document *Here*. Si un signe moins (-) suit immédiatement les caractères de réacheminement <<, le shell supprime les tabulations au début de chaque ligne avant de passer la ligne à la *commande*.

Le shell crée un fichier temporaire contenant le document Here et ??? y effectue des substitutions de variables et de commandes avant de passer le fichier à la commande. Il vérifie les noms de fichier qui font partie des lignes de commande dans les substitutions de commandes. Pour interdire toute substitution, marquez un caractère de *chaîneeof* :

```
commande << \chaîneeof
```

Le document Here est particulièrement utile pour un faible volume de données, qu'il est plus judicieux de placer directement dans une procédure shell plutôt que dans un fichier distinct (scripts d'édition, par exemple). Vous pouvez, par exemple, entrer :

```
cat <<- xyz
  Ce message sera affiché à
  l'écran sans tabulations en tête.
  xyz
```

Cette fonction est très utile dans les procédures shell.

Pipes et filtres

UNIX permet de chaîner des commandes de sorte que la sortie standard de la première soit l'entrée standard de la suivante, et ainsi de suite. Une série de commandes ainsi chaînées est appelée un *pipeline*. L'opération reliant les commandes est appelée *chaînage (pipe)*. Il s'agit d'une fonction essentielle d'UNIX, qui permet de créer une commande complexe et puissante, à partir de plusieurs commandes simples.

Vous pouvez vous servir d'un pipeline pour définir la sortie d'une commande comme entrée d'une autre. Séparez les commandes concernées par le symbole | (barre verticale).

Une commande qui traite le résultat d'une autre et envoie le résultat final vers la sortie standard, est appelée un *filtre*. Les filtres peuvent être utilisés isolément, mais sont particulièrement utiles dans les pipelines. Les plus courants sont :

- sort
- more
- pg

Par exemple, la commande **ls** affiche le contenu du répertoire courant en un flot ininterrompu. Si les informations ne tiennent pas sur un écran, certaines ne sont pas visibles. Pour obtenir un affichage page par page, chaînez les commandes **ls** et **pg** :

```
ls | pg
```

La sortie de la commande **ls** devient l'entrée de la commande **pg**. Pour afficher l'écran suivant, appuyez sur Entrée.

Les pipelines fonctionnent dans une seule direction (de gauche à droite). Chaque commande est exécutée comme un process distinct, tous les process pouvant être

exécutés simultanément. Un process est suspendu lorsqu'il n'a plus de données à lire ou que le tube (pipe) vers le process suivant est plein.

La commande **grep** recherche dans un fichier les lignes comportant une chaîne de caractères donnée. Pour afficher tous les fichiers créés ou modifiés en juillet (July), entrez :

```
ls -l | grep Jul
```

La sortie de la commande **ls** est ici l'entrée de la commande **grep**.

Affichage de la sortie d'un programme avec copie dans un fichier (commande tee)

La commande **tee**, dans le cadre d'un chaînage, lit l'entrée standard d'un programme, puis envoie le résultat vers la sortie standard et, simultanément, le copie dans le ou les fichiers spécifiés. Vous avez ainsi la possibilité de visualiser immédiatement la sortie tout en la conservant dans un fichier pour un usage ultérieur.

Ainsi, pour visualiser et sauvegarder simultanément le résultat d'une commande, entrez :

```
ps -ef | tee program.ps
```

La sortie standard de la commande **ps -ef** est affichée sur la station de travail tandis qu'une copie de cette sortie est enregistrée dans le fichier `program.ps`. Si ce fichier existe déjà, il est écrasé, sauf si vous avez spécifié l'option **noclobber** de la commande intégrée **set**.

Pour visualiser et sauvegarder simultanément le résultat d'une commande dans un fichier existant, entrez :

```
ls -l | tee -a program.ls
```

La sortie standard de la commande **ls -l** est affichée sur la station de travail tandis qu'une copie de cette sortie est ajoutée à la fin du fichier `program.ls`. Si ce fichier n'existe pas, il est créé, sauf si vous avez spécifié l'option **noclobber** de la commande intégrée **set**.

Le système affiche des informations semblables aux suivantes, les mêmes informations étant enregistrées dans le fichier `program.ls` :

```
-rw-rw-rw- 1 jones  staff  2301  Sep 19  08:53 161414
-rw-rw-rw- 1 jones  staff  6317  Aug 31  13:17 def.rpt
-rw-rw-rw- 1 jones  staff  5550  Sep 10  14:13 try.doc
```

Reportez-vous à la commande **tee** dans le manuel *AIX Commands Reference*.

Effacement de l'écran (commande clear)

La commande **clear** efface messages et entrées clavier affichés.

À l'invite, entrez :

```
clear
```

Le système efface l'écran et affiche l'invite.

Reportez-vous à la commande **clear** dans le manuel *AIX Commands Reference*.

Envoi d'un message vers la sortie standard (commande echo)

La commande **echo** permet d'afficher des messages à l'écran.

Par exemple, pour inscrire un message sur la sortie standard, entrez, à l'invite :

```
echo Insérez une disquette SVP . . . . .
```

Le système affiche :

```
Insérez une disquette SVP. . . .
```

Pour combiner commande **echo** et caractères joker, entrez, à l'invite :

```
echo Les fichiers-de sauvegarde sont : *.bak
```

Le système affiche le message Les fichiers de sauvegarde sont : suivi des fichiers suffixés `.bak` du répertoire courant.

Reportez-vous à la commande **echo** dans le manuel *AIX Commands Reference*.

Ajout d'une ligne de texte à un fichier (commande echo)

La commande **echo**, associée à un opérateur de réacheminement avec ajout, permet d'ajouter une ligne de texte à un fichier.

Par exemple, entrez, à l'invite :

```
echo Ne pas oublier la sauvegarde hebdomadaire>  
  
>notes
```

Le message `Ne pas oublier la sauvegarde hebdomadaire` est inséré à la fin du fichier `notes`.

Reportez-vous à la commande **echo** dans le manuel *AIX Commands Reference*.

Copie de l'écran dans un fichier (commandes capture et script)

La commande **capture**, qui émule un terminal VT100, permet d'effectuer une copie d'écran dans un fichier.

La commande **script** a la même fonction, à ceci près qu'elle n'émule pas de terminal VT100.

Les deux commandes permettent d'obtenir des copies papier des dialogues au terminal.

Par exemple, pour capturer l'écran d'un terminal et émuler un VT100, entrez, à l'invite :

```
capture screen.01
```

Le système affiche un écran semblable à celui-ci :

```
Commande capture lancée. Fichier : screen.01.  
Utilisez ^P pour obtenir un cliché d'écran vers le fichier  
screen.01.  
Emulation d'un terminal vt100.  
Pour continuer, appuyez sur une touche.
```

Une fois les données entrées et le contenu de l'écran copié (via un cliché), interrompez la commande **capture** en appuyant sur Ctrl-D ou en entrant `exit`. Le système affiche un écran semblable à celui-ci :

```
Commande capture exécutée. Fichier : screen.01.  
Emulation d'un terminal vt100 TERMINEE.
```

Pour afficher le contenu du fichier, lancez la commande **cat**.

Par exemple, pour capturer l'écran d'un terminal, entrez, à l'invite :

```
script
```

Le système affiche un écran semblable à celui-ci :

```
Commande script lancée sur le fichier typescript.
```

Tout ce qui est affiché à l'écran est copié dans le fichier `typescript`.

Pour interrompre la commande **script**, appuyez sur Ctrl-D ou entrez `exit`. Le système affiche un écran semblable à celui-ci :

Commande script exécutée sur le fichier typescript.

Pour afficher le contenu du fichier, lancez la commande **cat**.

Reportez-vous aux commandes **capture** et **script** dans le manuel *AIX Commands Reference*.

Affichage de texte en gros caractères (commande banner)

La commande **banner** affiche du texte ASCII en gros caractères. Chaque ligne ne peut comporter plus de 10 chiffres ou caractères (majuscules ou minuscules).

Par exemple, entrez, à l'invite :

```
banner SALUT !
```

Le message `SALUT !` s'affiche en grand sur l'écran.

Reportez-vous à la commande **banner** dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives au réacheminement des entrées/sorties

>	Réachemine les sorties standard vers un fichier.
<	Définit un fichier comme source des entrées standard.
> >	Ajoute les sorties standard à un fichier.
	Regroupe des process dans un pipeline.
banner	Affiche des textes ASCII en gros caractères sur la sortie standard.
capture	Effectue une copie des écrans du terminal dans un fichier.
clear	Efface l'écran du terminal.
echo	Ecrit les chaînes de caractères sur la sortie standard.
script	Enregistre dans un fichier les entrées/sorties du terminal.
tee	Affiche la sortie standard d'un programme et la copie dans un fichier.

Chapitre 5. Systèmes de fichiers et répertoires

Un *système de fichiers* est constitué d'un groupe de répertoires et des fichiers qu'ils contiennent. Il est généralement représenté par une arborescence : le répertoire racine, symbolisé par une barre oblique(/), définit un système de fichiers et apparaît au sommet de l'arborescence. Les branches descendantes, issues du répertoire racine, contiennent des fichiers ou des sous-répertoires. Ces ramifications définissent des chemins d'accès uniques à tous les objets du système de fichiers.

Les fichiers sont regroupés dans des *répertoires*. Ces derniers sont rarement indépendants les uns des autres : les structurer en système de fichiers permet de les organiser.

Un *fichier* est un ensemble de données, susceptible d'être lu et dans lequel il est possible d'écrire. Un programme, un texte, des données, une unité, etc., constituent des fichiers. Commandes, imprimantes, terminaux, courrier et programmes d'application y sont stockés. L'accès à ces divers éléments est de ce fait uniformisé : le système y gagne en souplesse et en simplicité.

Ce chapitre traite des points suivants :

- Système de fichiers, page 5-2.
 - Types de systèmes de fichiers, page 5-2.
 - Structure des systèmes de fichiers, page 5-2.
 - Espace disponible sur un système de fichiers (commande **df**), page 5-4.
- Répertoires : généralités, on page 5-5.
 - Types de répertoires, page 5-5.
 - Organisation des répertoires, page 5-6.
 - Conventions d'appellation des répertoires, page 5-6.
 - Chemins d'accès aux répertoires, page 5-6.
 - Abréviations propres aux répertoires, page 5-7.
- Procédures de gestion des répertoires, page 5-8.
 - Création d'un répertoire (commande **mkdir**), page 5-8.
 - Déplacement ou changement de nom d'un répertoire (commande **mvdir**), page 5-8.
 - Affichage du répertoire courant (commande **pwd**), page 5-9.
 - Passage à un autre répertoire (commande **cd**), page 5-9.
 - Copie d'un répertoire (commande **cp**), page 5-9.
 - Affichage du contenu d'un répertoire (commandes **li** et **ls**), page 5-10.
 - Suppression ou retrait d'un répertoire (commande **rmdir**), page 5-12.
 - Comparaison entre répertoires (commande **dircmp**), page 5-13.
- Récapitulatif des commandes relatives aux systèmes de fichiers et aux répertoires, page 5-14.

Systemes de fichiers

Un *systeme de fichiers* est une structure hiérarchique (arborescence de fichiers) de fichiers et de répertoires. Ce type de structure ressemble à un arbre renversé avec les racines au sommet et les branches en bas. Données et programmes y sont organisés en groupes via des répertoires, ce qui permet de manipuler simultanément plusieurs fichiers et répertoires.

Certaines tâches sont exécutées de façon plus efficace sur un système de fichiers que sur chaque répertoire d'un système de fichiers. Par exemple, vous pouvez sauvegarder, déplacer ou protéger tout un système de fichiers.

Le système de fichiers de base s'appelle *systeme de fichiers journalisé*. Il exploite les techniques de journalisation de base de données pour maintenir sa cohérence structurelle, ce qui prévient tout risque d'altération en cas d'arrêt anormal du système.

Nombre de tâches de gestion du système sont liées aux systèmes de fichiers, notamment :

- l'affectation d'espace aux systèmes de fichiers sur les volumes logiques,
- la création de systèmes de fichiers,
- la mise à la disposition des utilisateurs de l'espace du système de fichiers,
- le contrôle de l'utilisation de l'espace du système de fichiers,
- la sauvegarde des systèmes de fichiers pour protéger les données en cas de panne du système,
- le maintien de la cohérence des systèmes de fichiers.

Ces tâches incombent à l'administrateur du système.

Ce chapitre traite des points suivants :

- Types de systèmes de fichiers, page 5-2.
- Structure des systèmes de fichiers, page 5-2.
- Espace disponible sur un système de fichiers (commande **df**), page 5-4.

Types de systemes de fichiers

AIX prend en charge plusieurs types de systèmes de fichiers. Par exemple :

Systeme de fichiers journalisé	Type de base. Il accepte l'ensemble des commandes relatives aux systèmes de fichiers.
Systeme de fichiers réseau (NFS)	Type de système de fichiers permettant d'accéder aux fichiers résidant sur des machines distantes comme s'il s'agissait de fichiers locaux.
Systeme de fichiers CD-ROM	Type de système de fichiers permettant d'accéder au contenu d'un CD-ROM via les interfaces habituelles du système (ouverture, lecture et fermeture).

Structure des systemes de fichiers

Sur les systèmes autonomes, les systèmes de fichiers suivants résident, par défaut, sur les unités associées :

/dev/hd1/	/home
/dev/hd2/	/usr
/dev/hd3/	/tmp
/dev/hd4/	/(racine)
/dev/hd9var	/var

L'arborescence des fichiers présente les caractéristiques suivantes :

- Les fichiers partageables par les machines de la même architecture matérielle se trouvent dans le système de fichiers **/usr**.
- Les fichiers texte partageables, indépendants de l'architecture (pages de manuel, par exemple) se trouvent dans le répertoire **/var**.
- Le système de fichiers **/(racine)** contient les fichiers et les répertoires essentiels à l'exploitation du système. Par exemple :
 - répertoire des unités (**/dev**)
 - points de montage des systèmes de fichiers sur le système de fichiers racine, **/mnt**, par exemple.
- Le système de fichiers **/home** est le point de montage des répertoires personnels de l'utilisateur.
- Pour les serveurs, le répertoire **/export** contient les fichiers de pagination d'espace, les systèmes de fichiers racine client (non partagés), les répertoires de vidage, personnels et **/usr/share** des clients sans disque, ainsi que les répertoires **/usr** exportés.

La liste suivante propose des informations sur le contenu de certains sous-répertoires du système de fichiers / (racine).

/bin	Lien symbolique au répertoire /usr/bin . Dans les anciens systèmes de fichiers UNIX, le répertoire /bin contenait les commandes utilisateur qui résident désormais dans le répertoire /usr/bin .
/dev	Contient les noeuds unité pour les fichiers spéciaux des unités locales. Le répertoire /dev contient les fichiers spéciaux des unités de bande, des imprimantes, des partitions de disque et des terminaux.
/etc	Contient les fichiers de configuration propres à chaque machine. Exemples : <ul style="list-style-type: none">• /etc/hosts• /etc/passwd <p>Le répertoire /etc contient les fichiers généralement utilisés pour l'administration du système. La plupart des commandes qui résidaient dans le répertoire /etc résident désormais dans le répertoire /usr/sbin. Toutefois, pour des raisons de compatibilité, il contient des liens symboliques aux nouveaux emplacements de certains des fichiers exécutables. Exemples :</p> <ul style="list-style-type: none">• /etc/chown est un lien symbolique à /usr/bin/chown.• /etc/exportvg est un lien symbolique à /usr/sbin/exportvg.
/export	Contient les répertoires et les fichiers d'un serveur, réservés aux clients distants.
/home	Sert de point de montage au système de fichiers contenant les répertoires personnels de l'utilisateur. Le système de fichiers /home contient les fichiers et les répertoires utilisateur. Sur les machines autonomes, un système de fichiers distinct est monté sur le répertoire /home . Dans un réseau, un serveur peut contenir des fichiers utilisateur qui doivent être accessibles par plusieurs machines. Dans ce cas, la copie serveur du répertoire /home est télémontée sur un système de fichiers /home local.
/lib	Lien symbolique au répertoire /usr/lib qui comprend les bibliothèques indépendantes de l'architecture au format lib*.a .

/sbin	Contient les fichiers requis pour l'amorçage de la machine et le montage du système de fichiers /usr . La plupart des commandes utilisées au moment de l'amorçage proviennent du système de fichiers du disque RAM de l'image d'amorçage. De ce fait, très peu de commandes se trouvent dans le répertoire /sbin .
/tmp	Sert de point de montage du système de fichiers contenant les fichiers temporaires générés par le système.
/u	Lien symbolique au répertoire /home .
/usr	Sert de point de montage au système de fichiers contenant les fichiers invariables partageables par les machines (fichiers exécutables et documentation ASCII, par exemple).
/var	Sert de point de montage des fichiers variables (dépendant de la machine). Le système de fichiers /var est configuré comme un système de fichiers dans la mesure où les fichiers qu'il contient tendent à grossir. Par exemple, c'est un lien symbolique au répertoire /usr/tmp qui contient les fichiers de travail temporaires.

Espace disponible sur un système de fichiers (commande df)

La commande **df** affiche des informations sur l'espace total et l'espace disponible sur un système de fichiers. Spécifiez, dans le paramètre *SystèmeFichier*, le nom de l'unité où réside le système de fichiers, le répertoire sur lequel il est monté ou son chemin d'accès relatif. A défaut, la commande affiche des informations sur tous les systèmes de fichiers montés. Si vous indiquez un fichier ou un répertoire, elle affiche des informations sur le système de fichiers sur lequel réside le fichier.

Normalement, la commande **df** se sert des comptes libres du superbloc. Dans certaines circonstances exceptionnelles, ces comptes peuvent être en erreur. C'est ce qui se produit si un système de fichiers est activement modifié pendant l'exécution de la commande **df**, par exemple.

Reportez-vous à la commande **df** dans le manuel *AIX Commands Reference*.

Remarque : Sur certains systèmes de fichiers distants, tels que les systèmes de fichiers réseau (NFS), si le serveur ne fournit pas d'informations, les colonnes sont vides.

Par exemple, pour afficher des informations sur tous les systèmes de fichiers montés, entrez :

```
df
```

Si votre système est configuré de sorte que les répertoires **/usr**, **/site** et **/usr/venus** résident sur des systèmes de fichiers distincts, la commande **df** génère un résultat semblable au suivant :

Système fichiers	512-blocs	disponibles	%used	iused	%iused	Monté sur
/dev/hd4	20480	13780	32%	805	13%	/
/dev/hd2	385024	15772	95%	27715	28%	/usr
/dev/hd9var	40960	38988	4%	115	1%	/var
/dev/hd3	20480	18972	7%	81	1%	/tmp
/dev/hd1	4096	3724	9%	44	4%	/home

Par exemple, pour afficher l'espace disponible sur le système de fichiers où réside votre répertoire courant, entrez :

```
df .
```

Répertoires : généralités

Un *répertoire* est un fichier qui ne contient que les informations d'accès à d'autres fichiers et répertoires. Par conséquent, il occupe moins d'espace que les autres types de fichiers. Organiser le système par le biais de répertoires lui confère souplesse et puissance : les fichiers et répertoires peuvent en effet être regroupés pour créer une hiérarchie modulaire. A la différence des autres types de fichiers, les répertoires ont leur propre jeu de commandes.

Les répertoires contiennent des entrées. Chacune d'entre elles contient le nom et le numéro d'*i-node* (index node) du fichier. Pour optimiser l'utilisation de l'espace disque, les données d'un fichier sont réparties sur plusieurs zones de la mémoire. L'*i-node* indique les adresses et l'enchaînement de tous les blocs de données associés à un fichier. L'*i-node* enregistre également d'autres informations : date et heure de la dernière ouverture/modification, droits d'accès, nombre de liens, propriétaire, type de fichier, etc. Vous pouvez, via la commande **In**, associer plusieurs noms d'un fichier à un même *i-node*.

Les répertoires contenant souvent des informations réservées à certains utilisateurs, l'accès aux répertoires peut être protégé. Vous pouvez, en définissant des droits d'accès, contrôler et limiter l'accès aux répertoires, et désigner les utilisateurs autorisés à en modifier le contenu. Pour en savoir plus, reportez-vous à la section "Modes d'accès aux fichiers et aux répertoires", page 9-4.

Ce chapitre traite des points suivants :

- Types de répertoires, page 5-5.
- Organisation des répertoires, page 5-6.
- Conventions d'appellation des répertoires, page 5-6.
- Chemins d'accès aux répertoires, page 5-6.
- Abréviations propres aux répertoires, page 5-7.
- Procédures de gestion des répertoires, page 5-8.

Types de répertoires

Les répertoires peuvent être définis par le système d'exploitation, l'administrateur ou les utilisateurs. Les répertoires système contiennent des fichiers système particuliers (commandes, par exemple). Le répertoire **/**(**racine**), défini par le système, constitue le sommet de la hiérarchie du système de fichiers. Il est en général composé des répertoires suivants :

/dev	fichiers spéciaux pour unités d'E/S.
/etc	fichiers d'initialisation et d'administration du système.
/home	répertoires de connexion des utilisateurs du système.
/tmp	fichiers temporaires, supprimés à l'issue d'un nombre de jours donné.
/usr	répertoires lpp , include et programmes système.
/usr/bin	programmes exécutables utilisateur.

Certains répertoires, comme le répertoire de connexion ou le répertoire personnel (**\$HOME**), sont définis et personnalisés par l'administrateur. Lorsque vous vous connectez au système, vous vous trouvez dans le répertoire de connexion (qui est alors le répertoire courant).

Les répertoires que vous créez sont appelés des répertoires utilisateur. Ils vous permettent d'organiser et de maintenir vos fichiers.

Organisation des répertoires

Les répertoires contiennent des fichiers, des sous-répertoires ou une combinaison des deux. Un sous-répertoire est un répertoire contenu dans un autre, qui est alors son parent.

Pour que le système puisse localiser chaque répertoire, ceux-ci contiennent une entrée relative à leur répertoire parent, .. (point point), et une entrée pour le répertoire lui-même, . (point). Dans la plupart des listes de répertoires, ces fichiers sont cachés.

Arborescence des répertoires

Les structures composées de répertoires parents, de sous-répertoires et de fichiers sont des *systèmes de fichiers*. Ces structures sont souvent comparées à un arbre à l'envers. Le répertoire racine, symbolisé par une barre oblique (/), constitue la base et est représenté en haut de l'arborescence. Les sous-répertoires et les fichiers représentent les branches.

La structure d'un système de fichiers peut rapidement devenir complexe : essayez de ne pas la surcharger. Veillez en outre à donner aux fichiers et aux répertoires des noms "parlants" : votre travail en sera facilité.

Répertoire parent

Chaque répertoire, à l'exception du répertoire /(racine), a un parent et peut avoir un ou plusieurs enfants. Dans l'exemple illustré, **C** est le parent de **E**, et **C** l'enfant de /(racine).

Répertoire personnel

Lorsque vous vous connectez au système, vous vous trouvez dans votre *répertoire personnel* (répertoire de connexion). Il est défini par l'administrateur pour chaque utilisateur. C'est dans ce répertoire que vous conservez vos fichiers. Normalement, les répertoires que vous créez pour votre usage personnel sont des sous-répertoires de votre répertoire personnel. Pour revenir à tout moment à votre répertoire personnel, entrez la commande **cd** à l'invite.

Répertoire de travail

Vous travaillez toujours dans le cadre d'un répertoire. Ce répertoire, quel qu'il soit, est appelé répertoire *courant* ou *de travail*. La commande **pwd** (present working directory) affiche le nom du répertoire courant. La commande **cd** permet de passer d'un répertoire de travail à un autre.

Conventions d'appellation des répertoires

Le nom d'un répertoire doit être unique dans le répertoire qui le contient. Ainsi, à chaque répertoire ne correspond qu'un seul chemin d'accès dans le système de fichiers. Les noms de répertoires suivent les mêmes conventions que les fichiers, explicitées à la section "Conventions d'appellation des fichiers", page 6-3.

Chemins d'accès aux répertoires

L'accès à chaque fichier ou répertoire d'un système de fichiers se fait par un chemin unique, appelé *chemin d'accès*, qui en indique l'emplacement. Le chemin d'accès spécifie l'emplacement d'un répertoire ou d'un fichier dans le système de fichiers.

Remarque : Les chemins d'accès ne peuvent dépasser 1023 caractères.

Deux types de chemins d'accès sont associés au système de fichiers :

chemin d'accès absolu	chemin à partir du répertoire /(racine). Les chemins d'accès absolus commencent toujours par le signe / (barre oblique).
chemin d'accès relatif	chemin à partir du répertoire courant vers les sous-répertoires et fichiers en aval.

Un chemin d'accès absolu est constitué de la liste complète des noms de répertoire ou de fichier situés en amont, à partir du répertoire **/ (racine)**. Quel que soit l'endroit où vous vous trouvez dans le système de fichiers, vous pouvez atteindre un répertoire ou un fichier en spécifiant son chemin d'accès absolu. Ce chemin doit obligatoirement commencer par le signe / (barre oblique), symbole du répertoire racine. Le chemin d'accès **/A/D/9** est le chemin absolu vers **9**. La première / (barre oblique) représente le répertoire **/ (racine)**, point de départ de la recherche. Le reste du chemin oriente la recherche vers **A**, puis **D** et enfin **9**.

Il existe deux fichiers de même nom (**9**). Cela est possible car le chemin d'accès absolu donne à chaque fichier un nom unique à l'intérieur du système de fichiers. Les chemins **/A/D/9** et **/C/E/G/9** spécifient deux fichiers distincts appelés (**9**).

En revanche, les chemins d'accès relatifs indiquent l'accès à un répertoire ou à un fichier basé sur le répertoire de travail courant. Pour ce type de chemin d'accès, vous pouvez utiliser la notation **..** (point point) pour remonter dans la hiérarchie du système de fichiers. Ces deux points successifs représentent le répertoire parent. De ce fait, ils ne commencent pas par le signe / (barre oblique). Ils donnent accès à un fichier du répertoire courant ou à un fichier ou répertoire situé en amont ou en aval du répertoire courant dans le système de fichiers. Si **D** est le répertoire courant, le chemin relatif d'accès à **10** est **F/10**, mais le chemin absolu est toujours **/A/D/F/10**. De même, le chemin d'accès relatif à **3** est **../B/3**.

Vous pouvez également utiliser la notation **.** (point). La notation par point est couramment employée pour exécuter des programmes qui doivent lire le nom du répertoire courant.

Abréviations propres aux répertoires

Les abréviations constituent un moyen simple pour spécifier certains répertoires. En voici la liste.

Abréviation	Signification
.	Répertoire de travail courant
..	Répertoire "au-dessus" (parent) du répertoire de travail courant.
~	Répertoire personnel (inexact pour le shell Bourne).
\$HOME	Répertoire personnel (valable pour tous les shells).

Procédures de gestion des répertoires

Nombre d'opérations peuvent être exécutées sur les répertoires et leur contenu.

Cette section présente les différentes commandes, avec des exemples:

- Création d'un répertoire (commande **mkdir**), page 5-8.
- Déplacement ou changement de nom d'un répertoire (commande **mvdir**), page 5-8.
- Affichage du répertoire courant (commande **pwd**), page 5-9.
- Passage à un autre répertoire (commande **cd**), page 5-9.
- Copie d'un répertoire (commande **cp**), page 5-9.
- Affichage du contenu d'un répertoire (commandes **li** et **ls**), page 5-10.
- Suppression ou retrait d'un répertoire (commande **rmdir**), page 5-12.
- Comparaison entre répertoires (commande **dircmp**), page 5-13.

Création d'un répertoire (commande **mkdir**)

La commande **mkdir** crée un ou plusieurs répertoires, spécifiés par le paramètre *répertoire*. Chaque répertoire ainsi créé contient les entrées standard **.** (point) et **..** (point point). Vous pouvez préciser des droits d'accès au répertoire via l'indicateur **-m mode**.

Tout répertoire est créé à l'intérieur du répertoire courant (de travail), sauf si vous spécifiez un autre emplacement via un chemin d'accès absolu.

Par exemple, pour créer un répertoire appelé `Test` dans le répertoire courant, doté des droits d'accès par défaut, entrez :

```
mkdir Test
```

Pour créer un répertoire appelé `Test` doté des droits `rxwxr-xr-x` dans le répertoire `/home/demo/sub1`, entrez :

```
mkdir -m 755 /home/demo/sub1/Test
```

Pour créer un répertoire appelé `Test`, doté des droits d'accès par défaut, dans le répertoire `/home/demo/sub2`, entrez :

```
mkdir -p /home/demo/sub2/Test
```

L'indicateur **-p** crée les répertoires `/home/`, `/home/demo` et `/home/demo/sub2` s'ils n'existent pas.

Reportez-vous à la commande **mkdir** dans le manuel *AIX Commands Reference*.

Déplacement ou changement de nom d'un répertoire (commande **mvdir**)

La commande **mvdir** déplace ou renomme des répertoires.

Par exemple, pour déplacer un répertoire, entrez :

```
mvdir book manuel
```

Tous les répertoires `book` sont déplacés dans le répertoire `manuel`, si celui-ci existe. Sinon, le répertoire `book` est renommé `manuel`.

>Par exemple, pour déplacer et renommer un répertoire, entrez :

```
mvdir book3 proj4/manuel
```

`book3` est déplacé dans `proj4`, lequel est renommé `manuel` (si `manuel` n'existe pas déjà).

Reportez-vous à la commande **mvdir** dans le manuel *AIX Commands Reference*.

Affichage du répertoire courant (commande pwd)

La commande **pwd** écrit, sur l'unité de sortie standard, le chemin d'accès absolu du répertoire courant (en commençant par le répertoire **/**(racine)). Tous les répertoires sont séparés par une barre oblique (/). Le répertoire **/**(racine) est représenté par la première barre /, le dernier répertoire indiqué étant le répertoire courant.

Par exemple, pour afficher le répertoire courant, entrez :

```
pwd
```

Le chemin d'accès complet au répertoire s'affiche, sous la forme :

```
/home/thomas
```

Reportez-vous à la commande **pwd** dans le manuel *AIX Commands Reference*.

Passage à un autre répertoire (commande cd)

La commande **cd** permet de passer à un autre répertoire. Vous devez détenir sur ce dernier des droits d'exécution (recherche).

Si vous omettez de spécifier le paramètre *répertoire*, la commande **cd** vous ramène à votre répertoire de connexion (**\$HOME** dans les environnements **ksh** et **bsh**, **\$home** dans l'environnement **csh**). Si vous spécifiez un chemin d'accès complet, le répertoire désigné devient le répertoire courant. Un chemin d'accès complet commence par une barre oblique (/) indiquant le répertoire **/**(racine), un point (.) indiquant le répertoire courant ou un double point (..) indiquant le répertoire parent. Si vous spécifiez un chemin d'accès relatif, la commande le recherche en fonction des chemins précisés dans la variable shell **\$CDPATH** (ou la variable **csh \$cdpath**). Cette variable a la même syntaxe et sensiblement la même valeur sémantique que la variable shell **\$PATH** (ou la variable **csh \$path**).

Par exemple, pour revenir à votre répertoire personnel, entrez :

```
cd
```

Pour passer au répertoire `/usr/include`, entrez :

```
cd /usr/include
```

`/usr/include` devient le répertoire courant :

Pour passer au sous-répertoire `sys`, entrez :

```
cd sys
```

Si le répertoire courant `/usr/include` contient bien le sous-répertoire `sys`, `/usr/include/sys` devient le répertoire courant.

Pour remonter d'un niveau dans l'arborescence des répertoires, entrez :

```
cd ..
```

Ce nom spécial `..` (point point), référence le répertoire immédiatement "au-dessus" du répertoire courant.

Reportez-vous à la commande **cd** dans le manuel *AIX Commands Reference*.

Copie d'un répertoire (commande cp)

La commande **cp** ou **copy** copie le *FichierSource* ou le *RépertoireSource* dans le *FichierCible* ou le *RépertoireCible*, en écrasant le cas échéant le contenu de *FichierCible*, sans avertissement. Si le *FichierCible* existe, son contenu est écrasé lors de la copie. Pour copier plusieurs fichiers source, la cible doit être un répertoire.

Pour copier le *FichierSource* dans un répertoire, indiquez, dans le paramètre *RépertoireCible*, un chemin d'accès à ce répertoire. Les fichiers conservent leur nom d'origine, sauf si vous en spécifiez d'autres à la fin du chemin d'accès. Si vous lui associez l'indicateur **-r** ou **-R**, la commande **cp** copie des répertoires entiers dans d'autres.

Par exemple, pour copier tous les fichiers d'un répertoire dans un autre, entrez :

```
cp /home/janet/clients/* /home/nick/customers
```

Les fichiers du répertoire `clients` sont copiés dans le répertoire `customers`.

Pour copier un répertoire, avec ses fichiers et ses sous-répertoires, dans un autre, entrez :

```
cp -R /home/nick/clients /home/nick/customers
```

Le répertoire `clients`, avec ses fichiers et ses sous-répertoires, est copié dans le répertoire `customers/clients`.

Reportez-vous à la commande **cp** dans le manuel *AIX Commands Reference*.

Affichage du contenu d'un répertoire (commandes **li** et **ls**)

Pour afficher le contenu d'un répertoire, vous disposez des deux commandes **li** et **ls**.

Commande **li**

La commande **li** affiche des informations sur chaque *Fichier* indiqué dans chacun des *Répertoires* spécifiés. Si *Fichier* est un fichier archivé, spécifiez l'indicateur **-Rq** pour afficher également son contenu. Si vous ne spécifiez pas un *Fichier* ou un *Répertoire*, la commande **li** affiche le contenu du répertoire courant.

Lorsque **li** affiche le contenu d'un répertoire, les entrées de fichiers dont le nom commence par un `.` (point) ne sont pas répertoriées sauf si vous avez utilisé l'indicateur **-a**.

La commande **li** affiche les noms de fichiers et de répertoires dans l'ordre alphabétique. Les caractères de contrôle des noms de fichiers sont affichés sous leur forme développée (par exemple, `^ D, \177`).

Par exemple, pour afficher la liste de tous les fichiers du répertoire courant, entrez :

```
li
```

Vous pouvez lister les fichiers et répertoires séparés par des fichiers, des fichiers exécutables, des liens symboliques et des répertoires. Les fichiers exécutables sont entre chevrons (`< >`), les liens symboliques sont entre arobares (`@`), et les répertoires entre crochets (`[]`). Pour visualiser cette liste, entrez :

```
li -v
```

Pour afficher la liste de tous les fichiers et répertoires du répertoire courant, y compris ceux dont les noms commencent par un point (`.`), entrez :

```
li -a
```

Par exemple, pour afficher des informations complètes, entrez :

```
li -l chap1 .profile
```

Une liste complète avec des informations détaillées sur les fichiers `chap1` et `.profile` s'affiche.

Pour afficher des informations complètes sur un répertoire, entrez :

```
li -d -l . manuel manuel/chap1
```

Une liste complète d'informations sur les répertoires `.` (point) et `manuel`, et sur le fichier `manuel/chap1`.

Pour afficher le contenu de chaque répertoire d'une arborescence, entrez :

```
li -R manuel
```

La liste de tous les répertoires dérivés de `manuel` s'affiche.

Reportez-vous à la commande **li** dans le manuel *AIX Commands Reference*.

Commande ls

La commande **ls** écrit, sur l'unité de sortie standard, le contenu des *Répertoires* ou le nom des *Fichiers* spécifiés, ainsi que les informations demandées via les indicateurs. A défaut de paramètre *Fichier* ou *Répertoire*, elle affiche le contenu du répertoire courant.

Par défaut, la commande **ls** affiche toutes les informations par ordre alphabétique des noms de fichiers. Si la commande est lancée par un utilisateur racine, l'indicateur par défaut est **-A**, et affiche toutes les entrées sauf . (point) et .., (point point). Pour afficher toutes les entrées des fichiers, y compris celles commençant par un . (point), utilisez la commande **ls -a**.

Pour le format de sortie, vous avez le choix entre afficher :

- une entrée par ligne, avec l'indicateur **-l**.
- des entrées sur plusieurs colonnes, avec l'indicateur **-C** ou **-x**, **-C** étant défini par défaut pour une sortie sur un terminal tty.
- des entrées séparées par des virgules, avec l'indicateur **-m**.

Pour déterminer le nombre de caractères par ligne sur l'unité de sortie, la commande **ls** se sert de la variable d'environnement **\$COLUMNS**. Si celle-ci n'est pas définie, la commande lit le fichier **terminfo**. Si aucune de ces méthodes ne donne de résultat, elle adopte par défaut la valeur 80.

Les informations affichées avec les indicateurs **-e** et **-l** sont interprétées comme suit :

Si le premier caractère est :

d	l'entrée est un répertoire.
b	l'entrée est un fichier bloc spécial.
c	l'entrée est un fichier caractères spécial.
l	l'entrée est un lien symbolique.
p	l'entrée est un fichier FIFO (first-in, first-out) spécial.
s	l'entrée est un socket local.
-	l'entrée est un fichier ordinaire.

Les 9 caractères suivants sont regroupés en tranches de 3. Les 3 premiers concernent les droits du propriétaire, les 3 suivants, les droits des autres utilisateurs du groupe, et les 3 derniers, les droits de tous les autres utilisateurs ayant accès au fichier. Chaque caractère d'un groupe de 3 correspond aux droits d'accès en lecture, écriture et exécution. Le droit d'exécution sur un répertoire confère le droit d'y rechercher un fichier donné.

Les droits sont indiqués comme suit :

r	accès en lecture.
t	seul le propriétaire du répertoire ou du fichier peut supprimer ou renommer un fichier dans ce répertoire, même si les autres utilisateurs ont un droit d'écriture sur ce répertoire.
w	accès en écriture (édition).
x	accès en exécution (recherche).
-	accès correspondant refusé.

Les informations affichées avec l'indicateur **-e** sont les mêmes que celles affichées avec l'indicateur **-l**, sauf qu'un 11ème caractère est ajouté :

- +** indique un fichier non doté d'informations de sécurité étendues. Par exemple, le fichier peut comporter des attributs **ACL**, **TCB** ou **TP** dans le mode.
- indique un fichier non doté d'informations de sécurité étendues.

Lorsque la taille des fichiers d'un répertoire est affichée, la commande `ls` affiche le décompte total de blocs, blocs indirects compris.

Par exemple, pour afficher la liste de tous les fichiers du répertoire courant, entrez :

```
ls -a
```

Cette liste comporte tous les fichiers, y compris :

- les fichiers `.` (point),
- les fichiers `..` (point point),
- les autres fichiers commençant ou non par un `.` (point)

Par exemple, pour afficher des informations complètes, entrez :

```
ls -l chap1 .profile
```

Une longue liste donnant des informations complètes sur `chap1` et `.profile` est affichée.

Pour afficher des informations complètes sur un répertoire, entrez :

```
ls -d -l . manuel manuel/chap1
```

Une longue liste concernant les répertoires `.` (point) et `manuel`, et sur le fichier `manuel/chap1`. Sans indicateur **-d**, la commande afficherait la liste des fichiers des répertoires `.` et `manuel`, et non des informations complètes sur les répertoires eux-mêmes.

Reportez-vous à la commande **ls** dans le manuel *AIX Commands Reference*.

Suppression ou retrait d'un répertoire (commande `rmdir`)

La commande **rmdir** supprime du système le répertoire spécifié par le paramètre *Répertoire*. Ce répertoire doit être vide (il ne peut contenir que `.` et `..`) et vous devez avoir accès en écriture à son répertoire parent. Lancez la commande **ls -a** *Répertoire* pour vérifier s'il est vide.

Par exemple, pour vider ou supprimer un répertoire, tapez :

```
rm mydir/* mydir/.  
rmdir mydir
```

Le contenu de `monrép` est supprimé, puis le répertoire lui-même. La commande **rm** affiche un message d'erreur concernant la suppression des répertoires `.` (point) et `..` (point point), puis la commande **rmdir** supprime le répertoire.

Notez que la commande `rm monrép/* monrép/.*` supprime d'abord les fichiers dont le nom ne commence pas par un point, puis les autres. Vous ne connaîtrez peut-être pas l'existence de fichiers commençant par un point : la commande **ls** ne les affiche qu'assortie de l'indicateur **-a**.

Par exemple, pour supprimer le répertoire `/tmp/jones/demo/monrép` et ses sous-répertoires, entrez :

```
cd /tmp  
rmdir -p jones/demo/mydir
```

Le répertoire `jones/demo/monrép` est supprimé du répertoire `/tmp`. Si un répertoire n'est pas vide ou que vous ne détenez pas le droit d'écriture requis, la commande est interrompue et un message d'erreur s'affiche.

Reportez-vous à la commande **rmdir** dans le manuel *AIX Commands Reference*.

Comparaison entre répertoires (commande **dircmp**)

La commande **dircmp** compare les deux répertoires spécifiés par *Répertoire1* et *Répertoire2* et écrit des informations relatives à leur contenu sur l'unité de sortie standard. La commande compare d'abord les noms des fichiers de chaque répertoire. Lorsqu'elle rencontre deux fichiers portant le même nom, elle compare leur contenu.

En sortie, la commande **dircmp** liste les fichiers non communs aux deux répertoires, puis les fichiers portant le même nom, mais de contenu différent, et enfin, si aucun indicateur n'est spécifié, les fichiers identiques (même nom, même contenu).

Par exemple, pour récapituler les différences entre les fichiers des répertoires `proj.ver1` et `proj.ver2`, entrez :

```
dircmp proj.ver1 proj.ver2
```

Le récapitulatif généré liste les différences entre les répertoires `proj.ver1` et `proj.ver2`. Il liste séparément les fichiers propres à chaque répertoire et les fichiers communs aux deux. Ces derniers sont en outre différenciés selon que leur contenu est identique ou non.

Par exemple, pour afficher les différences entre les fichiers des répertoires `proj.ver1` et `proj.ver2`, entrez :

```
dircmp -d -s proj.ver1 proj.ver2
```

L'indicateur **-s** élimine l'affichage des fichiers identiques. L'indicateur **-d** affiche une liste **diff** des fichiers différents trouvés dans les deux répertoires.

Reportez-vous à la commande **dircmp** dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives aux systèmes de fichiers et aux répertoires

Systèmes de fichiers

df	Donne des informations sur l'espace disponible dans les systèmes de fichiers.
-----------	---

Abréviations des répertoires

.	Répertoire de travail courant.
..	Répertoire "au-dessus" (parent) du répertoire de travail courant.
~	Répertoire personnel (inexact pour le shell Bourne).
\$HOME	Répertoire personnel (valable pour tous les shells).

Procédures de gestion des répertoires

cd	Passe à un autre répertoire.
cp	Copie des fichiers ou des répertoires.
dircmp	Compare deux répertoires, ainsi que le contenu de leurs fichiers communs.
li	Liste le contenu d'un répertoire.
ls	Affiche le contenu d'un répertoire.
mkdir	Crée un ou plusieurs répertoires.
mvdir	Déplace (ou renomme) un répertoire.
pwd	Affiche le chemin d'accès au répertoire de travail.
rmdir	Supprime un répertoire.

Chapitre 6. Fichiers

Dans le système d'exploitation, toutes les entrées/sorties (E/S) d'informations passent par des fichiers. Ceci permet de standardiser l'accès aux logiciels et au matériel. Une entrée est créée chaque fois qu'un fichier est modifié, et une sortie constituée chaque fois que le contenu d'un fichier est lu ou transféré vers un autre fichier. Par exemple, lorsque qu'une sortie papier d'un fichier est demandée, le système lit le contenu du fichier et le copie dans le fichier représentant l'imprimante.

Ce chapitre traite des points suivants :

- Types de fichiers, page 6-2.
 - Conventions d'appellation des fichiers, page 6-3.
 - Chemins d'accès aux fichiers, page 6-3.
 - Caractères joker et métacaractères, page 6-3.
 - Recherche générique et expressions régulières, page 6-4.
- Procédure de gestion de fichier, page 6-5.
 - Suppression de fichiers (commandes **del** et **rm**), page 6-5.
 - Déplacement et changement de nom d'un fichier (commande **mv**), page 6-6.
 - Copie de fichiers (commande **cp**), page 6-7.
 - Recherche de fichiers (commande **find**), page 6-8.
 - Affichage du type d'un fichier (commande **file**), page 6-9.
 - Affichage du contenu d'un fichier (commandes **pg**, **more**, **page** et **cat**), page 6-9.
 - Recherche de chaînes dans un fichier texte (commande **grep**), page 6-10.
 - Tri de fichiers texte (commande **sort**), page 6-10.
 - Comparaison de fichiers (commande **diff**), page 6-11.
 - Décompte des mots, des lignes et des octets d'un fichier (commande **wc**), page 6-12.
 - Affichage des premières lignes d'un fichier (commande **head**), page 6-12.
 - Affichage des dernières lignes d'un fichier (commande **tail**), page 6-12.
 - Coupe de sections de fichiers texte (commande **cut**), page 6-13.
 - Collage de sections de fichiers texte (commande **paste**), page 6-13.
 - Numérotation des lignes d'un fichier texte (commande **nl**), page 6-14.
 - Suppression de colonnes dans un fichier texte (commande **colrm**), page 6-15.
- Liaison entre fichiers et répertoires, page 6-16.
 - Types de liens, page 6-16.
 - Liaison de fichiers (commande **ln**), page 6-17.
 - Suppression de fichiers liés, page 6-18.
- Fichiers DOS, page 6-19.
 - Copie de fichiers DOS dans des fichiers AIX, page 6-19.
 - Copie de fichiers AIX dans des fichiers DOS, page 6-19.
 - Suppression de fichiers DOS, page 6-20.
 - Contenu d'un répertoire DOS, page 6-20.
- Récapitulatif des commandes relatives aux fichiers, page 6-21.

Types de fichiers

Il existe trois types de fichiers de base :

standard	fichier de données (texte, binaire ou exécutable).
répertoire	fichier contenant des informations permettant d'accéder à d'autres fichiers.
spécial	fichier de chaînage FIFO (first-in, first-out) ou unité physique.

Les types de fichier reconnus par le système relèvent obligatoirement d'une de ces trois catégories. Il existe cependant à l'intérieur de ces catégories de nombreuses variantes.

Fichiers standard

Les plus courants sont les fichiers standard, appelés aussi ordinaires. Ces fichiers contiennent des données.

Fichiers texte

Les fichiers texte sont des fichiers standard, contenant des données lisibles par l'utilisateur, stockées en ASCII. Vous pouvez les afficher et les imprimer. Les lignes d'un fichier texte ne doivent pas comporter de caractères **NUL**, et ne doivent pas excéder la longueur de **{LINE_MAX}** octets, caractère de nouvelle ligne compris.

Le terme *fichier texte* n'interdit pas la présence de caractères de contrôle et autres caractères non imprimables (autres que **NUL**). En conséquence, les utilitaires standard d'affichage des fichiers texte, en entrée ou en sortie, soit sont capables de traiter ces caractères spéciaux, soit décrivent explicitement les limites de leur action.

Fichiers binaires

Les fichiers binaires sont des fichiers standard, interprétés par le système. Les fichiers binaires peuvent être des fichiers exécutables donnant des instructions au système. Les commandes et les programmes sont stockées dans des fichiers binaires. La conversion textes ASCII/codes binaires est effectuée par des compilateurs.

longueur inférieure à **{LINE_MAX}** octets, pas de caractères **NUL**, terminées par un caractère de nouvelle ligne.

Fichiers répertoire

Un répertoire est un fichier qui ne contient que les informations d'accès à d'autres fichiers et répertoires. Par conséquent, il occupe moins d'espace que les autres types de fichiers et confère au système de fichiers souplesse et puissance. Chaque entrée de répertoire représente un fichier ou un sous-répertoire. Elle contient le nom et le numéro d'i-node du fichier. Cet i-node, qui pointe sur le noeud d'index unique associé au fichier, décrit l'emplacement des données associées au fichier. A la différence des autres types de fichiers, les répertoires ont leur propre jeu de commandes.

Pour en savoir plus, reportez-vous à Répertoires : généralités, page 5-5.

Fichiers spéciaux

Les fichiers spéciaux définissent les unités pour le système, et les fichiers temporaires générés par les process. Il existe trois types de fichiers spéciaux : FIFO (first-in, first-out), bloc et caractère. Les fichiers FIFO sont aussi appelés pipes : créés par un process pour assurer temporairement la communication avec un autre process, ils sont détruits dès la fin du premier process. Les fichiers bloc et caractères définissent des unités.

A chaque fichier sont associés des droits d'accès (dits modes d'accès), déterminant les utilisateurs habilités à le lire, le modifier ou l'exécuter.

Reportez-vous à "Modes d'accès aux fichiers et aux répertoires", page 9-4.

Conventions d'appellation des fichiers

Il ne peut y avoir, dans un même répertoire, deux fichiers enregistrés sous le même nom. Ainsi, à chaque fichier ne correspond qu'un seul chemin d'accès dans le système de fichiers. Pour nommer un fichier, voici quelques règles à respecter :

- Le nom ne peut excéder 255 caractères (lettres, chiffres et traits de soulignement).
- Le système différencie majuscules et minuscules : `FICHIERA`, `FiCHiera` et `fichiera` désignent des fichiers distincts, même s'ils résident dans le même répertoire.
- Choisissez des noms "parlants".
- Les noms de répertoires doivent respecter les mêmes conventions.
- Certains caractères ont une signification spéciale pour le système d'exploitation, il est donc déconseillé de les utiliser dans les noms de fichiers. En voici une liste (non exhaustive) :

`/ \ " ' * ; - ? [] () ~ ! $ { } < > # @ & |`

- Un fichier dont le nom commence par un `.` (point) n'apparaît pas dans les listes de répertoires normales. Pour les afficher, assortissez la commande `ll` ou `ls` de l'indicateur `-a`.

Chemins d'accès aux fichiers

Le chemin d'accès à un fichier ou à un répertoire est constitué de la liste des répertoires situés en amont dans l'arborescence des répertoires.

Tous les chemins étant issus du répertoire racine `/`, chaque fichier est associé de façon unique à ce répertoire, par un chemin qualifié d'absolu. Un chemin d'accès absolu commence toujours par le symbole `/` (barre oblique). Dans l'exemple, le chemin d'accès absolu du fichier `h` est `/B/C/h`. Notez qu'il existe deux fichiers portant le nom `g`. Les chemins d'accès absolus à ces fichiers étant différents (`/B/g` et `/B/C/g`), le système distingue les deux fichiers. Le chemin d'accès est composé de répertoires, à l'exception du dernier élément, qui peut être un fichier.

Remarque : Les chemins d'accès ne peuvent dépasser 1023 caractères.

Caractères joker et métacaractères

Les caractères joker permettent de spécifier facilement un ensemble de noms de fichiers ou de répertoires, par le biais d'un seul caractère. Il existe deux caractères joker : `*` (astérisque) et `?` (point d'interrogation). Les métacaractères sont : `[]` (crochets), `-` (tiret) et `!` (point d'exclamation).

Caractère joker `*`

L'astérisque remplace une séquence ou une chaîne quelconque de caractères. La chaîne peut éventuellement être vide. Si, par exemple, votre répertoire contient les fichiers :

```
1test 2test afile1 afile2 bfile1 file file1 file10 file2 file3
```

et que vous voulez référencer les fichiers commençant par `file`, spécifiez :

```
file*
```

La sélection portera sur les fichiers : `file file1 file10 file2 file3`.

Pour référencer les fichiers contenant le mot `file`, spécifiez :

```
*file*
```

La sélection portera sur les fichiers : `afile1 afile2 bfile1 file file1 file10 file2 file3`.

? Caractère joker

Le point d'interrogation ? remplace un caractère quelconque. Il peut s'agir de n'importe quel caractère unique.

Pour référencer les fichiers commençant par `file`, suivi d'un seul caractère, spécifiez :

```
file?
```

La sélection portera sur les fichiers : `file1 file2 file3`.

Pour référencer les fichiers commençant par `file`, suivi de deux caractères, spécifiez :

```
file??
```

La sélection portera sur le fichier : `file10`

Métacaractères shell []

Les métacaractères sont une alternative aux caractères joker. Il s'agit d'encadrer les caractères souhaités par des crochets []. La méthode est semblable à l'utilisation de ?, mais là, vous précisez les caractères qui doivent correspondre. Vous pouvez aussi, à l'aide d'un tiret, spécifier un intervalle de valeurs. Pour spécifier toutes les lettres de l'alphabet, indiquez `[:alpha:]`. Pour spécifier toutes les lettres minuscules de l'alphabet, indiquez `[:lower:]`.

Pour référencer les fichiers se terminant par 1 ou 2, spécifiez :

```
*file[12]
```

La sélection portera sur les fichiers : `afile1 afile2 file1 file2`.

Pour référencer les fichiers commençant par un chiffre, spécifiez :

La sélection portera sur les fichiers : `1test 2test`

Pour référencer les fichiers ne commençant pas par un a, spécifiez :

```
[!a]*
```

La sélection portera sur les fichiers : `1test 2test bfile1 file file1 file10 file2 file3`

Recherche générique et expressions régulières

Les expressions régulières permettent de sélectionner des chaînes dans un ensemble spécifique de chaînes de caractères. Leur usage est généralement associé à un traitement de texte.

Les expressions régulières peuvent représenter un large éventail de chaînes possibles. Alors que ces expressions peuvent être diversement interprétées selon l'environnement local, les fonctions d'internationalisation assurent l'invariance du contexte au niveau des différents environnements locaux.

Voici quelques exemples :

Recherche générique	Expression régulière
*	.*
?	.
[!a]	[^a]
[abc]	[abc]
[:alpha:]	[:alpha:]

Reportez-vous à la commande **awk** dans le manuel *AIX Commands Reference*.

Procédures de gestion de fichier

Travailler sur des fichiers couvre de multiples tâches. Le plus souvent, vous créez un fichier texte à l'aide d'un éditeur. Les éditeurs standard d'UNIX sont **vi** et **ed**. AIX inclut également son propre éditeur de texte, **INed**. Choisissez celui qui vous convient le mieux.

Vous pouvez également créer des fichiers en réacheminant des entrées/sorties, comme décrit dans "Réacheminement des entrées/sorties", page 4-1. La sortie d'une commande peut ainsi être envoyée dans un nouveau fichier ou ajoutée à un fichier existant.

Les fichiers créés et modifiés peuvent ensuite être copiés dans un autre répertoire, renommés pour en distinguer les différentes versions. Vous serez sans doute également amené à créer de nouveaux répertoires si vous travaillez sur plusieurs projets.

Il vous faudra aussi supprimer certains fichiers. Et parfois libérer de l'espace dans vos répertoires en supprimant des fichiers devenus inutiles ou obsolètes. La suppression de fichiers devenus inutiles ou obsolètes libérera de l'espace sur votre système.

Ce chapitre traite des points suivants :

- Suppression de fichiers (commandes **del** et **rm**), page 6-5.
- Déplacement et changement de nom d'un fichier (commande **mv**), page 6-6.
- Copie de fichiers (commande **cp**), page 6-7.
- Recherche de fichiers (commande **find**), page 6-8.
- Affichage du type de fichier (commande **file**), page 6-9.
- Affichage du contenu d'un fichier (commandes **pg**, **more** et **cat**), page 6-9.
- Recherche de chaînes dans un fichier texte (commande **grep**), page 6-10.
- Tri des fichiers texte (commande **sort**), page 6-10.
- Comparaison de fichiers (commande **diff**), page 6-11.
- Décompte des mots, des lignes et des octets d'un fichier (commande **wc**), page 6-12.
- Affichage des premières lignes d'un fichier (commande **head**), page 6-12.
- Affichage des dernières lignes d'un fichier (commande **tail**), page 6-12.
- Coupe de sections de fichiers texte (commande **cut**), page 6-13.
- Collage de sections de fichiers texte (commande **paste**), page 6-13.
- Numérotation des lignes d'un fichier texte (commande **nl**), page 6-14.
- Suppression de colonnes dans un fichier texte (commande **colrm**), page 6-15.

Suppression de fichiers (commandes **del** et **rm**)

Lorsqu'un fichier devient inutile, vous pouvez le supprimer par la commande **del**, **rm** ou **delete**. La commande **del** demande confirmation avant de supprimer un fichier. Vous pouvez ainsi changer d'avis. Ceci n'est pas le cas des commandes **rm** et **delete**. Ces trois commandes permettent de supprimer un groupe de fichiers ou de sélectionner dans une liste les fichiers à supprimer.

Commande **del**

Attention : La commande **del** ignore la protection de fichiers, permettant au propriétaire d'un fichier de supprimer un fichier protégé en écriture. Toutefois, pour supprimer un fichier, vous devez détenir les droits d'accès en écriture au répertoire contenant le fichier. Appuyer sur la touche Entrée correspond à répondre oui : soyez vigilant de ne pas supprimer de fichiers de façon accidentelle.

La commande **del** affiche la liste des fichiers spécifiés et vous demande de confirmer leur suppression. Pour confirmer, appuyez sur Entrée ou tapez `y` (ou son équivalent local). Toute autre réponse revient à répondre non (fichiers non supprimés).

Remarque : La commande **del** ne supprime pas de répertoires. Reportez-vous à la commande **rmdir** pour en savoir plus sur la suppression des répertoires.

Par exemple, pour supprimer le fichier `chap1.bak`, entrez :

```
del chap1.bak
```

Le message suivant s'affiche :

```
del: Voulez-vous supprimer chap1.bak? Dans l'affirmative, tapez o
ou appuyez sur Entrée.
```

Appuyez sur Entrée ou tapez `o` pour confirmer. Appuyez sur n'importe quelle autre touche pour annuler.

Reportez-vous à la commande **del** dans le manuel *AIX Commands Reference*.

Commande **rm**

La commande **rm** ou **delete** supprime d'un répertoire les entrées du ou des fichier(s) spécifié(s). Aucun droit d'accès en écriture ou en lecture au fichier à supprimer n'est requis. Toutefois, vous devez détenir les droits d'accès en écriture au répertoire contenant le fichier.

Par exemple, pour supprimer le fichier `monfich`, entrez :

```
rm monfich
```

Pour supprimer un par un les fichiers du répertoire `mydir`, entrez :

```
rm -i mydir/*
```

A l'affichage d'un fichier, tapez `y` pour confirmer la suppression ou appuyez sur Entrée pour l'annuler.

Remarque : Les touches ne sont pas les mêmes que pour la commande **del**.

Reportez-vous à la commande **rm** dans le manuel *AIX Commands Reference*.

Déplacement et changement de nom d'un fichier (commande **mv**)

La commande **mv** déplace fichiers et répertoires d'un répertoire dans un autre, ou renomme un fichier ou un répertoire. Si vous effectuez un déplacement sans spécifier de nouveau nom, les noms d'origine sont conservés.

Attention : Si vous omettez l'indicateur `-i`, la commande **mv** peut écraser de nombreux fichiers. L'indicateur `-i` demande confirmation avant d'écraser un fichier, ce qui n'est pas le cas de `-f`. Si `-f` et `-i` sont tous deux spécifiés, c'est le dernier indiqué qui est prioritaire.

Déplacer un fichier avec la commande **mv**

Par exemple, pour déplacer un fichier dans un autre répertoire et le renommer, entrez :

```
mv intro manual/chap1
```

Le fichier `intro` est transféré dans `manual/chap1`. Le nom `intro` disparaît du répertoire courant et le fichier apparaît sous le nom `chap1` dans le répertoire `manual`. Consultez l'**avertissement** précédent.

Pour transférer un fichier sans le renommer, entrez :

```
mv chap3 manual
```

Le fichier `chap3` est transféré dans `manual/chap3`. Consultez l'**avertissement** précédent.

Renommer un fichier avec la commande mv

Par exemple, pour renommer un fichier, entrez-

```
mv appendix apndx.a
```

Le fichier `appendix` s'appelle désormais `apndx.a`. Si un fichier de même nom existe déjà, il est écrasé par le contenu de `appendix`. Consultez l'**avertissement** précédent.

Reportez-vous à la commande **mv** dans le manuel *AIX Commands Reference*.

Copie de fichiers (commande cp)

La commande **cp** ou **copy** copie le *FichierSource* ou le *RépertoireSource* dans le *FichierCible* ou le *RépertoireCible*, en écrasant le cas échéant le contenu de *FichierCible*, sans avertissement. Si le *FichierCible* existe, son contenu est écrasé lors de la copie. Vous ne pouvez copier plusieurs *FichiersSource* que dans un répertoire.

Si un fichier de même nom existe à l'emplacement cible, son contenu sera écrasé par le fichier copié. Prenez donc l'habitude d'attribuer un nouveau nom au fichier copié: vous éviterez tout risque de perte de données.

Pour copier le *FichierSource* dans un répertoire, indiquez, dans le paramètre *RépertoireCible*, un chemin d'accès à ce répertoire. Les fichiers conservent leur nom d'origine, sauf si vous en spécifiez d'autres à la fin du chemin d'accès. Si vous lui associez l'indicateur **-r** ou **-R**, la commande **cp** copie des répertoires entiers dans d'autres.

Les fichiers unité peuvent également être copiés. Spécifiez de préférence l'indicateur **-R**. Les fichiers seront ainsi recréés sous le nouveau chemin. Si vous spécifiez **-r**, la commande tente de copier les fichiers en les convertissant en fichiers standard.

Par exemple, pour copier un fichier de répertoire courant, entrez :

```
cp prog.c prog.bak
```

Le fichier `prog.c` est copié dans `prog.bak`. Si celui-ci n'existe pas, il est créé. S'il existe, il est remplacé par la copie de `prog.c`.

Pour copier un fichier du répertoire courant dans un autre répertoire, entrez :

```
cp jones /home/nick/clients
```

Le fichier `jones` est copié dans le répertoire `/home/nick/clients/jones`.

Par exemple, pour copier tous les fichiers d'un répertoire dans un autre, entrez :

```
cp /home/janet/clients/* /home/nick/customers
```

Les fichiers du répertoire `clients` sont copiés dans le répertoire `customers`.

Pour copier un ensemble donné de fichiers dans un autre répertoire, entrez :

```
cp jones lewis smith /home/nick/clients
```

Les fichiers `jones`, `lewis` et `smith` sont copiés du répertoire courant dans le répertoire `/home/nick/clients`.

Pour copier des fichiers en utilisant des caractères joker, entrez :

```
cp programs/*.c.
```

Tous les fichiers du répertoire `programs` suffixés `.c` sont copiés dans le répertoire courant (spécifié par le point `.` unique). Notez que l'extension `c` et le point doivent être séparés par un espace.

Reportez-vous à la commande **cp** dans le manuel *AIX Commands Reference*.

Recherche de fichiers (commande find)

La commande **find** parcourt récursivement l'arborescence, pour chaque *Chemin* spécifié, à la recherche des fichiers correspondant à une expression booléenne, sur laquelle nous reviendrons. Le résultat de la commande **find** dépend des termes du paramètre *Expression*.

Par exemple, pour afficher tous les fichiers `.profile` du système de fichiers, entrez :

```
find / -name .profile
```

L'intégralité du système de fichiers est balayé et les fichiers `profile` trouvés sont affichés, avec leur chemin d'accès complet. La barre oblique (/) indique à la commande **find** d'explorer le répertoire racine (/) et tous ses sous-répertoires. Pour gagner du temps, essayez de limiter ce type de recherche aux répertoires susceptibles de contenir les fichiers qui vous intéressent.

Par exemple, pour afficher les fichiers du répertoire courant dotés du code d'accès `0600`, entrez :

```
find . -perm 0600
```

Seuls les fichiers accessibles en lecture-écriture par leur *seul* propriétaire sont affichés. Le point (.) indique à la commande **find** d'explorer le répertoire courant et ses sous-répertoires. Pour en savoir plus sur les codes d'accès, reportez-vous à la commande **chmod**.

Par exemple, pour rechercher, dans plusieurs répertoires, les fichiers dotés de codes d'accès donnés, entrez :

```
find manual clients proposals -perm -0600
```

Seuls les fichiers accessibles en lecture-écriture, et éventuellement avec d'autres droits d'accès, par leur *-seul* propriétaire sont affichés. Les répertoires `manual`, `clients` et `proposals`, et leurs sous-répertoires, sont explorés. Dans l'exemple précédent (`-perm 0600`), seuls les fichiers dont le code d'accès est exactement `0600` (`-perm 0600`) sont sélectionnés. Ici, sont sélectionnés les fichiers dont le code d'accès est supérieur à égal à `0600` (`-perm -0600`) : les codes `0622` et `2744` sont ainsi également concernés.

Par exemple, pour afficher tous les fichiers du répertoire courant, modifiés au cours des dernières 24 heures, entrez :

```
find . -ctime 0
```

Par exemple, pour rechercher les fichiers standard dotés de liens multiples, entrez :

```
find . -type f -links +1
```

Les fichiers standard (`-type f`) dotés de plusieurs liens (`-links +1`) sont affichés.

Remarque : A chaque répertoire sont associés au moins deux liens : l'entrée dans son répertoire parent et sa propre . (point) entrée. Pour en savoir plus, reportez-vous à la commande **ln**.

Par exemple, pour imprimer les chemins d'accès de tous les fichiers du répertoire courant et de ses sous-répertoires, sauf des répertoires `SCCS` et des fichiers de ces répertoires, entrez :

```
find . -name SCCS -prune
```

Par exemple, pour rechercher tous les fichiers de 414 octets exactement, entrez :

```
find . -size 414c
```

Reportez-vous à la commande **find** dans le manuel *AIX Commands Reference*.

Affichage du type d'un fichier (commande file)

La commande **file** parcourt les fichiers spécifiés par le paramètre *Fichier* ou **-f ListeFichier**, y exécute une série de tests, tente de les classer selon leur type, puis écrit le résultat sur l'unité de sortie standard.

Pour un fichier ASCII, la commande **file** examine les 512 premiers octets pour en déterminer la langue. S'il ne s'agit pas d'un fichier ASCII, la commande **file** poursuit l'analyse pour déterminer s'il s'agit d'un fichier de données binaire ou d'un fichier texte contenant des caractères étendus.

Si le paramètre *Fichier* indique un fichier exécutable ou un module objet, avec un numéro de version supérieure à 0, la commande **file** affiche le numéro de version.

La commande **file** identifie, via le fichier **/etc/magic**, les fichiers comportant un chiffre "magique", c'est-à-dire une constante, numérique ou de type chaîne, indiquant le type du fichier.

Par exemple, pour afficher le type des informations du fichier `monfich`, entrez :

```
file monfich
```

Le type de `monfich` (répertoire, données, texte ASCII, programme source C, archive, etc.) est affiché.

Par exemple, pour afficher le type de chacun des fichiers de `filenames.lst` (liste de noms de fichiers), entrez :

```
file -f filenames.lst
```

Le type de chaque fichier de la liste `filenames.lst` est affiché (sous réserve que chaque fichier apparaisse seul sur une ligne).

Par exemple, pour créer le fichier `filenames.lst`, de sorte qu'il contienne les noms de tous les fichiers du répertoire courant, entrez :

```
ls > filenames.lst
```

Modifiez ensuite `filenames` à votre guise.

Reportez-vous à la commande **file** dans le manuel *AIX Commands Reference*.

Affichage du contenu d'un fichier (commandes pg, more, page et cat)

Les commandes **pg**, **more** et **page** permettent de visualiser le contenu d'un fichier et d'en contrôler l'affichage. La commande **cat** affiche le contenu d'un ou de plusieurs fichiers. Combinée à la commande **pg**, elle permet d'afficher ce contenu écran par écran.

Vous pouvez également consulter le contenu des fichiers via les procédures de réacheminement. Pour en savoir plus, reportez-vous à "Réacheminement des entrées/sorties : généralités", page 4-1.

Commande pg

La commande **pg** lit les noms de fichiers spécifiés par le paramètre *Fichier* et les affiche page par page sur l'unité de sortie standard. Si vous spécifiez **-** (moins) comme paramètre *Fichier*, ou ne spécifiez aucun paramètre, la commande **pg** lit les entrées standard. Chaque écran est suivi d'une invite : appuyer sur Entrée affiche l'écran suivant. Des sous-commandes permettent de revenir à un écran antérieur.

Par exemple, pour consulter page par page le contenu du fichier `monfich`, entrez :

```
pg monfich
```

Reportez-vous à la commande **pg** dans le manuel *AIX Commands Reference*.

Commande **more** ou **page**

La commande **more** ou **page** affiche du texte en continu, écran par écran. Une pause entre chaque affichage permet d'afficher au bas de l'écran *nomfichier* et le pourcentage de lecture effectué (*monfich (7%)*, par exemple). Si vous appuyez sur Entrée, la commande **more** affiche une ligne de plus. Si vous appuyez sur la barre d'espace, la commande **more** affiche l'écran de texte suivant.

Remarque : Sur certains modèles de terminaux, la commande **more** efface l'écran (au lieu de le faire avancer d'un ligne) et affiche l'écran de texte suivant.

Par exemple, pour visualiser le fichier *monfich*, entrez :

```
more monfich
```

Appuyez sur la barre d'espace pour passer à l'écran suivant.

Reportez-vous à la commande **more** dans le manuel *AIX Commands Reference*.

Commande **cat**

La commande **cat** lit en séquence les paramètres *Fichier* et les écrit sur l'unité de sortie standard.

Par exemple, pour afficher le contenu du fichier *notes*, entrez :

```
cat notes
```

Si le fichier dépasse 24 lignes, le début défile sans que vous ayez le temps de le lire. Pour obtenir un affichage page par page, utilisez la commande **pg**.

Par exemple, pour afficher le contenu des fichiers *notes*, *notes2* et *notes3*, entrez :

```
cat notes notes2 notes3
```

Reportez-vous à la commande **cat** dans le manuel *AIX Commands Reference*.

Recherche de chaînes dans un fichier texte (commande **grep**)

La commande **grep** recherche l'expression générique spécifiée par le paramètre *Trame* et écrit chaque ligne correspondant à l'expression sur l'unité de sortie standard.

Par exemple, pour rechercher dans le fichier *pgm.s* les lignes correspondant à une expression contenant des caractères joker et des métacaractères (*, ^, ?, [,], , , { et }), ici les lignes commençant par une lettre (majuscule ou minuscule), entrez :

```
grep "^[a-zA-Z]" pgm.s
```

Toutes les lignes de *pgm.s* commençant par une lettre sont affichées.

Pour afficher toutes les lignes du fichier *sort.c* ne correspondant pas à la trame, entrez :

```
grep -v bubble sort.c
```

Toutes les lignes du fichier *sort.c* ne contenant pas le mot *bubble* sont affichées.

Pour afficher les lignes de la sortie de la commande **ls** contenant la chaîne *staff*, entrez :

```
ls -l | grep staff
```

Reportez-vous à la commande **grep** dans le manuel *AIX Commands Reference*.

Tri des fichiers texte (commande **sort**)

La commande **sort** trie par ordre alphabétique les lignes du (des) fichier(s) spécifié(s) par le paramètre *Fichier* et écrit le résultat sur l'unité de sortie standard. Si *Fichier* spécifie plusieurs fichiers, la commande les concatène et trie le fichier résultant.

Remarque : La commande **sort** différencie les majuscules des minuscules et place normalement les majuscules avant les minuscules (mais cela dépend de l'environnement local).

Soit le fichier `names` contenant :

```
marta
denise
joyce
endrica
melanie
```

et le fichier `states` contenant :

```
texas
colorado
ohio
```

Pour trier le contenu du fichier `names`, entrez :

```
sort names
```

Le système affiche un écran semblable à celui-ci :

```
denise
endrica
joyce
marta
melanie
```

Pour trier le contenu des deux fichiers `names` et `states`, entrez :

```
sort names states
```

Le système affiche un écran semblable à celui-ci :

```
colorado
denise
endrica
joyce
marta
melanie
ohio
texas
```

Pour remplacer le contenu du fichier `names` d'origine par son contenu trié, entrez :

```
sort -o names names
```

Le contenu du fichier `names` est remplacé par son contenu trié.

Reportez-vous à la commande **sort** dans le manuel *AIX Commands Reference*.

Comparaison de fichiers (commande **diff**)

La commande **diff** compare les fichiers texte. Elle peut comparer des fichiers isolés ou regroupés dans des répertoires.

Exécutée sur des fichiers standard, la commande **diff**, lorsqu'elle porte sur des fichiers texte dans des répertoires différents, indique les lignes à modifier pour que les fichiers soient identiques.

Par exemple, pour comparer deux fichiers, entrez :

```
diff chap1.bak chap1
```

Les différences entre les fichiers `chap1.bak` et `chap1` s'affichent.

Pour comparer deux fichiers sans tenir compte des blancs, entrez :

```
diff -w prog.c.bak prog.c
```

Deux lignes ne se distinguant que par le nombre d'espaces et de tabulations sont considérées comme identiques par la commande **diff -w**.

Reportez-vous à la commande **diff** dans le manuel *AIX Commands Reference*.

Décompte des mots, des lignes et des octets d'un fichier (commande **wc**)

Par défaut, la commande **wc** compte le nombre de lignes, de mots et d'octets des fichiers spécifiés par le paramètre *Fichier*. A défaut, c'est le texte en entrée standard qui est pris en compte. Le résultat est envoyé en sortie standard, avec cumul des décomptes de tous les fichiers. Si des indicateurs sont spécifiés, leur ordre détermine celui de la sortie. Toute chaîne de caractères délimitée par des espaces, des tabulations ou des caractères ligne suivante définit un *mot*.

Si vous indiquez des fichiers sur la ligne de commande, leur nom est inscrit avec le décompte.

Par exemple, pour afficher les décomptes en lignes, en mots et en octets du fichier `chap1`, entrez :

```
wc chap1
```

Le nombre de lignes, de mots et d'octets du fichier `chap1` est affiché.

Par exemple, pour afficher les seuls décomptes en octets et en mots, entrez :

```
wc -cw chap*
```

Le nombre d'octets et de mots des fichiers dont le nom commence par `chap` est affiché.

Reportez-vous à la commande **wc** dans le manuel *AIX Commands Reference*.

Affichage des premières lignes d'un fichier (commande **head**)

La commande **head** écrit sur l'unité de sortie standard les premières lignes des fichiers spécifiés ou de l'entrée standard. En l'absence d'indicateur, ce sont les 10 premières lignes qui sont affichées par défaut.

Par exemple, pour afficher les 5 premières lignes du fichier `Test`, entrez :

```
head -5 Test
```

Reportez-vous à la commande **head** dans le manuel *AIX Commands Reference*.

Affichage des dernières lignes d'un fichier (commande **tail**)

La commande **tail** écrit le fichier *Fichier* sur l'unité de sortie standard, à partir du point spécifié.

Par exemple, pour afficher les 10 dernières lignes du fichier `notes`, entrez :

```
tail notes
```

Par exemple, pour préciser la ligne (en commençant par la fin) à partir de laquelle lire le fichier `notes`, entrez :

```
tail -20 notes
```

Par exemple, pour afficher le fichier `notes` page par page, à partir du 200ème octet, entrez :

```
tail -c +200 notes | pg
```

Par exemple, pour surveiller la taille du fichier `accounts`, entrez :

```
tail -f accounts
```

Par exemple, pour afficher les 10 dernières lignes du fichier `accounts`, entrez : La commande **tail** continue à afficher les lignes au fur et à mesure où elles sont ajoutées dans le fichier `accounts`. L'affichage se poursuit tant que vous n'appuyez pas sur Ctrl-C pour l'interrompre.

Reportez-vous à la commande **tail** dans le manuel *AIX Commands Reference*.

Coupe de sections de fichiers texte (commande cut)

La commande **cut** écrit sur l'unité de sortie standard les octets, les caractères ou les champs spécifiés, à partir de chaque ligne d'un fichier.

Par exemple, pour afficher certains champs de chaque ligne d'un fichier :

```
cut -f1,5-d: /etc/passwd
```

Cette commande affiche les champs nom de connexion et nom utilisateur complet du fichier mot de passe du système. Il s'agit des 1er et 5ème champs (`-f1, 5`) séparés par des deux points (`-d:`).

Par exemple, pour un fichier `/etc/passwd` semblable à :

```
su:*:0:0:User with special privileges:/:usr/bin/sh
daemon:*:1:1::/etc:
bin:*:2:2::/usr/bin:
sys:*:3:3::/usr/src:
adm:*:4:4:System Administrator:/var/adm:/usr/bin/sh
pierre:*:200:200:Pierre Harper:/home/pierre:/usr/bin/sh
joan:*:202:200:Joan Brown:/home/joan:/usr/bin/sh
```

la commande **cut** génère :

```
su:User with special privileges
daemon:
bin:
sys:
adm:System Administrator
pierre:Pierre Harper
joan:Joan Brown
```

Reportez-vous à la commande **cut** dans le manuel *AIX Commands Reference*.

Collage de sections de fichiers texte (commande paste)

La commande **paste** fusionne les lignes de plusieurs fichiers (12 au maximum) dans un seul.

Par exemple, soit le fichier `names` contenant le texte :

```
rachel!
jerry
mark
linda
scott
```

le fichier `places` contenant le texte :

```
New York
Austin
Chicago
Boca Raton
Seattle
```

et le fichier `dates` contenant le texte :

```
February 5
March 13
June 21
July 16
November 4
```

Vous pouvez regrouper les fichiers `names`, `places` et `dates`, en entrant :

```
paste>names places dates > npd
```

Cette commande crée le fichier `npd` contenant les données du fichier `names` en première colonne, celles du fichier `places` en deuxième colonne et celles du fichier `dates` en troisième colonne : Le fichier `npd` contient désormais :

```
rachel      New York    February 5
jerry       Austin     March 13
mark        Chicago    June 21
linda       Boca Raton July 16
scott       Seattle    November 4
```

Les données nom, lieu et date sont séparées, sur chaque ligne, par une tabulation. les taquets de tabulation sont positionnés toutes les 8 colonnes.

Pour séparer les colonnes par un caractère autre qu'une tabulation, entrez :

```
paste -d"!@" names places dates > npd
```

Les caractères `!` et `@` séparent alternativement les données. Si les fichiers `names`, `places` et `dates` sont les mêmes que dans l'exemple 1, le fichier `npd` apparaît désormais sous la forme :

```
rachel!New York@February 5
jerry!Austin@March 13
mark!Chicago@June 21
linda!Boca Raton@July 16
scott!Seattle@November 4
```

Par exemple, pour afficher le répertoire courant en quatre colonnes, entrez :

```
ls | paste - - - -
```

Chaque signe `-` (moins) indique à la commande **paste** de créer une colonne contenant des données issues de l'unité d'entrée standard. La première ligne est affichée dans la première colonne, la deuxième ligne dans la deuxième colonne, etc.

Reportez-vous à la commande **paste** dans le manuel *AIX Commands Reference*.

Numérotation des lignes d'un fichier texte (commande `nl`)

La commande **nl** lit le fichier spécifié (entrée standard par défaut), en numérote les lignes et écrit celles-ci sur la sortie standard.

Par exemple, pour numéroté les seules lignes non vides, entrez :

```
nl chap1
```

Les lignes non vides du fichier `chap1` s'affichent, numérotées.

Par exemple, pour numéroté toutes les lignes :

```
nl -ba chap1
```

Toutes les lignes du fichier `chap1`, y compris les lignes vides, sont numérotées.

Reportez-vous à la commande **nl** dans le manuel *AIX Commands Reference*.

Suppression de colonnes dans un fichier texte (commande colrm)

La commande **colrm** supprime d'un fichier les colonnes spécifiées. L'entrée est lue sur l'unité d'entrée standard. La sortie est envoyée vers l'unité de sortie standard.

Lancée avec un seul paramètre, la commande supprime de chaque ligne toutes les colonnes à partir de la colonne spécifiée. Lancée avec deux paramètres, ce sont les colonnes comprises entre les deux colonnes spécifiées qui sont supprimées.

Remarque : La numérotation des colonnes commence à 1.

Par exemple, pour supprimer des colonnes du fichier `text.fil`, entrez :

```
colrm 6 < text.fil
```

Si `text.fil` contient :

```
123456789
```

la commande **colrm** affiche :

```
12345
```

Reportez-vous à la commande **colrm** dans le manuel *AIX Commands Reference*.

Liaison entre fichiers et répertoires

Un *lien* est une connexion entre un nom de fichier et un numéro de référence de noeud d'index (i-node), représentation interne d'un fichier. Les entrées de répertoire pouvant comporter des noms de fichiers appariés avec des i-nodes, chaque entrée de répertoire constitue un lien. Le numéro d'i-node identifie le fichier lui-même, et non son nom. Par le biais des liens, chaque i-node ou fichier peut être référencé par plusieurs noms.

Par exemple, l'i-node 798 contient un mémo sur les ventes de Juin de l'agence Omaha. L'entrée de répertoire correspondante est la suivante :

Numéro d'i-node	Abréviation de nom de fichier
798	memo

Cette information étant liée à celle des répertoires `sales` et `omaha`, des liaisons sont établies pour assurer la mise en commun des informations. Des liaisons sont créées vers ces répertoires à l'aide de la commande **In**. Le fichier est alors référencé sous trois noms :

Numéro d'i-node	Nom de fichier
798	memo
798	sales/june
798	omaha/junesales

Si vous lancez la commande **pg** ou **cat** pour afficher le contenu de ces fichiers, vous obtiendrez les mêmes informations. Et si vous modifiez l'un de ces trois fichiers, le contenu des deux autres est modifié en conséquence.

Cette section traite des points suivants :

- Types de liens, page 6-16.
- Liaison de fichiers (commande **In**), page 6-17.
- Suppression de fichiers liés, page 6-18.

Types de liens

Les liens sont créés via la commande **In**. Il en existe deux types :

lien fixe	Donne accès aux données d'un fichier par le biais d'un nouveau nom de fichier. Un lien fixe assure l'existence d'un fichier. Si le dernier lien est supprimé, l'i-node et ses données le sont également. Les liens fixes ne peuvent être créés qu'entre fichiers d'un même système de fichiers.
lien symbolique	Donne accès aux données d'autres système de fichiers par le biais d'un nouveau nom de fichier. Un lien symbolique est un type particulier de fichier, contenant un chemin d'accès. Lorsqu'un process rencontre un lien symbolique, il recherche éventuellement ce chemin. Les liens symboliques ne protègent pas les fichiers de leur suppression du système de fichiers.

Remarque : L'utilisateur qui crée un fichier en reste propriétaire quel que soit le nombre de liens qui y sont rattachés. Lui seul (ou l'utilisateur racine) est habilité à en définir le mode d'accès. Le fichier peut néanmoins être modifié à partir d'un nom de fichier lié, sous réserve de détenir les droits d'accès ad hoc.

Un fichier ou un répertoire existe tant qu'il existe un lien fixe avec son i-node. Le nombre de liens à chaque fichier et sous-répertoire apparaît dans la liste détaillée affichée par la commande **ls -l**. Tous les liens fixes sont équivalents pour le système : l'ordre de leur création est indifférent.

Liaison de fichiers (commande **ln**)

Lier des fichiers via la commande **ln** permet de travailler sur les mêmes données en plusieurs endroits. Créer un lien revient à donner d'autres noms au fichier d'origine. L'utilisation de liens permet, par exemple, de partager un fichier volumineux (base de données, liste de publipostage, etc.) entre plusieurs utilisateurs sans le copier. l'espace disque n'est pas gaspillé et toute modification apportée au fichier est immédiatement répercutée sur tous les fichiers liés.

La commande **ln** lie le *FichierSource* au *FichierCible* ou au fichier de même nom du répertoire *RépertoireCible*. Par défaut, **ln** crée des liens fixes. Pour créer des liens symboliques via la commande **ln**, spécifiez l'indicateur **-s**.

Vous ne pouvez associer qu'un fichier à un nouveau nom. Vous pouvez cependant en associer plusieurs à un répertoire.

Le paramètre *FichierCible* est facultatif. A défaut, la commande **ln** crée un fichier dans le répertoire courant, portant le même nom que *FichierSource*.

Remarque : Vous devez absolument spécifier l'indicateur **-s** pour lier des fichiers entre systèmes de fichiers.

Par exemple, pour créer un autre lien avec le fichier `chap1`, entrez :

```
ln -f chap1 intro
```

Le fichier `chap1` est lié au nom `intro`. Avec l'indicateur **-f**, le nom `intro` est créé s'il n'existe pas. S'il existe, le fichier est remplacé par un lien à `chap1`. Les deux noms `chap1` et `intro` référencent alors le même fichier. Toute modification apportée à l'un des fichiers est reportée dans l'autre.

Par exemple, pour lier le fichier `index` au même nom dans le répertoire `manual`, entrez :

```
ln index manual
```

`index` est lié au nom `manual/index`.

Par exemple, pour lier plusieurs fichiers à des noms dans d'autres répertoires, entrez :

```
ln chap2 jim/chap3 /home/manual
```

`chap2` est lié à `/home/manual/chap2`, et `jim/chap3` à `/home/manual/chap3`.

Par exemple, la commande :

```
ln manual/* .
```

lie tous les fichiers du répertoire `manual` au répertoire courant, `.` (point), en conservant leurs noms d'origine.

Remarque : Vous devez séparer l'astérisque du point par un espace.

Par exemple, pour créer un lien symbolique, entrez :

```
ln -s /tmp/toc toc
```

Le lien symbolique, `toc`, est créé dans le répertoire courant. Le fichier `toc` pointe sur le fichier `/tmp/toc`. Si celui-ci existe, la commande **cat** `toc` affiche son contenu.

Pour obtenir le même résultat sans préciser le paramètre *FichierCible*, entrez :

```
ln -s /tmp/toc
```

Reportez-vous à la commande **ln** dans le manuel *AIX Commands Reference*.

Suppression de fichiers liés

La commande **rm** ou **del** supprime le lien du nom de fichier que vous indiquez. Le fichier lui-même peut ne pas être supprimé, s'il existe sous un autre nom : ce n'est qu'à la suppression du dernier lien à l'i-node que les données sont supprimées. Le numéro d'i-node correspondant est alors libéré.

Reportez-vous aux commandes **del** et **rm** dans le manuel *AIX Commands Reference*.

Fichiers DOS

AIX permet d'exploiter des fichiers DOS. Il vous suffit de copier les fichiers DOS voulus sur une disquette. Si vous lancez les commandes adéquates, le système lit les fichiers en les intégrant à un répertoire AIX. Une sauvegarde sur disquette DOS est ensuite effectuée.

Remarque : Les caractères joker * et ? (astérisque et point d'interrogation) ne sont pas admis pour ces commandes (bien qu'ils soient acceptés par le shell AIX). Si vous omettez l'extension d'un fichier, le fichier recherché est supposé ne pas avoir d'extension.

Cette section traite des points suivants :

- Copie de fichiers DOS dans des fichiers AIX, page 6-19.
- Copie de fichiers AIX dans des fichiers DOS, page 6-19.
- Suppression de fichiers DOS, page 6-20.
- Contenu d'un répertoire DOS, page 6-20.

Copie de fichiers DOS dans des fichiers AIX

La commande **dosread** copie le fichier DOS spécifié dans le fichier AIX spécifié.

Remarque : Les conventions d'appellation des fichiers DOS s'appliquent, à une exception près : le caractère barre oblique inversée (\) ayant une signification particulière pour AIX, il doit être remplacé par une barre oblique simple (/) pour délimiter les sous-répertoires dans un chemin d'accès DOS.

Par exemple, pour copier le fichier texte `chap1.doc` d'une disquette DOS dans le système de fichiers AIX, entrez :

```
dosread -a chap1.doc chap1
```

Le fichier texte DOS `.DOC` de l'unité par défaut **/dev/fd0** est copié dans le fichier AIX `chap1`, dans le répertoire courant.

Par exemple, pour copier un fichier binaire d'une disquette DOS dans le système de fichiers AIX, entrez :

```
dosread -D/dev/fd1 /survey/test.dta /home/fran/testdata
```

Le fichier de données DOS `.DTA` sur **/dev/fd1** est copié dans le fichier AIX `/home/fran/testdata`.

Reportez-vous à la commande **dosread** dans le manuel *AIX Commands Reference*.

Copie de fichiers AIX dans des fichiers DOS

La commande **doswrite** copie le fichier AIX spécifié dans le fichier DOS spécifié.

Remarque : Les conventions DOS d'appellation des fichiers s'appliquent, à une exception près : le caractère barre oblique inversée (\) ayant une signification particulière pour AIX, il doit être remplacé par une barre oblique simple (/) pour délimiter les sous-répertoires dans un chemin d'accès DOS.

Par exemple, pour copier le fichier texte `chap1` du système de fichiers AIX sur une disquette DOS, entrez :

```
doswrite -a chap1 chap1.doc
```

Le fichier AIX `chap1` du répertoire courant est copié dans le fichier texte DOS `.DOC` sur **/dev/fd0**.

Par exemple, pour copier le fichier binaire `/survey/test.dta` du système de fichiers AIX sur une disquette DOS, entrez :

```
doswrite -D/dev/fd1 /home/fran/testdata /survey/test.dta
```

Le fichier de données AIX `/home/fran/testdata` est copié dans le fichier DOS `\SURVEY\TEST.DTA` sur **/dev/fd1**.

Reportez-vous à la commande **doswrite** dans le manuel *AIX Commands Reference*.

Suppression de fichiers DOS

La commande **dosdel** supprime le fichier DOS spécifié.

Remarque : Les conventions d'appellation des fichiers DOS s'appliquent, à une exception près : le caractère barre oblique inversée (`\`) ayant une signification particulière pour AIX, il doit être remplacé par une barre oblique simple (`/`) pour délimiter les sous-répertoires dans un chemin d'accès DOS.

La commande **dosdel** convertit en majuscules les minuscules apparaissant dans le nom du fichier ou du répertoires avant de vérifier le disque. Tous les chemins d'accès étant supposés absolus (et non relatifs), la barre oblique (`/`) initiale est inutile.

Par exemple, pour supprimer le fichier DOS `file.ext` sur l'unité par défaut (**/dev/fd0**), entrez :

```
dosdel file.ext
```

Reportez-vous à la commande **dosdir** dans le manuel *AIX Commands Reference*.

Contenu d'un répertoire DOS

La commande **dosdir** affiche des informations sur les fichiers et les répertoires DOS spécifiés.

Remarque : Les conventions d'appellation des fichiers DOS s'appliquent, à une exception près : le caractère barre oblique inversée (`\`) ayant une signification particulière pour AIX, il doit être remplacé par une barre oblique simple (`/`) pour délimiter les sous-répertoires dans un chemin d'accès DOS.

La commande **dosdir** convertit en majuscules les minuscules apparaissant dans le nom du fichier ou du répertoires avant de vérifier le disque. Tous les chemins d'accès étant supposés absolus (et non relatifs), la barre oblique (`/`) initiale est inutile.

Par exemple, pour afficher les fichiers DOS d'un répertoire sur **/dev/fd0**, entrez :

```
dosdir
```

La commande affiche le nom des fichiers, et la taille de l'espace disponible.

```
PG3-25.TXT  
PG4-25.TXT  
PG5-25.TXT  
PG6-25.TXT  
Free space: 312320 bytes
```

Reportez-vous à la commande **dosdir** dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives aux fichiers

*	caractère joker, remplace un nombre quelconque de caractères.
?	caractère joker, remplace un caractère unique quelconque.
[]	métacaractère, correspond aux caractères entre crochets.

Procédures de gestion de fichier

cat	Concatène ou affiche des fichiers.
cmp	Compare deux fichiers.
colrm	Supprime des colonnes d'un fichier.
cp	Copie des fichiers.
cut	Ecrit les octets, les caractères ou les champs sélectionnées de chaque ligne d'un fichier.
del	Supprime des fichiers (avec confirmation).
diff	Compare des fichiers texte.
file	Détermine le type d'un fichier.
find	Recherche les fichiers correspondant à une expression.
grep	Recherche les fichiers correspondant à un modèle.
head	Affiche les premières lignes ou les premiers octets d'un ou de plusieurs fichiers.
more	Affiche le texte en continu écran par écran.
mv	Déplace des fichiers.
nl	Numérote les lignes d'un fichier.
pg	Formate des fichiers en fonction de l'écran.
rm	Supprime (dissocie) des fichiers ou des répertoires.
paste	Fusionne les lignes de plusieurs fichiers ou les lignes suivantes d'un fichier.
page	Affiche le texte en continu écran par écran.
sort	Trie des fichiers, les fusionne et détermine si des fichiers sont triés ou non.
tail	Ecrit un fichier sur l'unité de sortie standard, à partir d'un point donné.
wc	Décompte le nombre de lignes, de mots et d'octets d'un fichier.

Liaison de fichiers et de répertoires

ln	Lie fichiers et répertoires.
-----------	------------------------------

Fichiers DOS

dosdel	Supprime des fichiers DOS.
dosdir	Affiche les fichiers DOS du répertoire.
dosread	Copie des fichiers DOS dans des fichiers AIX.
doswrite	Copie des fichiers AIX dans des fichiers DOS.

Chapitre 7. Imprimantes, travaux d'impression et files d'attente

Vous pouvez, en fonction de l'imprimante, contrôler la présentation et les caractéristiques des sorties papier. L'imprimante, d'une part, la console et l'unité centrale, d'autre part, peuvent se trouver dans des lieux distincts. L'imprimante peut être directement connectée à un système local, mais un travail d'impression peut également être envoyé via un réseau à un système distant.

Pour gérer les travaux d'impression avec un maximum d'efficacité, le système place chaque travail dans une file d'attente, jusqu'à ce que l'imprimante soit disponible. Un ou plusieurs travaux peuvent ainsi se trouver en file d'attente : tandis qu'un travail est en cours d'impression, le système traite le travail suivant de la file, et ce, jusqu'à ce que tous les travaux soient imprimés.

AIX Guide to Printers and Printing contient des informations sur les imprimantes, les travaux d'impression et les files d'attente.

Ce chapitre traite des points suivants :

- Terminologie, page 7-2.
- Lancement d'un travail d'impression (commande **qprt**), page 7-4.
- Annulation d'un travail d'impression (commande **qcan**), page 7-7.
- Vérification de l'état d'un travail (commande **qchk**), page 7-8.
- Conditions de l'état de l'imprimante, page 7-9
- Définition de la priorité d'un travail (commande **qpri**), page 7-10.
- Blocage et libération d'un travail (commande **qhld**), page 7-11
- Déplacement d'un travail vers une autre file d'attente (commande **qmov**), page 7-12
- Formatage des fichiers à imprimer (commande **pr**), page 7-13.
- Impression de fichiers ASCII sur une imprimante PostScript, page 7-14.
- Automatisation de la conversion ASCII–PostScript, page 7-16.
- Annulation de la détermination automatique du type de fichier, page 7-16.
- Récapitulatif des commandes relatives à l'impression, page 7-17.

Terminologie

Voici une définition des principaux termes relatifs à l'impression.

Travail d'impression

Un *travail d'impression* est une unité de travail destinée à l'imprimante. Il peut être constitué d'un ou de plusieurs fichiers, selon la demande émise. Le système affecte un numéro unique à chaque travail qu'il exécute.

File d'attente

La *file d'attente* est l'endroit vers lequel sont dirigés les travaux d'impression. Il s'agit d'une strophe du fichier **/etc/qconfig** dont le nom est celui de la file d'attente, qui pointe sur l'unité de file d'attente associée. Voici un exemple :

```
Msal:
    device = lp0
```

`Msal` est le nom de la file d'attente, `lp0`, le nom de l'unité.

Unité de file d'attente

La strophe relative à l'*unité de file d'attente* suit normalement celle relative à la file d'attente locale dans le fichier **/etc/qconfig**. Elle indique le fichier **/dev** (unité imprimante) destinataire de l'impression et le programme expéditeur à utiliser. Voici un exemple :

```
lp0:
    file = /dev/lp0
    header = never
    trailer = never
    access = both
    backend = /usr/lpd/piobe
```

`lp0` est le nom de l'unité, les autres lignes définissant son mode d'utilisation.

Remarque : Plusieurs unités de file d'attente peuvent être associées à une seule file.

qdaemon

qdaemon est un process d'arrière-plan qui contrôle les files d'attente. Il est généralement activé à la mise sous tension du système.

Spouleur d'impression

Le *spouleur* n'est pas réservé aux travaux d'impression. Il offre une fonction de traitement différé générique, permettant la mise en file d'attente de différents types de travaux, y compris bien sûr des travaux d'impression.

Le spouleur ne "sait" normalement pas quel type de travail est mis en attente. Lorsque l'administrateur définit une file d'attente, son objet est déterminé par le programme expéditeur spécifié pour la file d'attente. Par exemple, si le programme expéditeur est la commande **piobe** (programme d'E/S de l'imprimante), il s'agit d'une file d'attente d'impression. De même, si le programme expéditeur est un compilateur, il s'agit d'une file d'attente de travaux à compiler. Lorsque la commande **qdaemon** du spouleur sélectionne un travail de la file, il exécute le travail en appelant le programme expéditeur – spécifié par l'administrateur lors de la définition de la file d'attente.

Le spouleur principal contrôle la commande **enq**. Bien que vous puissiez faire directement appel à cette commande pour mettre un travail en file d'attente, vous disposez pour ce faire de trois commandes frontales : **lp**, **lpr** et **qprt**. Une demande émise par le biais d'une de ces commandes est d'abord transmise au programme **enq**, lequel place les informations sur le fichier dans la file d'attente, en vue de son traitement par le process **qdaemon**.

Imprimante réelle

Une *imprimante réelle* est l'imprimante matérielle connectée à un port série ou parallèle à une adresse matérielle unique. Le pilote de l'unité d'impression dans le noyau communique avec la partie matérielle de l'imprimante et constitue une interface entre l'imprimante réelle et une imprimante virtuelle, mais ne connaît pas le concept d'imprimante virtuelle.

Imprimante virtuelle

Une *imprimante virtuelle* est un ensemble d'attributs qui définissent une vue logicielle d'une imprimante réelle. Cette vue ne référence que les flux de données de haut niveau (ASCII, PostScript, etc.) reconnus par l'imprimante. Elle ne comporte aucune information sur la connexion matérielle entre l'imprimante et l'hôte, ou le protocole de transfert de et vers l'imprimante. Les imprimantes virtuelles sont définies par le gestionnaire du système.

Imprimantes locales et distantes

Lorsque vous branchez une imprimante sur un nœud ou sur un hôte, elle est dite *imprimante locale*. Un *système d'impression à distance* donne aux nœuds non directement connectés accès à l'imprimante.

Pour disposer des fonctions d'impression à distance, les nœuds individuels doivent être connectés à un réseau via le protocole TCP/IP (Transmission Control Protocol/Internet Protocol) et prendre en charge les applications TCP/IP requises.

Programme expéditeur de l'imprimante

Le *programme expéditeur de l'imprimante* est un ensemble de programmes appelés par la commande **qdaemon** du spouleur pour traiter un travail en attente d'impression. Ce programme :

- reçoit de la commande **qdaemon** la liste des fichiers à imprimer ;
- applique les attributs imprimante et de formatage issus de la base de données, éventuellement remplacés par les indicateurs spécifiés sur la ligne de commande ;
- initialise l'imprimante avant d'imprimer un fichier ;
- lance au besoin les filtres requis pour convertir le flux des données dans un format pris en charge par l'imprimante ;
- fournit les filtres requis pour le formatage simple des documents ASCII ;
- assure le support des caractères nationaux ;
- transmet les données d'impression filtrées au pilote de l'unité d'impression ;
- génère les pages de début et de fin ;
- génère les exemplaires multiples ;
- signale les incidents (absence de papier, intervention requise, erreur imprimante, etc.) ;
- signale les incidents décelés par les filtres ;
- effectue un nettoyage après l'annulation d'un travail d'impression ;
- fournit un environnement d'impression, personnalisable par l'administrateur système.

Lancement d'un travail d'impression (commande `qprt`)

Pour lancer un travail d'impression, vous disposez des commandes `qprt` et `smit`.

- le nom du fichier à imprimer,
- le nom de la file d'attente d'impression,
- le nom du tiroir de sortie,
- le nombre d'exemplaires à imprimer,
- si une copie du fichier doit être effectuée sur l'hôte distant,
- si le fichier doit être effacé après impression,
- si une notification sur l'état du travail doit être envoyée,
- si une notification sur l'état du travail doit être envoyée via la messagerie,
- si l'impression doit être continue ou non,
- le nom utilisateur destiné à l'étiquette "Delivery To",
- le message d'accusé de réception console pour les impressions à distance,
- le message d'accusé de réception fichier pour les impressions à distance,
- le niveau de priorité.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Commande `qprt`

La commande `qprt` crée et met en file d'attente un travail pour imprimer le(s) fichier(s) spécifié(s). Un travail d'impression peut porter sur plusieurs fichiers, qui sont alors imprimés dans l'ordre indiqué sur la ligne de commande.

Pour imprimer un fichier, vous devez pouvoir y accéder en lecture. Pour supprimer un fichier après impression, vous devez détenir le droit d'écriture sur le répertoire qui le contient.

Elle respecte la syntaxe :

-b <i>Nombre</i>	Marge inférieure. Nombre de lignes blanches à réserver en bas de page.
-B <i>Valeur</i>	Mode de séparation des pages (papier en continu séparé par des perforations). La variable <i>Valeur</i> est constituée de deux caractères. Le premier est relatif aux pages d'en-tête. Le second caractère est relatif aux pages de fin. Ces caractères sont les suivants :
a	Imprime une page (en-tête ou fin) pour chaque fichier du travail d'impression.
n	N'imprime pas de page d'en-tête ni de fin.
g	Imprime une seule page (en-tête ou fin) pour chaque travail (ensemble de fichiers).

Par exemple, l'indicateur **-B ga** imprime une page d'en-tête au début de chaque travail d'impression, et une page de fin à la fin de chaque fichier de chaque travail.

Remarque : Dans le cas d'une impression à distance, la valeur par défaut est déterminée par la file d'attente distante sur le serveur.

-e Option	Impression en gras (double frappe).
	+ impression en gras
	! impression en taille normale
-E Option	impression en double hauteur
	+ impression en double hauteur
	! impression en taille normale
-f Fichier	Filtre (1 caractère) à appliquer au(x) fichier(s) avant impression : p , qui appelle le filtre pr , ou n , qui traite les sorties générées par la commande troff .
-i Nombre	Indentation de lignes (en nombre d'espaces). La variable <i>Nombre</i> doit être inférieure à la largeur de page spécifiée via l'indicateur -w .
-K Option	Impression condensée
	+ impression condensée
	! impression normale
-l Nombre	Hauteur de page (en nombre de lignes). Si vous indiquez 0 pour <i>Nombre</i> , le paramètre est ignoré et la sortie est considérée comme une seule page continue. La hauteur inclut les marges inférieure et supérieure, et indique la hauteur de papier imprimable.
-L Option	Passage à la ligne ou troncature des lignes plus longues que la largeur de page.
	+ Passage à la ligne des lignes longues.
	! Lignes longues tronquées au niveau de la marge droite.
-N Nombre	Nombre d'exemplaires à imprimer. En l'absence de spécification de cet indicateur, un seul exemplaire est imprimé.
-p Nombre	Pas d'impression en <i>Nombre</i> de caractères par pouce. Les valeurs habituelles de <i>Nombre</i> sont 10 et 12. Le pas réel d'impression dépend également de la valeur des indicateurs -K (condensée) et -W (double largeur).
-P FileAttente [:UnitéFile Attente]	File d'attente d'impression et (facultatif) unité de file d'attente (par défaut, imprimante par défaut).
-Q Valeur	Taille du papier. La <i>Valeur</i> à indiquer dépend de l'imprimante. Les valeurs classiques sont : 1 (letter), 2 (legal), etc. Reportez-vous au manuel de l'imprimante.
-t Nombre	Marge supérieure. Nombre de lignes blanches à réserver en tête de page.
-w Nombre	Largeur de page en nombre de caractères (défini par la variable <i>Nombre</i>). La largeur indiquée doit être supérieure à la valeur d'indentation spécifiée par l'indicateur -i .
-W Option	impression en double hauteur
	+ impression en double largeur
	! impression normale
-z Valeur	Imprime avec une rotation d'autant de quarts de tour qu'indiqué par <i>Valeur</i> (dans le sens horaire). La hauteur (-l) et la largeur (-w) de page sont automatiquement modifiées en conséquence.
	0 Portrait
	1 Paysage à droite
	2 Portrait à l'envers
	3 Paysage à gauche

-= TiroirSortie	Spécifie le tiroir de sortie destinataire d'un travail d'impression. La liste des valeurs possibles est indiquée ci-dessous. Toutefois, les tiroirs de sortie autorisés dépendent de l'imprimante.
0	Tiroir supérieur de l'imprimante.
1-49	Tiroirs à grande capacité 1 – 49
-# Valeur	Fonction spéciale.
j	Affiche le numéro du travail spécifié.
h	Met le travail en file d'attente, mais à l'état HELD (bloqué).
v	Valide les valeurs des indicateurs du programme expéditeur de l'imprimante. Cette validation sert notamment à détecter les indicateurs légaux, au moment de la soumission d'un travail. A défaut, un travail assorti d'un indicateur incorrect ne sera interrompu qu'ultérieurement, en cours de traitement.

Par exemple, pour imprimer le fichier `myfile` sur la première imprimante disponible, configurée pour la file d'attente par défaut, avec les valeurs par défaut, entrez :

```
qprt myfile
```

Par exemple, pour imprimer le fichier `somefile` sur une file d'attente donnée, avec des indicateurs spécifiques, à valider au moment de la soumission, entrez :

```
qprt -f p -e + -Pfastest -# v somefile
```

Le fichier `somefile` passe par le filtre de la commande **pr** (indicateur **-f p**) et est imprimé en gras (indicateur **-e +**) sur la première imprimante disponible, configurée pour la file d'attente **fastest** (indicateur **-Pfastest**).

Par exemple, pour imprimer `myfile` sur du papier de taille "legal", entrez :

```
qprt -Q2 myfile
```

Pour imprimer trois exemplaires de chacun des fichiers `new.index.c`, `print.index.c` et `more.c` sur la file d'attente `Msp1`, entrez :

```
qprt -PMsp1 -N 3 new.index.c print.index.c more.c
```

Pour imprimer trois exemplaires de la concaténation des trois fichiers `new.index.c`, `print.index.c` et `more.c`, entrez :

```
cat new.index.c print.index.c more.c | qprt -PMsp1 -N 3
```

Remarque : AIX prend également en charge la commande d'impression UNIX BSD (**lpr**) et celle d'UNIX System V (**lp**). Reportez-vous aux commandes **lpr** et **lp** dans le manuel *AIX Commands Reference*.

Reportez-vous à la commande **qprt** dans le manuel *AIX Commands Reference*.

Commande smit

Vous pouvez également lancer la commande **qprt** par le biais de **smit**. A l'invite, tapez :

```
smit qprtet appuyez sur Entrée.
```

Annulation d'un travail d'impression (commande **qcan**)

Pour annuler un travail dans la file d'attente, vous disposez du raccourci Web-based System Manager ou des commandes **qcan** et **smit**. Lorsque vous annulez un travail, vous devez indiquer le nom de sa file d'attente et le numéro du travail.

Cette procédure s'applique aux impressions locales et à distance.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Web-based System Manager Raccourci

Pour annuler un travail d'impression avec le raccourci Web-based System Manager, tapez :

```
wsm printers
```

Dans le conteneur Print Queues, sélectionnez le travail d'impression, puis servez-vous des menus pour le supprimer de la file d'attente.

Commande **qcan**

La commande **qcan** permet d'annuler un travail (via son numéro) d'une file d'attente locale ou distante, ou tous les travaux d'une file d'attente locale. Pour connaître le numéro d'un travail, lancez la commande **qchk**.

Elle respecte la syntaxe :

```
qcan -PFileAttente -x NuméroTravail
```

Reportez-vous à la commande **qcan** dans le manuel *AIX Commands Reference*.

Par exemple, pour annuler le travail numéro 123 quelle que soit l'imprimante sur laquelle il se trouve, entrez :

```
qcan -x 123
```

Pour annuler tous les travaux de la file d'attente de l'imprimante lp0, entrez :

```
qcan -X -Plp0
```

Remarque : AIX prend également en charge la commande d'annulation UNIX BSD (**lprm**) et celle d'UNIX System V (**cancel**). Reportez-vous aux commandes **lprm** et **cancel** dans le manuel *AIX Commands Reference*.

Commande **smit**

Pour annuler un travail d'impression avec SMIT, tapez :

```
smit qcan
```

Vérification de l'état d'un travail (commande qchk)

Pour afficher des informations sur l'état d'un travail, d'une file d'attente, d'une imprimante ou d'un utilisateur, vous disposez du raccourci Web-based System Manager, et des commandes **qchk** et **smit**.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Raccourci Web-based System Manager

Pour vérifier l'état d'un travail d'impression avec le raccourci Web-based System Manager, tapez :

```
wsm printers
```

Dans le conteneur Print Queues, sélectionnez le travail d'impression, puis vérifiez son état à l'aide des menus.

Commande qchk

La commande **qchk** permet d'afficher des informations sur l'état d'un travail, d'une file d'attente ou d'un utilisateur.

Elle respecte la syntaxe :

```
qchk -P FileAttente -# NuméroTravail -u Propriétaire
```

Reportez-vous à la commande **qchk** dans le manuel *AIX Commands Reference*.

Par exemple, pour afficher la file d'attente par défaut, entrez :

```
qchk -q
```

Par exemple, pour afficher l'état complet de toutes les files d'attente jusqu'à ce qu'elles soient vides, avec mise à jour de l'écran toutes les 5 secondes, tapez :

```
qchk -A -L -w 5
```

Par exemple, pour afficher l'état de la file d'attente lp0, entrez :

```
qchk -P lp0
```

Pour afficher l'état du travail 123, entrez :

```
qchk -# 123
```

Pour vérifier l'état de tous les travaux dans toutes les files d'attente, entrez :

```
qchk -A
```

Remarque : AIX prend également en charge la commande de contrôle de file d'attente UNIX BSD (**lpq**) et celle d'UNIX System V (**lpstat**). Reportez-vous aux commandes **lpq** et **lpstat** dans le manuel *AIX Commands Reference*.

Commande smit

Pour vérifier l'état d'un travail d'impression à l'aide de SMIT, entrez :

```
smit qchk
```

Etat de l'imprimante

Voici quelques états possibles :

DEV_BUSY	Indique que : <ul style="list-style-type: none">• plusieurs files d'attente sont affectées à l'unité d'impression (lp0) et une file d'attente utilise actuellement l'unité ;• qdaemon a tenté d'utiliser l'unité d'impression (lp0), mais une autre application l'utilise déjà. Intervention : Attendez que la file d'attente ou l'application libère l'unité d'impression, ou annulez le travail ou le process qui utilise le port imprimante.
DEV_WAIT	Indique que la file d'attente ne peut disposer de l'imprimante car celle-ci est hors ligne, manque de papier, accuse un bourrage, etc., ou encore que son câble est détaché, endommagé ou mal branché. Intervention : Remédiez à la cause de l'incident. Vous serez peut-être obligé de retirer au préalable les travaux de la file d'attente. Une file d'attente à l'état DEV_WAIT passe à l'état DOWN au bout d'un délai défini (en secondes).
DOWN	Une file d'attente à l'état DEV_WAIT passe généralement à l'état DOWN . Cette situation se produit lorsque le pilote d'imprimante ne peut signaler la présence de l'imprimante, par manque de système de signalisation correct. Certaines imprimantes étant incapables de signaler au système de files d'attente qu'elles sont hors ligne transmettent alors le signal hors tension. Or, qu'une unité d'impression soit ou semble hors tension déclenche la mise à l'état DOWN de la file d'attente. Intervention : Remédiez à la cause de l'incident et demandez à l'administrateur système de vous fournir la sauvegarde de la file d'attente. Elle <i>doit</i> être placée manuellement avant toute utilisation.
HELD	Indique un travail à l'état bloqué. Autrement dit, il ne sera traité par le spouleur qu'une fois libéré.
QUEUED	Indique qu'un fichier d'impression se trouve en file d'attente, prêt à être imprimé.
READY	Indique que tout est prêt pour mettre en file d'attente et imprimer un travail.
RUNNING	Indique qu'un fichier est en cours d'impression.

Définition de la priorité d'un travail (commande qpri)

Pour modifier la priorité d'un travail, vous disposez du raccourci Web-based System Manager ou des commandes **qpri** et **smit**. Vous ne pouvez définir la priorité des travaux que sur des files d'attente locales. Plus le chiffre est élevé, plus la priorité du travail est grande. La priorité par défaut est de 15, la priorité maximale est de 20 pour un utilisateur normal, de 30 pour un utilisateur racine ou un membre du groupe printq (groupe 0).

Remarque : Vous ne pouvez définir la priorité d'un travail d'impression à distance.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Raccourci Web-based System Manager

Pour vérifier l'état d'un travail d'impression avec le raccourci Web-based System Manager, tapez :

```
wsm printers
```

Dans le conteneur Print Queues, sélectionnez le travail d'impression, puis les menus pour définir la priorité pour un travail donné dans une file d'attente locale.

Commande qpri

La commande **qpri** permet de modifier la priorité d'un travail que vous avez soumis. Si vous disposez des droits de l'utilisateur racine ou que vous appartenez au groupe printq, vous pouvez redéfinir la priorité de n'importe quel travail de la file d'attente.

Elle respecte la syntaxe :

```
qpri -# NuméroTravail -a Priorité
```

Par exemple, pour passer à 18 la priorité du travail 123, entrez :

```
qpri -# 123 -a 18
```

Pour définir la priorité d'un travail local au moment de sa soumission, entrez :

```
qpri -PFileAttente -R Priorité Fichier
```

Reportez-vous à la commande **qpri** dans le manuel *AIX Commands Reference*.

Commande smit

Pour modifier l'état d'un travail d'impression avec SMIT, tapez :

```
smit qpri
```

Blocage et libération d'un travail (commande qhld)

Un travail envoyé en file d'attente peut être bloqué via le raccourci Web-based System Manager, ou via les commandes **qhld** ou **smit**. Les mêmes commandes permettent de le libérer ultérieurement.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Web-based System Manager Raccourci

Pour bloquer ou libérer un travail d'impression avec le raccourci Web-based System Manager, tapez :

```
wsm printers
```

Dans le conteneur Print Queues, sélectionnez le travail d'impression, puis servez-vous des menus pour le bloquer ou le libérer pour l'impression.

Commande qhld

La commande **qhld** permet de bloquer un travail envoyé en file d'attente. Vous pouvez bloquer un travail donné ou l'ensemble des travaux d'une file d'attente. Pour connaître le numéro d'un travail, lancez la commande **qchk**.

Elle respecte la syntaxe :

```
qhld [ -r ] {[ -#NoTrav ] [ -PFileAttente ] [ -uUtilisateur ] }
```

Reportez-vous à la commande **qhld** dans le manuel *AIX Commands Reference*.

Par exemple, pour bloquer le travail 452 quelle que soit la file d'attente sur laquelle il se trouve, entrez :

```
qhld -#452
```

Pour bloquer tous les travaux de la file d'attente `hp2`, entrez :

```
qhld -Php2
```

Pour libérer le travail 452 quelle que soit la file d'attente sur laquelle il se trouve, entrez :

```
qhld -#452 -r
```

Pour libérer tous les travaux de la file d'attente `hp2`, entrez :

```
qhld -Php2 -r
```

Commande smit

Pour bloquer ou libérer un travail d'impression avec SMIT, tapez :

```
smit qhld
```

Déplacement d'un travail vers une autre file d'attente (commande **qmov**)

Après avoir envoyé un travail d'impression dans une file d'attente, vous pouvez décider de déplacer ce travail vers une autre file d'attente. Pour cela, ayez recours au raccourci Web-based System Manager ou encore aux commandes **qmov** ou **smit**.

Prérequis

- Pour les travaux locaux, l'imprimante doit être physiquement reliée à votre système.
- Pour les travaux à distance, le système doit être configuré pour communiquer avec le serveur d'impression distant.

Raccourci Web-based System Manager

Pour déplacer un travail d'impression vers une autre file d'attente à l'aide du raccourci Web-based System Manager, entrez :

```
wsm printers
```

Dans le conteneur Print Queues, sélectionnez le travail d'impression, puis servez-vous des menus pour le déplacer d'une file d'attente vers une autre.

Commande **qmov**

La commande **qmov** permet de déplacer un travail dans une autre file d'attente. Vous pouvez déplacer un travail donné ou tous les travaux d'une file d'attente, ou encore tous les travaux soumis par un utilisateur donné. Pour connaître le numéro d'un travail, lancez la commande **qchk**.

Elle respecte la syntaxe :

```
qmov  
-m  
NouvFile {[ -#NuméroTravail ] [ -PFile ] [ -uUtilis ] }
```

Reportez-vous à la commande **qmov** dans le manuel *AIX Commands Reference*.

Par exemple, pour déplacer le travail 280 dans la file d'attente `hp2`, entrez :

```
qmov -mhp2 -#280
```

Par exemple, pour déplacer tous les travaux de la file d'attente `hp4D` dans la file d'attente `hp2`, entrez :

```
qmov -mhp2 -Php4D
```

Commande **smit**

Pour déplacer un travail d'impression avec SMIT, tapez :

```
smit qmov
```

Formatage des fichiers à imprimer (commande **pr**)

La commande **pr** permet d'effectuer un formatage simple des fichiers à imprimer. Chaînez la sortie de la commande **pr** à la commande **qprt** pour formater votre texte.

Voici quelques indicateurs de la commande **pr** :

-d	Génère une sortie en double espacement.
-h "Chaîne"	Affiche la chaîne spécifiée, entre guillemets (" "), au lieu du nom du fichier, comme en-tête de page. Indicateur et chaîne doivent être séparés par un espace.
l Lignes	Redéfinit la hauteur de page (66 lignes, par défaut) au nombre de <i>Lignes</i> indiqué. Si ce nombre est inférieur à la somme des lignes de l'en-tête et du bas de page, ces derniers sont supprimés (même effet que l'indicateur -t).
-m	Fusionne des fichiers. La sortie standard est formatée de sorte que la commande pr écrive une ligne de chaque fichier spécifié par une variable <i>Fichier</i> , côte à côte, en colonnes de texte de même taille (sur la base du nombre total de colonnes). Cet indicateur est incompatible avec l'indicateur -Colonne .
-n [Largeur] [Caractère]	Numérote les lignes selon le format spécifié par la variable <i>Largeur</i> . La valeur par défaut est de 5 chiffres. Si <i>Caractère</i> (caractère non numérique quelconque) est spécifié, il est ajouté au numéro de ligne, pour marquer la séparation entre ce numéro et le reste de la ligne. Le caractère de séparation par défaut est le caractère ASCII TAB.
-o Décalage	Indente chaque ligne du nombre d'espaces indiqué par <i>Décalage</i> . Le nombre total de caractères par ligne est la somme de la largeur de page et du décalage. La valeur par défaut de <i>Décalage</i> est 0.
-sCaractère	Sépare les colonnes par le <i>Caractère</i> au lieu d'insérer des espaces. Le caractère de séparation par défaut est le caractère ASCII TAB.
-t	N'affiche ni les cinq lignes d'en-tête, ni les cinq lignes de bas de page. S'arrête à la dernière ligne de la page sans créer d'espace jusqu'à la fin de la page.
-w Largeur	Nombre de colonnes par ligne. La valeur par défaut est 72, générant une sortie multi-colonne de même taille. Aucune autre limite n'est imposée. Si l'indicateur -w n'est pas spécifié et que l'indicateur -s l'est, la largeur par défaut est de 512 colonnes.
-Colonne	Nombre de colonnes. La valeur par défaut est 1. Cette option est incompatible avec l'indicateur -m . Les indicateurs -e et -i sont supposés actifs pour les sorties multicolonnées. Une colonne de texte ne doit pas dépasser la largeur de la page (voir indicateur -l). Lorsque cet indicateur est combiné à l'indicateur -t , optez pour le nombre de lignes minimal pour la sortie.
+Page	Commence par afficher le numéro de page spécifié par <i>Page</i> . La valeur par défaut est 1.

Par exemple, pour imprimer le fichier `prog.c` avec en-têtes et numéros de page, entrez :

```
pr prog.c | qprt
```

Des en-têtes sont inclus dans le fichier `prog.c`, qui est passé à la commande **qprt**. L'en-tête est constitué de la date de dernière modification du fichier, de son nom et du nombre de pages.

Par exemple, pour indiquer le titre du fichier `prog.c`, entrez :

```
pr -h "PROGRAMME PRINCIPAL" prog.c | qprt
```

`prog.c` est imprimé sous le titre `PROGRAMME PRINCIPAL` (à la place du nom de fichier).
Date de modification et numéro de page sont toujours imprimés.

Par exemple, pour imprimer le fichier `word.lst` sur plusieurs colonnes, entrez :

```
pr -3 word.lst | qprt
```

Le fichier `word.lst` est imprimé sur trois colonnes.

Pour imprimer plusieurs fichiers côte à côte :

```
pr -m -h "Membres et visiteurs" member.lst visitor.lst | qprt
```

Les fichiers `member.lst` et `visitor.lst` sont imprimés côte à côte sous le titre `Membres et visiteurs`.

Pour modifier le fichier `prog.c` en vue d'un usage ultérieur, entrez :

```
pr -t -e prog.c > prog.notab.c
```

Les tabulations du fichier `prog.c` sont remplacées par des espaces, et le résultat placé dans `prog.notab.c`. Les tabulations sont positionnées aux colonnes 9, 17, 25, 33, etc. L'indicateur `-e` indique à la commande `pr` de remplacer les tabulations, l'indicateur `-t`, de supprimer les en-têtes de page.

Pour imprimer le fichier `myfile` sur deux colonnes, en texte de 7 points, entrez :

```
pr -l66 -w172 -2 myfile | qprt -z1 -p7
```

Reportez-vous à la commande `pr` dans le manuel *AIX Commands Reference*.

Impression de fichiers ASCII sur une imprimante PostScript

Le système de formatage de texte TFS (Text Formatting System) contient le filtre "enscript" qui convertit les fichiers d'impression ASCII en PostScript, pour impression sur une imprimante PostScript. Ce filtre est appelé par la commande `qprt -da` lors de la soumission d'un travail à une file d'attente PostScript.

Prérequis

- L'imprimante doit être physiquement reliée à votre système.
- L'imprimante doit être configurée et définie.
- La partie transcript des services TFS doit être installée.

Plusieurs indicateurs peuvent être associés à la commande `qprt`, pour personnaliser la sortie de fichiers ASCII soumis à une file d'attente PostScript.

<code>-1+</code>	Ajoute des en-têtes de page.
<code>-2+</code>	Formate la sortie sur deux colonnes.
<code>-3+</code>	Imprime les en-têtes de page, les dates et les numéros de page dans un style sophistiqué. On parle quelquefois de mode "gaudy".
<code>-4+</code>	Imprime le fichier, même s'il comporte des caractères non imprimables.
<code>-5+</code>	Liste les caractères absents d'une police.
<code>-h chaîne</code>	Indique la chaîne constituant l'en-tête de page. A défaut, l'en-tête reprend le nom du fichier, la date de sa dernière modification et le nombre de pages.
<code>-l valeur</code>	Nombre maximal de lignes par page. Un nombre de lignes moindre peut être effectivement imprimé, si la taille (en points) des caractères l'exige.

- L! Tronque les lignes plus longues que la largeur de page.
- p Taille en points. La valeur par défaut est de 10, sauf en mode deux colonnes avec rotation (-2+ -z1), pour lequel la valeur par défaut est de 7.
- s Style de police. En l'absence de cet indicateur, la police Courier est utilisée par défaut. Valeurs possibles :

- Courier-Oblique
- Helvetica
- Helvetica-Oblique
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- NewCenturySchlbk-Italic
- Optima
- Optima-Oblique
- Palatino-Roman
- Palatino-Italic
- Times-Roman
- Times-Italic.

Remarque : L'imprimante doit disposer de la police spécifiée.

- z1 Imprime en mode paysage (rotation de 90 degrés).

Par exemple, pour envoyer le fichier ACSII `myfile.ascii` sur l'imprimante PostScript `Msp1`, entrez :

```
qprt -da -PMsp1 myfile.ascii
```

Pour envoyer le fichier ACSII `myfile.ascii` sur l'imprimante PostScript `Msp1` et l'imprimer en Helvetica, entrez :

```
qprt -da -PMsp1 -sHelvetica myfile.ascii
```

Pour envoyer le fichier ASCII `myfile.ascii` sur l'imprimante PostScript `Msp1` et l'imprimer en corps 9, entrez :

```
qprt -da -PMsp1 -p9 myfile.ascii
```

Automatisation de la conversion ASCII–PostScript

La plupart des applications qui génèrent des fichiers d'impression PostScript suivent la convention d'appellation qui veut qu'un fichier d'impression PostScript commence toujours par les caractères %!. Pour configurer le système de sorte qu'il repère les fichiers d'impression ASCII soumis à une file d'attente PostScript et qu'il les convertisse automatiquement en fichiers PostScript, procédez comme suit.

1. A l'invite, entrez :

```
smit chpq
```

2. Indiquez une file d'attente PostScript ou sélectionnez–en une à l'aide de la fonction List.
3. Sélectionnez l'option **Printer Setup**.
4. Basculez la valeur de l'option **AUTOMATIC detection of print file TYPE to be done?** sur **yes**.

Les commandes suivantes sont alors équivalentes : elles convertissent tout fichier ASCII en fichier PostScript et l'impriment sur une imprimante PostScript. Ainsi, pour convertir et imprimer le fichier `myfile.ascii`, entrez l'une des commandes suivantes :

```
qpirt -Pps myfile.ps myfile.ascii
```

```
lpr -Pps myfile.ps myfile.ascii
```

```
lp -dps myfile.ps myfile.acsii
```

`ps` étant une file d'attente d'impression PostScript.

Annulation de la détermination automatique du type de fichier

Deux cas requièrent de passer outre la détermination automatique du type de fichier pour une impression PostScript.

Par exemple, pour imprimer le fichier PostScript `myfile.ps` (qui ne commence pas par les caractères %!), entrez :

```
qpirt -ds -Pps myfile.ps
```

Pour imprimer le listing source du fichier PostScript `myfile.ps` (qui commence par %!), entrez :

```
qpirt -da -Pps myfile.ps
```

Récapitulatif des commandes relatives à l'impression

cancel	Annule les demandes sur une imprimante ligne.
lp	Envoie une demande à une imprimante ligne.
lpq	Examine la file d'attente du spouleur.
lpr	Met en file d'attente les travaux d'impression.
lprm	Retire les travaux de la file d'attente de l'imprimante ligne.
lpstat	Affiche des informations sur l'état de l'imprimante ligne.
pr	Envoie un fichier vers l'unité de sortie standard.
qcan	Annule un travail d'impression.
qchk	Affiche l'état d'une file d'attente d'impression.
qhld	Bloque/libère un travail d'impression.
qmov	Déplace un travail vers une autre file d'attente.
qpri	Définit la priorité d'un travail dans la file d'attente.
qprt	Lance un travail d'impression.

Chapitre 8. Fichiers de sauvegarde et supports de stockage

Une fois que votre système fonctionne, il vous faut réfléchir aux moyens de sauvegarder systèmes de fichiers, répertoires et fichiers. Ces deux derniers représentent un investissement non négligeable en matière de temps et d'efforts. Sachant qu'il est toujours possible d'effacer ou de modifier un fichier – intentionnellement ou accidentellement, adopter une stratégie de sauvegarde efficace s'impose. En cas de problème, vous aurez ainsi toujours la ressource de restaurer la version la plus récente de vos fichiers et systèmes de fichiers.

Remarque : Si un disque "plante", toutes les données qu'il contient sont définitivement détruites. Le seul moyen de recouvrer ces données est de les restaurer à partir d'une copie de sauvegarde.

Il existe différentes méthodes de sauvegarde. Le plus souvent, vous opterez pour une sauvegarde simple, c'est-à-dire une copie du fichier, du répertoire ou du système de fichiers, permettant un transfert ou une restauration en cas de destruction ou de modification accidentelle. Il est également possible d'archiver une copie d'un ou de plusieurs fichiers, ou d'une base de données, pour un usage ultérieur, à des fins d'historique ou pour recouvrer les données en cas de perte ou de détérioration. L'archivage est généralement réservé à certains types de données.

Ce chapitre traite des points suivants :

- Stratégie de sauvegarde, page 8-2
 - Supports de sauvegarde, page 8-3
- Formatage de disquettes (commande **format** ou **fdformat**), page 8-5
- Vérification de l'intégrité du système de fichiers (commande **fsck**), page 8-6
- Copie de ou sur des disquettes (commande **flcopy**), page 8-6
- Copie de fichiers sur bande ou disque (commande **cpio -o**), page 8-7
- Copie de fichiers à partir de bande ou de disque (commande **cpio -i**), page 8-7
- Copie de et sur des bandes (commande **tcopy**), page 8-9
- Vérification de l'intégrité d'une bande (commande **tapechk**), page 8-9
- Compression de fichiers (commandes **compress** et **pack**), page 8-9
- Décompression de fichiers (commandes **uncompress** et **unpack**), page 8-11
- Sauvegarde de fichiers (commande **backup**), page 8-12
- Restauration de fichiers (commande **restore**), page 8-14
- Archivage de fichiers (commande **tar**), page 8-16
- Récapitulatif des commandes relatives à la sauvegarde, page 8-17

Stratégie de sauvegarde

Il n'existe pas une, mais plusieurs stratégies de sauvegarde – les utilisateurs ayant chacun leurs impératifs et leurs souhaits. Un système de sauvegarde adapté à un système mono-utilisateur a peu de chances de convenir sur un système servant 5 ou 10 utilisateurs. De même, une politique adaptée à un système où les fichiers sont modifiés quotidiennement sera inefficace sur un système où la fréquence de modification des données est faible. Vous seul êtes à même de déterminer la stratégie qui convient le mieux.

Garantissez-vous contre les pertes majeures

Votre système peut-il continuer à fonctionner si un seul disque tombe en panne ? Pouvez-vous recouvrer vos données si tous les disques tombent en panne ? Pouvez-vous reconstituer votre système si vous perdez vos disquettes ou vos bandes de sauvegarde, qu'elles sont brûlées ou volées ? Un tel acharnement du sort paraît peu vraisemblable, mais vous n'êtes à l'abri d'aucune des éventualités évoquées : réfléchissez donc au moyen de recouvrer l'usage de votre système dans chacun des cas de figure.

Vérifiez régulièrement vos sauvegardes

Supports et unités de sauvegarde, même les plus fiables, ne sont pas à l'abri d'une défaillance. Une bibliothèque conséquente de bandes ou de disquettes n'est d'aucune utilité si vous ne pouvez les relire et les transférer sur un disque. Pour vérifier que vos sauvegardes sont exploitables, affichez régulièrement leur table des matières (via **restore -T**, ou **tar -t** pour les bandes d'archives). Si vous effectuez vos sauvegardes sur disquettes et que vous disposez de plusieurs unités de disquettes, tentez de les lire sur une unité autre que celle qui a servi à créer la sauvegarde. Vous pouvez également créer un second lot de disquettes de sauvegarde de niveau 0. Si vous utilisez un dérouleur en continu, lancez régulièrement la commande **tapechk** pour effectuer un contrôle de cohérence rudimentaire sur la bande.

Conservez les anciennes sauvegardes

Prévoyez de recycler régulièrement les supports de sauvegarde, mais pas tous. Il peut parfois s'écouler plusieurs mois avant que vous (ou un autre utilisateur du système) vous aperceviez qu'un fichier important a disparu ou qu'il est endommagé. Vous serez alors soulagé de savoir qu'il en existe une sauvegarde. Voici un exemple de planning de recyclage de disquettes et des bandes de sauvegarde :

- Une fois par semaine, recyclage de toutes les disquettes de sauvegarde quotidiennes, excepté celles du vendredi.
- Une fois par mois, recyclage de toutes les disquettes des vendredis, excepté le dernier vendredi du mois (les sauvegardes du dernier vendredi du mois restent disponibles).
- Une fois par trimestre, recyclage de toutes les disquettes mensuelles, excepté celles du dernier mois. Conservez indéfiniment la dernière disquette de chaque mois, éventuellement dans un lieu distinct.

Vérifiez les systèmes de fichiers avant de les sauvegarder

Une sauvegarde effectuée sur un système de fichiers endommagé sera probablement inutilisable. Avant d'effectuer une sauvegarde, prenez l'habitude de vérifier l'intégrité du système de fichiers à l'aide de la commande **fsck**.

Vérifiez que les fichiers que vous sauvegardez ne sont pas en cours d'utilisation.

Le système ne doit pas être en cours d'exploitation lorsque vous lancez vos sauvegardes. Si c'est le cas, les fichiers risquent d'être modifiés, et votre sauvegarde de ne pas être à jour.

Sauvegardez votre système avant d'y effectuer des changements majeurs.

Il est toujours plus prudent de sauvegarder l'intégralité du système avant de tester un élément matériel, de procéder à une réparation ou d'installer des nouvelles unités, de nouveaux programmes ou autres.

Divers

Lors de l'élaboration de votre stratégie de sauvegarde, prenez également en compte :

- La fréquence de modification des données : comme nous l'avons souligné plus haut, les données propres au système d'exploitation sont peu sujettes à modification et ils donc inutile de les sauvegarder trop souvent. Les données utilisateur, à l'inverse, sont généralement souvent modifiées et vous devez les sauvegarder en conséquence.
- Le nombre d'utilisateurs sur le système : le volume à sauvegarder et la fréquence des sauvegarde en dépendent directement.
- La difficulté de recréer les données : pensez que certaines données ne peuvent être recréées si elles n'ont pas fait l'objet d'une sauvegarde.

Quelle qu'elle soit, une stratégie est indispensable. Prévoyez des sauvegardes fréquentes et régulières. Recouvrer des données perdues peut s'avérer difficile si vous n'avez pas défini de stratégie de sauvegarde.

Supports de stockage

Il existe différents types de supports de stockage. Les supports disponibles dépendent de la configuration matérielle et logicielle de votre système. Les supports les plus utilisés sont les disquettes 5 1/4 pouces, les bandes 8 mm, les bandes 9 pistes et les disquettes 3 1/2 pouces.

Avertissement : Exécuter la commande **backup** détruit toutes les données antérieurement stockées sur le support de sortie sélectionné.

Disquettes

Les disquettes sont le support de sauvegarde standard. C'est-à-dire que, sauf spécification autre de votre part via la commande **backup -f**, la commande **backup** envoie automatiquement sa sortie vers l'unité **/dev/rfd0**, à savoir l'unité de disquette. Pour effectuer une sauvegarde sur l'unité de bande par défaut, indiquez **/dev/rmt0**.

Maniez vos disquettes avec précaution. Chaque élément d'information occupe un espace tellement infime que la moindre rayure, poussière, miette, particule de tabac... peut rendre l'information inutilisable. Veillez à :

- ne pas toucher les surfaces d'enregistrement ;
- conserver les disquettes à l'abri de toute source de champ magnétique et électromagnétique (téléphone, dictaphone, calculatrice, etc.) ;
- d'éviter les températures extrêmes (plage recommandée : entre 10°C et 60°C) ;
- prendre soin des disquettes pour prévenir toute perte de données ;
- effectuer régulièrement des copies de sauvegarde.

Avertissement : Disquettes et unité de disquette doivent être du même type, faute de quoi vous risquez de perdre des données. Si vous introduisez une disquette incorrecte dans l'unité de disquette 3 1/2 pouces, les données de la disquette risquent d'être détruites.

Les disquettes 3 1/2 pouces compatibles sont les suivantes :

- disquette 1 Mo (stockant environ 720 ko de données),
- disquette 2 Mo (stockant environ 1,44 Mo de données).

Bandes

Les bandes, du fait de leur grande capacité et de leur durée de vie, sont bien adaptées à certaines tâches : elles sont souvent utilisées lorsqu'il s'agit de stocker des fichiers volumineux ou des ensembles de fichiers (archivage de systèmes de fichiers, par exemple). Elles servent également à transférer des fichiers d'un système à un autre. Elles sont en revanche peu adaptées au stockage des fichiers souvent sollicités : d'autres supports offrent un temps d'accès bien inférieur.

Les fichiers bande sont créés via des commandes telles que **backup**, **cpio** et **tar**, qui ouvrent un fichier, y écrivent et le ferment.

Formatage de disquettes (commande **format** ou **fdformat**)

Avertissement : Le formatage d'une disquette détruit toutes les données qu'elle contient.

Vous pouvez formater des disquettes dans l'unité spécifiée par le paramètre *Unité* (**/dev/rfd0**, par défaut) via les commandes **format** et **fdformat**. La commande **format** détermine le type d'unité de disquette :

- 5 1/4 pouces faible densité (360 ko), contenant 40 x 2 pistes, de 9 secteurs chacune.
- 5 1/4 pouces haute densité (1,2 Mo) contenant 80 x 2 pistes, de 15 secteurs chacune.
- 3 1/2 pouces faible densité (720 ko) contenant 80 x 2 pistes, de 9 secteurs chacune.
- 3 1/2 pouces haute densité (2,88 Mo) contenant 80 x 2 pistes, de 36 secteurs chacune.

Tous les secteurs ont une taille de 512 octets (quel que soit le type de disquette).

Sauf spécification contraire via le paramètre *Unité*, la commande **format** formate une disquette en haute densité.

De même, la commande **fdformat** formate une disquette en faible densité, sauf si l'indicateur **-h** est spécifié. Le paramètre *Unité* indique l'unité où se trouve la disquette à formater (**/dev/rfd0** pour l'unité 0, par exemple).

Avant de formater une disquette, les commandes **format** et **fdformat** demandent confirmation. Vous pouvez ainsi annuler l'opération au besoin.

Par exemple, pour formater une disquette dans l'unité **/dev/rfd0**, entrez :

```
format -d /dev/rfd0
```

Par exemple, pour formater une disquette sans vérifier les secteurs, entrez :

```
format -f
```

Par exemple, pour formater une disquette 360 ko dans une unité 5 1/4 pouces, 1,2 Mo de l'unité **/dev/rfd1**, entrez :

```
format -l -d /dev/rfd1
```

Par exemple, pour forcer le formatage en haute densité d'une disquette avec la commande **fdformat**, entrez :

```
fdformat -h
```

Reportez-vous à la commande **format** dans le manuel *AIX Commands Reference*.

Vérification de l'intégrité du système de fichiers (commande **fsck**)

La commande **fsck** permet de vérifier et de remédier interactivement aux incohérences des systèmes de fichiers. Il convient de l'exécuter sur tous les systèmes de fichiers dans le cadre de l'initialisation du système. Vous devez pouvoir lire le fichier unité sur lequel réside le système de fichiers (unité **/dev/hd0**, par exemple). Le plus souvent, le système de fichiers est cohérent, et la commande **fsck** se contente d'afficher des informations concernant le nombre de fichiers, les blocs utilisés, les blocs libres, etc. Si le système de fichiers présente une incohérence, la commande **fsck** la précise et demande l'autorisation d'y remédier. La commande **fsck** tente de corriger l'anomalie en évitant toute action susceptible de provoquer une perte de données valides. Dans certains cas, toutefois, la commande **fsck** recommande la destruction d'un fichier endommagé.

Avertissement : Lancez toujours la commande **fsck** après un incident système. L'intervention peut détruire quelques données. Par défaut, pour chaque correction, le système demande confirmation ou infirmation à l'opérateur (*yes* ou *no*). Si vous ne détenez pas le droit d'écriture sur un fichier endommagé, la commande **fsck** considère que votre réponse est négative (*no*), quelle que soit votre réponse.

Par exemple, pour vérifier tous les systèmes de fichiers par défaut, entrez :

```
fsck
```

Sous cette forme, la commande **fsck** vous demande confirmation avant toute modification sur un système de fichiers.

Par exemple, pour corriger automatiquement les problèmes mineurs sur tous les systèmes de fichiers par défaut, entrez :

```
fsck -p
```

Par exemple, pour vérifier le système de fichiers */dev/hd1*, entrez :

```
fsck /dev/hd1
```

Le système de fichiers non monté se trouvant sur l'unité */dev/hd1* est vérifié.

Remarque : La commande **fsck** n'opère aucune correction sur un système de fichiers monté.

Reportez-vous à la commande **fsck** dans le manuel *AIX Commands Reference*.

Copie de ou sur des disquettes (commande **flcopy**)

La commande **flcopy** permet de copier une disquette (ouverte comme **/dev/rfd0**) dans le fichier *floppy* créé sur le répertoire courant. Le message *Change floppy, hit return when done* s'affiche quand nécessaire. La commande **flcopy** copie ensuite le fichier *floppy* sur la disquette.

Par exemple, pour copier */dev/rfd1* dans le fichier *floppy* du répertoire courant, entrez :

```
flcopy -f /dev/rfd1 -r
```

Pour copier les 100 premières pistes de la disquette, entrez :

```
flcopy -f /dev/rfd1 -t 100
```

Reportez-vous à la commande **flcopy** dans le manuel *AIX Commands Reference*.

Copie de fichiers sur bande ou disque (commande **cpio -o**)

La commande **cpio -o** permet de lire les chemins d'accès aux fichiers sur l'unité d'entrée standard et de copier ces fichiers sur l'unité de sortie standard, avec chemins d'accès et informations d'état. Les chemins d'accès ne peuvent dépasser 128 caractères. Évitez de fournir à la commande **cpio** des noms de chemins composés uniquement de fichiers liés : pour peu que la mémoire soit insuffisante, les informations de liaison risquent d'être perdues.

Par exemple, pour copier sur disquette les fichiers du répertoire courant, suffixés `.c`, entrez :

```
ls *.c | cpio -ov >/dev/rfd0
```

L'indicateur **-v** affiche le nom de chaque fichier.

Par exemple, pour copier sur disquette le répertoire courant et tous ses sous-répertoires, entrez :

```
find . -print | cpio -ov >/dev/rfd0
```

L'arborescence dont le sommet est le répertoire courant (`.`) est sauvegardée, avec tous ses sous-répertoires et ses fichiers. Pour obtenir plus rapidement le même résultat, entrez :

```
find . -cpio /dev/rfd0 -print
```

Le paramètre `-print` affiche le nom de chaque fichier copié.

Reportez-vous à la commande **cpio** dans le manuel *AIX Commands Reference*.

Copie de fichiers à partir de bande ou de disque (commande **cpio-i**)

La commande **cpio -i** lit sur l'unité d'entrée standard un fichier archive créé via la commande **cpio -o** et copie les fichiers dont le nom correspond au paramètre *Trame*. Ces fichiers sont copiés dans l'arborescence de répertoire courante. Vous pouvez indiquer plusieurs paramètres *Trame*, en utilisant les conventions de notation décrites à la commande **ksh**. La valeur par défaut de *Trame* est un astérisque (*), qui sélectionne tous les fichiers du répertoire courant. Une expression du type `[a-z]` définit un intervalle (de "a" à "z") selon l'ordre de tri courant.

Remarque : Les expressions `"*.c"` et `"*.o"` doivent être entre guillemets, de sorte que le shell ne considère pas l'astérisque (*) comme un caractère générique. Il s'agit ici d'un cas particulier où la commande **cpio** decode elle-même les caractères génériques.

Par exemple, pour afficher la liste des fichiers sauvegardés sur disquette via la commande **cpio**, entrez :

```
cpio -itv </dev/rfd0
```

La table des matières des données précédemment sauvegardées sur le fichier `/dev/rfd0` au format de la commande **cpio** s'affiche. La liste est semblable à celle obtenue par la commande **li -l** (répertoire complet). Pour n'afficher que les noms des chemins d'accès, ne spécifiez que les indicateurs **-it**.

Par exemple, pour copier d'une disquette les fichiers précédemment sauvegardés via la commande **cpio**, entrez :

```
cpio -idmv </dev/rfd0
```

Les fichiers précédemment sauvegardés dans le fichier `/dev/rfd0` via la commande **cpio** sont recopiés dans le système de fichiers (indicateur **-i**). L'indicateur **-d** permet à la commande de créer les répertoires requis pour la sauvegarde d'un répertoire.

L'indicateur **-m** maintient l'heure de la dernière modification, effective au moment de la sauvegarde des fichiers. L'indicateur **-v** affiche le nom des fichiers copiés.

Par exemple, pour copier d'une disquette une sélection de fichiers, entrez :

```
cpio -i "*.c" "*.o" </dev/rfd0
```

Les fichiers suffixés `.c` ou `.o` sont copiés de la disquette.

Reportez-vous à la commande **cpio** dans le manuel *AIX Commands Reference*.

Copie de et sur des bandes (commande `tcopy`)

La commande `tcopy` permet de copier des bandes magnétiques.

Par exemple, pour copier à partir d'une bande continue sur une bande 9 pistes, entrez :

```
tcopy /dev/rmt0 /dev/rmt8
```

Reportez-vous à la commande `tcopy` dans le manuel *AIX Commands Reference*.

Vérification de l'intégrité d'une bande (commande `tapechk`)

La commande `tapechk` permet d'effectuer une vérification rudimentaire de cohérence sur un dérouleur de bande en continu. Certains dysfonctionnements peuvent être décelés par simple lecture de la bande. La commande `tapechk` permet de lire la bande au niveau des fichiers.

Par exemple, pour vérifier les trois premiers fichiers d'un dérouleur de bande en continu, entrez :

```
tapechk 3
```

Reportez-vous à la commande `tapechk` dans le manuel *AIX Commands Reference*.

Compression de fichiers (commandes `compress` et `pack`)

Les commandes `compress` et `pack` permettent de compresser des fichiers en vue de leur stockage, les commandes `uncompress` et `unpack`, de développer les fichiers restaurés. Le processus de compression/décompression n'est pas rapide, mais il permet un gain significatif d'espace sur les supports de stockage.

Il existe plusieurs méthodes de compression d'un système de fichiers :

- commande `backup` assortie de l'option `-p`,
- commande `compress` ou `pack`.

Comprimer des données répond généralement à deux soucis :

- Economiser les ressources de stockage et d'archivage du système :
 - Comprimez les systèmes de fichiers avant de les sauvegarder pour préserver l'espace disponible sur bande.
 - Comprimez les fichiers journaux créés par les scripts shell exécutés la nuit : il est très simple de programmer le script pour qu'il comprime un fichier avant qu'il quitte.
 - Comprimez les fichiers peu sollicités. Par exemple, les fichiers d'un utilisateur absent pour longtemps peuvent être comprimés et placés dans un fichier archive `tar` sur disque ou bande, et restaurés le moment venu.
- Economiser du temps et de l'argent lors de l'envoi de fichiers via un réseau.

Remarques :

1. L'espace de travail peut se révéler insuffisant pour exécuter la commande. Celle-ci crée en effet les fichiers compressés avant de supprimer les fichiers non compressés ; elle requiert donc environ 50% d'espace de plus que la taille totale des fichiers à compresser.
2. Il peut y avoir échec de la compression si un fichier est déjà compressé. Si `compress` ne peut compresser un fichier, elle échoue.

Commande compress

La commande **compress** réduit la taille des fichiers en utilisant le codage Lempel-Zev. Chaque fichier d'origine (paramètre *Fichier*) est placé dans un fichier compressé, suffixé **.z**. Ce fichier conserve les attributs du fichier d'origine (propriétaire, droits d'accès, date et heure de dernière modification, etc.). A défaut de fichiers spécifiés, la commande s'applique à l'intégralité de l'unité d'entrée standard, le résultat étant envoyé sur l'unité de sortie standard. Si la compression est sans effet sur un fichier (taille non réduite), un message est inscrit sur la sortie d'erreur standard et le fichier d'origine n'est pas remplacé.

La commande **uncompress** restaure les fichiers dans leur forme initiale.

Le pourcentage de compression dépend de la taille de l'entrée, du nombre de bits par code spécifié par la variable *Bits* et de la répartition des sous-chaînes communes. En général, source et texte simple sont réduits de 50 à 60%. La commande **compress** est plus rapide et génère des fichiers souvent plus compacts que la commande **pack**, laquelle utilise le codage Huffman.

Par exemple, pour comprimer le fichier `foo` et inscrire le pourcentage correspondant sur l'unité d'erreur standard, entrez :

```
compress -v foo
```

Reportez-vous à la commande **compress** dans le manuel *AIX Commands Reference*.

Commande pack

La commande **pack** stocke le(s) fichier(s) spécifié(s) par le paramètre *Fichier* sous forme comprimée, via le codage Huffman. Le fichier en entrée est remplacé par un fichier compressé, dont le nom dérive de celui du fichier d'origine (*Fichier.z*), doté des mêmes attributs (modes d'accès, dates d'accès et de modification, propriétaire, etc.). Le nom du fichier d'entrée ne doit pas dépasser 253 octets, pour réserver de l'espace au suffixe **.z**. Si la commande **pack** aboutit, le fichier d'origine est supprimé. Les fichiers comprimés peuvent être restaurés via la commande **unpack**.

Si la commande ne peut créer un fichier plus petit, elle s'interrompt et signale qu'elle ne peut gagner d'espace. (Cet échec s'observe généralement avec des fichiers trop petits ou avec une répartition uniforme de caractères.) L'espace gagné dépend de la taille du fichier et de la fréquence de répartition des caractères. Un algorithme de décodage formant la première partie des fichiers **z**, les fichiers de moins de 3 blocs ne peuvent être réduits. Les fichiers texte sont réduits de 25 à 40%.

La valeur en sortie de la commande **pack** est le nombre de fichiers qu'elle n'a pu comprimer. Ce qui se produit lorsque :

- le fichier est déjà comprimé,
- le nom du fichier en entrée comporte plus de 253 octets,
- le fichier est doté de liens,
- le fichier est un répertoire,
- le fichier ne peut être ouvert,
- aucun bloc de stockage n'est sauvegardé par l'opération,
- un fichier nommé *Fichier.z* existe déjà,
- le fichier **.z** ne peut être créé,
- une erreur d'E/S s'est produite au cours du traitement.

Par exemple, pour comprimer les fichiers `chap1` et `chap2`, entrez :

```
pack chap1 chap2
```

Les fichiers `chap1` et `chap2` sont compressés et remplacés par les fichiers **chap1.z** et **chap2.z**. La commande **pack** affiche le pourcentage de réduction de taille de chaque fichier.

Reportez-vous à la commande **pack** dans le manuel *AIX Commands Reference*.

Décompression de fichiers (commandes uncompress et unpack)

Les commandes **uncompress** et **unpack** permettent de décompresser des fichiers.

Commande uncompress

La commande **uncompress** permet de décompresser les fichiers précédemment compressés par la commande **compress**. Tout fichier compressé spécifié par la variable *Fichier* est remplacé par sa version développée. Cette version porte le même nom que la version compressée, sans l'extension **.Z**. Ce fichier conserve les attributs du fichier d'origine (propriétaire, droits d'accès, date et heure de dernière modification, etc.). A défaut de fichier(s) spécifié(s), l'entrée standard est décompressée sur la sortie standard.

Semblable à la commande **uncompress**, la commande **zcat** écrit toujours la sortie développée sur l'unité de sortie standard.

Par exemple, pour décompresser le fichier `foo`, entrez :

```
uncompress foo
```

Reportez-vous à la commande **uncompress** dans le manuel *AIX Commands Reference*.

Commande unpack

La commande **unpack** décompresse les fichiers créés par la commande **pack**. Pour chaque fichier spécifié, la commande **unpack** recherche le fichier *Fichier.z* compressé. Si le fichier est comprimé, **unpack** le remplace par sa version décompressée. La commande **unpack** renomme le nouveau fichier en supprimant le suffixe **.z** de *Fichier*. Le nouveau fichier conserve les attributs du fichier d'origine (propriétaire, modes, heures d'accès et de modification, etc.).

La commande **unpack** n'opère que sur les fichiers suffixés **.z**. Aussi, si vous spécifiez un nom de fichier ne se terminant pas par **.z**, la commande lui ajoute le suffixe et recherche un fichier avec ce suffixe.

La valeur en sortie de la commande **unpack** est le nombre de fichiers qu'elle n'a pu décompresser. Ce qui se produit lorsque :

- le nom du fichier (hors extension **.z**) dépasse 253 octets,
- le fichier ne peut être ouvert,
- le fichier n'est pas compressé,
- un fichier décompressé portant le même nom existe déjà,
- le fichier décompressé ne peut être créé.

Remarque : La commande **unpack** inscrit un avertissement sur l'unité de sortie standard si le fichier qu'elle décompresse est doté de liens. Le fichier nouvellement décompressé est affecté d'un numéro d'i-node différent du fichier compressé d'origine. Toutefois, les autres fichiers liés à l'i-node du fichier compressé d'origine existent toujours et sont toujours comprimés.

Par exemple, pour décompresser les fichiers `chap1.z` et `chap2`, entrez :

```
unpack chap1.z chap2
```

Les fichiers `chap1.z` et `chap2.z` sont décompressés et remplacés par les fichiers `chap1` et `chap2`. Notez que vous pouvez spécifier les fichiers avec ou sans le suffixe **.z**.

Reportez-vous à la commande **unpack** dans le manuel *AIX Commands Reference*.

Sauvegarde de fichiers (commande backup)

Avertissement : Si vous tentez de sauvegarder un système de fichiers monté, un message s'affiche. La sauvegarde se poursuit, mais des incohérences risquent d'apparaître dans le système de fichiers. Cet avertissement ne concerne pas le système de fichiers racine (/).

Vous pouvez créer des copies de vos fichiers sur un support de stockage, tel qu'une bande magnétique ou une disquette, avec la commande **backup** ou **smit**. Ces copies respectent l'un des deux formats de sauvegarde suivants :

- fichiers spécifiques sauvegardés par noms, via l'indicateur **-i** ;
- intégralité de systèmes de fichiers sauvegardés par numéros d'i-node, via les paramètres **-Niveau** et **SystèmeFichiers**.

Remarques

1. Il y a toujours un risque d'endommagement des données lorsqu'un fichier est modifié pendant la sauvegarde du système. Réduisez donc au minimum les activités sur le système pendant la procédure de sauvegarde.
2. Si vous effectuez une sauvegarde sur bande 8 mm alors que la taille de bloc de l'unité est défini à 0 (zéro), il sera impossible de procéder à une restauration directement à partir de la bande. Dans ce cas, reportez-vous aux procédures spéciales décrites à la section concernant la commande **restore**.

Avertissement : Vérifiez que les indicateurs que vous spécifiez sont compatibles avec le support de sauvegarde. Pensez aussi que si vous tentez de sauvegarder un système de fichiers monté, vous risquez des incohérences.

Commande backup

Par exemple, pour sauvegarder par noms des fichiers de votre répertoire \$HOME, entrez :

```
find $HOME -print | backup -i -v
```

L'indicateur **-i** précise au système de lire sur l'entrée standard les noms des fichiers à sauvegarder. La commande **find** génère la liste des fichiers du répertoire **\$HOME**. Cette liste devient, par chaînage, l'entrée standard de la commande **backup**. L'indicateur **-v** affiche un état de la progression de la sauvegarde. Les fichiers sont sauvegardés sur l'unité de sauvegarde par défaut du système local.

Par exemple, pour sauvegarder le système de fichiers racine, entrez :

```
backup -0 -u /
```

Le niveau (0) et la barre oblique (/) instruisent le système qu'il s'agit de sauvegarder le système de fichiers racine (/). La sauvegarde a lieu dans le fichier **/dev/rfd0**. L'indicateur **-u** signifie au système qu'il doit mettre à jour l'article relatif au niveau de sauvegarde courant, dans le fichier **/etc/dumpdates**.

Par exemple, pour sauvegarder tous les fichiers du système de fichiers racine (/), modifiés depuis la dernière sauvegarde de niveau 0, entrez :

```
backup -1 -u /
```

Reportez-vous à la commande **backup** dans le manuel *AIX Commands Reference*.

Commande smit

Vous pouvez également lancer la commande **backup** via **smit**.

1. A l'invite, entrez :

```
smit backup
```

2. Entrez le chemin d'accès au répertoire sur lequel est normalement monté le système de fichiers, dans le champ chemin d'accès complet au REPERTOIRE :

```
/home/bill
```

3. Dans les champs UNITE ou FICHIER DE SAUVEGARDE, entrez le nom de l'unité de sortie (ci-dessous le nom d'une unité de bande magnétique) :

```
/dev/rmt0
```

4. Appuyez sur Tab pour basculer la valeur du champ facultatif SIGNALEMENT de chaque étape de la sauvegarde, si vous souhaitez afficher des messages d'erreur.
5. Dans un environnement de gestion de système, observez la valeur par défaut du champ nombre MAX. de blocs à écrire sur la sauvegarde : ce champ ne concerne pas les sauvegardes sur bande.
6. Appuyez sur Entrée pour sauvegarder le répertoire ou le système de fichiers nommé.
7. Lancez la commande **restore -t**. Si elle génère une erreur, vous devez reprendre la sauvegarde à zéro.

Restauration de fichiers (commande restore)

Les commandes **restore** et **smit** permettent de lire les fichiers sauvegardés sur un support par la commande **backup** et de les restaurer sur le système local.

Remarques :

1. La restauration doit être effectuée par la même méthode que celle utilisée pour la sauvegarde. Par exemple, un système de fichiers sauvegardé par noms doit être restauré par noms.
2. Lorsque la sauvegarde occupe plusieurs disquettes, la commande **restore** lit celle qui est montée, puis vous invite à insérer la suivante. Appuyez sur Entrée une fois la disquette insérée, pour poursuivre la restauration.

Commande restore

Par exemple, pour afficher la liste des fichiers précédemment sauvegardés, entrez :

```
restore -T
```

Les informations sont lues à partir de l'unité de sauvegarde par défaut **/dev/rfd0**. S'il s'agit de fichiers simples, seul leur nom est affiché. S'il s'agit de systèmes de fichiers, leur numéro d'i-node apparaît également.

Par exemple, pour restaurer des fichiers sur le système de fichiers principal, entrez :

```
restore -x -v
```

L'indicateur **-x** extrait tous les fichiers du support de sauvegarde et les restaure à leur place dans le système de fichiers. L'indicateur **-v** affiche un état de la progression des opérations à mesure que les fichiers sont restaurés. S'il s'agit d'un système de fichiers, les fichiers sont affichés avec leur numéro d'i-node. Sinon, seul leur nom est affiché.

Par exemple, pour copier le fichier `/home/mike/manual/chap1`, entrez :

```
restore -xv /home/mike/manual/chap1
```

Le fichier `/home/mike/manual/chap1` est extrait du support de sauvegarde et restauré. Le nom `/home/mike/manual/chap1` doit être affichable par la commande **restore -T**.

Par exemple, pour copier tous les fichiers dans le répertoire `manual`, entrez :

```
restore -xdv manual
```

Le répertoire `manual` et les fichiers qu'il contient sont restaurés. Si le répertoire n'existe pas, il est créé dans le répertoire courant.

Reportez-vous à la commande **restore** dans le manuel *AIX Commands Reference*.

Commande smit

Vous pouvez également lancer la commande **restore** via **smit**.

1. A l'invite, entrez :

```
smit restore
```

2. Renseignez le champ Répertoire cible. Vous devez y préciser le répertoire dans lequel doivent résider les fichiers restaurés.

3. Passez au champ UNITE ou FICHIER DE SAUVEGARDE, entrez le nom de l'unité de sortie (ci-dessous le nom d'une unité de bande magnétique) :

```
/dev/rmt0
```

Si l'unité n'est pas disponible, un message semblable au suivant s'affiche :

```
Cannot open /dev/rmtX, no such file or directory.
```

Ce message indique que le système ne peut atteindre le pilote d'unité car il n'existe pas de fichiers pour `rmtX` dans le répertoire `/dev`. Les seuls éléments à l'état `available` se trouvent dans `/dev`.

4. Pour le champ NBRE de blocs à lire en une entrée, nous vous conseillons de conserver la valeur par défaut.
5. Appuyez sur Entrée pour restaurer le système de fichiers ou le répertoire spécifié.

Archivage de fichiers (commande tar)

La commande **tar** permet d'écrire des fichiers dans un fichier archive ou de les extraire. Sauf spécification autre, la commande recherche les archives sur l'unité par défaut (bande, généralement).

Lorsque vous écrivez dans un fichier archive, la commande **tar** passe par un fichier temporaire (**/tmp/tar***) et garde en mémoire une table des fichiers dotés de plusieurs liens. Un message d'erreur est émis si la commande **tar** ne peut créer ce fichier temporaire ou que la mémoire disponible est insuffisante pour contenir les tables de liens.

Par exemple, pour écrire les fichiers `file1` et `file2` dans un nouveau fichier archive, sur l'unité de bande par défaut, entrez :

```
tar -c file1 file2
```

Pour extraire tous les fichiers du répertoire `/tmp` du fichier archive de l'unité de bande `/dev/rmt2`, et imputer la durée de l'opération au temps de modification, entrez :

```
tar -xm -f/dev/rmt2 /tmp
```

Pour afficher le nom des fichiers du fichier archive sur disque `out.tar`, à partir du répertoire courant, entrez :

```
tar -vtf out.tar
```

Reportez-vous à la commande **tar** dans le manuel *AIX Commands Reference*.

Reportez-vous à la commande **tar** dans le manuel *AIX Commands Reference*.

Récapitulatif des commandes relatives à la sauvegarde

backup	Sauvegarde des fichiers et des systèmes de fichiers.
compress	Comprime et décomprime des données.
cpio	Copie des fichiers vers et à partir de supports de stockage et de répertoires.
fdformat	Formate des disquettes.
flcopy	Copie de et sur disquettes.
format	Formate des disquettes.
fsck	Vérifie la cohérence du système de fichiers et remédie aux problèmes en mode interactif.
pack	Comprime des fichiers.
restore	Copie des fichiers ou des systèmes de fichiers, préalablement sauvegardés via la commande backup , à partir d'une unité locale.
tapechk	Contrôle le dérouleur de bande en continu.
tar	Manipule les archives.
tcopy	Copie une bande magnétique.
uncompress	Comprime et décomprime des données.
unpack	Décomprime des fichiers.

Voir aussi

Commandes : généralités

Process : généralités

Réacheminement des entrées/sorties

Systemes de fichiers

Répertoires :

Fichiers : généralités

Sécurité du système et des fichiers

Chapitre 9. Sécurité du système et des fichiers

La sécurité informatique, comme toute forme de sécurité, vise à protéger. A protéger les informations stockées sur un ordinateur, précieuses à plusieurs titres. La sécurité informatique doit répondre à un triple objectif :

- Intégrité** La valeur d'une information dépend de sa fiabilité. Des données modifiées à mauvais escient perdent tout ou partie de leur intérêt.
- Confidentialité** La valeur de beaucoup d'informations tient au fait qu'elles sont secrètes.
- Disponibilité** Les informations doivent être disponibles.

Il est bon de planifier et de mettre en œuvre votre stratégie en matière de sécurité avant de commencer à exploiter votre système. Il est en effet très long de la modifier après coup-: mieux vaut donc prévenir que guérir.

Ce chapitre traite des points suivants :

- Menaces sur la sécurité, page 9-2.
 - Sécurité de base, page 9-2.
- Propriété des fichiers et groupes d'utilisateurs, page 9-4.
 - Changement du propriétaire d'un fichier ou d'un répertoire (commande **chown**), page 9-4.
 - Modes d'accès aux fichiers et aux répertoires, page 9-4.
 - Affichage d'informations sur le groupe (commande **lsgroup**), page 9-6.
 - Modification des droits d'accès aux fichiers et aux répertoires (commande **chmod**), page 9-8.
- Listes de contrôle des accès (ACL), page 9-9.
 - Affichage des informations de contrôle des accès (commande **aclget**), page 9-12.
 - Définition du contrôle des accès (commande **aclput**), page 9-12.
 - Modification des informations de contrôle des accès (commande **acledit**), page 9-13.
- Verrouillage du terminal (commande **lock** or **xlock**), page 9-14.
- Contrôle anti-virus (commande **virscan**), page 9-15.
- Récapitulatif des commandes relatives à la sécurité, page 9-17.

Menaces sur la sécurité

Trois types de comportement sont susceptibles de porter atteinte à la sécurité d'un système :

Négligence	La sécurité des informations est souvent mise à mal par simple négligence des utilisateurs autorisés : si vous divulguez votre mot de passe, aucun mécanisme de sécurité, aussi sophistiqué soit-il, ne pourra empêcher un accès illégal à votre compte et à vos données.
Indiscrétion	Nombre de problèmes de sécurité sont le fait d'indiscrètes - utilisateurs autorisés explorant le système à la recherche de données mal protégées.
Pénétration	Il s'agit là d'une "attaque" délibérée contre le système. Il s'agit souvent d'une pénétration effectuée par quelqu'un ayant étudié de près les protections en place et ayant réussi à en déceler les failles.

Ce dernier type d'action représente le plus grand danger pour les données, mais les problèmes liés à la négligence et à l'indiscrétion ne doivent pas être sous-estimés.

Sécurité de base

Voici quelques règles de base, applicables à tous les systèmes, pour assurer une sécurité minimale.

Sauvegardes

Si vous disposez d'une sauvegarde à jour du système, vous êtes armé pour faire face sans trop de dégâts à un problème système. Des sauvegardes physiquement sûres, fiables et à jour sont un élément de sécurité essentiel. Votre stratégie de sauvegarde doit être documentée, avec notamment des informations concernant :

- la fréquence des sauvegardes,
- le type de sauvegarde (système, données, par incréments),
- le mode de contrôle des bandes de sauvegarde,
- le mode de stockage des bandes de sauvegarde.

Pour en savoir plus, reportez-vous à "Fichiers de sauvegarde et supports de stockage", page 8-1.

Identification et authentification

Identification et authentification établissent votre identité. Vous êtes invité à vous connecter à votre système. Pour cela, vous devez fournir votre nom utilisateur et, éventuellement, un mot de passe (sur un système sûr, tous les comptes non affectés d'un mot de passe doivent être invalidés). Si le mot de passe est correct, vous accédez au compte correspondant et disposez des droits et des privilèges afférents.

Le mot de passe étant l'unique protection de votre compte, il est important de bien le choisir et de le protéger des indiscrétions. Tenter de deviner un mot de passe est souvent la base d'une tentative d'accès illégal. Pour une protection efficace des mots de passe, il n'y aucune confusion entre les données de sécurité et les données utilisateur. Les mots de passe, codés, et les autres données de sécurité se trouvent dans le fichier **/etc/security/passwd**. Ce fichier ne doit être accessible que par l'utilisateur racine : en restreignant ainsi cet accès, un utilisateur malveillant ne peut décrypter les mots de passe via un programme qui boucle sur un ensemble de mots de passe possibles.

Il reste possible de deviner un mot de passe en tentant la connexion à un compte : si le mot de passe est banal ou qu'il n'est pas modifié assez souvent, l'opération risque d'aboutir.

ID de connexion

Le système d'exploitation identifie également les utilisateurs par leur ID de connexion. Cet ID permet au système de suivre toutes les actions d'un utilisateur. Après connexion d'un utilisateur, mais avant l'exécution de son programme initial, le système affecte l'ID de connexion du process à l'ID utilisateur dans la base de données. Tous les process suivants de la session sont affectés de cet ID. Il est ainsi possible de garder trace de toutes les activités exécutées sous cet ID.

Au cours d'une session, un utilisateur peut modifier son ID utilisateur effectif, son ID utilisateur réel, son ID groupe effectif, son ID groupe réel et l'ID groupe complémentaire, mais non son ID de connexion.

Terminaux sans surveillance

Tous les systèmes sont vulnérables si des terminaux sont laissés connectés sans surveillance. Le cas le plus grave étant celui d'un administrateur laissant sans surveillance son terminal connecté sous l'identité de l'utilisateur racine. En règle générale, les utilisateurs doivent se déconnecter avant d'abandonner leur terminal.

Vous pouvez forcer la déconnexion d'un terminal, à l'issue d'un délai d'inactivité, via les paramètres **TMOUT** (shell Korn, **ksh**) ou **TIMEOUT** (shell Bourne, **bsh**) du fichier **/etc/profile**. Le paramètre **TMOUT** fonctionne dans le shell Korn (**ksh**), et le paramètre **TIMEOUT** avec le shell Bourne (**bsh**). Pour en savoir plus sur le paramètre **TMOUT**, reportez-vous à Substitution de paramètres dans le shell Korn ou POSIX, page 11-17 Pour en savoir plus sur le paramètre **TIMEOUT**, reportez-vous à Substitution de variables et de noms de fichiers dans le shell Bourne, page 11-83

Dans l'exemple suivant, extrait d'un fichier **.profile**, le terminal est automatiquement déconnecté au bout d'une heure d'inactivité :

```
TO=3600
echo "Setting Autologout to $TO"
TIMEOUT=$TO
TMOUT=$TO
export TIMEOUT TMOUT
```

Remarque : Les utilisateurs peuvent affecter d'autres valeurs aux paramètres **TMOUT** et **TIMEOUT** dans le fichier **.profile** de leur répertoire personnel, qui prennent alors le pas sur les valeurs définies dans le fichier **/etc/profile**.

Propriété des fichiers et groupes d'utilisateurs

A l'origine, le propriétaire d'un fichier est identifié par l'ID de l'utilisateur ayant créé le fichier. C'est lui qui détermine qui peut lire, écrire (modifier) ou exécuter le fichier. La commande **chown** permet de changer le propriétaire d'un fichier.

Chaque ID utilisateur est rattaché à un ID groupe unique. Les groupes sont créés par l'administrateur au moment de la configuration du système. Lorsqu'un fichier est créé, le système attribue des droits à l'ID de l'utilisateur créateur, à l'ID du groupe auquel il appartient et à un groupe dit "autres", constitué de tous les autres utilisateurs. Lancez la commande **id** pour afficher votre ID utilisateur (UID), votre ID groupe (GID) et le nom de tous les groupes dont vous êtes membre.

Dans les listes de fichiers (affichées, par exemple par la commande **ls** ou **ls -l**), les trois groupes d'utilisateurs apparaissent toujours dans l'ordre : utilisateur, groupe et autres. Pour connaître les groupes associés à un ID utilisateur, lancez la commande **groups**.

Changement du propriétaire d'un fichier ou d'un répertoire (commande **chown**)

La commande **chown** permet de changer le propriétaire de vos fichiers.

Assortie de l'indicateur **-R**, la commande **chown** balaye récursivement la structure arborescente des répertoires à partir du répertoire spécifié. Chaque fois qu'elle rencontre un lien symbolique, le propriétaire du fichier ou du répertoire sur lequel pointe le lien est changé (la propriété du lien lui-même reste inchangée).

Remarque : Seul l'utilisateur racine est habilité à changer le propriétaire d'un fichier dont il n'est pas propriétaire. Avec l'indicateur **-f**, aucune erreur n'est signalée.

Par exemple, pour changer le propriétaire du fichier `program.c`, entrez :

```
chown jim program.c
```

Les droits d'accès à `program.c` sont désormais reportés sur `jim`. En tant que propriétaire, `jim` peut ensuite lancer la commande **chmod** pour autoriser ou interdire l'accès à `program.c`.

Par exemple, pour attribuer la propriété de tous les fichiers du répertoire `/tmp/src` à `john` et leur associer le groupe `build`, entrez :

```
chown -R john:build /tmp/src
```

Reportez-vous à la commande **chown** dans le manuel *AIX Commands Reference*.

Modes d'accès aux fichiers et aux répertoires

Chaque fichier a un propriétaire. Il s'agit du créateur du fichier, pour les nouveaux fichiers. C'est lui qui définit le mode d'accès au fichier. Les modes d'accès déterminent les droits accordés aux autres utilisateurs sur ce fichier. Seuls le propriétaire d'un fichier ou l'utilisateur racine peuvent modifier les droits d'accès à ce fichier.

Il existe trois classes d'utilisateurs : utilisateur/propriétaire, groupe et tous les autres. Les droits accordés à ces catégories sont une combinaison des trois modes : lecture, écriture et exécution. Le créateur d'un fichier se voit attribuer par défaut les droits de lecture, d'écriture et d'exécution sur ce fichier, les deux autres catégories étant autorisées à lire et à exécuter le fichier. Le tableau ci-dessous illustre les droits accordés par défaut à chaque catégorie.

Classes	Lecture	Ecriture	Exécution
Propriétaire	Oui	Oui	Oui
Groupe	Oui	Non	Oui
Autres	Oui	Non	Oui

Un fichier peut être lu (r), écrit (w) ou exécuté (x). Le système détermine qui détient des droits et le niveau d'autorisation accordé pour chaque activité. Ces droits ont une double représentation dans le système : symbolique et numérique.

Représentation symbolique des droits d'accès

La représentation symbolique des droits d'accès est la suivante :

r	Droit de lecture : les utilisateurs sont autorisés à consulter le fichier.
w	Droit d'écriture : les utilisateurs sont autorisés à modifier le contenu du fichier.
x	Droit d'exécution : les utilisateurs sont autorisés à exécuter le fichier (il doit s'agir bien entendu d'un fichier exécutable, c'est-à-dire, le plus souvent d'un fichier contenant un programme). Pour les répertoires, ce droit donne la possibilité d'explorer leur contenu.

Par exemple, des droits `rwxr-xr-x` sur un fichier donnent aux trois catégories d'utilisateurs accès en lecture et en exécution à ce fichier, et le droit d'écriture au seul propriétaire du fichier. Il s'agit de la représentation symbolique des droits par défaut.

La commande **ls**, assortie de l'indicateur **-l** (L minuscule), affiche une liste complète du répertoire courant. Les 10 premiers caractères de cette liste indiquent le type de fichier et les droits associés. La commande indique également le propriétaire et le groupe associés à chaque fichier et à chaque répertoire.

Le premier caractère correspond au type du fichier, les neufs suivants aux droits détenus par chaque catégorie d'utilisateurs. Voici les types de fichiers possibles :

-	fichiers standard
d	répertoire
b	fichiers bloc spéciaux
c	fichiers caractères spéciaux
p	fichiers tube spéciaux
l	liens symboliques
s	sockets

Voici un exemple de liste obtenue par la commande **ls -l** :

```
-rwxrwxr-x 2 janet acct 512 Mar 01 13:33 january
```

Le premier caractère - (tiret) indique un fichier standard. Les caractères 2 à 4, `rwx`, représentent les droits d'accès de l'utilisateur (lecture, écriture et exécution). Les caractères 5 à 7, `rwx`, indiquent les droits d'accès du groupe (lecture, écriture et exécution). Les caractères 8 à 10, `r-x`, indiquent les droits d'accès des autres utilisateurs (lecture et exécution). Les tirets à la 9ème position indiquent l'absence de droit.

`janet` est la propriétaire du fichier et `acct` est le nom de son groupe. `512` est la taille du fichier (en octets), `Mar 01 13:33` sont les date et l'heure de dernière modification, et `january` est le nom du fichier. `2` est le nombre de liens associés au fichier.

Répresentation numérique des droits d'accès

Numériquement, les droits d'accès sont représentés par 4 pour la lecture, 2 pour l'écriture et 1 pour l'exécution. La somme des droits (comprise entre 1 et 7) représente les droits d'accès de chaque catégorie (utilisateur, groupe et autres). Voici un tableau récapitulatif :

Total	Lecture	Ecriture	Exécution
0	–	–	–
1	–	–	1
2	–	2	–
3	–	2	1
4	4	–	–
5	4	–	1
6	4	2	–
7	4	2	1

A la création d'un fichier, les droits par défaut sont représentés par 755, soit : accès en lecture, écriture et exécution (4+2+1=7) pour l'utilisateur, accès en lecture et exécution (4+1=5) pour le groupe, et lecture et exécution (4+1=5) pour les autres. Vous pouvez modifier les droits d'accès à vos fichiers par la commande **chmod** (change mode).

Affichage d'informations sur le groupe (commande **lsgroup**)

La commande **lsgroup** affiche les attributs de tous les groupes du système ou ceux des groupes spécifiés. Si elle ne parvient pas à lire un attribut (ou plusieurs), elle affiche autant d'informations que possible, sous la forme d'une série d'expressions *Attribut=Valeur*, séparées par un espace.

Par exemple, pour afficher la liste de tous les groupes du système, entrez :

```
lsgroup ALL
```

Le système affiche les groupes, leur ID et la liste des utilisateurs du groupe, comme suit :

```
system 0 arne, pubs, ctw, geo, root, chucka, noer, su, dea,
         backup, build, janice, denise
staff 1 john, ryan, flynn, daveb, jzitt, glover, maple, ken
        gordon, mbrady
bin 2 root, bin
sys 3 root, su, bin, sys
```

Pour afficher des attributs spécifiques pour tous les groupes, vous avez le choix entre deux styles de listes :

- Série d'expressions *Attribut=Valeur*, séparées par un espace (style par défaut). C'est le style par défaut. Par exemple, pour afficher la liste des ID et des utilisateurs de tous les groupes du système, entrez :

```
lsgroup -a id users ALL | pg
```

Une liste semblable à la suivante s'affiche :

```
system id=0
users=arne, pubs, ctw, geo, root, chucka, noer, su, dea, backup, build
staff id=1 users=john, ryan, flynn, daveb, jzitt, glover, maple, ken
```

- Format strophes. Par exemple, pour afficher la liste des ID et des utilisateurs de tous les groupes du système au format strophe, entrez :

```
lsgroup -a -f id users ALL | pg
```

Une liste semblable à la suivante s'affiche :

```
system:
  id=0
  users=pubs,ctw,geo,root,chucka,noer,su,dea,backup,build

staff:
  id=1
  users=john,ryan,flynn,daveb,jzitt,glover,maple,ken

bin:
  id=2
  users=root,bin

sys:
  id=3
  users=root,su,bin,sys
```

Pour afficher tous les attributs d'un groupe spécifique, vous avez également le choix entre deux styles :

- Série d'expressions `Attribut=Valeur`, séparées par un espace. C'est le style par défaut. Par exemple, pour afficher la liste des attributs du groupe système, entrez :

```
lsgroup system
```

Une liste semblable à la suivante s'affiche :

```
system id=0 users=arne,pubs,ctw,geo,root,chucka,noer,su,dea,
backup,build,janice,denise
```

- Format strophes. Par exemple, pour afficher la liste de tous les attributs du groupe `bin` au format strophes, entrez :

```
lsgroup -f system
```

Une liste semblable à la suivante s'affiche :

```
system:
  id=0
  users=arne,pubs,ctw,geo,root,chucka,noer,su,dea,
backup,build,janice,denise
```

Pour afficher des attributs spécifiques d'un groupe donné, entrez :

```
lsgroup -a Attributs Groupe
```

Par exemple, pour afficher la liste des ID et des utilisateurs du groupe `bin`, entrez :

```
lsgroup -a id users bin
```

Une liste semblable à la suivante s'affiche :

```
bin id=2 users=root,bin
```

Reportez-vous à la commande **lsgrupp** dans le manuel *AIX Commands Reference*.

Modification des droits d'accès aux fichiers et aux répertoires (commande `chmod`)

La commande `chmod` permet de modifier les droits d'accès en lecture, en écriture et en exécution à des fichiers spécifiés, et les droits d'exploration des répertoires spécifiés.

Par exemple, pour ajouter un droit d'accès aux fichiers `chap1` et `chap2`, entrez :

```
chmod g+w chap1 chap2
```

Le droit d'accès en écriture aux fichiers `chap1` et `chap2` est octroyé aux membres du groupe.

Pour effectuer plusieurs changements de droits au niveau du répertoire `mydir`, entrez :

```
chmod go-w+x mydir
```

Ce qui retire (-) aux membres du groupe (**g**) et aux autres (**o**) le droit de créer ou de supprimer des fichiers (**w**) dans `mydir` mais les autorise (+) à explorer `mydir` et à l'utiliser (**x**) dans un chemin d'accès. Cette commande équivaut à la séquence :

```
chmod g-w mydir
chmod o-w mydir
chmod g+x mydir
chmod o+x mydir
```

Par exemple, pour réserver à son propriétaire l'usage de la procédure shell `cmd` comme commande, entrez :

```
chmod u=rwx,go= cmd
```

Ceci octroie au propriétaire du fichier (**u=rwx**) le droit d'accès en lecture, en écriture et en exécution. Dans le même temps, ceci interdit au groupe et aux autres tout accès **go=** à `cmd`.

Par exemple, pour modifier les droit afférents au fichier `text` en utilisant la représentation numérique, entrez :

```
chmod 644 text
```

Ce qui accorde au propriétaire le droit d'accès en lecture et en écriture, les autres se contentant de la lecture seule.

Reportez-vous à la commande `chmod` dans le manuel *AIX Commands Reference*.

Listes de contrôle des accès (ACL)

Le contrôle des accès est assuré par des ressources protégées, qui déterminent qui est habilité à accéder à ces ressources. Le système d'exploitation permet le contrôle discrétionnaire et l'accès sélectif : le propriétaire d'une ressource peut accorder le droit d'accès en lecture ou en écriture à cette ressource ; le détenteur de droits sur une ressource peut transférer ces droits à d'autres utilisateurs. Ce système de sécurité autorise donc la circulation des informations, sous contrôle des utilisateurs : le propriétaire d'une ressource d'information définit les droits d'accès à l'objet.

Les utilisateurs disposent de droits d'accès de type utilisateur sur les seuls objets qui leur appartiennent. D'une façon générale, les utilisateurs se voient octroyer les droits de leur groupe ou les droits par défaut sur une ressource. La principale tâche au niveau de l'administration du contrôle des accès est de définir les membres des groupes, l'appartenance à un groupe déterminant de fait l'accès aux fichiers du système (autres que ceux créés par l'utilisateur).

Des droits étendus peuvent être définis, modifiant les droits de base attribués aux individus et aux groupes. Par le biais de ces droits étendus, vous pouvez autoriser ou interdire l'accès à un fichier sans modifier les droits de base.

Remarque : Une liste de contrôle des accès ne peut pas excéder une page mémoire (environ 4096 octets).

Les listes de contrôle des accès sont maintenues par les commandes **aclget**, **acledit** et **aclput**.

Bien que la commande **chmod**, en mode numérique (en notation octale), permette de définir les droits de base et les attributs, la sous-routine **chmod**, appelée par la commande, désactive les droits étendus : si vous utilisez le mode numérique de la commande **chmod** sur un fichier doté d'une liste ACL, les droits étendus sont désactivés. Le mode symbolique de la commande **chmod** ne désactive pas ces droits. Pour en savoir plus sur ces modes, reportez-vous à la commande **chmod**.

Droits d'accès de base

Les droits de base sont constitués des droits d'accès attribués traditionnellement au propriétaire du fichier, au groupe associé et aux autres utilisateurs. Il s'agit des droits d'accès : en lecture (r), en écriture (w) et en exécution/recherche(x).

Dans une liste de contrôle des accès, les droits de base sont au format suivant, le paramètre *Mode* étant exprimé sous la forme rwx (un tiret indiquant l'absence de droit) :

```
droit d'accès de base
owner (nom) : Mode
group (groupe) : Mode
others : Mode
```

Attributs

Trois attributs peuvent être ajoutés à une liste de contrôle des accès : **setuid** (SUID), **setgid** (SGID) et **savetext** (SVTX), sous la forme :

```
attributs : SUID, SGID, SVTX
```

Droits d'accès étendus

Les droits étendus sont un moyen pour le propriétaire d'un fichier d'affiner les droits afférents à ce fichier. Ils permettent de modifier les droits de base (utilisateur, groupe et autres) en accordant, en supprimant ou en spécifiant des droits spécifiques pour des individus, des groupes ou des combinaisons de groupes.

Les mots-clés **accès accordé**, **accès refusé** et **spécification** sont définis comme suit :

- accès accordé** Accorde à l'utilisateur ou au groupe le droit spécifié sur le fichier.
- accès refusé** Retire à l'utilisateur ou au groupe le droit spécifié sur le fichier.
- spécification** Définit précisément les droits de l'utilisateur ou du groupe sur le fichier.

Si un droit est refusé à un utilisateur par le biais de **accès refusé** ou de **spécification**, ce droit ne peut être rétabli par aucune autre entrée.

La liste de contrôle (ACL) doit être activée (mot-clé **activé**) pour que les droits étendus prennent effet. La valeur par défaut est **désactivé**.

Dans une liste ACL, les droits étendus apparaissent sous la forme :

```
droits d'accès étendus
  activé | désactivé
    accès accordé  Mode  InfoUtilisateur...:
    accès refusé   Mode  InfoUtilisateur...:
    spécification  Mode  InfoUtilisateur...:
```

Placez chacune des entrées **accès accordé**, **accès refusé** et **spécification** sur une ligne distincte. Le paramètre *Mode* est exprimé sous la forme **rwX** (un tiret indiquant l'absence de droit). Le paramètre *InfoUtilisateur* s'exprime sous la forme **u:NomUtilisateur** ou **g:NomGroupe**, ou encore par une combinaison de **u:NomUtilisateur** et **g:NomGroupe**.

Remarque : Si vous spécifiez plusieurs noms dans une entrée, celle-ci ne peut être utilisée dans une décision de contrôle d'accès, un process n'ayant qu'un seul ID utilisateur.

Exemple de liste de contrôle des accès

Voici un exemple d'ACL :

```
attributes: SUID
base permissions:
  owner(frank): rw-
  group(system): r-x
  others: ---
extended permissions:
  enabled
  permit rw- u:dhs
  deny r-- u:chas, g:system
  specify r-- u:john, g:gateway, g:mail
  permit rw- g:account, g:finance
```

Voici la signification des différents éléments de cette liste :

- La première ligne indique que le bit **setuid** est activé.
- La deuxième ligne, qui introduit les droits de base, est facultative.
- Les trois lignes suivantes précisent ces droits. Les noms du propriétaire et du groupe (entre parenthèses) sont donnés à titre d'information : les modifier n'a pas d'incidence sur le propriétaire réel du fichier, pas plus que sur le groupe auquel appartient le fichier. Seules les commandes **chown** et **chgrp** permettent de modifier ces attributs.
- La ligne suivante, qui introduit les droits étendus, est facultative.
- La ligne suivante indique que les droits étendus qui suivent sont activés.

- Les quatre dernières lignes correspondent aux droits étendus : la première accorde à l'utilisateur `dhs` l'accès en lecture (r) et en écriture (w) au fichier.
- La deuxième interdit l'accès en lecture (r) à l'utilisateur `chas` lorsqu'il est membre du groupe `system`.
- La troisième accorde à l'utilisateur `john` l'accès en lecture (r) tant qu'il est membre des deux groupes `gateway` et `mail`. S'il n'appartient plus à ces deux groupes, l'accès lui est refusé.
- La dernière ligne, enfin, accorde à tout utilisateur membre des deux groupes `account` et `finance` l'accès en lecture (r) et en écriture (w).

Remarque : Plusieurs droits étendus peuvent être appliqués à un process, les interdits primant sur les autorisations.

Reportez-vous à la commande **acedit** dans le manuel *AIX Commands Reference*.

Autorisations d'accès

La gestion des droits d'accès est du ressort du propriétaire de la ressource concernée. Les ressources sont protégées par des bits d'accès, intégrés au mode de l'objet. Ces bits définissent les droits du propriétaire de l'objet, ceux du groupe correspondant et ceux des autres utilisateurs (regroupés par défaut dans une classe). Le système d'exploitation gère trois droits d'accès (lecture, écriture et exécution), qui peuvent être accordés séparément.

Lorsqu'un utilisateur se connecte à un compte (via une commande **login** ou **su**), les ID utilisateur et groupe attribués au compte sont associés aux process de cet utilisateur et en déterminent les droits d'accès.

Pour les fichiers, les répertoires, les tubes nommés, les files d'attente de messages, les segments de mémoire partagée, les sémaphores et les unités (fichiers spéciaux), les accès sont définis comme suit :

- Pour chaque entrée (ACE) de la liste de contrôle (ACL), la liste des identificateurs est comparée aux identificateurs du process. Si elles sont identiques, le process se voit attribuer les droits et les interdits définis pour cette entrée. Les correspondances logiques pour les droits comme pour les interdits sont calculées pour chaque entrée concernée de l'ACL. Si le process demandeur ne correspond à aucune entrée de l'ACL, il se voit attribuer les droits associés à l'entrée par défaut.
- Si le droit d'accès demandé est autorisé (compris dans la somme des droits) et non interdit (compris dans la somme des interdits), le droit demandé est accordé. Sinon, il est refusé.

Un process doté de l'ID utilisateur 0 est dit process utilisateur racine. Ces process ont généralement tous les droits. Mais si un process racine demande le droit d'exécution sur un programme, celui-ci ne lui est accordé que s'il est détenu par au moins un utilisateur.

La liste des identificateurs d'une ACL correspond à un process si tous ces identificateurs correspondent à l'identificateur effectif – de même type – du process demandeur. Un identificateur de type USER "correspond" à un process s'il est identique à l'ID utilisateur effectif du process. Un identificateur de type GROUPE "correspond" s'il est identique à l'ID groupe effectif du process ou à l'un des ID des groupes complémentaires. Ainsi, une liste ACE avec la liste d'identificateurs :

```
USER:fred, GROUP:philosophers, GROUP:yankee_fans
```

correspond au process dont l'ID utilisateur est `fred`, et l'ensemble de groupes :

```
philosophers, philanthropists, yankee_fans, good_sports
```

mais pas au process dont l'ID utilisateur est `fred`, et l'ensemble de groupes :

```
philosophers, iconoclasts, redsox_fans, poor_sports
```

Notez qu'une ACE avec la liste suivante correspond aux deux process :

```
USER:fred, GROUP:philosophers
```

En d'autres termes, la liste d'identificateurs de l'ACE fonctionne comme un ensemble de conditions, à respecter pour que l'accès spécifié soit accordé.

Tous les contrôles d'accès pour ces objets sont effectués au niveau de l'appel système, au moment du premier accès aux objets. Dans la mesure où l'accès aux objets SVIPC n'est pas nominatif, les contrôles sont effectués à chaque accès. Pour les objets dotés de noms de systèmes de fichiers, il convient de "résoudre" le nom de l'objet réel, soit relativement (par rapport au répertoire de travail du process), soit absolument (par rapport au répertoire racine du process). Toute résolution de nom commence par la recherche de l'un de ces deux éléments.

Le mécanisme de contrôle d'accès discrétionnaire assure un contrôle effectif de l'accès aux ressources et assure une protection distincte de la confidentialité et de l'intégrité des informations. Les mécanismes de contrôle gérés par le propriétaire ne sont effectifs que s'ils sont définis par les utilisateurs. Tous les utilisateurs doivent maîtriser le mécanisme d'octroi et de refus de droits d'accès.

Affichage des informations de contrôle des accès (commande **aclget**)

La commande **aclget** d'afficher les informations de contrôle des accès à un fichier : notamment, attributs, droits de base et droits étendus.

Par exemple, pour afficher les informations de contrôle des accès au fichier `status`, entrez :

```
aclget status
```

Les attributs et les droits (de base et étendus) associés au fichier sont affichés. Un exemple est donné à la section "Exemple de liste de contrôle des accès", page 9-10.

Par exemple, pour sauvegarder les informations de contrôle des accès au fichier `plans` dans le fichier `acl11` (en vue d'une édition ultérieure), entrez :

```
aclget -o acl11 plans
```

Reportez-vous à la commande **aclget** dans le manuel *AIX Commands Reference*.

Définition du contrôle des accès (commande **aclput**)

La commande **aclput** permet de définir les informations de contrôle des accès à un fichier.

Remarque : Une liste de contrôle des accès ne peut pas excéder une page mémoire (environ 4096 octets).

Par exemple, pour définir les informations de contrôle des accès au fichier `status` à partir des informations enregistrées dans le fichier `acldefs`, entrez :

```
aclput -i acldefs status
```

Par exemple, pour définir les informations de contrôle des accès au fichier `status` comme étant identiques à celles du fichier `plans`, entrez :

```
aclget plans | aclput status
```

Reportez-vous à la commande **aclput** dans le manuel *AIX Commands Reference*.

Modification des informations de contrôle des accès (commande **acledit**)

La commande **acledit** permet de modifier les informations de contrôle des accès. Elle affiche les informations courantes et permet au propriétaire du fichier de les modifier. Une confirmation est demandée, avant toute modification permanente.

Remarque : La variable d'environnement **EDITOR** doit être spécifiée avec son chemin d'accès complet, faute de quoi la commande **acledit** échoue.

Les attributs et les droits (de base et étendus) associés au fichier sont affichés. Un exemple est donné à la section "Exemple de liste de contrôle des accès", page 9-10.

Par exemple, pour modifier les informations de contrôle des accès au fichier `plans`, entrez :

```
acledit plans
```

Reportez-vous à la commande **acledit** dans le manuel *AIX Commands Reference*.

Verrouillage du terminal (commande lock ou xlock)

La commande **lock** permet de verrouiller votre terminal. Dans l'intervalle, le terminal est bloqué et ne sera déverrouillé qu'à la seconde entrée du mot de passe. Cette commande vous invite à entrer un mot de passe, le lit et vous invite à l'indiquer une seconde fois. Le délai par défaut est de 15 minutes, mais vous pouvez le modifier via l'indicateur *-Nombre*.

Remarque : Si vous travaillez sous AIXwindows, utilisez la commande **xlock** de la même manière.

Par exemple, pour verrouiller votre terminal sous contrôle de mot de passe, entrez :

```
lock
```

Vous êtes invités par deux fois à entrer votre mot de passe, de sorte que le système puisse le vérifier. Si vous ne retapez pas votre mot de passe dans un délai de 15 minutes, la commande est abandonnée.

Par exemple, pour réserver un terminal sous contrôle de mot de passe, avec un délai de 10 minutes, entrez :

```
lock -10
```

Reportez-vous à la commande **lock** ou **xlock** dans le manuel *AIX Commands Reference*.

Contrôle antivirus (commande **virscan**)

La commande **virscan** permet de détecter nombre de virus informatiques courants. Elle examine les fichiers exécutables, à la recherche des signatures de virus connues lors de la mise à disposition de cette version du programme. Une *signature* est une trame de bits spécifiques d'un virus. Les signatures de virus sont obtenues par "inversion de traitement" sur des échantillons de virus. La recherche de virus via le programme **virscan** ne peut avoir lieu que sur des fichiers au format exécutable d'origine (non cryptés et non paqués).

Remarque : La commande **virscan** *ne détruit pas* les virus, n'inhibe pas leur propagation et ne restaure pas les programmes ou les données contaminés. Si vous suspectez un virus, prévenez l'administrateur système ou le service approprié.

La commande **virscan** examine les fichiers à la recherche de trames de bits correspondant aux signatures des virus connus. Le risque demeure cependant que de nouveaux virus viennent infester votre système (il n'existe pas de solution miracle permanente) aussi est-il recommandé d'effectuer des sauvegardes régulières et d'être prudent lors de l'acquisition et de la mise en service de nouveaux logiciels.

Par exemple, pour effectuer un contrôle sur le système de fichiers `/usr`, entrez :

```
virscan /usr
```

Pour afficher l'aide en ligne sur la commande **virscan**, entrez **virscan** sans argument. Pour afficher la liste des options ligne de commande disponibles, entrez :

```
virscan -h
```

Reportez-vous à la commande **virscan** dans le manuel *AIX Commands Reference*.

Format des fichiers signature

Pour repérer les virus, la commande **virscan** se sert des signatures de virus trouvées, le cas échéant, dans les fichiers **virsig.lst** et **addenda.lst**.

Les lignes de commentaire des deux fichiers commencent par un astérisque (*). Ces lignes sont ignorées par la commande **virscan**. Ces commentaires n'ont qu'une fonction informative.

Le corps du fichier est constitué d'entrées qui indiquent à la commande **virscan** ce qu'il convient de faire avec chaque virus. Chaque entrée est composée de trois lignes :

1. La première est constituée d'une chaîne hexadécimale, examinée par la commande **virscan** pour déterminer si un virus particulier se trouve dans le fichier. Voici quelques recommandations concernant les signatures :
 - Utilisez 24 chiffres hexadécimaux au minimum (soit au moins 12 octets) pour chaque signature ajoutée au fichier **addenda.lst**, davantage si possible. Des signatures plus courtes risquent d'induire des rapports inexacts.
 - Extrayez la signature d'une zone de code, et non d'une zone de données, pour minimiser le risque de fausses alertes.
 - Testez les nouvelles signatures sur un certain nombre de programmes avant de les utiliser à grande échelle, pour vérifier que les programmes courants ne génèrent pas de fausses alertes sur ces signatures.

Pour indiquer qu'une position d'octet dans la signature peut prendre une valeur quelconque, spécifiez deux points d'interrogation (??) à la place d'une paire de caractères hexadécimaux (représentant un octet de la signature). Ne comptez pas les octets ?? lorsque vous vérifiez les règles relatives à la longueur de la chaîne. Aucun examen sur les variations de chaîne n'est effectué sur une signature contenant une sous-chaîne ??, même en présence de l'indicateur **-m**. En outre, toute valeur FF dans la signature est traitée comme un octet ??.

2. La deuxième entrée est un message, en minuscules ou en majuscules, affiché si la chaîne est détectée dans les fichiers spécifiés, qui indique qu'un virus a été repéré.
3. La troisième entrée est constituée d'une ou de plusieurs chaînes indiquant ce que doit faire la commande **virscan** en présence de la signature. Ces chaînes sont les suivantes :

EXE	Le message doit être affiché si la signature est repérée dans un fichier exécutable.
Offset	Suivi d'une chaîne (délimitée par des espaces), obligatoirement numérique, est utilisé comme valeur de décalage (entier) dans le fichier objet dans lequel la signature du virus est supposée se trouver. Aucun message n'est affiché si la signature est repérée ailleurs.
No mutants	Indique à la commande virscan de ne pas rechercher de variations de ce virus même en présence de l'indicateur -m . Si une signature semble générer de fausses alertes lorsque l'indicateur -m est spécifié, cette chaîne les élimine.

Par exemple, supposons qu'un nouveau virus, Purple Virus, infeste les fichiers exécutables, et que vous avez découvert la chaîne suivante dans tous les exemplaires du virus :

```
EA6061626364786566676869716A6B6C6D6E516FC0C1C8C958D6F1
```

Pour que ce virus soit détecté par la commande **virscan**, ajoutez la ligne suivante au fichier **addenda.lst** :

```
*
* Entry for the nonexistent Purple Virus (just an example)
*
EA6061626364786566676869716A6B6C6D6E516FC0C1C8C958D6F1
A file on this disk may have the Purple Virus.
(EXE)
```

Ajout de signatures de virus

Pour rechercher des signatures autres que celles répertoriées dans le fichiers **virsig.lst** :

1. Créez un fichier **addenda.lst** contenant les nouvelles signatures.
2. Placez ce fichier dans le répertoire **/usr/lib/security/scan**, de sorte que la commande **virscan** puisse le trouver et le charger avec le fichier **virsig.lst**.
3. Spécifiez l'option **-v** pour vérifier que **virscan** charge correctement le fichier **addenda.lst**.

Le fichier **addenda.lst** a le même format que le fichier **virsig.lst**.

Récapitulatif des commandes relatives à la sécurité

acledit	Modifie les données de contrôle d'accès à un fichier.
aclget	Affiche les données de contrôle d'accès à un fichier.
aclput	Définit les données de contrôle d'accès à un fichier.
chmod	Modifie les droits d'accès.
chown	Change le propriétaire d'un fichier.
lock	Réserve un terminal.
lsgroup	Affiche les attributs de groupes.
virscan	Examine les fichiers à la recherche de virus.
xlock	Verrouille l'écran X local (jusqu'à l'entrée d'un mot de passe).

Chapitre 10. Personnalisation de l'environnement utilisateur

AIX met à votre disposition plusieurs commandes et fichiers d'initialisation vous permettant de personnaliser à votre gré votre environnement de travail.

Il est également possible de personnaliser certaines ressources par défaut des applications utilisées sur votre système. Les valeurs par défaut sont lancées par le programme au démarrage. Si vous les modifiez, vous devez quitter le programme et le relancer pour que les nouvelles valeurs soient prises en compte.

Common Desktop Environment 1.0: Advanced User's and System Administrator's Guide fournit des informations détaillées sur la personnalisation de CDE Desktop.

Ce chapitre traite des points suivants :

- Support AIX pour la X/Open UNIX95 Specification, page 10-2
- Fichiers de lancement du système : généralités, page 10-3.
 - Fichier `/etc/profile`
 - Fichier `/etc/environment`
 - Fichier `.profile`
 - Fichier `.env`
- Fichiers de lancement AIXwindows : généralités, page 10-6.
 - Fichier `.xinitrc`
 - Fichier `.Xdefaults`
 - Fichier `.mwmrc`
- Procédures de personnalisation, page 10-11.
 - Exportation de variables shell (commande shell **export**), page 10-11.
 - Changement de la police d'affichage (commande **chfont**), page 10-11.
 - Changement de l'affectation des touches de contrôle (commande **stty**), page 10-12.
 - Changement de l'invite système, page 10-12.
- Personnalisation du programme Windows InfoExplorer, page 10-13.
- Récapitulatif des commandes de personnalisation, page 10-16

Voir aussi

Système et environnement utilisateur

Commandes : généralités

Process : généralités

Systèmes de fichiers : généralités

Répertoires : généralités

Fichiers : généralités

Sécurité du système et des fichiers

Fichiers de lancement du système

Procédures de personnalisation

Personnalisation d'InfoExplorer (ASCII)

Personnalisation des shells

Shell Korn ou POSIX

Shell Bourne

Shell C

Support AIX pour la X/Open UNIX95 Specification

A partir d'AIX version 4.2.2, le système d'exploitation est conçu pour prendre en charge les X/Open UNIX95 Specification pour la portabilité des systèmes d'exploitation basés sur UNIX. De nombreuses interfaces ont été ajoutées, d'autres ont été améliorées pour répondre à ces spécifications. A partir de la version 4.2, AIX est encore plus ouvert et portable pour les applications.

En outre, la compatibilité avec les versions antérieures d'AIX est maintenue grâce à la création d'une nouvelle variable d'environnement, qui sert à configurer l'environnement sur la base du système, de l'utilisateur ou du process.

L'environnement AIX par défaut maintient la compatibilité avec les versions antérieures d'AIX. Pour obtenir un environnement conforme à la spécification UNIX95, la variable d'environnement **XPG_SUS_ENV** doit avoir la valeur **ON**. Si **XPG_SUS_ENV** n'a pas de valeur ou en a une autre, l'environnement par défaut d'AIX sera utilisé.

Lorsque la valeur de **XPG_SUS_ENV** est définie, les programmes de cet environnement fonctionnent avec l'environnement du système d'exploitation spécifié UNIX95. Il se peut que certaines applications compilées pour un environnement AIX (une version antérieure d'AIX, par exemple) ne fonctionnent pas avec la valeur de **XPG_SUS_ENV**.

Fichiers de lancement du système : généralités

Lorsque vous vous connectez, le shell définit votre environnement utilisateur en fonction des fichiers d'initialisation que vous avez configurés. C'est-à-dire, en fonction des variables d'environnement. L'environnement reste actif jusqu'à ce que vous vous déconnectiez.

Au moment de la connexion au système d'exploitation, le shell se sert de deux types de fichiers de profil. Il évalue les commandes de ces fichiers, puis exécute les commandes de configuration de l'environnement. Ces fichiers ont des fonctions similaires, si ce n'est que le fichier **/etc/profile** contrôle les variables de profil de tous les utilisateurs d'un système, tandis que le fichier **.profile** permet de personnaliser votre propre environnement.

Le shell évalue d'abord les commandes du fichier **/etc/profile**, puis exécute les commandes de configuration de votre environnement du fichier **/etc/environment**. Le shell vérifie ensuite si vous disposez d'un fichier **.profile** dans votre répertoire personnel. Si un fichier **.profile** existe, le fichier est lancé. Le fichier **.profile** indique également s'il existe un fichier d'environnement. Dans l'affirmative (le fichier s'appelle généralement **.env**), le système l'exécute et configure les variables d'environnement.

Les fichiers **/etc/profile**, **/etc/environment** et **.profile** sont exécutés une fois au moment de la connexion. Le fichier **.env**, lui, est exécuté chaque fois que vous ouvrez un nouveau shell ou une fenêtre.

Cette section traite des points suivants :

- Fichier **/etc/profile**
- Fichier **/etc/environment**
- Fichier **profile**
- Fichier **.env**

Fichier **/etc/profile**

Le fichier **/etc/profile** est le premier qu'utilise le système au moment de la connexion. Ce fichier contrôle des variables par défaut applicables à l'intégralité du système, telles que :

- les variables d'exportation,
- le masque de création de fichier (umask),
- les types de terminaux,
- les messages indiquant l'arrivée d'un nouveau courrier par la messagerie.

L'administrateur système configure le fichier **profile** pour tous les utilisateurs du système. Lui seul est habilité à le modifier.

Voici un exemple de fichier **/etc/profile** :

```
#Set file creation mask
unmask 022
#Tell me when new mail arrives
MAIL=/usr/mail/$LOGNAME
#Add my /bin directory to the shell search sequence
PATH=/usr/bin:/usr/sbin:/etc::
#Set terminal type
TERM=lft
#Make some environment variables global
export MAIL PATH TERM
```

Reportez-vous à ".profile File Format" dans le manuel *AIX Files Reference* pour en savoir plus sur le fichier **/etc/profile**.

Fichier `/etc/environment`

Le fichier `/etc/environment` est le deuxième utilisé par le système au moment de la connexion. Il contient des variables spécifiant l'environnement de base de tous les process. Lorsqu'un nouveau process est lancé, la sous-routine **exec** constitue un tableau des chaînes disponibles, sous la forme *Nom=Valeur*. Ce tableau est appelé environnement. Chaque nom défini par une chaîne étant appelé variable d'environnement ou variable shell. La sous-routine **exec** permet de définir en une seule fois l'ensemble de l'environnement.

Lorsque vous vous connectez, le système définit les variables d'environnement à partir du fichier **environment**, avant de lire votre profile de connexion, **.profile**. Voici les variables constitutives de l'environnement de base.

HOME	Chemin d'accès complet au répertoire de connexion ou HOME de l'utilisateur. Le programme login lui affecte le nom défini dans le fichier <code>/etc/passwd</code> .
LANG	Nom courant actuel. La variable LANG est initialement définie dans le fichier <code>/etc/profile</code> au moment de l'installation.
NLSPATH	Chemin d'accès complet aux catalogues de messages.
LOCPATH	Chemin d'accès complet à l'emplacement des tables NLS (National Language Support).
PATH	Séquence des répertoires à explorer par des commandes, telles que sh , time , nice et nohup , à la recherche d'une commande dont le chemin d'accès est incomplet.
TZ	Zone temps. La variable d'environnement TZ est initialement définie dans le fichier <code>/etc/profile</code> , le fichier de connexion système.

Reportez-vous à "environment File" dans le manuel *AIX Files Reference*.

Fichier `.profile`

Le troisième fichier utilisé par le système au moment de la connexion est le fichier **.profile**. Ce fichier, qui se trouve dans votre répertoire personnel (**\$HOME**), vous permet de personnaliser votre environnement de travail. Il s'agit d'un fichier caché : lancez la commande **li -a** pour le consulter.

Une fois que le programme **login** a intégré les variables **LOGNAME** (nom de connexion) et **HOME** (répertoire de connexion) à l'environnement, les commandes du fichier **\$HOME/.profile** (si présent) sont exécutées. Ce fichier contient votre propre profil, qui prime sur les variables définies dans le fichier `/etc/profile`. Le fichier **.profile** est souvent utilisé pour définir les variables d'environnement exportées et les modes du terminal. Adaptez à votre gré le fichier **.profile**. Il contrôle les paramètres par défaut suivants :

- shells à ouvrir,
- style de l'invite,
- son du clavier.

Voici un exemple de fichier **.profile** :

```
PATH=/usr/bin:/etc:/home/bin1:/usr/lpp/tps4.0/user::
epath=/home/gsc/e3:
export PATH epath
csh
```

Dans cet exemple, deux variables de chemin (**PATH** et **epath**) sont définies, puis exportées, et un shell C (**csh**) est ouvert.

Vous pouvez également vous servir du fichier **.profile** (ou, à défaut, du fichier `/etc/profile`) pour déterminer les variables du shell de connexion, ou pour personnaliser d'autres environnements shell. Vous pouvez ainsi, par exemple, utiliser les fichiers **.cshrc** et **.kshrc** pour personnaliser un shell C et un shell Korn (respectivement).

Fichier .env

Un quatrième fichier utilisé par le système au moment de la connexion est le fichier **.env**, sous réserve que votre fichier **.profile** contienne la ligne : `export ENV=$HOME/.env`

Le fichier **.env** permet de personnaliser vos propres variables d'environnement de travail. Comme il est caché, lancez la commande **li -a** pour le consulter. Ce fichier contient votre propre profil, qui prime sur les variables définies dans le fichier **/etc/environment**. Adaptez à votre gré le fichier **.env** – qui contrôle les paramètres par défaut suivants : Voici un exemple de fichier **.env** :

```
export myid=`id | sed -n -e 's/).*$/' -e 's/^.*(//p' ` \  
bold=`tput smso` \  
norm=`tput rmso` \  
#set prompt: login & system name (reverse video) & path (normal) \  
if [ $myid = root ] \  
then typeset -x PSCH='${bold}#:${norm}\${PWD}> ' \  
PS1='${bold}#:${norm}\${PWD}> ' \  
else typeset -x PSCH='>' \  
PS1='${bold}$LOGNAME@$UNAME:${norm}\${PWD}> ' \  
PS2='>' \  
PS3='#?' \  
fi \  
export PS1 PS2 PS3 \  
#setup my command aliases \  
alias ls="/bin/ls -CF" \  
d="/bin/ls -Fal | pg" \  
rm="/bin/rm -i" \  
up="cd .."
```

Remarque : Si vous modifiez le fichier **.env**, veillez à ce que les variables nouvellement définies n'entrent pas en conflit avec les variables standard telles que **MAIL**, **PS1**, **PS2** et **IFS**.

Fichiers de lancement AIXwindows : généralités

Lorsque vous travaillez dans un environnement graphique, le système utilise également certains fichiers d'initialisation.

Le mode de démarrage du serveur X et d'AIXwindows dépend de votre système informatique : renseignez-vous auprès de votre administrateur système. Le plus souvent, le serveur X et AIXwindows sont lancés via un script shell, exécuté automatiquement au moment de la connexion. Mais il vous faudra peut-être lancer l'un et/ou l'autre explicitement.

Si, une fois connecté, vous vous trouvez devant un écran simple, sans fenêtres, vous pouvez lancer X en entrant :

```
xinit
```

Si cette commande reste sans effet, vérifiez, avec l'aide de votre administrateur, que le répertoire X11, contenant les programmes exécutables, se trouve bien dans votre chemin de recherche. Ce chemin varie selon les systèmes.

Si, une fois connecté, vous vous trouvez avec une ou plusieurs fenêtres sans cadre, vous pouvez lancer AIXwindows Window Manager en entrant :

```
mwm &
```

Remarque : Avant de lancer cette commande, vérifiez que le pointeur se trouve dans une fenêtre affichant une invite système.

AIXwindows autorisant toutes sortes de personnalisations, tant par les programmeurs via des applications AIXwindows que par les utilisateurs, vous trouverez peut-être que les boutons de la souris ou d'autres fonctions ne fonctionnent pas exactement comme nous vous l'indiquons dans ce manuel. Il vous suffit dans ce cas de revenir à l'environnement AIXwindows par défaut en appuyant simultanément sur les quatre touches :

Alt–Ctrl–Maj–!

Vous pouvez annuler cette réinitialisation en réappuyant sur la même combinaison de touches. Si votre système ne reconnaît pas cette combinaison, vous pouvez également restaurer l'environnement par défaut via le menu racine.

Cette section traite des points suivants :

- Fichier **.xinitrc**
- Fichier **.Xdefaults**
- Fichier **.mwmrc**

Fichier **.xinitrc**

La commande **xinit** se sert d'un fichier script shell personnalisable qui liste les programmes client X à démarrer. Le fichier **.xinitrc** de votre répertoire personnel contrôle les fenêtres et les applications qui sont lancées en même temps qu'AIXwindows.

Pour lancer AIXwindows, la commande **xinit** recherche d'abord la variable d'environnement **\$XINITRC**. Si elle ne la trouve pas, elle recherche alors le script shell **\$HOME/.xinitrc**. Si elle ne le trouve pas non plus, elle lance le script shell **/usr/lib/X11/\$LANG/xinitrc**. Si **/usr/lib/X11/\$LANG/xinitrc** est introuvable, la commande recherche le script shell **/usr/lpp/X11/defaults/\$LANG/xinitrc**. En cas d'échec, elle recherche le script shell **/usr/lpp/X11/defaults/xinitrc**.

Le script shell **xinitrc** lance des commandes telles que **mwm** (AIXwindows Window Manager), **aixterm** et **xclock**.

La commande **xinit** :

- lance un serveur X sur l'écran courant ;
- définit la variable d'environnement **\$DISPLAY** ;
- exécute le fichier **xinitrc** pour lancer les programmes client X ;
- charge automatiquement l'extension DPS (Display PostScript).

Remarque : Pour empêcher le chargement automatique de **DPS**, vous devez modifier le fichier **/usr/lpp/X11/xinit**.

Voici la partie personnalisable du fichier **xinitrc** :

```
# This script is invoked by /usr/lpp/X11/bin/xinit
.
.
.
#*****
# Start the X clients. Change the following lines to          *
# whatever command(s) you desire!                            *
# The default clients are an analog clock (xclock), an lft    *
# terminal emulator (aixterm), the X Desktop Manager (xdt),  *
# and the Motif Window Manager (mwm).                        *
#*****
# If not X-Station then invoke the DPS copyright
if [ -z "$XSTATION" ]
then
    /usr/lpp/DPS/bin/copyright -x &
fi
aixterm =80x25+0-0 -fg Wheat -bg MidnightBlue &
if [ -z "$XSTATION" ]
then
    sleep 3          # allow for DPS copyright to show up
fi
if [ -x /usr/lpp/X11/bin/xdt3 ]
then
    /usr/lpp/X11/bin/xdt3 &
fi
exec mwm
```

Fichier **.Xdefaults**

Si vous travaillez dans une interface AIXwindows, vous pouvez, via le fichier **.Xdefaults**, personnaliser l'interface : couleurs, police, etc.

Plusieurs éléments d'une application basée sur des fenêtres sont contrôlés par des ensembles de variables appelées *ressources*. L'aspect, visuel ou fonctionnel, d'une ressource dépend de la valeur qui lui est attribuée. Il existe plusieurs types de valeurs : par exemple, une ressource contrôlant les couleurs peut se voir affecter la valeur prédéfinie *DarkSlateBlue* ou *Black*. Les ressources spécifiant des dimensions sont affectées de valeurs numériques, et d'autres de valeurs booléennes (*vrai* ou *faux*).

Si votre répertoire personnel ne contient pas de fichier **.Xdefaults**, vous pouvez en créer un à l'aide d'un éditeur texte quelconque. Vous avez ensuite toute latitude pour y définir les ressources de votre choix. Un fichier modèle, **Xdefaults.tmpl**, se trouve dans le répertoire **/usr/lpp/X11/defaults**.

Voici un extrait d'un fichier **.Xdefaults** :

```
*AutoRaise: on
*DeIconifyWarp: on
*warp:on
*TitleFont:andysans12
*scrollBar: true
*font: Rom10.500
Mwm*menu*foreground: black
Mwm*menu*background: CornflowerBlue
Mwm*menu*RootMenu*foreground: black
Mwm*menu*RootMenu*background: CornflowerBlue
Mwm*icon*foreground: grey25
Mwm*icon*background: LightGray
Mwm*foreground: black
Mwm*background: LightSkyBlue
Mwm*bottomShadowColor: Blue1
Mwm*topShadowColor: CornflowerBlue
Mwm*activeForeground: white
Mwm*activeBackground: Blue1
Mwm*activeBottomShadowColor: black
Mwm*activeTopShadowColor: LightSkyBlue
Mwm*border: black
Mwm*highlight:white

aixterm.foreground: green
aixterm.background: black
aixterm.fullcursor: true
aixterm.ScrollKey: on
aixterm.autoRaise: true
aixterm.autoRaiseDelay: 2
aixterm.boldFont:Rom10.500
aixterm.geometry: 80x25
aixterm.iconFont: Rom8.500
aixterm.iconStartup: false
aixterm.jumpScroll: true
aixterm.reverseWrap: true
aixterm.saveLines: 500
aixterm.scrollInput: true
aixterm.scrollKey: false
aixterm.title: AIX
```

Fichier **.mwmrc**

Les caractéristiques le plus souvent personnalisées correspondant à des ressources du fichier **.Xdefaults**. Toutefois, les affectations des touches et des boutons de la souris, de même que les définitions de menus du gestionnaire de fenêtres sont spécifiées dans un fichier complémentaire, **.mwmrc**, référencé par des ressources du fichier **.Xdefaults**.

Si votre répertoire personnel ne contient pas de fichier **.mwmrc**, vous pouvez le copier via la commande :

```
cp /usr/lib/X11/system.mwmrc .mwmrc
```

Les spécifications du fichier **.mwmrc** priment sur celles, globales, définies dans le fichier **system.mwmrc** : vous ne risquez pas d'interférer avec les spécifications des autres.

Voici un extrait d'un fichier **system.mwmrc** :

```
# DEFAULT mwm RESOURCE DESCRIPTION FILE (system.mwmrc)
#
# menu pane descriptions
#
# Root Menu Description
Menu RootMenu
{ "Root Menu"          f.title
  no-label             f.separator
  "New Window"        f.exec "aixterm &"
  "Shuffle Up"        f.circle_up
  "Shuffle Down"      f.circle_down
  "Refresh"           f.refresh
  no-label             f.separator
  "Restart"           f.restart
  "Quit"              f.quit_mwm
}

# Default Window Menu Description

Menu DefaultWindowMenu MwmWindowMenu
{ "Restore"    _R    Alt<Key>F5          f.normalize
  "Move"       _M    Alt<Key>F7          f.move
  "Size"       _S    Alt<Key>F8          f.resize
  "Minimize"   _n    Alt<Key>F9          f.minimize
  "Maximize"   _x    Alt<Key>F10        f.maximize
  "Lower"      _L    Alt<Key>F3          f.lower
  no-label     f.separator
  "Close"      _C    Alt<Key>F4          f.kill
}

# no acclerator window menu
Menu NoAccWindowMenu
{
  "Restore"    _R    f.normalize
  "Move"       _M    f.move
  "Size"       _S    f.resize
  "Minimize"   _n    f.minimize
  "Maximize"   _x    f.maximize
  "Lower"      _L    f.lower
  no-label     f.separator
  "Close"      _C    f.kill
}

Keys DefaultKeyBindings
{
  Shift<Key>Escape      icon|window      f.post_wmenu
  Meta<Key>space        icon|window      f.post_wmenu
  Meta<Key>Tab          root|icon|window f.next_key
  Meta Shift<Key>Tab    root|icon|window f.prev_key
  Meta<Key>Escape      root|icon|window f.next_key
  Meta Shift<Key>Escape root|icon|window f.prev_key
  Meta Ctrl Shift<Key>exclam root|icon|window f.set_behavior
}

#
# button binding descriptions
#
```

```

Buttons DefaultButtonBindings
{
  <Btn1Down>          frame|icon          f.raise
  <Btn3Down>          frame|icon          f.post_wmenu
  <Btn1Down>          root                f.menu  RootMenu
  <Btn3Down>          root                f.menu  RootMenu
  Meta<Btn1Down>     icon|window        f.lower
  Meta<Btn2Down>     window|icon        f.resize
  Meta<Btn3Down>     window              f.move
}

Buttons PointerButtonBindings
{
  <Btn1Down>          frame|icon          f.raise
  <Btn2Down>          frame|icon          f.post_wmenu
  <Btn3Down>          frame|icon          f.lower
  <Btn1Down>          root                f.menu  RootMenu
  Meta<Btn2Down>     window|icon        f.resize
  Meta<Btn3Down>     window|icon        f.move
}

#
#  END OF mwm RESOURCE DESCRIPTION FILE
#

```

Procédures de personnalisation

Il existe plusieurs moyens de personnaliser l'environnement système. Cette section traite des points suivants :

- Exportation de variables shell (commande shell **export**), page 10-11.
- Changement de la police d'affichage (commande **chfont**), page 10-11.
- Changement de l'affectation des touches de contrôle (commande **stty**), page 10-12.
- Changement de l'invite système, page 10-12.

Exportation de variables shell (commande shell **export**)

Une variable shell *locale* est une variable reconnue uniquement par le shell qui l'a créée. Si vous lancez un nouveau shell, toutes les variables définies pour l'ancien shell lui sont inconnues. Si vous souhaitez qu'il les reconnaisse, il vous faut les rendre *globales* en les exportant.

Vous pouvez, via la commande **export**, rendre globales des variables locales. Pour le faire automatiquement, exportez-les dans le fichier **.profile**.

Remarque : Les variables peuvent être exportées vers des shells enfant, mais pas vers des shells parent.

Par exemple, pour rendre globale la variable shell locale PATH, entrez :

```
export path
```

Par exemple, pour afficher toutes les variables exportées, entrez :

```
export
```

Le système affiche un écran semblable à celui-ci :

```
DISPLAY=unix:0
EDITOR=vi
ENV=$HOME/.env
HISTFILE=/u/denise/.history
HISTSIZ=500
HOME=/u/denise
LANG=En_US
LOGNAME=denise
MAIL=/usr/mail/denise
MAILCHECK=0
MAILMSG=**VOUS AVEZ DU COURRIER USE THE mail COMMAND TO SEE YOUR
MAILPATH=/usr/mail/denise?denise has mail !!!
MAILRECORD=/u/denise/.Outmail
PATH=/usr/ucb:/usr/lpp/X11/bin:/bin:/usr/bin:/etc:/u/denise:/u/de
nise/bin:/u/bin1
PWD=/u/denise
SHELL=/bin/ksh
```

Changement de la police d'affichage (commande **chfont**)

Vous pouvez modifier la police par défaut au lancement du système avec **chfont** ou **smit**. Une *palette de polices* est un fichier que le système utilise pour définir et identifier les polices disponibles.

Remarque : Seul un utilisateur racine est habilité à exécuter la commande **chfont**.

Commande **chfont**

Par exemple, pour passer à la cinquième police de la palette, entrez :

```
chfont -a5
```

La police ID 5 devient la police principale.

Par exemple, pour passer à la police Roman, en italique et en gras, en conservant la même taille, entrez :

```
chfont -n /usr/lpp/fonts/Itl14.snf /usr/lpp/fonts/Bld14.snf
/usr/lpp/fonts/Rom14.snf
```

Reportez-vous à la commande **chfont** dans le manuel *AIX Commands Reference*. Voir aussi "Liste des polices disponibles (commande **lsfont**)", page 2-4.

Commande **smit**

Vous pouvez également lancer la commande **chfont** via **smit**.

Pour sélectionner la police active, entrez :

```
smit chfont
```

Pour sélectionner la palette de polices, entrez :

```
smit chfontpl
```

Changement de l'affectation des touches de contrôle (commande **stty**)

Vous pouvez modifier l'affectation des touches de contrôle via la commande **stty**. Les changements valent jusqu'à la déconnexion suivante. Pour les rendre permanents, placez-les dans le fichier **.profile**.

Par exemple, pour définir Ctrl-Z comme touche d'interruption, entrez :

```
stty intr ^Z
```

Par exemple, pour restaurer toutes les touches de contrôle à leur valeur par défaut, entrez :

```
stty sane
```

Par exemple, pour afficher les paramètres courants, entrez :

```
stty -a
```

Reportez-vous à la commande **stty** dans le manuel *AIX Commands Reference*.

Changement de l'invite système

Le shell utilise trois variables d'invite :

- PS1** invite normale.
- PS2** invite à entrer des données supplémentaires.
- PS3** invite indiquant que vous détenez les droits de l'utilisateur racine.

Pour changer le contenu d'une invite, il suffit de modifier la variable correspondante. Les changements valent jusqu'à la déconnexion suivante. Pour les rendre permanents, placez-les dans le fichier **.env**.

Par exemple, pour afficher la valeur actuelle de la variable PS1, entrez :

```
echo "prompt is $PS1"
```

Le système affiche un écran semblable à celui-ci :

```
prompt is $
```

Par exemple, pour afficher l'invite `Ready>`, entrez :

```
PS1="Ready> "
```

Par exemple, pour que l'invite de suite devienne `Enter more->`, entrez :

```
PS2="Enter more->"
```

Par exemple, pour que l'invite racine soit `Root->`, entrez :

```
PS3="Root-> "
```

Personnalisation d'InfoExplorer (programme d'aide)

Vous pouvez personnaliser InfoExplorer en définissant à votre convenance valeurs par défaut et préférences. Vous avez ainsi la possibilité de choisir la base d'informations et les imprimantes à utiliser. Vous pouvez également spécifier des éléments comme le nombre d'événements consignés dans l'historique, le type d'impression, etc.

La principale différence entre valeur par défaut et préférence est que la plupart des valeurs par défaut sont initialisées par le programme InfoExplorer. Si vous modifiez une valeur par défaut (sauf les paramètres d'impression), il vous faut quitter le programme et le relancer. Les préférences, en revanche, sont immédiatement prises en compte.

Cette section traite des points suivants :

- Modification des paramètres par défaut d'InfoExplorer, page 10-13
- Choix des préférences dans InfoExplorer, page 10-14

Modification des paramètres par défaut d'InfoExplorer (Window)

Sous InfoExplorer, vous avez la possibilité de modifier nombre de paramètres par défaut. Ce sont ceux utilisés par InfoExplorer au moment du lancement. Sélectionnez **Defaults** dans le sous-menu **Options** du menu déroulant InfoExplorer pour définir les paramètres :

- taille des fenêtres,
- article de navigation,
- fichiers bookmark, history et note,
- fichier modèle de note,
- bases de données à explorer,
- imprimantes.

Modification de la taille des fenêtres

Lorsque vous lancez InfoExplorer, la taille des fenêtres est celle définie par défaut par le système. Vous pouvez personnaliser leur taille à votre convenance. Vous pouvez changer la taille des fenêtres de navigation et de lecture : sélectionnez l'option **Window Size** du menu déroulant Options. La nouvelle valeur est prise en compte pour la session en cours et pour les suivantes.

Pour plus de détails sur la procédure, reportez-vous à Setting Window Size with the InfoExplorer Program (Aide InfoExplorer). Pour des informations sur le changement de taille des fenêtres via le fichier .Xdefaults, reportez-vous à Setting Window Size with X Resources (Aide InfoExplorer).

Désignation de l'article de navigation par défaut

Par défaut, InfoExplorer affiche l'article Topic & Task Index comme article de navigation principal. Vous pouvez choisir un autre article : au lancement d'InfoExplorer, le nouvel article sera automatiquement affiché comme article de navigation.

Pour plus de détails sur la procédure, reportez-vous à **Setting Defaults (InfoExplorer Windows)**.

Désignation de fichiers par défaut

Si vous travaillez avec un stagiaire, vous souhaitez sans doute qu'il suive un chemin déterminé lorsqu'il accède à InfoExplorer. Pour gérer ce type d'action, vous pouvez désigner un fichier comme fichier d'historique par défaut: lorsque vous spécifiez ensuite l'option **.History**, ce fichier vous conduit (vous ou votre stagiaire) via le chemin désigné.

Vous pouvez de la même façon désigner un fichier bookmarks ou un fichier notes par défaut, et suivre les chemins correspondants dans les sessions suivantes.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (Aide InfoExplorer).

Désignation d'un fichier modèle de notes par défaut

Lorsque vous créez une note dans la documentation hypertexte, l'écran de l'éditeur apparaît vide : vous pouvez taper votre note. Si vous souhaitez vous servir d'un format particulier, vous pouvez spécifier un modèle de notes. Si vous en avez créé plusieurs, vous pouvez en désigner un comme modèle par défaut. Si vous utilisez ensuite l'option **Note**, InfoExplorer affiche le modèle par défaut.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (Aide InfoExplorer).

Modification de la base de données à explorer

Par défaut, InfoExplorer explore les bases chargées avec le programme. Mais il est parfois inutile de les explorer toutes. Pour mieux cibler les recherches, vous pouvez indiquer les bases qu'il convient d'explorer, pour la session en cours et les suivantes. Et, si vous changez d'avis, il est très simple de revenir à la sélection par défaut.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (Aide InfoExplorer).

Désignation des imprimantes par défaut

Le choix d'une imprimante dépend, entre autres, du type d'impression souhaité. **Pretty Print** fait appel au programme **troff**, qui convertit les documents en sorties exploitables par une photocomposeuse. Sous InfoExplorer, vous pouvez désigner l'imprimante par défaut pour chaque type d'impression : simple (texte), sophistiquée (avec enrichissements et caractéristiques de styles) ou graphique. Lorsque vous émettez une commande d'impression à partir d'un écran comportant un type donné d'impression ou un graphique, la sortie est automatiquement dirigée vers l'imprimante ad hoc.

Par exemple, l'option **Print Graphic** envoie automatiquement le travail d'impression vers l'imprimante désignée pour les impressions graphiques.

Remarque : L'option Pretty Print n'est active que si le programme de formatage de document **troff** est installé sur le système.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (Aide InfoExplorer).

Choix des préférences dans InfoExplorer (Window)

Pour définir vos préférences sous InfoExplorer, sélectionnez l'option **Preferences** du menu déroulant Customize.- Vous pourrez ainsi redéfinir :

- la taille de la police,
- l'affichage des graphiques,
- les articles rémanents,
- l'article d'impression,
- la taille de l'historique,
- la bibliothèque de prédilection.

Changement de la taille de la police

Vous pouvez changer la taille de la police d'affichage (moyenne, par défaut). Les tailles disponibles sont : X-small, small, large et X-large. Vous pouvez appliquer le changement à la seule session en cours ou aux sessions suivantes également.

Pour plus de détails sur la procédure, reportez-vous à Setting Preferences (Aide InfoExplorer).

Changement de l'affichage des graphiques

Certains articles hypertexte comportent un schéma. C'est à vous de décider si vous affichez ce graphique dans une fenêtre graphique distincte ou dans la fenêtre de lecture. Pour ce faire, sélectionnez l'une des options **Inline**, **Separate Window** ou **Don't Care** du menu déroulant **Customize**. Si vous sélectionnez l'option **Don't Care**, les schémas s'afficheront dans une fenêtre distincte, sauf s'ils ont été spécifiquement conçus pour la fenêtre de lecture.

Vous pouvez appliquer votre choix à la seule session en cours ou aux sessions suivantes également.

Pour plus de détails sur la procédure, reportez-vous à **Setting Preferences** (Aide InfoExplorer).

Articles rémanents

L'option **Auto-Hold** du menu déroulant **Customize** détermine si InfoExplorer ouvre une nouvelle fenêtre pour lire un nouvel article. Par défaut, InfoExplorer remplace le contenu de la fenêtre de lecture lorsque vous appelez un autre article. Pour que les articles restent ouverts lorsque vous vous déplacez dans la base de données, utilisez l'option **Auto-Hold**. Si vous activez l'option **Auto-Hold** (valeur **On**), vous pouvez suivre les liens sans perdre les articles de départ. Vous pouvez appliquer votre choix à la seule session en cours ou aux sessions suivantes également.

Pour plus de détails sur la procédure, reportez-vous à **Setting Preferences** (Aide InfoExplorer).

Désignation de l'article d'impression

Deux types d'impression sont proposés : simple et sophistiquée (pretty). **Simple Print** ne prend pas en charge les graphiques. **Pretty Print** fait appel au programme **troff**, qui convertit les documents en sorties exploitables par une photocomposeuse.

Pour plus de détails sur la procédure, reportez-vous à **Setting Preferences** (Aide InfoExplorer).

Modification de la taille de l'historique

Un historique peut comporter entre 0 (zéro) et 500 événements : un historique paramétré à 10 événements contient vos 10 derniers liens.

La valeur système par défaut est de 100 événements. Vous pouvez adapter cette taille par défaut à votre situation particulière. Vous pouvez choisir une autre valeur, applicable à la seule session en cours ou également aux sessions suivantes.

Pour plus de détails sur la procédure, reportez-vous à **Setting Preferences** (Aide InfoExplorer).

Choix d'une bibliothèque préférée

Vous pouvez désigner une bibliothèque comme "bibliothèque préférée" : elle sera appelée par InfoExplorer sans qu'il soit besoin de la spécifier sur la ligne de commande.

Pour plus de détails sur la procédure, reportez-vous à **Setting Preferences** (Aide InfoExplorer).

Récapitulatif des commandes de personnalisation

Fichiers de lancement du système

/etc/profile	Fichier système contenant les commandes exécutées par le système lorsque vous vous connectez.
/etc/environment	Fichier système contenant des variables spécifiant l'environnement de base de tous les process.
\$HOME/.profile	Fichier de votre répertoire personnel contenant des commandes primant, lorsque vous vous connectez, sur celles du profil système /etc/profile .
\$HOME/.env	Fichier de votre répertoire personnel contenant des commandes primant sur celles de l'environnement système /etc/environment et contenant des variables spécifiant l'environnement de base de tous les process.

AIXwindows Fichiers de lancement

\$HOME/.xinitrc	Fichier de votre répertoire personnel contrôlant les fenêtres et les applications lancées en même temps qu'AIXwindows.
\$HOME/.Xdefaults	Fichier de votre répertoire personnel contrôlant l'aspect visuel et fonctionnel des ressources AIXwindows.
\$HOME/.mwmrc	Fichier de votre répertoire personnel définissant les affectations des touches et des boutons de la souris, et la définition des menus pour le gestionnaire de fenêtres.

Procédures de personnalisation

PS1	Invite système normale.
PS2	Invite d'entrées complémentaires.
PS3	Invite racine.
chfont	Change la police d'affichage sur un écran (après relance du système).
stty	Définit, redéfinit et fait état des paramètres d'exploitation d'une station de travail.

Chapitre 11. Shells

Les *shells* assurent l'interface avec le système d'exploitation. Ils constituent la couche externe du système d'exploitation. Intégrant un langage de programmation, ils sont à même de contrôler process et fichiers, et de gérer et de contrôler d'autres programmes. Ils vous invitent à entrer des données, les interprètent et traitent les résultats générés par le système.

Ils assurent la communication avec le système d'exploitation. Communication qui s'effectue soit en mode interactif (les entrées au clavier sont immédiatement interprétées), soit via des scripts. Un *script shell* est une séquence de commandes (shell et système), enregistrée dans un fichier.

Lorsque vous vous connectez, le système recherche le programme shell à exécuter, l'exécute et affiche l'invite (généralement le signe \$). Lorsque vous tapez une commande et appuyez sur Entrée, le shell évalue la commande et tente de l'exécuter. Le résultat est ensuite dirigé soit vers l'écran, soit réacheminé, selon vos instructions. Et l'invite est réaffichée, dans l'attente de la commande suivante.

La *ligne de commande*, affichant l'invite shell, est celle sur laquelle vous entrez les commandes. Son format de base est le suivant :

```
$ Commande Argument(s)
```

Le shell interprète le premier mot (jusqu'au premier espace) comme étant le nom de la commande, et les mots suivants comme des arguments.

Ce chapitre traite des points suivants :

- Caractéristiques des shell, page 11-3
- Shell Korn ou POSIX, page 11-9
- Déclaration de caractères, page 11-12
- Mots réservés, page 11-14
- Alias de commande (shell Korn ou POSIX), page 11-15
- Substitution de paramètres (shell Korn ou POSIX), page 11-17
- Substitution de commandes (shell Korn ou POSIX), page 11-23
- Evaluation arithmétique (shell Korn ou POSIX), page 11-24
- Séparation de zones, page 11-25
- Substitution de noms de fichiers (shell Korn ou POSIX), page 11-26
- Réacheminement des entrées/sorties (shell Korn ou POSIX), page 11-28
- Etats de sortie, page 11-31
- Commandes du shell Korn ou POSIX, page 11-32
- Commandes intégrées du shell Korn ou POSIX, page 11-37
- Expressions conditionnelles, page 11-49
- Contrôle des travaux (shell Korn ou POSIX), page 11-51
- Edition en ligne (shell Korn ou POSIX), page 11-53
- Liste des commandes intégrées du shell Korn ou POSIX, page 11-61
- Liste des commandes intégrées du shell Bourne, page 11-63

- Liste des commandes intégrées du shell C, page 11-64
- Voir aussi shell Korn, page 11-67
- Voir aussi shell Bourne, page 11-68
- Voir aussi shell C, page 11-69
- Shell Bourne, page 11-70
- Shell restreint, page 11-72
- Commandes du shell Bourne, page 11-73
- Substitution de variables et de noms de fichiers (shell Bourne), page 11-83
- Réacheminement des entrées/sorties (shell Bourne), page 11-91
- Shell C, page 11-92
- Commandes du shell C, page 0
- Substitution de l'historique (shell C), page 11-105
- Substitution de l'alias (shell C), page 11-108
- Substitution de variables et de noms de fichiers (shell C), page 11-109
- Variables d'environnement (shell C), page 11-113
- Réacheminement des entrées/sorties (shell C), page 11-116
- Contrôle des travaux (shell C), page 11-118

Caractéristiques des shells

Communiquer avec le système via une interface shell présente les avantages suivants :

- **Substitution de caractères joker dans les noms de fichiers (correspondance)**

Exécute des commandes sur un groupe de fichiers, identifiés par des caractères joker (et non sur un fichier explicitement nommé).

Pour en savoir plus, reportez-vous à :

- Substitution de noms de fichiers (shell Korn ou POSIX), page 11-26.
- Substitution de variables et de noms de fichiers (shell Bourne), page 11-84.
- Substitution de nom de fichiers (shell C), page 11-111.

- **Traitement en arrière-plan**

Exécute les tâches lourdes en arrière-plan, libérant le terminal pour des traitements interactifs concurrents.

Pour en savoir plus, reportez-vous à la commande **bg** dans :

- Contrôle des travaux (shell Korn ou POSIX), page 11-51.
- Commandes intégrées (shell C), page 11-94.

Remarque : Le shell Bourne ne prend pas en charge le contrôle des travaux.

- **Alias de commandes**

Attribue un alias à une commande ou à une expression. Lorsqu'un alias figure dans une ligne de commande ou dans un script shell, le système remplace cet alias par le texte qu'il représente.

Pour en savoir plus, reportez-vous à :

- Alias de commandes (shell Korn ou POSIX), page 11-15
- Substitution d'alias (shell C), page 11-108.

Remarque : Le shell Bourne ne prend pas en charge les alias de commandes.

- **Historique des commandes**

Consigne dans un fichier historique les commandes lancées. Vous pouvez utiliser ce fichier pour appeler, modifier ou relancer une commande enregistrée.

Pour en savoir plus, reportez-vous à la commande **history** dans :

- Historique des commandes (shell Korn ou POSIX), page 11-11.
- Commandes intégrées (shell C), page 11-94.
- Substitution d'historique (shell C), page 11-105.

Remarque : Le shell Bourne ne prend pas en charge l'historique des commandes.

- **Substitution de noms de fichiers**

Génère automatiquement une liste de noms de fichiers sur une ligne de commande sur la base de caractères génériques.

Pour en savoir plus, reportez-vous à :

- Substitution de noms de fichiers (shell Korn ou POSIX), page 11-26.
- Substitution de variables et de noms de fichiers (shell Bourne), page 11-84.
- Substitution de nom de fichiers (shell C), page 11-111.

- **Réacheminement des entrées/sorties**

Lit des données à partir d'une source autre que le clavier et réachemine les sorties vers un fichier ou une unité autre que le terminal. Par exemple, les entrées destinées à un programme peuvent être extraites d'un fichier et réacheminées vers l'imprimante ou vers un autre fichier.

Pour en savoir plus, reportez-vous à :

- Réacheminement des entrées/sorties (shell Korn ou POSIX), page 11-28.
- Réacheminement des entrées/sorties (shell Bourne), page 11-91.
- Réacheminement des entrées/sorties (shell C), page 11-116.

- **Chaînage**

Relie des commandes entre elles pour former un programme complexe : la sortie standard de chaque programme constitue l'entrée standard du suivant.

Pour en savoir plus, reportez-vous à la définition de **pipeline** dans "Terminologie", page 11-5.

- **Substitution de variables shell**

Enregistre des données dans des variables définies par l'utilisateur et dans des variables shell prédéfinies.

Pour en savoir plus, reportez-vous à :

- Substitution de paramètres (shell Korn ou POSIX), page 11-17.
- Substitution de variables (shell Bourne), page 11-83.
- Substitution de variables (shell C), page 11-109.

Shells disponibles

Les shells fournis avec cette version du système d'exploitation sont les suivants :

- shell Korn (lancé par la commande **ksh**) ;
- shell Bourne (lancé par la commande **bsh**) ;
- shell restreint (version limitée du shell Bourne, lancé par la commande **Rsh**) ;
- shell POSIX (également appelé shell Korn, lancé par la commande **psh**) ;
- shell par défaut (lancé par la commande **sh**) ;
- shell C (lancé par la commande **csch**) ;
- shell sécurisé (version limitée du shell Korn, lancé par la commande **tsh**) ;
- shell distant (lancé par la commande **bsch**).

Le *shell de connexion*, est lancé lors de la connexion. Votre shell de connexion est défini dans le fichier **/etc/passwd**. Le shell Korn est le shell de connexion standard. Il présente une compatibilité ascendante avec le shell Bourne.

Le *shell par défaut* ou *shell standard* est le shell lié à et lancé par la commande **/usr/bin/sh**. Le shell Bourne est configuré comme shell par défaut et est un sous-ensemble du shell Korn.

Le *shell par défaut* ou *shell standard* est le shell lié à et lancé par la commande **/usr/bin/sh**. Le shell Korn, également appelé shell POSIX, est défini comme shell par défaut. Le shell POSIX est appelé par la commande **/usr/bin/psh** et lié à la commande **/usr/bin/ksh**.

Terminologie

Voici quelques définitions utiles :

blanc	Un des caractères de la classe de blancs définie dans la catégorie LC_CTYPE. Dans le shell POSIX, un blanc est soit une tabulation, soit un espace.	
commande intégrée	Commande que le shell exécute sans la rechercher ni créer de process distinct.	
commande	Séquence de caractères construite suivant la syntaxe du langage shell. Le shell lit chaque commande, puis exécute l'action demandée directement ou par l'intermédiaire d'utilitaires.	
commentaire	Mot préfixé par le signe # : tout caractère ou chaîne de caractères situé entre ce signe et le caractère de ligne suivante est ignoré.	
identificateur	Séquence de lettres, chiffres ou symboles de soulignements introduite par une lettre ou un symbole de soulignement. Le premier caractère d'un identificateur ne peut pas être un chiffre. Les identificateurs servent de noms pour les alias, les fonctions et les paramètres nommés.	
liste	Séquence d'un ou de plusieurs pipelines séparés par l'un des symboles suivants : ; (point-virgule), & (perluète), && (double perluète) ou (double barre verticale). La fin de la liste est indiquée par l'un des symboles suivants : ; (point virgule), & (perluète) ou & (barre verticale, perluète).	
	;	Le shell traite le pipeline précédent en mode séquentiel. Il exécute successivement les commandes jusqu'à la dernière, dont il attend la fin de l'exécution.
	&	Le shell traite le pipeline précédent en mode asynchrone. Il exécute successivement les commandes, traitant le pipeline en arrière-plan sans attendre la fin de son exécution.
	&	Le shell traite le pipeline précédent en mode asynchrone et établit un canal bilatéral vers le shell parent. Il exécute successivement les commandes, traitant le pipeline en arrière-plan sans attendre la fin de son exécution. Les commandes read -p et print -p permettent au shell parent de lire à partir de l'entrée standard et d'écrire vers la sortie standard de la commande générée dynamiquement. Une seule commande peut être activée à la fois.
	&&	Le shell traite la liste située après ce symbole si le résultat du pipeline précédent est la valeur 0 (zéro).
		Le shell traite la liste située après ce symbole si le résultat du pipeline précédent est une valeur différente de 0 (zéro).

Les symboles ; (point virgule), & (perluète) et |& (barre verticale+perluète) ont une priorité moindre que les symboles && (double perluète) et || (double barre verticale). Les symboles ,, & et |& ont la même priorité. De même, les symboles && et || ont la même priorité. Un ou plusieurs caractères ligne suivante peuvent remplacer le symbole point virgule pour délimiter deux commandes dans une liste.

Remarque : Le symbole |& n'est valide que sous le shell Korn.

métacaractère

Un métacaractère est un caractère qui a une signification spéciale pour le shell et marque toujours la fin d'un mot, sauf s'il est placé entre apostrophes. Les symboles considérés comme des métacaractères sont : | (barre verticale), & (perluète), ; (point virgule), < (inférieur à), > (supérieur à), ((parenthèse gauche),) (parenthèse droite), \$ (dollar), ` (apostrophe gauche), \ (barre oblique inverse), ' (apostrophe droit), " (guillemets), caractère de ligne suivante, espace et tabulation. Les caractères encadrés d'apostrophes, dits déclarés, sont interprétés littéralement. Sinon, c'est leur signification spéciale qui est prise en compte. (Dans le shell C, les métacaractères sont aussi appelés *métacaractères d'analyse sémantique*.)

liste d'affectation des paramètres

Liste de mots de la forme *Identificateur=Valeur*. Il doit y avoir le même nombre d'espaces de part et d'autre du signe = (égal) ou aucun espace.

Remarque : Dans le shell C, cette liste est de la forme **set** *Identificateur = Valeur*. Les espaces encadrant le signe = (égal) sont obligatoires.

pipeline

Séquence de commandes séparées par une barre verticale (|). Chaque commande (sauf éventuellement la dernière) est exécutée séparément, mais la sortie standard d'une commande constitue l'entrée standard de la suivante. Une liste de commandes entre parenthèses est exécutée comme simple commande d'un sous-shell distinct.

Si le pipeline n'est pas précédé du mot réservé !, son état de sortie est celui de la dernière commande. Sinon, c'est l'opposé (NON logique) de l'état de cette dernière commande. En d'autres termes, si la dernière commande renvoie zéro, l'état de sortie est 1. Si elle renvoie une valeur supérieure à zéro, l'état de sortie est zéro.

Le format d'un pipeline est le suivant :

```
[!] commande1 [ | commande2 ... ]
```

Remarque : Dans les premières versions du shell Bourne, un canal était symbolisé par un ^ (caret).

variable shell

Nom ou paramètre auquel est affectée une valeur. Pour affecter une valeur, entrez le nom de la variable suivi d'un signe égal (=) et de la valeur souhaitée. Vous pouvez remplacer le nom de la variable par la valeur affectée en préfixant le nom de la variable d'un signe \$ (dollar). Les variables sont particulièrement utiles pour abrégé un nom de chemin trop long : **\$HOME**, par exemple, donne un accès direct au répertoire personnel. Une variable prédéfinie est une variable dont la valeur est attribuée par le shell. Une variable définie par l'utilisateur est une variable dont la valeur est attribuée par l'utilisateur.

commande simple	Séquence de listes d'affectation et de réacheminement de paramètres, dans un ordre quelconque. Elle est éventuellement suivie de commandes, de mots et de réacheminements. Elle est terminée par ;, , &, , &&, & ou un caractère nouvelle ligne. Le nom de la commande est passé comme paramètre 0 (tel que défini par la sous-routine exec). La valeur d'une commande simple est son état en sortie (zéro si elle a abouti, différent de zéro, sinon). Reportez-vous à la sous-routine sigaction , sigvec ou signal dans le manuel <i>AIX Technical Reference, Volume 2: Base Operating System and Extensions</i> pour la liste des valeurs d'état en sortie.
sous-shell	Shell exécuté comme enfant du shell de connexion ou du shell courant.
caractère joker	(ou caractère générique) Symbole spécial affecté d'une valeur par le shell. Les caractères de base sont : ?, *, [ensemble], et ![ensemble]. Ils sont particulièrement utiles pour effectuer des substitutions de noms de fichiers.
mot	Séquence de caractères exempte de blancs. Les mots sont séparés par un ou plusieurs métacaractères.

Création et exécution d'un script shell

Les scripts shell sont des outils qui permettent de prendre en charge des commandes, des séquences de commandes ou des routines particulièrement complexes ou lourdes. Un script shell est un fichier contenant une ou plusieurs commandes. Pour exécuter la séquence de commandes, il suffit de taper le nom du fichier.

Vous créez un script shell à l'aide d'un éditeur de texte. Un script peut regrouper des commandes système ou des commandes shell intégrées.

Pour écrire un script shell, procédez comme suit.

1. Via un éditeur de texte, créez et sauvegardez un fichier. Vous pouvez y inscrire n'importe quelle combinaison de commandes système et de commandes shell.

Par convention, les scripts shell non destinés à être exploités par plusieurs utilisateurs sont enregistrés dans le répertoire **\$HOME/bin**.

Remarque : Le système d'exploitation ne prend pas en charge les sous-routines **setuid** et **setgid** dans un script shell.

1. Lancez la commande **chmod** pour limiter au seul propriétaire le droit d'exécuter le fichier. Par exemple, si le fichier s'appelle `script1`, entrez :

```
chmod u=rwx script1
```

2. Pour exécuter le script, entrez son nom sur la ligne de commande. Pour exécuter le script `script1`, entrez :

```
script1
```

Remarque : Vous pouvez exécuter un script shell sans l'avoir rendu exécutable, sous réserve que son nom soit précédé d'une commande shell (**ksh**, **bsh** ou **cs**) sur la ligne de commande. Par exemple, pour exécuter le fichier non-exécutable `script1` sous le shell Korn, entrez :

```
ksh script1
```

Spécification d'un shell pour un fichier script

Lorsque vous exécutez un script shell sous le shell Korn (shell POSIX) ou le shell Bourne, les commandes sont exécutées sous contrôle du shell courant – sauf spécification contraire explicite. Lorsque vous exécutez un script shell sous le shell C, les commandes sont exécutées sous contrôle du shell Bourne (**/usr/bin/bsh**) sauf si vous spécifiez un autre shell.

Pour spécifier un autre shell, vous devez l'inclure dans le script shell. Pour lancer un script shell exécutable sous un shell spécifique, entrez `#!Chemin` sur la première ligne de commande du script. Les caractères `#!` identifient le type de fichier. La variable *Chemin*, indique le chemin du shell sous lequel le script est exécuté. Par exemple, si la première ligne d'un script shell est `#!/usr/bin/bsh`, le script s'exécute sous le contrôle du shell Bourne.

Si vous précédez d'une commande shell le nom d'un fichier script (sur la ligne de commande), le shell ainsi spécifié prime sur celui éventuellement indiqué dans le script lui-même. Ainsi, la commande `ksh myfile` exécute le fichier **myfile** sous le shell Korn, même si la première ligne de **myfile** est `#!/usr/bin/csh`.

Voir aussi

Commandes : généralités

Process : généralités

Systèmes de fichiers :

Répertoires : généralités

Fichiers : généralités

Sécurité du système et des fichiers

Shell Korn ou POSIX

Shell Bourne

Shell C

Shell Korn ou POSIX

Le shell Korn est un interpréteur de commandes interactif et un langage de programmation. Il est conforme à l'environnement POSIX (Portable Operating System Interface for Computer Environments), standard international pour les systèmes d'exploitation. POSIX n'est pas un système d'exploitation, mais un *standard* ayant pour objet la portabilité des applications, au niveau source, entre plusieurs systèmes. Les caractéristiques POSIX sont construites au niveau supérieur du shell Korn. Le shell Korn (également appelé shell POSIX) offre, outre nombre des fonctions des shells Bourne et C (réacheminement des E/S, substitution de variables et de noms de fichiers, etc.), un certain nombre de fonctions spécifiques. En outre, il inclut un certain nombre de commandes supplémentaires et de caractéristiques de langages de programmation :

évaluation arithmétique	La commande intégrée let du shell Korn exécute des opérations arithmétiques sur des nombres entiers, sur n'importe quelle base entre 2 et 36. Pour en savoir plus, reportez-vous à Evaluation arithmétique, page 11-24.
Historique des commandes	Le shell Korn, ou POSIX, contient un fichier qui enregistre toutes les commandes entrées. Vous pouvez modifier n'importe quelle commande de ce fichier à l'aide d'un éditeur de texte, puis la réutiliser. Pour en savoir plus, reportez-vous à "Historique des commandes", page 11-11.
Fonction coprocess	La fonction coprocess permet d'exécuter des process en arrière-plan, de leur envoyer des informations et d'en recevoir.
Edition	Les options d'édition permettent de modifier la ligne de commande. Editeurs disponibles : emacs, gmacs et vi. "Pour en savoir plus, reportez-vous à "Edition en ligne", page 11-53.

Cette section traite des points suivants :

- Environnement du shell Korn, page 11-10.
- Lancement du shell, page 11-10.
- Historique des commandes (shell Korn ou POSIX), page 11-11.
- Déclaration de caractères, page 11-12.
- Mots réservés, page 11-14.
- Alias de commandes (shell Korn ou POSIX), page 11-15
- Substitution de paramètres (shell Korn ou POSIX) on page 11-17.
- Substitution de commandes (shell Korn ou POSIX) on page 11-23.
- Evaluation arithmétique (shell Korn ou POSIX), page 11-24.
- Séparation de zones, page 11-25.
- Substitution de noms de fichiers (shell Korn ou POSIX), page 11-26.
- Réacheminement des entrées/sorties (shell Korn ou POSIX), page 11-28.
- Etats de sortie, page 11-31
- Commandes du shell Korn ou POSIX, page 11-32.
- Liste des commandes intégrées du shell Korn ou POSIX, page 11-37.
- Expressions conditionnelles, page 11-49.
- Contrôle des travaux (shell Korn ou POSIX), page 11-51.

- Edition en ligne (shell Korn ou POSIX), page 11-53.
- Voir aussi Shell Korn

Environnement shell Korn

Toutes les variables (avec leurs valeurs) connues d'une commande lors de son lancement constituent son *environnement* : variables héritées du process parent et variables spécifiées comme paramètres-clés sur la ligne de commande. Les interactions entre le shell et l'environnement sont multiples. Lorsqu'il est lancé, le shell balaye l'environnement et crée un paramètre chaque fois qu'il trouve un nom en attribuant à ce paramètre la valeur correspondante et en le marquant pour exportation. Les commandes exécutées héritent de l'environnement.

Si vous modifiez les valeurs des paramètres shell ou si vous en créez de nouvelles à l'aide des commandes **export** ou **typeset -x**, ces paramètres sont intégrés à l'environnement. Pour une commande exécutée, l'environnement se présente donc comme suit : paires nom-valeur héritées par le shell (valeurs éventuellement modifiées par le shell courant), plus les paires résultant de la commande **export** ou **typeset -x**. La commande exécutée (sous-shell) "voit" les modifications apportées aux variables d'environnement héritées, mais doit exporter ces variables pour que ses shells et process enfants les voient.

Pour modifier l'environnement d'une fonction ou d'une commande simple, il suffit de la préfixer par une ou plusieurs affectations de paramètres, de la forme *Identificateur=Valeur*. Les deux expressions suivantes sont donc équivalentes (en ce qui concerne l'exécution de la commande):

```
TERM=450 arguments commande
(export TERM; TERM=450; arguments commande)
```

Si l'indicateur **-k** est spécifié, *tous* les arguments d'affectation de paramètres sont intégrés à l'environnement, même placés après le nom de commande. Dans l'exemple suivant, les premiers affichent **a=b c** suivis de **c** :

```
echo a=b c
set -k
echo a=b c
```

Remarque : Cette fonction est conçue pour les scripts des premières versions du shell. Elle est fortement déconseillée dans les nouveaux scripts.

Lancement du shell

Vous pouvez lancer le shell Korn via la commande **ksh** ou la commande **psh** (shell POSIX).

Si vous le lancez par la commande **exec**, et que le premier caractère de l'argument zéro (**\$0**) est **-** (signe moins), le shell est supposé être un shell de connexion. Le shell commence par lire les commandes du fichier **/etc/profile**, puis celles du fichier **.profile** du répertoire courant ou du fichier **\$HOME/.profile** (si les fichiers existent). Ensuite, il lit les commandes du fichier (s'il existe) nommé lors de la substitution de paramètre sur la valeur de la variable d'environnement **ENV**.

Si vous spécifiez le paramètre *Fichier [Paramètre]* en appelant le shell Korn ou POSIX, celui-ci exécute le fichier script identifié par *Fichier*, avec tous ses paramètres. Le fichier script doit être accessible en lecture. Les éventuels paramètres **setuid** et **setgid** sont ignorés. Le shell lit ensuite les commandes.

Remarque : Ne spécifiez pas de fichier script avec les indicateurs **-c** ou **-s**.

Pour en savoir plus sur les paramètres positionnels, reportez-vous à "Substitution de paramètres (shell Korn ou POSIX)", page 11-17.

Invite shell

En mode interactif, le shell affiche l'invite définie par le paramètre **PS1**, avant de lire une commande. Puis à chaque caractère ligne suivante entré, le shell affiche la deuxième invite (définie par le paramètre **PS2**) pour vous demander de compléter la commande.

Historique des commandes (shell Korn ou POSIX)

Le shell Korn ou POSIX enregistre les commandes entrées dans un fichier d'historique, dont le nom est défini par la variable **HISTFILE**. Si cette variable n'est ou ne peut être définie, le fichier **\$HOME/.sh_history** est utilisé par défaut. Si le fichier d'historique n'existe pas et que le shell Korn ne dispose pas des droits nécessaires pour le joindre, le shell utilise un fichier temporaire comme fichier d'historique. Le shell accède aux commandes de tous les shells interactifs via le même fichier d'historique, doté des droits d'accès adhoc.

Par défaut, le shell Korn ou POSIX sauvegarde le texte des 128 dernières commandes. La taille du fichier historique (variable **HISTSIZE**) n'est pas limitée, mais un fichier trop volumineux risque de ralentir le démarrage du shell Korn.

Substitution de l'historique des commandes

La commande intégrée **fc** permet d'afficher et de modifier tout ou partie du fichier d'historique. Le cas échéant, précisez le numéro ou l'intervalle de numéros (ou encore les premiers caractères) des commandes qui vous intéressent. Vous pouvez spécifier une ou plusieurs commandes.

Si vous ne précisez pas de programme d'édition en argument de la commande **fc**, c'est l'éditeur spécifié par la variable **FCEDIT** qui est utilisé. En cas de non définition de la variable **FCEDIT**, le fichier **/usr/bin/ed** est utilisé. Les commandes modifiées sont affichées et exécutées dès que vous quittez l'éditeur.

Le nom d'éditeur – (tiret) permet de sauter la phase d'édition et de réexécuter directement la commande. Vous pouvez, dans ce cas, passer par un paramètre de substitution de la forme *Ancien=Nouveau* pour modifier la commande avant de l'exécuter. Par exemple, si `r a` pour alias `fc -e -`, tapez `r bad=good c` : la commande la plus récente commençant par la lettre `c` est exécutée et remplace la première occurrence de la chaîne `bad` par la chaîne `good`.

Pour en savoir plus sur la commande d'historique, reportez-vous à "Affichage des commandes antérieures (commande shell history)", page 3-8 et à la commande **fc** dans le manuel *AIX Commands Reference*.

Déclaration de caractères

Si vous voulez que le shell Korn ou POSIX interprète un caractère comme étant un caractère normal (c'est-à-dire dépourvu de la signification qui lui est normalement associée), vous devez le déclarer. S'il s'agit d'un métacaractère, utilisez un des caractères de déclaration de la liste ci-après.

Un métacaractère est un caractère qui a une signification spéciale pour le shell : si vous omettez de le déclarer, le shell l'interprétera et le mot sera tronqué. Voici les métacaractères du shell Korn ou POSIX (à déclarer, le cas échéant) :

- | (barre verticale)
- & (perluète)
- ; (point virgule)
- <(inférieur à) et > (supérieur à)
- ((parenthèse gauche) et) (parenthèse droite)
- \$ (dollar)
- ` (apostrophe inverse) et ' (apostrophe)
- \ (barre oblique inverse)
- " (guillemets)
- caractère de nouvelle ligne
- espace
- tabulation.

Les caractères de déclaration sont : \ (barre oblique inverse), apostrophes et guillemets.

\ (barre oblique inverse)

Une barre oblique inverse (\) non déclarée préserve la valeur littérale du caractère suivant (sauf s'il s'agit du caractère nouvelle ligne). S'il s'agit du caractère nouvelle ligne, le shell l'interprète comme une continuation de ligne.

Apostrophes

Mettre des caractères entre apostrophes (' ') préserve leur valeur littérale. Ne placez pas d'apostrophe dans la chaîne encadrée.

Une barre oblique inverse ne permet pas d'insérer une apostrophe dans une chaîne de ce type : pour imbriquer des apostrophes, écrivez, par exemple : ' a \ ' ' b ' , pour obtenir a'b.

Guillemets

Mettre des caractères entre guillemets (" ") préserve leur valeur littérale – sauf pour les caractères \$ (dollar), ' (apostrophe inverse) et barre oblique inverse).

\$ Le signe dollar conserve sa fonction : développement de paramètres (forme de substitution de commandes) et développement arithmétique.

Les caractères entre guillemets se trouvant également entre \$(et le symbole) correspondant ne sont pas affectés par les guillemets, mais définissent la commande dont la sortie remplace \$ (...) lorsque le mot est développé.

A l'intérieur d'une chaîne délimitée par \$ { et }, le nombre d'apostrophes ou de guillemets (le cas échéant) doit être pair. Pour déclarer un littéral { ou }, utilisez une barre oblique inverse.

` L'apostrophe inverse conserve sa fonction : introduire l'autre forme de substitution de commande. La partie de la chaîne comprise entre la première apostrophe et la suivante (non précédée d'une barre oblique inverse) définit la commande dont la sortie remplace ` ... ` lorsque le mot est développé.

**** La barre oblique inverse conserve sa fonction d'échappement si elle est suivie de l'un des caractères suivants : \$, `, ", \ ou nouvelle ligne.

Pour insérer un guillemet dans une chaîne entre guillemets, faites-le précéder d'une barre oblique inverse. Avec des guillemets, si une barre oblique inverse est immédiatement suivie d'un caractère spécial, la barre est supprimée et le caractère suivant, interprété littéralement. Si la barre oblique inverse précède un caractère normal, elle demeure à sa place, et le caractère suivant est inchangé. Par exemple :

```
"\ $"    ->    $  
"\a"     ->    \a
```

Voici les règles applicables aux métacaractères et aux caractères de déclaration :

- La signification de \$* (dollar, astérisque) et de \$@: (dollar, arobas) est la même lorsqu'ils ne sont pas déclarés, qu'ils soient utilisés comme valeur d'affectation de paramètre ou comme nom de fichier.
- Utilisé comme argument de commande, "\$*" (guillemets, signe dollar, astérisque, guillemets) équivaut à "\$1 d\$2d...", d étant le premier caractère du paramètre IFS.
- "\$@" (guillemets doubles, dollar, arobas, guillemets doubles) équivaut à "\$1" "\$2"
- Entre apostrophes inverses (` `), le caractère barre oblique inverse déclare les caractères (barre oblique inverse), ' (apostrophe) et \$ (dollar). Si les apostrophes inverses se trouvent entre guillemets (" "), la double barre inverse déclare également le caractère guillemets.
- Les substitutions de commandes et de paramètres ne sont pas affectées par les guillemets ("").
- Les noms des fonctions et des commandes intégrées ne peuvent être déclarés. Pour annuler la signification spéciale des alias et des mots réservés, déclarez-en chaque caractère.

Mots réservés

Voici la liste des mots réservés du shell :

```
!          case    do
done       elif    else
esac      fi      for
function  if      in
select    then    time
until     while   {
}         [[     ]]
```

Un mot réservé n'est identifié que lorsqu'il n'est pas déclaré et qu'il constitue :

- le premier mot d'une commande ;
- le premier mot suivant un mot réservé autre que **case**, **for** ou **in** ;
- le troisième mot d'une commande **case** ou **for** (**in** est le seul mot possible).
- Voir aussi Shell Korn

Alias de commandes (shell Korn ou POSIX)

Le shell Korn, ou POSIX, permet de personnaliser les commandes en créant des alias. La commande **alias** définit comme alias un mot de la forme `Nom=Chaîne`. Lorsque vous utilisez un alias comme premier mot d'une ligne de commande, **ksh** vérifie s'il traite déjà un alias de même nom. Dans l'affirmative, il ne remplace pas le nom de l'alias. Dans la négative, il remplace le nom de l'alias par sa valeur.

Le premier caractère d'un nom d'alias peut être n'importe quel caractère imprimable à l'exclusion des caractères spéciaux. Les autres caractères sont les mêmes que ceux utilisés pour un identificateur. La chaîne de remplacement peut contenir n'importe quel texte shell correct, métacaractères compris.

Si le dernier caractère d'une valeur alias est un blanc, le shell vérifie aussi le mot qui suit l'alias. Les alias peuvent servir à redéfinir les commandes intégrées spéciales, mais pas les mots réservés. Les définitions d'alias ne sont pas conservées d'un appel de **ksh** à l'autre. Toutefois, si vous spécifiez **alias -x**, l'alias reste en vigueur pour les scripts appelés par leur nom, qui n'appellent pas de scripts distincts. Pour exporter une définition d'alias et en offrir l'accès aux process enfants, spécifiez **alias -x** et la définition de l'alias dans le fichier d'environnement.

Pour créer, lister et exporter des alias, lancez la commande **alias**. La commande **unalias** permet de les supprimer.

Format de création d'un alias :

```
alias Nom=Chaîne
```

où *Nom* est le nom de l'alias et *Chaîne*, sa valeur.

Les alias exportés suivants sont prédéfinis par le shell Korn, mais vous pouvez les inhiber ou les redéfinir. Nous vous conseillons toutefois de ne pas les modifier : vous risqueriez de perturber un tiers, habitué au fonctionnement des alias prédéfinis.

```
autoload='typeset -fu'  
false='let 0'  
functions='typeset -f'  
hash='alias -t'  
history='fc -l'  
integer='typeset -i'  
nohup='nohup '  
r='fc -e -'  
true=':'  
type='whence -v'
```

Les alias ne sont pas pris en charge par les appels non interactifs du shell Korn (**ksh**) (dans un script shell ou, comme dans l'exemple suivant; avec l'option **-c** associée à **ksh**).

```
ksh -c alias
```

Pour en savoir plus, reportez-vous à Création d'un alias de commande, page 3-10 et à la commande **alias** dans le manuel *AIX Commands Reference*.

Alias de trace

Les alias sont souvent utilisés pour abrégier les noms de chemins absolus. La fonction de création d'alias permet d'associer automatiquement la valeur d'un alias au chemin absolu de la commande correspondante. Ce type d'alias s'appelle alias de *trace*. Ce type d'alias accélère l'exécution des commandes, le shell n'ayant plus à rechercher dans la variable **PATH** le chemin absolu.

La commande **set -h** active le processus de *traçage*, de sorte que chaque fois qu'une commande est référencée, le shell définit un alias de trace. La valeur de cet alias devient indéfinie dès que vous réinitialisez la variable **PATH**.

Ces alias demeurent tracés, jusqu'à ce qu'une référence ultérieure les redéfinisse. Plusieurs alias de trace sont compilés dans le shell.

Substitution de tilde

Après exécution des substitutions d'alias, le shell examine les mots à la recherche de ceux qui commencent par un tilde (~) non déclaré. S'il en trouve, il vérifie le mot, jusqu'à la première barre oblique (/), pour voir s'il correspond à un nom utilisateur du fichier **/etc/passwd**. Si oui, il remplace le caractère ~ et le nom par le répertoire de connexion de l'utilisateur correspondant. Ce processus est appelé *substitution de tilde*.

S'il ne trouve pas de correspondance, le shell ne modifie pas le texte original. Le shell Korn effectue aussi des remplacements dans le cas particulier où le caractère ~ est le seul du mot ou est suivi du signe + (plus) ou du signe - (moins).

~	Remplacé par la variable HOME .
~+	Remplacé par la variable \$PWD (chemin absolu du répertoire précédent).
~-	Remplacé par la variable \$OLDPWD (chemin absolu du répertoire précédent).

En outre, le shell essaie d'effectuer une substitution de tilde lorsque la valeur d'un paramètre d'affectation de variable commence par un caractère ~.

- Voir aussi Shell Korn

Substitution de paramètres (shell Korn ou POSIX)

Le shell Korn, ou POSIX, autorise les substitutions de paramètres.

Cette section traite des points suivants :

- Paramètres (shell Korn), page 11-17.
- Substitution de paramètres, page 11-18.
- Paramètres spéciaux prédéfinis, page 11-19.
- Variables définies par le shell Korn ou POSIX, page 11-20
- Variables utilisées par le shell Korn ou POSIX, page 11-20.
- Voir aussi Shell Korn, page 11-67

Paramètres (shell Korn)

Il existe plusieurs catégories de paramètres :

- Identificateur constitué d'une combinaison des caractères* (astérisque), : (arobas), # (dièse), ? (point d'interrogation), – (tiret), \$ (dollar) et ! (point d'exclamation). Il s'agit des *paramètres spéciaux*.
- Argument désigné par un chiffre (*paramètre positionnel*).
- Paramètre désigné par un identificateur, doté d'une valeur et, éventuellement, d'attributs (*paramètres/variables nommés*).

La commande intégrée spéciale **typeset** affecte valeurs et attributs aux paramètres nommés. Vous trouverez la description des attributs admis par le shell Korn dans la section qui traite de la commande **typeset**. Les paramètres exportés transmettent valeurs et attributs à l'environnement.

Le format d'affectation d'une valeur est le suivant :

```
Nom=Valeur [ Nom=Valeur ] ...
```

Si l'attribut entier **-i** est associé au paramètre *Nom*, le paramètre *Valeur* fait l'objet d'une évaluation arithmétique. Pour en savoir plus, reportez-vous à "Évaluation arithmétique (shell Korn ou POSIX)", page 11-24.

Le shell offre une fonction de tableau à une dimension. Chaque élément du tableau est référencé par un indice, désigné par une expression arithmétique entre crochets([]). Pour affecter des valeurs à un tableau, utilisez la commande `set -A Nom Valeur ...`. Les indices doivent être compris entre 0 et 511. Les tableaux n'ont pas besoin d'être déclarés. Toute référence à un paramètre doté d'un indice valide est autorisée et permet, le cas échéant, la création d'un tableau. Référencer un tableau sans indice équivaut à référencer l'élément zéro.

La commande spéciale **set** sert à affecter des valeurs aux paramètres positionnels. A l'appel du shell, le paramètre **\$0** est positionné à partir de l'argument zéro. Le caractère \$ permet d'introduire des paramètres remplaçables.

Substitution de paramètres

Voici la liste des paramètres remplaçables :

<code>\${Paramètre}</code>	<p>Le shell lit tous les caractères à partir du signe \$ (signe dollar, accolade gauche) jusqu'au signe } (accolade droite) correspondant faisant partie d'un même mot, même si celui-ci contient des accolades ou des métacaractères. La valeur éventuelle du paramètre spécifié est remplacée. Les accolades sont obligatoires lorsque <i>Paramètre</i> est suivi d'une lettre, d'un chiffre ou d'un soulignement qui ne font pas partie du nom ou pour un paramètre nommé indexé.</p> <p>Un paramètre spécifié avec un ou plusieurs chiffres est un <i>paramètre positionnel</i>. Tout paramètre positionnel de plus d'un chiffre doit être mis entre accolades. Lorsque la valeur de la variable est un * (astérisque) ou un @ (arobas), chaque paramètre positionnel commençant par \$1, est remplacé (séparé par un caractère séparateur de zone). Lorsqu'un identificateur de tableau doté d'un indice * (astérisque) ou d'un @ (arobas) est utilisé, la valeur de chacun des éléments (séparé par un caractère séparateur de zone) est remplacé.</p>
<code>\${#Paramètre}</code>	<p>Lorsque la valeur de <i>Paramètre</i> est un * ou un @, le nombre de paramètres positionnels fait l'objet d'une substitution. Sinon, la longueur spécifiée par <i>Paramètre</i> est remplacée.</p>
<code>\${#Identificateur[*]}</code>	<p>Le nombre d'éléments du tableau spécifié par le paramètre <i>Identificateur</i> est remplacé.</p>
<code>\${Paramètre:-Mot}</code>	<p>Si <i>Paramètre</i> est défini et non nul, sa valeur est remplacée ; sinon, la substitution porte sur <i>Mot</i>.</p>
<code>\${Paramètre:=Mot}</code>	<p>Si <i>Paramètre</i> est défini et non nul, sa valeur est remplacée ; sinon, la substitution porte sur <i>Mot</i>. Les paramètres positionnels ne peuvent être affectés de cette façon.</p>
<code>\${Paramètre:?Mot}</code>	<p>Si <i>Paramètre</i> est défini et non nul, sa valeur est remplacée. Sinon, la valeur de <i>Mot</i> s'affiche et vous quittez le shell. Si vous avez omis de définir <i>Mot</i>, un message standard s'affiche.</p>
<code>\${Paramètre:+Mot}</code>	<p>Si <i>Paramètre</i> est défini et non nul, sa valeur est remplacée. Sinon, aucune substitution n'est effectuée.</p>
<code>\${Paramètre#Trame} \${Paramètre##Trame}</code>	<p>Si la <i>Trame</i> shell correspond aux premiers caractères de <i>Paramètre</i>, la valeur de remplacement est celle de <i>Paramètre</i> après suppression de la partie correspondante. Sinon, la valeur de <i>Paramètre</i> est remplacée. Dans le premier format, c'est la plus petite correspondance de caractères génériques qui est éliminée. Dans le second format, c'est la plus longue correspondance de caractères génériques qui est éliminée.</p>
<code>\${Paramètre%Trame} \${Paramètre%%Trame}</code>	<p>Si la <i>Trame</i> shell correspond aux premiers caractères de <i>Paramètre</i>, la valeur de remplacement est celle de <i>Paramètre</i> après suppression de la partie correspondante. Sinon, la valeur de <i>Paramètre</i> est remplacée. Dans le premier format, c'est la plus petite correspondance de caractères génériques qui est éliminée, dans le deuxième format, c'est la plus longue.</p> <p>Dans les expressions précédentes, la variable <i>Mot</i> n'est évaluée que si elle doit être utilisée comme chaîne remplacée. Dans l'exemple suivant, la commande pwd ne sera donc exécutée que si l'indicateur -d est nul ou non défini :</p> <pre>echo \${d:-\$(pwd)}</pre>

Remarque : Si le signe : (deux points) est omis, le shell vérifie uniquement si *Paramètre* est défini.

Paramètres spéciaux prédéfinis

Les paramètres automatiquement définis par le shell sont les suivants :

- @** Développe les paramètres positionnels, en commençant par **\$1**. Les paramètres sont séparés par des espaces.
Si vous encadrez **\$@** de guillemets ("), le shell interprète chaque paramètre positionnel comme une chaîne distincte. En l'absence de paramètre positionnel, le shell développe l'instruction en chaîne nulle non déclarée.
- *** Développe les paramètres positionnels, en commençant par **\$1**. Le shell sépare chaque paramètre par le premier caractère du paramètre **IFS**.
Si vous encadrez **\$*** de guillemets ("), le shell encadre de guillemets les valeurs du paramètre positionnel. Les valeurs sont séparées par le premier caractère du paramètre **IFS**.
- #** Spécifie le nombre (décimal) de paramètres positionnels transmis au shell, nom de la procédure shell elle-même exclu. Le paramètre **\$#** génère donc le paramètre positionnel ayant la valeur la plus élevée. Ce paramètre sert principalement à vérifier que le nombre d'arguments requis est présent.
- (tiret)** Fournit les indicateurs pour le lancement du shell ou de la commande **set**.
- ?** Valeur de sortie de la dernière commande exécutée (chaîne décimale). La plupart des commandes renvoient 0 pour indiquer qu'elles ont été correctement exécutées. Le shell renvoie la valeur courante de la variable **\$?**.
- \$** Numéro de process du shell. Un numéro de process étant unique, cette chaîne (de 5 chiffres au maximum) est souvent utilisée pour générer des noms uniques pour les fichiers temporaires.
L'exemple suivant illustre comment créer des fichiers temporaires dans un répertoire réservé à cet effet :
- ```
temp=$HOME/temp/$$
ls >$temp
.
.
.
rm $temp
```
- !** Numéro de process de la dernière commande d'arrière-plan invoquée.
- 0 (zéro)** Développe le nom du shell ou du script shell.

## Variables définies par le shell Korn ou POSIX

Les variables définies par le shell sont les suivantes :

|                     |                                                                                                                                                                                                                                                                                                                                                                        |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>_ (souligné)</b> | Indique initialement le nom de chemin du shell ou du script en cours d'exécution tel que transmis à l'environnement. Le dernier argument de la commande précédente lui est ensuite affecté. Ce paramètre n'est pas défini pour les commandes asynchrones. Il sert également à bloquer le nom du fichier <b>MAIL</b> correspondant pendant la vérification du courrier. |
| <b>ERRNO</b>        | Valeur définie par la dernière sous-routine non aboutie. Cette valeur, dépendante du système, sert à la mise au point.                                                                                                                                                                                                                                                 |
| <b>LINENO</b>       | Numéro de la ligne courante dans le script ou dans la fonction en cours d'exécution.                                                                                                                                                                                                                                                                                   |
| <b>OLDPWD</b>       | Répertoire de travail précédemment défini par la commande <b>cd</b> .                                                                                                                                                                                                                                                                                                  |
| <b>OPTARG</b>       | Valeur du dernier argument de l'option traitée par la commande intégrée standard <b>getopts</b> .                                                                                                                                                                                                                                                                      |
| <b>OPTIND</b>       | Valeur du dernier indice de l'option traitée par la commande intégrée standard <b>getopts</b> .                                                                                                                                                                                                                                                                        |
| <b>PPID</b>         | Numéro de process du shell parent.                                                                                                                                                                                                                                                                                                                                     |
| <b>PWD</b>          | Répertoire de travail courant défini par la commande <b>cd</b> .                                                                                                                                                                                                                                                                                                       |
| <b>RANDOM</b>       | Génère un nombre entier aléatoire compris entre 0 et 32767. La séquence des nombres aléatoires peut être initialisée en affectant une valeur numérique à la variable <b>RANDOM</b> .                                                                                                                                                                                   |
| <b>REPLY</b>        | Défini par l'instruction <b>select</b> et la commande intégrée standard <b>read</b> en l'absence d'arguments.                                                                                                                                                                                                                                                          |
| <b>SECONDS</b>      | Nombre de secondes écoulées depuis le renvoi de l'appel du shell. Lorsque cette variable est renseignée, la valeur renvoyée est la valeur affectée incrémentée du nombre de secondes écoulées depuis l'affectation.                                                                                                                                                    |

## Variables utilisées par le shell Korn ou POSIX

Les variables suivantes sont utilisées par le shell :

|                 |                                                                                                                                                                                                                                          |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CDPATH</b>   | Chemin d'accès à la commande <b>cd</b> (changement de répertoire).                                                                                                                                                                       |
| <b>COLUMNS</b>  | Largeur de la fenêtre d'édition, pour les modes d'édition du shell et l'impression des listes <b>select</b> .                                                                                                                            |
| <b>EDITOR</b>   | Si ce paramètre se termine par <b>emacs</b> , <b>gmacs</b> , ou <b>vi</b> , et que la variable <b>VISUAL</b> n'est pas définie par la commande <b>set</b> , l'option correspondante est activée.                                         |
| <b>ENV</b>      | Si cette variable est renseignée, une substitution de paramètre est exécutée sur cette valeur pour créer le nom de chemin du script exécuté à l'appel du shell. Ce fichier sert en général à définir les alias et les fonctions.         |
| <b>FCEDIT</b>   | Nom de l'éditeur par défaut pour la commande <b>fc</b> .                                                                                                                                                                                 |
| <b>FPATH</b>    | Chemin d'accès aux définitions de fonction. Il est exploré pour les fonctions assorties de l'indicateur <b>-u</b> et les commandes introuvables. Si un fichier exécutable est trouvé, il est lu et exécuté dans l'environnement courant. |
| <b>HISTFILE</b> | Si défini à l'appel du shell, nom du chemin d'accès au fichier enregistrant l'historique des commandes.                                                                                                                                  |

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>HISTSIZE</b>    | Si défini à l'appel du shell, nombre minimal (128, par défaut) de commandes précédemment entrées et accessibles par ce shell. La valeur par défaut est 128.                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>HOME</b>        | Répertoire de connexion, qui est le répertoire courant dès que la connexion est établie. Cette variable est initialisée par le programme <b>login</b> . La commande <b>cd</b> utilise par défaut la valeur du paramètre <b>\$HOME</b> . Utiliser cette variable plutôt qu'un nom de chemin explicite dans une procédure shell permet d'exécuter la procédure à partir d'un répertoire différent sans modification.                                                                                                                                 |
| <b>IFS</b>         | Séparateurs de zone internes (en général, espace, tabulation et nouvelle ligne), qui servent à séparer les mots d'une commande résultant d'une substitution de paramètre ou de commande, ou issus d'une commande <b>read</b> . Le premier caractère du paramètre <b>IFS</b> sépare les arguments pour la substitution <b>\$*</b> .                                                                                                                                                                                                                 |
| <b>LANG</b>        | Valeur par défaut des variables <b>LC_*</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>LC_ALL</b>      | Prime sur les valeurs des variables <b>LANG</b> et <b>LC_*</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>LC_COLLATE</b>  | Détermine la réaction d'une expression de type intervalle dans le cadre d'une recherche générique.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>LC_CTYPE</b>    | Définit la classification des caractères, la conversion majuscules/minuscules et autres attributs de caractères.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>LC_MESSAGES</b> | Langue d'affichage des messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>LINES</b>       | Longueur des colonnes pour l'affichage des listes de sélection. Les listes de sélection s'affichent verticalement jusqu'au deux tiers environ des lignes spécifiées par le paramètre <b>LINES</b> .                                                                                                                                                                                                                                                                                                                                                |
| <b>MAIL</b>        | Chemin d'accès au fichier utilisé par la messagerie pour détecter l'arrivée de courrier. Si vous indiquez le nom d'un fichier courrier et que le paramètre <b>MAILPATH</b> n'est pas défini, le shell informe l'utilisateur que du courrier est arrivé dans le fichier spécifié.                                                                                                                                                                                                                                                                   |
| <b>MAILCHECK</b>   | Fréquence (en secondes) des contrôles effectués par le shell sur d'éventuels changements apportés au délai de modification des fichiers (défini via les variables <b>MAILPATH</b> ou <b>MAIL</b> ). La valeur par défaut est 600 secondes. Lorsque le délai est écoulé, le shell effectue un contrôle avant l'affichage de l'invite suivante.                                                                                                                                                                                                      |
| <b>MAILPATH</b>    | Liste de noms de fichiers, séparés par des signes deux points. Si cette variable est renseignée, le shell informe l'utilisateur de toute modification des fichiers spécifiés, survenue au cours de l'intervalle, en secondes, spécifié par <b>MAILCHECK</b> . Chaque nom de fichier peut être suivi d'un <b>?</b> (point d'interrogation) et d'un message. Le message fera l'objet d'une substitution de paramètre avec la variable <b>\$_</b> définie comme nom de fichier modifié. Le message par défaut est <code>you have mail in \$_</code> . |
| <b>NLSPATH</b>     | Emplacement des catalogues de messages utilisés pour le traitement de <b>LC_MESSAGES</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>PATH</b>        | Chemin d'accès aux commandes, sous forme de liste de chemins à des répertoires, séparés par des deux-points. Lorsqu'il recherche une commande, le shell explore ces répertoires dans l'ordre spécifié. Le répertoire courant est représenté par une chaîne nulle.                                                                                                                                                                                                                                                                                  |
| <b>PS1</b>         | Chaîne constituant l'invite système principale ( <b>\$</b> par défaut), dont la valeur est développée pour la substitution de paramètres. Le caractère <b>!</b> (point d'exclamation) dans l'invite principale est remplacé par le numéro de commande.                                                                                                                                                                                                                                                                                             |

|               |                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PS2</b>    | Chaîne constituant l'invite système secondaire (> (supérieur à), par défaut.                                                                                                                                                                                                                                                                                                                                                              |
| <b>PS3</b>    | Chaîne constituant l'invite de sélection dans une boucle <b>select</b> (#? (dièse, point d'interrogation) par défaut).                                                                                                                                                                                                                                                                                                                    |
| <b>PS4</b>    | La valeur de cette variable (+, par défaut), développée pour la substitution de paramètres, précède chaque ligne des suivis d'exécution.                                                                                                                                                                                                                                                                                                  |
| <b>SHELL</b>  | Chemin d'accès au shell, enregistré dans l'environnement.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>TMOUT</b>  | Délai (en secondes) pendant lequel le shell reste inactif avant de quitter. Si ce délai est positif, le shell quitte en l'absence de commande entrée dans ce délai, après l'affichage de l'invite <b>PS1</b> . (Notez que le shell peut être compilé avec l'indication d'une limite maximale pour le délai.)<br><b>Remarque</b> : A l'expiration du délai imparti, une pause de 60-secondes s'écoule encore avant que le shell ne quitte. |
| <b>VISUAL</b> | Lorsque la valeur de cette variable se termine par emacs, gmacs ou vi, l'option correspondante est activée.                                                                                                                                                                                                                                                                                                                               |

Le shell attribue des valeurs par défaut aux paramètres **PATH**, **PS1**, **PS2**, **MAILCHECK**, **TMOUT** et **IFS**, tandis que les paramètres **HOME**, **SHELL**, **ENV** et **MAIL** *ne sont pas* définis par le shell (bien que le paramètre **HOME** soit défini par la commande **login**).

---

## Substitution de commandes (shell Korn ou POSIX)

Le shell Korn, ou POSIX, autorise la substitution de commandes.

Lors d'une substitution de commandes, le shell exécute une commande dans un environnement de sous-shell et remplace cette commande par son résultat. Pour exécuter une substitution de commande dans le shell Korn ou POSIX, procédez comme suit :

```
$(commande)
```

ou, avec des apostrophes inverses :

```
`commande`
```

**Remarque :** Bien que **ksh** accepte les apostrophes inverses, les normes XPG4 et POSIX les considèrent comme étant obsolètes. Ces dernières prônent l'utilisation de la syntaxe `$(commande)` par les applications portables.

Le shell exécute *commande* dans un environnement sous-shell, et remplace le texte de *commande*, encadré de ( ) ou d'apostrophes inverses, par le résultat standard de la commande, en supprimant les retours ligne à la fin de la substitution.

Dans l'exemple suivant, les signes \$( ) (dollar, parenthèses) entourant la commande indiquent la substitution du résultat de la commande **whoami** :

```
echo My name is: $(whoami)
```

La commande suivante a le même effet :

```
echo My name is: `whoami`
```

Dans les deux cas, le résultat (pour l'utilisateur *dee*) est le suivant :

```
My name is: dee
```

Vous pouvez également substituer des expressions arithmétiques en les mettant entre parenthèses. Par exemple, la commande :

```
echo Each hour contains=$((60 * 60)) seconds
```

génère :

```
Each hour contains 3600 seconds
```

Le shell Korn ou POSIX supprime tous les caractères ligne suivante situés à droite. Par exemple, si votre répertoire courant contient les fichiers *file1*, *file2*, et *file3*, la commande :

```
echo $(ls)
```

supprime les caractères ligne suivante et affiche :

```
file1 file2 file3
```

Si vous souhaitez conserver les caractères ligne suivante, mettez la commande entre guillemets (" ") :

```
echo "$(ls)"
```

## Evaluation arithmétique (shell Korn ou POSIX)

La commande intégrée **let** du shell Korn ou POSIX permet d'effectuer des calculs sur des nombres entiers. Le format des constantes est  $[Base]Nombre$ . Le paramètre *Base* est un nombre décimal compris entre 2 et 36, correspondant à la base de numérotation. Le paramètre *Nombre* est un nombre dans cette base. En l'absence de spécification du paramètre *Base*, le shell utilise une base de 10.

Les expressions arithmétiques adoptent la même syntaxe, la même priorité et la même associativité qu'en langage C. Tous les opérateurs entiers, à l'exception de **++** (double signe plus), **--** (tiret), **?:** (point d'interrogation, deux-points) et **,** (virgule), sont admis. Voilà la liste des opérateurs valides (par ordre de priorité décroissant) :

| Opérateurs arithmétiques (shell Korn)     |                                      |
|-------------------------------------------|--------------------------------------|
| Opérateur                                 | Définition                           |
| -                                         | Signe moins (unaire)                 |
| !                                         | Négation logique                     |
| ~                                         | Négation niveau bit                  |
| *                                         | Multiplication                       |
| /                                         | Division                             |
| %                                         | Reste                                |
| +                                         | Addition                             |
| -                                         | Soustraction                         |
| <<, >>                                    | Décalage arithmétique gauche, droite |
| <=, >=, <, >, ==, !=                      | Comparaison                          |
| &                                         | ET niveau bit                        |
| ^                                         | OU niveau bit exclus if              |
|                                           | OU niveau bit                        |
| &&                                        | ET logique                           |
|                                           | OU logique                           |
| = *=, /=, &= +=, -=, <<=, >>=, &=, ^=,  = | Affectation                          |

Nombre d'opérateurs arithmétiques (\*, &, <, >, par exemple) ont une signification particulière pour le shell Korn ou POSIX. Ils doivent être placés entre guillemets. Par exemple, pour multiplier la valeur courante de *y* par 5 et réaffecter la nouvelle valeur à *y*, utilisez l'expression :

```
let "y = y * 5"
```

Mis entre guillemets, le caractère\* (astérisque) perd sa signification spéciale.

Vous pouvez regrouper des opérations dans le cadre de la commande **let**. Par exemple, l'expression :

```
let "z = q * (z - 10)"
```

multiplie *q* par la valeur de *z* diminuée de 10.

Si l'évaluation ne porte que sur une seule expression, le shell Korn ou POSIX propose un deuxième format de la commande **let**. Le shell traite les commandes entre doubles parenthèses (( )) comme des expressions déclarées. Ainsi, l'expression :

```
((x = x / 3))
```

équivalent à :

```
let "x = x / 3"
```

Dans une expression arithmétique, les paramètres nommés sont référencés par leur nom, sans appel à la syntaxe de substitution de paramètre, et évalués comme des expressions arithmétiques.

Pour spécifier une représentation interne entière d'un paramètre nommé, utilisez la commande intégrée **typeset** assortie de l'indicateur **-i**. L'évaluation arithmétique portera sur la valeur de chaque affectation à un paramètre nommé. Si vous n'indiquez pas de base de numération, celle-ci est définie par la première affectation du paramètre (elle servira lors de la substitution de paramètres).

---

## Séparation de zones

Après exécution d'une substitution de commandes, le shell Korn balaye les résultats des substitutions, à la recherche des séparateurs de zones définis dans la variable **IFS** (Internal Field Separator). A chaque occurrence d'un séparateur, le shell partage la ligne en mots distincts. Il conserve les arguments nuls explicites (" " ou " "), mais élimine les implicites (résultant de paramètres non renseignés).

- Si la valeur d'**IFS** est un espace, une tabulation, un caractère nouvelle ligne, ou qu'elle n'est pas définie, toute séquence d'espaces, de tabulations, de caractères nouvelle ligne est ignorée si elle se trouve au début ou à la fin de l'entrée, ou délimite une zone si elle se trouve à l'intérieur de l'entrée. Ainsi, l'entrée suivante génère deux zones, **school** et **days** :

```
<newline><space><tab>school<tab><tab>days<space>
```

- Sinon, et si la valeur d'**IFS** est non nulle, les règles suivantes sont applicables (dans l'ordre). Le terme "espace **IFS**" désigne toute séquence (zéro ou plusieurs occurrences) de caractères définie dans **IFS** (par exemple, si **IFS** contient espace/virgule/tabulation, toute combinaison d'espaces et de tabulations constitue un espace **IFS**).
  - a. Un espace **IFS** en début ou en fin d'entrée est ignoré.
  - b. Chaque occurrence d'un caractère **IFS** autre que l'espace **IFS**, associé à un espace **IFS** adjacent, délimite une zone.
  - c. Des espaces **IFS** de longueur non nulle délimitent une zone.

---

## Substitution de noms de fichiers (shell Korn ou POSIX)

Pour procéder à une substitution de noms de fichier, le shell Korn, ou POSIX, balaye les mots de la commande spécifiée par *Mot*, à la recherche de certains caractères. Si un mot contient les caractères \* (astérisque), ? (point d'interrogation) ou [ (crochet gauche), et en l'absence de l'indicateur **-f**, le shell le considère comme une trame. Il les remplace par les noms de fichiers correspondant à la trame, triés selon l'ordre en vigueur dans l'environnement local. Si aucun nom de fichier ne correspond, le mot reste tel quel.

Lorsque le shell utilise une trame pour la substitution de noms de fichiers, les caractères . (point) et / (barre oblique) doivent correspondre explicitement.

**Remarque :** Dans d'autres cas de recherche sur la base d'une trame, le shell Korn ne traite pas les caractères distinctement.

Voici les substitutions générées :

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| *     | Correspond à n'importe quelle chaîne, chaîne nulle comprise.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| ?     | Correspond à un seul caractère (quelconque).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| [...] | Correspond à n'importe lequel des caractères entre crochets. Une paire de caractères séparés par un - (trait d'union) correspond à n'importe quel caractère compris (selon l'ordre local en vigueur) entre ces deux valeurs. Si le premier caractère qui suit le [ (crochet gauche) d'ouverture est un ! (point d'exclamation), n'importe quel caractère à l'exclusion de ceux entre crochets correspond. Pour inclure un - (trait d'union) dans l'ensemble de caractères considéré, placez-le au début ou à la fin de l'ensemble. |

Vous pouvez également utiliser la notation `[:classecar:]` pour faire correspondre les noms de fichier d'un intervalle donné : tout caractère de la *classe* est réputé correspondre. La définition du contenu des classes de caractères est indiqué via la catégorie **LC\_CTYPE** de la sous-routine **setlocale**. Toutes les classes de caractères spécifiées dans l'environnement local sont reconnues.

Voici quelques unes de ces classes :

- **alnum**
- **alpha**
- **cntrl**
- **digit**
- **graph**
- **lower**
- **print**
- **punct**
- **space**
- **upper**
- **xdigit**

Par exemple, `[:upper:]` correspond à toute lettre majuscule.

Le shell Korn prend en charge l'extension de noms de fichiers basée sur le classement des éléments ou des symboles, et des classes d'équivalence.

Une *ListeTrames* est une liste d'une ou de plusieurs trames, séparées par une barre verticale (|). Les trames composées sont formées à partir des caractères :

|                       |                                                                   |
|-----------------------|-------------------------------------------------------------------|
| <i>?(ListeTrames)</i> | Correspond éventuellement à l'une des trames spécifiées.          |
| <i>*(ListeTrames)</i> | Correspond à zéro ou plusieurs occurrences des trames spécifiées. |
| <i>+(ListeTrames)</i> | Correspond à une ou plusieurs occurrences des trames spécifiées.  |
| <i>@(ListeTrames)</i> | Correspond exactement à l'une des trames spécifiées.              |
| <i>!(ListeTrames)</i> | Correspond à toute trame, à l'exclusion de celles spécifiées.     |

L'utilisation de trames souffre quelques restrictions. Si le premier caractère d'un nom de fichier est un . (point), il ne peut être mis en correspondance qu'avec une trame commençant aussi par un point. Par exemple, \* (astérisque) correspond aux fichiers `myfile` et `yourfile`, mais pas aux fichiers `.myfile` et `.yourfile`. Pour ces fichiers, utilisez, par exemple :

```
. *file
```

Si la recherche n'aboutit pas, c'est la trame elle-même qui est renvoyée en sortie.

Excluez des noms de fichiers et de répertoires les caractères \* (astérisque), ? point d'interrogation), [ (crochet gauche) et ] (crochet droit) : ils risquent de provoquer une boucle infinie lors des recherches sur la base de trames.

## Suppression des caractères de déclaration

Les caractères de déclaration \ (barre oblique inverse), ' (apostrophe) et " (guillemets) présents dans le mot d'origine, sont supprimés – sauf s'ils sont eux-mêmes déclarés.

- Voir aussi Shell Korn

---

## Réacheminement des entrées/sorties (shell Korn ou POSIX)

Avant d'exécuter une commande, le shell C balaye la ligne d'entrée à la recherche de caractères de réacheminement. Ces derniers lui indiquent de réacheminer les entrées et les sorties. Ils peuvent se trouver n'importe où dans une commande simple, ou la précéder ou la suivre. Ils ne sont pas transmis à la commande appelée.

Le shell exécute les substitutions de commandes et de paramètres avant de prendre en compte les paramètres *Mot* et *Chiffre*, sauf indication spéciale. La substitution de nom de fichier ne se produit que si la trame correspond à un seul fichier et que les blancs ne sont pas interprétés.

|                             |                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;Mot</code>        | Le fichier <i>Mot</i> est l'entrée standard (descripteur de fichier 0).                                                                                                                                                                                                                                                                                                                                                              |
| <code>&gt;Mot</code>        | Le fichier <i>Mot</i> est l'entrée standard (descripteur de fichier 1). Si le fichier n'existe pas, il est créé. Si le fichier existe et que l'option <b>noclobber</b> est activée, une erreur est générée ; sinon, le fichier est tronqué à la longueur zéro.                                                                                                                                                                       |
| <code>&gt; Mot</code>       | Semblable à la commande <code>&gt;Mot</code> , sauf que cette instruction de réacheminement prime sur l'option <b>noclobber</b> .                                                                                                                                                                                                                                                                                                    |
| <code>&gt; &gt;Mot</code>   | Le fichier <i>Mot</i> est la sortie standard. S'il existe, le shell y ajoute la sortie (après recherche du caractère de fin de fichier). Si le fichier n'existe pas, il est créé.                                                                                                                                                                                                                                                    |
| <code>&lt;&gt;Mot</code>    | Ouvre en lecture et en écriture le fichier <i>Mot</i> , en tant qu'entrée standard.                                                                                                                                                                                                                                                                                                                                                  |
| <code>&lt;&lt;[-]Mot</code> | Lit chaque ligne de l'entrée shell jusqu'à en trouver une ne contenant que la valeur de <i>Mot</i> ou un caractère de fin de fichier. Le shell n'effectue aucune substitution (paramètre, commande ou nom de fichier) sur le fichier spécifié. Le document résultant, appelé <i>document here</i> , devient l'entrée standard. Si l'un des caractères de <i>Mot</i> est déclaré, les caractères du document ne sont pas interprétés. |

Le document here est traité comme un seul mot, commençant après le caractère nouvelle ligne suivant, et continuant jusqu'à une ligne ne contenant que le délimiteur, sans espaces finaux. Le document here, s'il existe, est alors démarré. Le format est le suivant :

```
[n]<<mot
 document here
délimateur
```

Si un des caractères de *mot* est déclaré, le délimiteur est créé par suppression des guillemets de *mot*. Les lignes du document here ne sont pas développées. Sinon, le délimiteur est *mot* lui-même. Si aucun caractère de *mot* n'est déclaré, toutes les lignes du document here sont développées pour procéder au développement des paramètres, aux substitutions de commandes et au développement arithmétique.

Le shell effectue une substitution de paramètre sur les données réacheminées. Pour empêcher le shell d'interpréter les caractères (barre oblique inverse), \$ (dollar), et ' (apostrophe), ainsi que le premier caractère de *Mot*, faites-les précéder du caractère

Si un signe – (moins) est ajouté à la suite de <<, le shell élimine toutes les tabulations à gauche du paramètre *Mot* et du document.

|                                |                                                                                              |
|--------------------------------|----------------------------------------------------------------------------------------------|
| <code>&lt;&amp;Chiffre</code>  | Duplique l'entrée standard à partir du descripteur de fichier spécifié par <i>Chiffre</i> .  |
| <code>&gt;&amp; Chiffre</code> | Duplique la sortie standard à partir du descripteur de fichier spécifié par <i>Chiffre</i> . |
| <code>&lt;&amp;-</code>        | Ferme l'entrée standard.                                                                     |
| <code>&gt;&amp;-</code>        | Ferme la sortie standard.                                                                    |
| <code>&lt;&amp;p</code>        | Transfère l'entrée du coprocess vers l'entrée standard.                                      |
| <code>&gt;&amp;p</code>        | Transfère la sortie du coprocess vers la sortie standard.                                    |

Lorsqu'un chiffre précède l'une des options de réacheminement, le descripteur de fichier indiqué est spécifié par le chiffre (et non par la valeur par défaut 0 ou 1). dans l'exemple suivant, le shell ouvre le descripteur de fichier 2 pour écriture comme duplicata du descripteur de fichier 1 :

```
... 2>&1
```

L'ordre de spécification des réacheminements est important. Le shell évalue chaque réacheminement selon l'association (*DescripteurFichier, Fichier*) au moment de l'évaluation. Dans l'exemple :

```
... 1>Fichier 2>&1
```

le descripteur de fichier 1 est associé à *Fichier*. Le shell associe le descripteur de fichier 2 au fichier associé au descripteur de fichier 1 (*Fichier*). Si l'on inversait l'ordre des réacheminements, le descripteur de fichier 2 serait associé au terminal (dans l'hypothèse où le descripteur de fichier 1 l'était) et le descripteur de fichier 1, associé à *Fichier*.

Lorsqu'une commande est suivie d'un signe & (perluète) et que la fonction de contrôle des travaux n'est pas active, l'entrée standard par défaut de la commande est le fichier vide, */dev/null*. Sinon, l'environnement d'exécution d'une commande contient les descripteurs de fichier du shell appelant, modifiés par les spécifications d'entrée et de sortie.

Pour en savoir plus, reportez-vous à "Réacheminement des entrées/sorties", page 4-1

## Fonction coprocess

Le shell Korn, ou POSIX, permet d'exécuter en arrière-plan une ou plusieurs commandes. Ces commandes, lancées à partir d'un script shell, sont appelées *coprocess*. Ces derniers sont particulièrement utiles pour communiquer avec un programme.

Pour désigner un coprocess, placez un opérateur **|&** (barre verticale, perduète) après la commande. L'entrée et la sortie standard de la commande sont dirigées vers votre script.

Lorsque vous travaillez avec un coprocess, vous devez :

- inclure un caractère nouvelle ligne à la fin de chaque message ;
- envoyer chaque message en sortie vers la sortie standard ;
- effacer la sortie standard après chaque message.

L'exemple suivant montre comment passer une entrée à un coprocess et en recevoir les résultats :

```
echo "Initial process"
./FileB.sh |&
read -p a b c d
echo "Read from coprocess: $a $b $c $d"
print -p "Passed to the coprocess"
read -p a b c d
echo "Passed back from coprocess: $a $b $c $d"

FileB.sh
echo "The coprocess is running"
read a b c d
echo $a $b $c $d
```

La sortie standard qui en résulte est :

```
Initial process
Read from coprocess: The coprocess is running
Passed back from coprocess: Passed to the coprocess
```

La commande **print -p** permet d'écrire dans le coprocess. La commande **read -p** permet d'y lire.

### Réacheminement des entrées/sorties du coprocess

La fonction de réacheminement des E/S permet de réaffecter l'entrée et la sortie standard d'un coprocess à un descripteur de fichier numéroté. Par exemple, la commande :

```
exec 5>&p
```

transfère l'entrée du coprocess vers le descripteur de fichier 5.

Vous pouvez ensuite, via la syntaxe de réacheminement standard ou par appel d'un autre coprocess, réacheminer la sortie de la commande vers le coprocess. Vous pouvez également démarrer un autre coprocess. La sortie des deux coprocess est connectée au même tube, lisible via la commande **read -p**. Pour mettre fin au coprocess, tapez :

```
read -u5
```

---

## Etats de sortie

Si le shell détecte une erreur (de syntaxe, par exemple), il renvoie un état de sortie non nul. Sinon, il renvoie l'état de sortie de la dernière commande exécutée. Si le numéro de la ligne comportant l'erreur est supérieur à 1, il est également indiqué (entre crochets [ ]), après le nom de la fonction ou de la commande. Le shell fait état des erreurs d'exécution en indiquant le nom de la fonction ou de la commande en cause, ainsi que le numéro d'erreur correspondant.

Avec un shell non interactif, une erreur rencontrée par une commande, intégrée ou non, génère un message de diagnostic, comme indiqué dans le tableau ci-après.

| Erreur                                                                     | Commande intégrée | Autres fonctions |
|----------------------------------------------------------------------------|-------------------|------------------|
| Erreur de syntaxe du langage shell                                         | quitte            | quitte           |
| Erreur de syntaxe sur une opération (erreur sur une option ou un opérande) | quitte            | ne quitte pas    |
| Erreur de réacheminement                                                   | quitte            | ne quitte pas    |
| Erreur d'affectation de variable                                           | quitte            | ne quitte pas    |
| Erreur de développement                                                    | quitte            | quitte           |
| Commande non trouvée                                                       | non applicable    | quitte parfois   |
| Script dot non trouvé                                                      | quitte            | non applicable   |

Si une erreur indiquée "quitte (parfois)" se produit dans un sous-shell, le sous-shell est (parfois) quitté, avec un état de sortie non nul, mais pas le script contenant le sous-shell.

Dans tous les cas indiqués sur le tableau, un shell interactif émet un message de diagnostic vers la sortie standard, sans quitter.

Pour en savoir plus sur le réacheminement, reportez-vous à "Réacheminement des entrées/sorties (shell Korn ou POSIX)."

- Voir aussi Shell Korn

---

## Commandes du shell Korn ou POSIX

Une commande du shell Korn est l'un des éléments suivants :

- Commande simple, page 11-5
- Pipeline, page 11-5
- Listes d'historique, page 11-5
- Commande composée, page 11-33
- Fonctions, page 11-35.

Lorsque vous émettez une commande shell Korn ou POSIX, le shell l'évalue et :

- effectue toutes les substitutions indiquées ;
- détermine si la commande contient une / (barre oblique). Dans l'affirmative, exécute le programme nommé par le chemin spécifié.

Dans la négative, le shell effectue les opérations suivantes :

- Il détermine s'il s'agit d'une commande intégrée spéciale. Dans l'affirmative, il exécute la commande dans le process shell courant.

Pour en savoir plus, reportez-vous à "Commandes intégrées du shell Korn ou POSIX", page 11-37.

- Il compare la commande aux fonctions définies par l'utilisateur. Si elle correspond à une de ces fonctions, les paramètres positionnels sont sauvegardés et réinitialisés par les arguments de l'appel de **fonction**. Lorsque la fonction s'achève ou émet un retour, la liste des paramètres positionnels est restaurée, et l'éventuel traitement d'interruption défini pour **EXIT** dans la fonction est exécuté. La valeur d'une fonction est celle de la dernière commande exécutée. Une fonction est exécutée dans le process shell courant.
- Si le nom de la commande correspond à celui d'une commande intégrée standard, cette commande est appelée.

Pour en savoir plus, reportez-vous à "Commandes intégrées du shell Korn ou POSIX", page 11-37.

- Il crée un process et tente d'exécuter la commande par le biais de la commande **exec** (s'il ne s'agit ni d'une commande intégrée, ni d'une fonction définie par l'utilisateur).

Le shell Korn, ou POSIX, explore les répertoires du chemin d'accès à la recherche d'un fichier exécutable. La variable shell **PATH** définit le chemin d'accès au répertoire contenant la commande. Les autres répertoires sont séparés par un signe : (deux points). Le chemin par défaut est `/usr/bin:` (indiquant le répertoire **/usr/bin**, et le répertoire courant, dans cet ordre). Le répertoire courant est spécifié par des deux points contigus, ou par un caractère deux points au début ou à la fin de la liste des chemins.

Si le fichier est doté des droits d'exécution, mais n'est pas un répertoire ni un fichier **a.out**, le shell suppose qu'il contient les commandes shell. Le process shell en cours génère dynamiquement un sous-shell qui lit le fichier. Tous les alias non exportés, toutes les fonctions et tous les paramètres indiqués sont supprimés du fichier. Si le fichier de commande shell est accessible en lecture, ou que les bits **setuid** ou **setgid** sont définis dans le fichier, le shell lance un agent qui définit les droits d'accès et exécute le shell, le fichier de commande shell étant passé comme un fichier ouvert. Toute commande entre parenthèses est exécutée dans un sous-shell et les quantités non exportées ne sont pas supprimées.

Cette section traite des points suivants :

- Commandes composées du shell Korn, page 11-33.
- Fonctions, page 11-35.
- Commandes intégrées (shell Korn ou POSIX), page 11-37.
- Expressions conditionnelles, page 11-49.
- Voir aussi Shell Korn

## Commandes composées du shell Korn

Une commande composée est une liste de commandes simples, un pipeline ou une commande commençant par un mot réservé. Vous serez souvent amené à vous servir de commandes composées telles que **if**, **while** et **for** lorsque vous écrirez des scripts shell.

### Liste des commandes composées du shell Korn ou POSIX

**for** *Identificateur* [**in** *Mot ...*] ;**do** *Liste* ;**done**

A chaque itération de la commande **for**, *Identificateur* prend la valeur du mot suivant de la liste **in** *Mot ...*. A défaut de cette liste, c'est la commande **do** *Liste* qui est exécutée pour chaque paramètre positionnel défini. L'exécution se termine lorsqu'il n'y a plus de mots dans la liste. Pour en savoir plus sur les paramètres positionnels, reportez-vous à "Substitution de paramètres (shell Korn ou POSIX)", page 11-17.

**select** *Identificateur* [**in** *Mot ...*] ;**do** *Liste* ;**done**

La commande **select** dirige vers la sortie d'erreur standard (descripteur de fichier 2) l'ensemble de mots spécifié, chaque mot étant précédé d'un numéro. Si la liste **in** *Mot ...* est omise, ce sont les paramètres positionnels qui sont utilisés. L'invite **PS3** s'affiche et une ligne est lue à partir de l'entrée standard. Si cette ligne correspond à l'un des numéros des mots de la liste, *Identificateur* prend la valeur du mot correspondant à ce numéro.

Si la ligne lue est vide, la liste de sélection s'affiche à nouveau. Sinon, *Identificateur* prend la valeur nulle. Le contenu de la ligne lue est sauvegardé dans le paramètre **REPLY**. Le paramètre *Liste* est exécuté pour chaque sélection jusqu'à ce qu'un caractère d'interruption ou de fin-de-fichier soit rencontré. Pour en savoir plus sur les paramètres positionnels, reportez-vous à "Substitution de paramètres (shell Korn ou POSIX)", page 11-17.

**case** *Mot* **in** [( *Trame* | *Trame* ... .) *Liste* ;;] ... **esac**

Une commande **case** exécute la *Liste* associée à la première *Trame* correspondant au *Mot*. Le format des trames est le même que celui utilisé pour les substitutions de noms de fichiers.

**if** *Liste* ;**then** *Liste* [**elif** *Liste* ;**then** *Liste*] ... [**else** *Liste*] ;**fi**

*Liste* spécifie une liste de commandes à exécuter. Le shell exécute d'abord la commande **if** *Liste*. Si cette commande renvoie un état de sortie nul, il exécute la commande **then** *Liste*. Sinon, il exécute les commandes spécifiées par le paramètre *Liste* qui suit **elif**.

Si la dernière commande de **elif** *Liste* renvoie la valeur zéro, la commande **then** *Liste* est exécutée. Si la dernière commande de **then** *Liste* renvoie la valeur zéro, la commande **else** *Liste* est exécutée. Si aucune des commandes spécifiées par les paramètres *Liste* pour **else** ou **then** n'est exécutée, **if** renvoie un état de sortie nul.

**while** *Liste* ;**do** *List* ;**done**

**while** *Liste* ;**do** *Liste* ;**done**

*Liste* spécifie une liste de commandes à exécuter. *Liste* spécifie une liste de commandes, exécutées itérativement par la commande **while**. Si la commande renvoie une valeur nulle, le shell exécute la commande **do** *Liste*. Si la commande renvoie une valeur non nulle, la boucle se termine. Si aucune des commandes de **do** *Liste* n'est exécutée, la commande **while** renvoie une valeur nulle. La commande **until** a le même effet que la commande **while**, à ceci près que le test de fin de boucle n'est pas exécuté.

(*Liste*)

*Liste* spécifie une liste de commandes à exécuter. Le shell exécute le paramètre *Liste* dans un environnement distinct.

**Remarque** : Si l'imbrication exige la présence de deux parenthèses contiguës, vous devez insérer un espace entre elles afin de différencier la commande de l'évaluation arithmétique.

(*Liste*)

*Liste* spécifie une liste de commandes à exécuter. Le paramètre *Liste* est simplement exécuté.

**Remarque** : Contrairement aux parenthèses (), les accolades {} indiquent les mots réservés (utilisés à des fins particulières, et non des ID déclarés par l'utilisateur). Pour être reconnus, ces mots réservés doivent se trouver au début d'une ligne ou après un point-virgule (;).

[[*Expression*]]

Évalue le paramètre *Expression*. Si elle est vraie, la commande renvoie un état de sortie nul.

**function** *Identificateur* {*Liste* ;} ou **function** *Identificateur* () {*Liste* ;}

Définit une fonction référencée par *Identificateur*. Le corps de la fonction est constitué par la liste des commandes entre accolades {}. Les parenthèses () constituent deux opérateurs, aussi le panachage de caractères blancs avec *Identificateur*, (et) est-il autorisé, quoique non obligatoire.

**time** *Pipeline*

Exécute le paramètre *Pipeline*. Le délai écoulé, le temps utilisateur et le temps système sont envoyés sur la sortie standard.

## Fonctions

Le mot réservé **function** définit les fonctions shell. Le shell lit et enregistre les fonctions en interne, les noms d'alias étant résolus à la lecture de la fonction. Le shell exécute les fonctions de la même manière que les commandes, les arguments étant passés comme paramètres positionnels. Pour en savoir plus sur les paramètres positionnels, reportez-vous à "Substitution de paramètres (shell Korn ou POSIX", page 11-17.

Le shell Korn ou POSIX exécute les fonctions dans l'environnement à partir duquel elles sont appelées. Les éléments ci-après sont tous communs à la fonction et au script appelant (des effets secondaires peuvent en résulter):

- valeurs et attributs des variables (sauf utilisation de **typeset** dans la fonction pour déclarer une variable locale) ;
- répertoire de travail ;
- alias, définitions de fonctions et attributs ;
- paramètre spécial \$ ;
- fichier ouverts.

Les éléments non communs à la fonction et au script appelant (aucun effet secondaire) sont les suivants :

- paramètres positionnels ;
- paramètre spécial # ;
- variables de la liste d'affectation de la fonction appelée ;
- variables déclarées via **typeset** dans la fonction ;
- options ;
- traitements d'interruption (toutefois, les signaux ignorés du script appelant sont également ignorés par la fonction).

**Remarque :** Dans les versions antérieures du shell Korn, les traitements d'interruption (autres que **EXIT** et **ERR**) étaient communs à la fonction et au script appelant.

Un traitement d'interruption sur **0** ou **EXIT** défini dans le corps d'une fonction est exécuté une fois la fonction achevée, dans l'environnement à partir duquel a été appelée la fonction. Un traitement d'interruption défini à l'extérieur d'une fonction est exécuté après sortie du shell Korn. Dans les versions antérieures du shell Korn, aucun traitement d'interruption sur **0** ou **EXIT** défini à l'extérieur du corps de la fonction n'était exécuté après sortie de la fonction.

Erreurs de syntaxe et affectations de variables suivent les règles définies au paragraphe "Commandes intégrées du shell Korn ou POSIX".

Pour exécuter une commande composée, il suffit de spécifier son nom comme celui d'une commande simple, les opérandes de la commande faisant temporairement office de paramètres positionnels, pendant la durée de l'exécution. Le paramètre spécial # est modifié pour refléter le nombre d'opérandes. Le paramètre spécial 0 reste inchangé.

La commande spéciale **return** permet de reprendre la main à partir des appels de la fonction. Si des erreurs surviennent à l'intérieur des fonctions, le contrôle est repris par le programme appelant.

Les ID fonctions sont listés via l'option **-f** ou **+f** de la commande spéciale **typeset**. L'option **-f** liste également le texte des fonctions. Les fonctions sont désactivées par l'option **-f** de la commande spéciale **unset**.

Généralement, les fonctions sont désactivées lorsque le shell exécute un script shell. La commande **typeset** assortie de l'option **-xf** permet d'exporter une fonction vers des scripts exécutés, sans appel séparé du shell. Les fonctions qui doivent être définies par appel séparé du shell doivent être spécifiées dans le fichier **ENV** via la commande **typeset** assortie de l'option **-xf**.

Une fonction renvoie un état de sortie nul si sa déclaration a échoué, un état positif sinon. L'état de sortie d'un appel de fonction est celui de la dernière commande exécutée par la fonction.

---

## Liste des commandes intégrées du shell Korn ou POSIX

Les commandes spéciales sont intégrées au shell Korn ou POSIX, et exécutées dans le process shell. Sauf indication contraire, la sortie est dirigée vers le descripteur de fichier 1, avec un état de sortie nul si la commande est exempte d'erreurs de syntaxe. Le réacheminement des entrées/sorties est autorisé. Il existe deux types de commandes intégrées : les *commandes intégrées spéciales* et les *commandes intégrées standard*.

La Liste des commandes intégrées du shell Korn ou POSIX répertorie ces commandes par ordre alphabétique.

Voici les principales différences entre une commande standard et une commande spéciale :

- Une erreur de syntaxe dans une commande spéciale n'entraîne pas forcément l'arrêt du shell, contrairement à ce qui se passe avec une commande standard. Une erreur de syntaxe ne générant pas d'arrêt du shell renvoie un état de sortie nul.
- Les affectations de variables spécifiées pour une commande spéciale restent en vigueur après exécution de la commande, ce qui n'est pas le cas pour les commandes standard.
- Les réacheminements d'E/S sont traités après les affectations de paramètres.

En outre, les mots sous forme d'une affectation de paramètre, qui suivent une commande spéciale **export**, **readonly** ou **typeset**, sont développés selon les règles applicables à l'affectation de paramètres. Ainsi, la substitution de tilde est effectuée après le signe = (égal), et la séparation de mot et la substitution de noms de fichiers ne sont pas exécutées.

### Description des commandes spéciales

Voici les commandes intégrées spéciales du shell Korn :

|          |        |          |         |
|----------|--------|----------|---------|
| :        | eval   | newgrp   | shift   |
| .        | exec   | readonly | times   |
| break    | exit   | return   | trap    |
| continue | export | set      | typeset |
|          | unset  |          |         |

**:** [*Argument ...*] Ne développe que les arguments. Utilisée lorsqu'une commande est requise, comme la condition *then* dans une instruction **if**, mais que cette commande n'exécute aucune action.

**.** *Fichier* [*Argument ...*] Lit le fichier spécifié et exécute les commandes, dans l'environnement shell courant. Le chemin de recherche, spécifié par la variable **PATH**, donne accès au répertoire contenant le fichier. Les éventuels arguments sont traités comme des paramètres positionnels. A défaut, les paramètres positionnels restent inchangés. L'état de sortie est celui de la dernière commande exécutée. Pour en savoir plus sur les paramètres positionnels, reportez-vous à "Substitution de paramètres (shell Korn ou POSIX)", page 11-17.

**Remarque :** La commande **.** *Fichier* [*Argument ...*] lit l'intégralité du fichier avant d'exécuter une commande. Aussi les commandes **alias** et **unalias** du fichier ne peuvent-elles s'appliquer aux fonctions définies dans le fichier.

**break**[*n*] Quitte, le cas échéant, la boucle **for**, **while**, **until** ou **select**. Si vous spécifiez *n*, la commande reprend le traitement à la *nième* boucle. La valeur de *nest* est un entier supérieur ou égal à 1.

**continue**[*n*] Reprend à l'itération suivante de la boucle **for**, **while**, **until** ou **select**. Si vous spécifiez *n*, la commande reprend à la *nième* boucle. La valeur de *nest* est un entier supérieur ou égal à 1.

|                                                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>eval</b> [ <i>Argument...</i> ]                      | Lit les arguments spécifiés comme entrée et exécute la ou les commandes qui en résultent.                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>exec</b> [ <i>Argument...</i> ]                      | Exécute, à la place du shell, la commande spécifiée par l'argument (sans créer de nouveau process). Le process courant peut être perturbé par des arguments en entrée ou en sortie. Si vous omettez un argument, la commande <b>exec</b> modifie les descripteurs de fichier comme préconisé par la liste de réacheminement des entrées/sorties. Dans ce cas, tous les numéros de descripteurs de fichier supérieurs à 2 ouverts par ce mécanisme sont fermés dès qu'un autre programme est appelé. |
| <b>exit</b> [ <i>n</i> ]                                | Quitte le shell, l'état de sortie étant soit celui indiqué par le paramètre <i>n</i> (entier décimal compris entre 0 et 255), soit celui de la dernière commande exécutée. La sortie du shell est également induite par le caractère fin de fichier, à moins que l'option <b>ignoreeof</b> de la commande spéciale <b>set</b> ne soit active.                                                                                                                                                       |
| <b>export -p</b> [ <i>Nom</i> [= <i>Valeur</i> ]] ...   | <p>Marque les noms spécifiés, pour exportation automatique vers l'environnement des commande ultérieures.</p> <p><b>-p</b></p> <p>Ecrit sur la sortie standard les noms et valeurs de toutes les variables exportées, selon le format :</p> <pre>"export %s= %s\n", &lt;nom&gt; &lt;valeur&gt;</pre>                                                                                                                                                                                                |
| <b>newgrp</b> [ <i>Groupe</i> ]                         | <p>Equivalent à la commande <b>exec/usr/bin/newgrp</b> [<i>Groupe</i>].</p> <p><b>Remarque :</b> Cette commande ne génère pas de retour.</p>                                                                                                                                                                                                                                                                                                                                                        |
| <b>readonly -p</b> [ <i>Nom</i> [= <i>Valeur</i> ]] ... | <p>Marque en lecture seule le <i>Nom</i> spécifié. Aucune autre affectation ultérieure n'est possible.</p> <p><b>-p</b></p> <p>Ecrit sur la sortie standard les noms et valeurs de toutes les variables exportées, selon le format :</p> <pre>"export %s= %s\n", &lt;nom&gt; &lt;valeur&gt;</pre>                                                                                                                                                                                                   |
| <b>return</b> [ <i>n</i> ]                              | Provoque le renvoi d'une fonction shell au script appelant. L'état de retour est spécifié par la variable <i>n</i> ou, à défaut, par la dernière commande exécutée. Si vous appelez la commande <b>return</b> en dehors d'une fonction ou d'un script, elle équivaut à la commande <b>exit</b> .                                                                                                                                                                                                    |

**set** [+|-abCefhkmnostuvx] [+|-o *Option*]... [+|-A *Nom*] [*Argument* ...]

En l'absence de spécification d'options ou d'arguments, la commande **set** écrit les noms et les valeurs des variables shell dans l'ordre défini par l'environnement local. Lorsque des options sont spécifiées, les attributs du shell sont modifiés en conséquence (voir ci-après). Lorsque des arguments sont spécifiés, des paramètres positionnels sont définis ou deviennent indéfinis (voir ci-après). Les indicateurs de cette commande sont les suivants :

- A** Affectation de tableau. Annule la définition du paramètre *Nom* et affecte séquentiellement les valeurs à partir de la liste *Argument*. Si **+A** est défini, le paramètre *Nom* n'est pas annulé en premier.
- a** Exporte automatiquement tous les autres paramètres définis.
- b** Avertit l'utilisateur, en mode asynchrone, de la fin des travaux en arrière-plan.
- C** Equivalents à `set -o noclobber`.
- e** Exécute le traitement d'interruption **ERR**, le cas échéant, et sort lorsqu'une commande renvoie un état de sortie non nul. Ce mode est désactivé au cours de la lecture des profils.
- f** Désactive la substitution de nom de fichier.
- h** Désigne chaque commande comme alias tracé, à la première occurrence de la commande.
- k** Place tous les arguments d'affectation de paramètre dans l'environnement de la commande, et pas uniquement les arguments précédant le nom de commande.
- m** Exécute les travaux d'arrière-plan dans un process indépendant et affiche une ligne lorsqu'ils sont terminés. L'état de sortie des travaux d'arrière-plan est indiqué dans un message de fin. Sur les systèmes avec contrôle des travaux, cet indicateur est activé automatiquement pour les shells interactifs (reportez-vous à "Contrôle des travaux (shell Korn ou POSIX)", page 11-51.)
- n** Lit les commandes à la recherche d'erreurs de syntaxe, mais ne les exécute pas. Cet indicateur est ignoré par les shells interactifs.
- oOption** Affiche les valeurs de l'option courante si vous n'avez pas spécifié d'argument. Vous pouvez définir plusieurs options sur une même ligne de commande **ksh**. Si l'indicateur **+o** est défini, l'option spécifiée est désactivée. Les arguments disponibles, spécifiés par la variable *Option*, sont les suivants :
  - allexport** Identique à l'indicateur **-a**.
  - errexit** Identique à l'indicateur **-e**.
  - bgnice** Exécute tous les travaux d'arrière-plan avec une priorité inférieure. C'est le mode par défaut.
  - emacs** Appelle un éditeur en ligne du type emacs pour l'entrée des commandes.

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>gmacs</b>      | Appelle un éditeur en ligne de type gmacs pour l'entrée des commandes.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>ignoreeof</b>  | Ne quitte pas le shell lorsqu'un caractère fin de fichier est rencontré. Pour quitter le shell, lancez la commande <b>exit</b> ou appuyez sur Ctrl-D plus de 11 fois.                                                                                                                                                                                                                                                                                                                                                     |
| <b>keyword</b>    | Identique à l'indicateur <b>-k</b> .<br><b>Remarque</b> : Cet indicateur étant prévu pour assurer la compatibilité avec le shell Bourne, il est fortement déconseillé de l'utiliser.                                                                                                                                                                                                                                                                                                                                      |
| <b>markdirs</b>   | Ajoute une barre oblique (/) à tous les noms de répertoire générés par la substitution de nom de fichier.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>monitor</b>    | Identique à l'indicateur <b>-m</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>noclobber</b>  | Empêche la procédure de réacheminement de tronquer les fichiers. Lorsque vous spécifiez cette option, le symbole de réacheminement (> ) doit être suivi d'une barre verticale pour tronquer le fichier.                                                                                                                                                                                                                                                                                                                   |
| <b>noexec</b>     | Identique à l'indicateur <b>-n</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>noglob</b>     | Identique à l'indicateur <b>-f</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>nolog</b>      | Empêche que les définitions de fonction ne soient enregistrées dans le fichier historique.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>nounset</b>    | Identique à l'indicateur <b>-u</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>privileged</b> | Identique à l'indicateur <b>-p</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>trackall</b>   | Identique à l'indicateur <b>-h</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>verbose</b>    | Identique à l'indicateur <b>-v</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>vi</b>         | Appelle le mode insertion d'un éditeur en ligne de type vi pour l'entrée des commandes. Entrer le caractère d'échappement 033 met l'éditeur en mode transfert. Un retour envoie la ligne.                                                                                                                                                                                                                                                                                                                                 |
| <b>viraw</b>      | Traite chaque caractère comme s'il était tapé en mode vi.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>xtrace</b>     | Identique à l'indicateur <b>-x</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>-p</b>         | Désactive le traitement du fichier <b>\$HOME/.profile</b> et utilise le fichier <b>/etc/suid_profile</b> à la place du fichier <b>ENV</b> . Ce mode est activé chaque fois que l'ID utilisateur (UID) ou l'ID groupe (GID) effectif est différent du véritable UID ou du véritable GID. Désactiver cette option positionne l'UID ou le GID effectifs à la valeur de l'UID et du GID réels.<br><b>Remarque</b> : Le système n'admet pas l'option <b>-p</b> car il ne prend pas en charge les scripts shell <b>setuid</b> . |

- s** Effectue le tri lexicographique des paramètres positionnels.
- t** Quitte après lecture et exécution d'une seule commande.  
**Remarque :** Cet indicateur étant prévu pour assurer la compatibilité avec le shell Bourne, il est fortement déconseillé de l'utiliser.
- u** Traite les paramètres non définis comme des erreurs lors de la substitution.
- v** Affiche les lignes d'entrée au fur et à mesure de leur lecture.
- x** Affiche les commandes et leurs arguments au fur et à mesure de leur exécution.
- Désactive les indicateurs **-x** et **-v** et met fin à l'examen des arguments des indicateurs.
- Empêche toute modification des indicateurs. Cette option est très utile pour affecter au paramètre **\$1** une valeur commençant par un signe **-** (moins). Si aucun argument ne suit cet indicateur, les paramètres positionnels ne sont pas définis.

Faire précéder un des indicateurs de la commande **set** d'un signe **+** (plus) plutôt que d'un signe **-** (signe moins) désactive l'indicateur. Utilisez ces indicateurs à l'appel du shell. L'ensemble des indicateurs courant se trouve dans le paramètre **\$-**. A moins d'avoir spécifié l'indicateur **-A**, les autres arguments sont des paramètres positionnels affectés, dans l'ordre, à **\$1**, **\$2**, etc. En l'absence d'arguments, les noms et valeurs de tous les paramètres nommés sont dirigés vers la sortie standard.

**shift [n]**

Renomme les paramètres positionnels, de **\$<sub>n+1</sub>** ... à **\$1** .... La valeur par défaut du paramètre **n** est 1. Le paramètre **n** est une expression arithmétique représentée par un nombre non négatif inférieur ou égal au paramètre **\$#**.

**times**

Affiche les temps utilisateur et système cumulés pour le shell et les process exécutés à partir du shell.

## **trap** [*Commande*] [*Signal*] ...

Exécute la commande spécifiée dès réception par le shell du signal ou des signaux spécifiés. Le paramètre *Commande* est lu une fois à l'activation du traitement d'interruption et une fois à son exécution. Le paramètre *Signal* peut être un numéro ou le nom du signal. Les commandes de traitement d'interruption sont exécutées dans l'ordre des numéros de signal. Toute tentative de définir un traitement d'interruption sur un signal qui a été ignoré en entrée du shell courant est sans effet.

Si la commande est un signe – (moins), tous les traitements d'interruption sont restaurés à leurs valeurs d'origine. Si vous avez omis la commande et que le premier signal est un signal numérique, la commande **ksh** restaure les valeurs d'origine du ou des paramètres *Signal*.

**Remarque** : Si vous avez omis la commande et que le premier signal est un nom symbolique, le signal est interprété comme une commande.

Si la valeur de *Signal* est **ERR**, la commande spécifiée est exécutée chaque fois qu'une commande a un état de sortie non nul. Si la valeur de signal est **DEBUG**, la commande spécifiée est exécutée après chaque commande. Si la valeur de *Signal* est **0** ou **EXIT** et que la commande **trap** est définie à l'intérieur d'une fonction, l'action correspondante est exécutée à la fin de la fonction. Si la valeur de *Signal* est **0** ou **EXIT** et que la commande **trap** est définie à l'extérieur de la fonction, l'action correspondante est exécutée à la sortie du shell. La commande **trap** sans arguments affiche la liste des commandes associées à chaque numéro de signal.

Pour la liste complète des valeurs de *Signal*, utilisées dans la commande **trap** sans préfixe **SIG**, reportez-vous aux sous-routines **sigaction**, **sigvec** et **signal** dans le manuel *AIX Technical Reference, Volume 2: Base Operating System and Extensions*.

## **typeset** [+HLRZfirtux[*n*]] [*Nom*[=*Valeur*]] ...

Définit les attributs et les valeurs des paramètres shell. Lorsqu'il est appelé à l'intérieur d'une fonction, une nouvelle instance du paramètre *Nom* est créée. Le type et la valeur du paramètre sont restaurés à la fin de la fonction. Les indicateurs de la commande **typeset** sont les suivants :

- H** Assure le mappage AIX-fichier hôte sur des machines non AIX.
- L** Justifie à gauche et supprime les blancs à gauche du paramètre *Valeur*. Lorsque *n* est non nul, il définit la longueur de la zone ; sinon, c'est la valeur de sa première affectation qui le détermine. Une fois affecté, le paramètre est rempli à droite par des blancs ou tronqué, si nécessaire, pour s'ajuster à la zone. Lorsque l'indicateur **-Z** est défini, les zéros à gauche sont supprimés. L'indicateur **-R** est désactivé.
- R** Justifie à droite et remplit avec des blancs à gauche. Lorsque *n* est non nul, il définit la longueur de la zone ; sinon, c'est la valeur de sa première affectation qui le détermine. La zone reste remplie par des blancs ou est tronquée à partir de la fin lorsque le paramètre est réaffecté. L'indicateur **L** est désactivé.

- Z** Justifie à droite et remplit avec des zéros à gauche lorsque le premier caractère significatif est un chiffre et que l'indicateur **-L** n'est pas défini. Lorsque *n* est non nul, il définit la longueur de la zone ; sinon, c'est la valeur de sa première affectation qui le détermine.
- f** Indique que les noms se réfèrent à la fonction, et non aux paramètres. Aucune affectation ne peut être effectuée, les seuls indicateurs utilisables sont **-t**, **-u** et **-x**. L'indicateur **-t** active le suivi de la fonction. L'indicateur **-u** déclare non définie la fonction. Lorsque la fonction est référencée, le système recherche sa définition dans la variable **FPATH**. L'indicateur **-x** maintient en vigueur la définition de la fonction dans tous les scripts shell qui ne constituent pas un appel séparé de la commande **ksh**.
- i** Identifie le paramètre comme nombre entier, ce qui accélère le traitement arithmétique. Lorsqu'une valeur non nulle est affectée au paramètre *n*, cette valeur définit la base arithmétique en sortie.
- l** Convertit tous les caractères majuscules en minuscules. L'indicateur de conversion des majuscules **-u** est désactivé.
- r** Marque en lecture seule les noms spécifiés par *Nom*. Aucune autre affectation ultérieure n'est possible.
- t** Etiquette les paramètres spécifiés. Les étiquettes, éventuellement définies par l'utilisateur, n'ont aucune signification spéciale pour le shell.
- u** Convertit tous les caractères minuscules en majuscules. L'indicateur **-l** est désactivé.
- x** Marque pour exportation automatique vers l'environnement des commandes ultérieures le nom spécifié par *Nom*.

Un signe + (plus) à la place d'un signe – (moins) désactive les indicateurs de la commande **typeset**. Si vous ne spécifiez pas de paramètres *Nom* mais que vous spécifiez des indicateurs, la liste des noms de paramètres marqués (et éventuellement, leurs valeurs) s'affiche. (Si vous ne voulez pas que leurs valeurs soient affichées, utilisez + à la place de –.) A défaut de noms et d'indicateurs, les noms et attributs de tous les paramètres s'affichent.

**unset [-fv] Nom ...** Annule les valeurs et les attributs définis pour les paramètres indiqués par la liste *Nom*. Si **-v** est spécifié, *Nom* référence un nom de variable : le shell supprime sa définition et le retire de l'environnement. La définition des variables en lecture seule ne peut être supprimée. Supprimer la définition des variables **ERRNO**, **LINENO**, **MAILCHECK**, **OPTARG**, **OPTIND**, **RANDOM**, **SECONDS**, **TMOU**T et \_ (souligné) leur fait perdre leur signification spéciale, même si elles sont affectées par la suite.

Si **-f** est spécifié, *Nom* fait référence à un nom de fonction : le shell supprime la définition de la fonction.

## Description des commandes intégrées standard

Voici les commandes intégrées standard du shell Korn :

|         |         |           |         |
|---------|---------|-----------|---------|
| alias   | fg      | print     | ulimit  |
| bg      | getopts | pwd       | umask   |
| cd      | jobs    | read      | unalias |
| command | kill    | setgroups | wait    |
| echo    | let     | test      | whence  |
| fc      |         |           |         |

**alias** [-t] [-x] [*Alias*[=*Chaîne*]] ...

Crée ou redéfinit les alias, ou écrit les définitions d'alias existantes sur la sortie standard.

Pour en savoir plus, reportez-vous à la commande **alias** dans le manuel *AIX Commands Reference*.

**bg** [*IDTravail*...]

Place chaque travail spécifié en arrière-plan. Le travail courant est placé en arrière-plan si vous avez omis le paramètre *IDTravail*. Pour en savoir plus sur le contrôle des travaux, reportez-vous à "Contrôle des travaux (shell Korn ou POSIX)", page 11-51.

Pour en savoir plus sur l'exécution en arrière-plan, reportez-vous à la commande **bg** dans le manuel *AIX Commands Reference*.

**cd** [*Argument*]

**cd** *Ancien Nouveau*

Cette commande a deux formats. Premier format : le répertoire spécifié par *Argument* devient le répertoire courant. Si *Argument* a la valeur - (signe moins), vous passez au répertoire précédent. Par défaut, *Argument* est la variable shell **HOME**, la variable **PWD** ayant la valeur du répertoire courant.

La variable shell **CDPATH** définit le chemin d'accès au répertoire contenant la valeur d'*Argument*. Les autres répertoires sont séparés par un signe : (deux points). Le chemin vide (qui est le chemin par défaut) désigne le répertoire courant. Ce répertoire est affiché immédiatement après le signe égal ou entre des deux-points dans la liste des chemins. Si l'argument commence par une barre oblique (/), le chemin n'est pas utilisé. Sinon, tous les répertoires du chemin sont explorés.

Second format : le *Nouveau* nom remplace l'*Ancien* nom du répertoire courant, **PWD**, et tente de passer à ce répertoire.

**command** [-p] *Commande* [*Argument* ...]

**command** [-v | -V] *Commande*

La commande **command** amène le shell à traiter la commande spécifié comme une commande simple, éliminant la fonction shell de consultation.

Pour en savoir plus, reportez-vous à la commande **command** dans le manuel *AIX Commands Reference*.

**echo** [*Chaîne*...]

Ecrit les chaînes de caractères sur la sortie standard. Reportez-vous à la description de la commande **echo**. L'indicateur **-n** n'est pas pris en charge.

**fc** [-r] [-e *Editeur*] [*Premier* [*Dernier*]]

**fc** -l [-n] [-r] [*Premier* [*Dernier*]]

**fc** -s [*Ancien=Nouveau*] [*Premier*]

Affiche le contenu du fichier d'historique des commandes ou appelle un éditeur pour modifier et réexécuter les commandes précédemment définies dans le shell.

Pour en savoir plus, reportez-vous à la commande **fc** dans le manuel *AIX Commands Reference*.

**fg** [*IDTravail*]

Amène à l'avant-plan le travail spécifié. En l'absence de spécification de travail, la commande amène le travail courant à l'avant-plan.

Pour en savoir plus sur l'exécution en arrière-plan, reportez-vous à la commande **fg** dans le manuel *AIX Commands Reference*.

**getopts** *ChaîneOption Nom* [*Argument ...*]

Contrôle les options légales du paramètres *Argument*.

Pour en savoir plus, reportez-vous à la commande **getopts** dans le manuel *AIX Commands Reference*.

**jobs** [-l | -n | -p] [*IDTravail ...*]

Affiche l'état des travaux lancés dans l'environnement shell courant. En l'absence de spécification d'un travail spécifique, l'état de tous les travaux actifs est affiché. Si un travail est signalé terminé, le shell supprime l'ID correspondant de la liste des travaux reconnus par l'environnement shell courant.

Pour en savoir plus, reportez-vous à la commande **jobs** dans le manuel *AIX Commands Reference*.

Pour envoyer un signal aux process :

**kill** [-s { *NomSignal* | *NuméroSignal* } ] *IDProcess...*

**kill** [ -*NomSignal* | -*NuméroSignal* ] *IDProcess...*

Pour afficher la liste des noms de signaux :

**kill** -l [*EtatSortie* ]

Envoie un signal (par défaut, **SIGTERM**) à un process en cours. En général, ce dernier s'arrête alors. Pour arrêter un process, indiquez son identificateur (PID) dans la variable *IDProcess*. Le shell fait état des PID de chaque process exécuté en arrière-plan (sauf si vous avez lancé plusieurs process via un pipeline, auquel cas il fait état du numéro de dernier process). Pour déterminer le PID des commandes, vous disposez également de la commande **ps**.

Pour en savoir plus, reportez-vous à la commande **kill** dans le manuel *AIX Commands Reference*.

**let** *Expression...*

Evalue les expressions arithmétiques spécifiées. L'état de sortie est 0 si la valeur de la dernière expression est non nulle, 1 sinon. Pour en savoir plus, reportez-vous à "Évaluation arithmétique (shell Korn ou POSIX)", page 11-24.

**print** [-Rnprsu[n]] [*Argument ...*]

Imprime la sortie shell. En l'absence d'indicateurs, ou en présence des indicateurs - (moins) ou — (tiret), les arguments sont dirigés vers la sortie standard comme décrit par la commande **echo**.  
Action des indicateurs :

- R** Affiche en mode brut (les conventions d'échappement de la commande **echo** sont ignorées). Tous les arguments et indicateurs autres que **-n** sont affichés.
- n** Empêche l'adjonction d'un caractère nouvelle ligne à la sortie.
- p** Transmet les arguments au tube du process exécuté via |& (barre, perluète), et non à la sortie standard.
- r** Affiche en mode brut. Les conventions d'échappement de la commande **echo** sont ignorées.
- s** Transmet les arguments au fichier d'historique et non à la sortie standard.
- u** Spécifie un numéro de descripteur de fichier *n* de 1 chiffre, dans lequel placer la sortie. La valeur par défaut est 1.

**pwd**Equivalent à **print -r - \$PWD**.

**Remarque** : La commande interne **pwd** du shell Korn ne prend pas en charge les liens symboliques.

**read** [-prsu[n]] [*Nom?Invite*] [*Nom...*]

Lit l'entrée shell. Chaque ligne lue est divisée en zones, séparées par les caractères définis dans la variable **IFS**.

Pour en savoir plus, reportez-vous à la commande **read** dans le manuel *AIX Commands Reference*.

**setgroups**

Exécute la commande **/usr/bin/setgroups** qui fonctionne comme un shell distinct. Pour en savoir plus, reportez-vous à la commande **setgroups**. Il existe toutefois une différence. Si la commande intégrée **setgroups** appelle un sous-shell, la commande **setgroups** remplace le shell en cours d'exécution. Comme la commande intégrée n'est prise en charge que pour une question de compatibilité, les scripts doivent utiliser le chemin d'accès absolu **/usr/bin/setgroups** plutôt que la commande shell intégrée.

**test**

Identique à [*expression*]. Pour en savoir plus, reportez-vous à "Expressions conditionnelles", page 11-49.

## **ulimit [-HSacdfmst] [Limite]**

Définit ou affiche les limites des ressources utilisateur, telles que définies dans le fichier `/etc/security/limits`. Ce fichier contient six limites par défaut :

```
fsize = 2097151
core = 2048
cpu = 3600
data = 131072
rss = 65536
stack = 8192
```

Ces valeurs sont utilisées par défaut lorsqu'un nouvel utilisateur intègre le système. Elles sont définies par la commande **mkuser** lorsque l'utilisateur est ajouté au système, ou modifiées par la commande **chuser**.

Les limites sont d'ordre logiciel ou matériel. Un utilisateur peut, via la commande **ulimit**, modifier une limite logicielle, jusqu'au maximum défini par la limite matérielle. Les limites matérielles ne peuvent être modifiées que par un utilisateur racine.

Nombre de systèmes ne sont pas configurés avec toutes les limites : une ressource est limitée lorsque le paramètre *Limite* correspondant est défini. Ce dernier peut être un chiffre propre à la ressource, ou la valeur `unlimited`. La commande **ulimit** accepte les indicateurs suivants :

- H** Indique qu'une limite matérielle est définie pour cette ressource. Seul l'utilisateur racine est habilité à augmenter une limite matérielle (tout utilisateur peut la diminuer).
- S** Indique qu'une limite logicielle est définie pour cette ressource. Une limite logicielle peut être augmentée à concurrence de la limite matérielle fixée. A défaut d'indicateur **-H** ou **-S**, la limite s'applique aux deux options.
- a** Liste toutes les limites définies.
- c** Nombre de blocs de 512 octets utilisables pour les vidages mémoire.
- d** Taille (en ko) de la zone de données.
- f** Nombre de blocs de 512 octets de fichiers écrits par les process enfants (aucune limite n'est imposée pour la lecture de ces fichiers).
- m** Taille de la mémoire physique (en ko).
- n** Limite du nombre de descripteurs de fichiers pouvant être ouvert par un process.
- s** Taille de la zone pile (en ko).
- t** Temps (en secondes) alloué à chaque process.

Si vous omettez la variable *Limit*, la limite affectée à la ressource courante s'affiche. Il s'agit de la limite logicielle si vous n'avez pas précisé l'indicateur **-H**. Si vous spécifiez plus d'une ressource, le nom et l'unité de la limite sont indiqués avant sa valeur. En l'absence de spécification d'option, l'indicateur **-f** est utilisé par défaut. Si vous changez la valeur et que vous n'avez pas précisé les indicateurs **-H** ou **-S**, définissez les limites matérielles et logicielles.

Pour plus d'informations, reportez-vous à **getrlimit**, **setrlimit** ou **vlimit** dans le manuel *AIX Technical Reference, Volume 1: Base Operating System and Extensions*.

**umask [-S] [Masque]** Détermine les droits d'accès au fichier. Cette valeur, associée aux droits du process créateur, définit les droits sur un fichier, à sa création (valeur par défaut 022). Si Masque est omis, la commande **umask** affiche, sur la sortie standard, le masque de création du mode fichier de l'environnement shell courant.

Pour en savoir plus, reportez-vous à la commande **umask** dans le manuel *AIX Commands Reference*.

**unalias -a**

**unalias NomAlias...** Supprime la définition des alias spécifiés, ou celle de tous les alias si **-a** est spécifié. Ces définitions sont supprimées de l'environnement shell courant.

Pour en savoir plus, reportez-vous à la commande **unalias** dans le manuel *AIX Commands Reference*.

**wait [IDProcess...]** Attend le travail spécifié et se termine. En l'absence de spécification d'un travail, la commande attend tous les process enfants actifs. L'état de sortie de la commande est celui du process attendu.

Pour en savoir plus, reportez-vous à la commande **wait** dans le manuel *AIX Commands Reference*.

**whence [-pv] Nom ...** Indique l'interprétation à donner à chaque nom spécifié, lorsqu'utilisé comme nom de commande. Utilisé sans autre indicateur, **whence** affiche le chemin d'accès absolu, s'il existe, qui correspond à chaque nom.

**-p** Effectue une recherche sur les chemins d'accès au(x) nom(s), même s'il s'agit d'alias, de fonctions ou de mots réservés.

**-v** Génère un état plus fourni (type de chaque nom).

---

## Expressions conditionnelles

Une expression conditionnelle associée à la commande composée `[[` (double crochet) permet de tester les attributs des fichiers et de comparer des chaînes. La substitution de nom de fichier et le fractionnement des mots ne s'appliquent pas aux mots entre `[[` et `]]` (double crochets). Chaque expression est constituée d'une ou de plusieurs expressions binaires ou unaires.

|                                        |                                                                                                                                                                                                                       |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-a Fichier</code>                | Vrai si le fichier spécifié est un lien symbolique pointant sur un fichier existant.                                                                                                                                  |
| <code>-b Fichier</code>                | Vrai si le fichier existe et est un fichier bloc spécial.                                                                                                                                                             |
| <code>-c Fichier</code>                | Vrai si le fichier existe et est un fichier caractère spécial.                                                                                                                                                        |
| <code>-d Fichier</code>                | Vrai si le fichier existe et est un répertoire.                                                                                                                                                                       |
| <code>-e Fichier</code>                | Vrai si le fichier existe.                                                                                                                                                                                            |
| <code>-f Fichier</code>                | Vrai si le fichier existe et est un fichier ordinaire.                                                                                                                                                                |
| <code>-g Fichier</code>                | Vrai si le fichier existe et que le bit <b>setgid</b> est défini.                                                                                                                                                     |
| <code>-h Fichier</code>                | Vrai si le fichier existe et est un lien symbolique.                                                                                                                                                                  |
| <code>-k Fichier</code>                | Vrai si le fichier existe et que son bit collant est défini.                                                                                                                                                          |
| <code>-n Chaîne</code>                 | Vrai si la chaîne est de longueur non nulle.                                                                                                                                                                          |
| <code>-o Option</code>                 | Vrai si l'option spécifiée est active.                                                                                                                                                                                |
| <code>-p Fichier</code>                | Vrai si le fichier existe et qu'il s'agit d'un fichier FIFO spécial ou d'un tube.                                                                                                                                     |
| <code>-r Fichier</code>                | Vrai si le fichier existe et qu'il est lisible par le process courant.                                                                                                                                                |
| <code>-s Fichier</code>                | Vrai si le fichier existe et qu'il est de taille supérieure à 0.                                                                                                                                                      |
| <code>-t<br/>DescripteurFichier</code> | Vrai si le numéro de descripteur de fichier est ouvert et associé à un terminal.                                                                                                                                      |
| <code>-u Fichier</code>                | Vrai si le fichier existe et que le bit <b>setuid</b> est défini.                                                                                                                                                     |
| <code>-w Fichier</code>                | Vrai si le fichier existe et que le bit d'écriture est activé. Le fichier peut toutefois ne pas être inscriptible s'il se trouve sur un système de fichiers en lecture seule, même si le test renvoie la valeur vrai. |
| <code>-x Fichier</code>                | Vrai si le fichier existe et que l'indicateur <b>execute</b> est défini. S'il existe et qu'il s'agit d'un répertoire, le process courant est autorisé à l'explorer.                                                   |
| <code>-z Chaîne</code>                 | Vrai si la chaîne est de longueur nulle.                                                                                                                                                                              |
| <code>-L Fichier</code>                | Vrai si le fichier existe et est un lien symbolique.                                                                                                                                                                  |
| <code>-O Fichier</code>                | Vrai si le fichier existe et qu'il est la propriété de l'ID utilisateur effectif de ce process.                                                                                                                       |
| <code>-G Fichier</code>                | Vrai si le fichier existe et qu'il est la propriété de l'ID groupe effectif de ce process.                                                                                                                            |
| <code>-S Fichier</code>                | Vrai si le fichier existe et qu'il s'agit d'un socket.                                                                                                                                                                |
| <code>Fichier1 -nt<br/>Fichier2</code> | Vrai si <i>Fichier1</i> existe et qu'il est plus récent que <i>Fichier2</i> .                                                                                                                                         |
| <code>Fichier1 -ot<br/>Fichier2</code> | Vrai si <i>Fichier1</i> existe et qu'il est moins récent que <i>Fichier2</i> .                                                                                                                                        |
| <code>Fichier1 -ef<br/>Fichier2</code> | Vrai si <i>Fichier1</i> et <i>Fichier2</i> existent et qu'ils font référence au même fichier.                                                                                                                         |
| <code>Chaîne1 = Chaîne2</code>         | Vrai si <i>Chaîne1</i> est égale à <i>Chaîne2</i> .                                                                                                                                                                   |

|                                        |                                                                                                 |
|----------------------------------------|-------------------------------------------------------------------------------------------------|
| <i>Chaîne1 !=<br/>Chaîne2</i>          | Vrai si <i>Chaîne1</i> n'est pas égale à <i>Chaîne2</i> .                                       |
| <i>Chaîne = Trame</i>                  | Vrai si la chaîne correspond à la trame.                                                        |
| <i>Chaîne != Trame</i>                 | Vrai si la chaîne ne correspond pas à la trame.                                                 |
| <i>Chaîne1 &lt; Chaîne2</i>            | Vrai si <i>Chaîne1</i> précède <i>Chaîne2</i> , au sens de la valeur ASCII de leurs caractères. |
| <i>Chaîne1 &gt; Chaîne2</i>            | Vrai si <i>Chaîne1</i> suit <i>Chaîne2</i> , au sens de la valeur ASCII de leurs caractères.    |
| <i>Expression1 -eq<br/>Expression2</i> | Vrai si <i>Expression1</i> est égale à <i>Expression2</i> .                                     |
| <i>Expression1 -ne<br/>Expression2</i> | Vrai si <i>Expression1</i> n'est pas égale à <i>Expression2</i> .                               |
| <i>Expression1 -lt<br/>Expression2</i> | Vrai si <i>Expression1</i> est inférieure à <i>Expression2</i> .                                |
| <i>Expression1 -gt<br/>Expression2</i> | Vrai si <i>Expression1</i> est supérieure à <i>Expression2</i> .                                |
| <i>Expression1 -le<br/>Expression2</i> | Vrai si <i>Expression1</i> est inférieure ou égale à <i>Expression2</i> .                       |
| <i>Expression1 -ge<br/>Expression2</i> | Vrai si <i>Expression1</i> est supérieure ou égale à <i>Expression2</i> .                       |

**Remarque :** Dans les expressions ci-dessus, si la variable *Fichier* est semblable à */dev/fd/n*, *n* étant un entier, le test s'applique au fichier ouvert dont le numéro de descripteur est *n*.

Vous pouvez créer une expression composée à partir de ces primitives en vous servant de l'une des expressions suivantes.

|                                               |                                                                                  |
|-----------------------------------------------|----------------------------------------------------------------------------------|
| <i>(Expression)</i>                           | Vrai si l'expression spécifiée est vraie. Utilisée pour grouper des expressions. |
| <i>! Expression</i>                           | Vrai si l'expression spécifiée est fausse.                                       |
| <i>Expression1 &amp;&amp;<br/>Expression2</i> | Vrai si <i>Expression1</i> et <i>Expression2</i> sont simultanément vraies.      |
| <i>Expression1   <br/>Expression2</i>         | Vrai si <i>Expression1</i> ou <i>Expression2</i> est vraie.                      |

---

## Contrôle des travaux (shell Korn ou POSIX)

Le shell Korn ou POSIX offre une fonction de gestion des séquences de commandes, ou travaux (*jobs*). Lorsque vous exécutez la commande spéciale **set -m**, le shell Korn associe un travail à chaque pipeline. Il maintient une table des travaux courants, affichée via la commande **jobs**, et leur affecte des nombres entiers de faible valeur.

Lorsqu'un travail est lancé en arrière-plan avec un caractère **&** (perluète), le shell affiche une ligne du type :

```
[1] 1234
```

indiquant que le travail, en arrière-plan, porte le numéro 1 et est constitué d'un process (de haut niveau) dont l'ID est 1234.

Lorsqu'un travail est en cours et que vous souhaitez exécuter une autre tâche, appuyez sur les touches Ctrl-Z : un signal **STOP** est envoyé au travail en cours. Le shell indique alors que le travail a été arrêté et affiche une invite. Vous pouvez alors intervenir sur l'état du travail (par exemple, le placer en arrière-plan via la commande **bg**), exécuter d'autres commandes et, éventuellement, le rappeler en avant-plan via la commande **fg**. La séquence Ctrl-Z agit immédiatement et est interprétée comme un arrêt par le shell – qui rejette alors toutes les sorties en attente et les entrées non lues.

Un travail exécuté en arrière-plan s'arrête dès qu'il tente de lire à partir du terminal. Les travaux d'arrière-plan génèrent habituellement une sortie : cette option peut être désactivée via la commande **stty tostop**. Si vous l'activez, les travaux en arrière-plan sont interrompus dès qu'ils tentent de générer une sortie ou de lire une entrée.

Vous pouvez référencer un travail dans le shell Korn de plusieurs façons : par son ID process (PID), mais aussi par :

|                  |                                                                                          |
|------------------|------------------------------------------------------------------------------------------|
| <b>%Numéro</b>   | Référence le travail au numéro donné.                                                    |
| <b>%Chaîne</b>   | Référence les travaux dont la ligne de commande commence par la variable <i>Chaîne</i> . |
| <b>% ?Chaîne</b> | Référence les travaux dont la ligne de commande contient la variable <i>Chaîne</i> .     |
| <b>%%</b>        | Référence le travail courant.                                                            |
| <b>%+</b>        | Equivalent à %%.                                                                         |
| <b>%-</b>        | Référence le travail précédent.                                                          |

Le shell identifie immédiatement tout changement d'état d'un processus. Il vous informe dès qu'un travail est bloqué, empêchant toute progression du traitement. Il vous informe avant même d'afficher une invite, afin de ne pas perturber votre travail.

Lorsque le mode Monitor est activé, chaque travail d'arrière-plan achevé déclenche les traitements d'interruption définis pour le **CHLD**.

Si vous tentez de quitter le shell (commande `exit` ou séquence Ctrl-D) alors que des travaux sont arrêtés ou en cours, le système affiche le message : `Travaux arrêtés. Travaux en cours d'exécution`. Lancez la commande **jobs** pour prendre connaissance de ces travaux. Si vous tentez encore de quitter le shell, il met fin aux travaux en cours et arrêtés, sans nouvel avertissement.

## Traitement des signaux

Les signaux **SIGINT** et **SIGQUIT** d'une commande appelée sont ignorés si la commande est suivie d'une **&** (perluète) et que l'option **monitor** est inactive. Sinon, ils prennent les valeurs héritées du shell parent.

Si un signal, pour lequel un traitement d'interruption a été défini, est reçu pendant que le shell attend l'achèvement d'une commande en arrière-plan, le traitement d'interruption n'est exécuté qu'une fois la commande terminée. Ainsi, un traitement d'interruption sur un signal **CHILD** n'est exécuté qu'une fois le travail en arrière-plan terminé.

- Voir aussi Shell Korn

---

## Edition en ligne (shell Korn ou POSIX)

Pour entrer vos commandes, vous tapez une ligne de commande à partir d'un terminal et la faites suivre d'un caractère de nouvelle ligne (RETOUR ou LIGNE SUIVANTE). Si vous activez une option d'édition en ligne (emacs, gmacs ou vi), vous pouvez modifier cette ligne de commande.

Voici les commandes d'appel des éditeurs :

|                     |                                                                           |
|---------------------|---------------------------------------------------------------------------|
| <b>set -o emacs</b> | Appelle le mode d'édition emacs et ouvre un éditeur en ligne style emacs. |
| <b>set -o gmacs</b> | Appelle le mode d'édition emacs et ouvre un éditeur en ligne style gmacs. |
| <b>set -o vi</b>    | Appelle le mode d'édition vi et ouvre un éditeur en ligne style vi.       |

Une option d'édition est automatiquement sélectionnée chaque fois que la valeur affectée à **VISUAL** ou à **EDITOR** se termine par l'un de ces noms d'option.

**Remarque :** Pour accéder aux fonctions d'édition, votre terminal doit accepter RETOUR comme retour chariot sans passage à la ligne et un espace doit remplacer (et non s'insérer avant) le caractère courant. Un espace doit remplacer (et non s'insérer avant) le caractère courant.

Chaque mode d'édition ouvre une fenêtre au niveau de la ligne courante. Sa longueur est définie par la variable **COLUMNS** ; ou, à défaut, égale à 80 espaces caractères. Si la ligne dépasse la longueur de fenêtre diminuée de deux caractères, le système affiche une marque à l'extrémité de la fenêtre. Lorsque le curseur atteint les limites de la fenêtre, celle-ci se recentre sur le curseur. Les marques affichées sont :

|   |                                                    |
|---|----------------------------------------------------|
| > | La ligne se prolonge à droite de la fenêtre.       |
| < | La ligne se prolonge à gauche de la fenêtre.       |
| * | La ligne se prolonge des deux côtés de la fenêtre. |

Les commandes de recherche de chaque mode d'édition donnent accès au fichier d'historique du shell Korn. Seules les chaînes sont mises en correspondance. Si le premier caractère de la chaîne est un ^ (caret), la recherche doit débuter au premier caractère de la ligne.

Cette section traite des points suivants :

- Mode d'édition emacs, page 11-53.
- Mode d'édition vi, page 11-57.
- Voir aussi Shell Korn, on page 11-67

### Mode d'édition emacs

Pour sélectionner le mode emacs, activez indifféremment l'option **emacs** ou l'option **gmacs** : les deux modes ne se différencient que par leur interprétation de la commande Ctrl-T. Pour éditer une ligne, positionnez le curseur et effectuez les corrections requises. Toutes les commandes d'édition sont des caractères de contrôle ou des séquences d'échappement.

Les commandes d'édition opèrent n'importe où sur la ligne (et pas seulement au début). Sauf indication contraire, n'appuyez ni sur Entrée ni sur la touche ligne suivante (flèche vers le bas) après une commande d'édition.

|                       |                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Ctrl-F</b>         | Avance le curseur d'un caractère.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Echap-F</b>        | Avance le curseur au mot suivant (un mot est une chaîne composée exclusivement de lettres, chiffres et traits de soulignement).                                                                                                                                                                                                                                                                                         |
| <b>Ctrl-B</b>         | Reculé le curseur d'un caractère.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Echap-B</b>        | Ramène le curseur au mot précédent.                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Ctrl-A</b>         | Amène le curseur au début de la ligne.                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Ctrl-E</b>         | Avance le curseur à la fin de la ligne.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Ctrl-] c</b>       | Avance le curseur au caractère indiqué.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Echap-Ctrl-] c</b> | Avance le curseur au caractère indiqué.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Ctrl-X Ctrl-X</b>  | Echange curseur et marque.                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>ERASE</b>          | Efface le caractère précédent. (Caractère d'effacement défini par l'utilisateur, via la commande <b>stty</b> , généralement la séquence Ctrl-H).                                                                                                                                                                                                                                                                        |
| <b>Ctrl-D</b>         | Efface le caractère courant.                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Echap-D</b>        | Supprime le mot courant.                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Echap-RetAr</b>    | Efface le mot précédent.                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Echap-H</b>        | Efface le mot précédent.                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Echap-Suppr</b>    | Efface le mot précédent. Si la touche Suppression est définie comme touche d'interruption, cette commande ne fonctionne pas.                                                                                                                                                                                                                                                                                            |
| <b>Ctrl-T</b>         | Transpose le caractère courant et le caractère suivant (mode emacs). Transpose les deux caractères précédents (mode gmacs).                                                                                                                                                                                                                                                                                             |
| <b>Ctrl-C</b>         | Met en majuscules le caractère courant.                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Echap-C</b>        | Met en majuscules le mot courant.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Echap-L</b>        | Met en minuscules le mot courant.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Ctrl-K</b>         | Efface la zone entre le curseur et la fin de la ligne. Précédé d'un paramètre numérique dont la valeur est inférieure à celle de la position courante du curseur, efface la zone entre la position indiquée par ce paramètre et le curseur. Précédé d'un paramètre numérique dont la valeur est supérieure à la position courante du curseur, efface la zone entre la position indiquée par ce paramètre et le curseur. |
| <b>Ctrl-W</b>         | Efface la zone entre le curseur et la marque.                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Echap-P</b>        | Repousse dans la pile la zone entre le curseur et la marque.                                                                                                                                                                                                                                                                                                                                                            |
| <b>KILL</b>           | Caractère défini par l'utilisateur via la commande <b>stty</b> , généralement la séquence Ctrl-G ou le caractère @ (arobas). Tue l'intégralité de la ligne courante. Une succession de deux caractères kill génère un saut de ligne à chaque nouveau caractère kill (utile sur un terminal papier).                                                                                                                     |
| <b>Ctrl-Y</b>         | Restaure le dernier élément supprimé de la ligne. (Restitue l'élément sur la ligne.)                                                                                                                                                                                                                                                                                                                                    |
| <b>Ctrl-L</b>         | Effectue un saut de ligne et affiche la ligne courante.                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Ctrl-@</b>         | (Caractère nul) Définit une marque.                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Echap-Espace</b>   | Définit une marque.                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Ctrl-J</b>         | (Nouvelle ligne) Exécute la ligne courante.                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Ctrl-M</b>         | (Retour) Exécute la ligne courante.                                                                                                                                                                                                                                                                                                                                                                                     |

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>EOF</b>                   | Traite le caractère fin de fichier (généralement, Ctrl-D) comme tel, uniquement si la ligne courante est nulle.                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Ctrl-P</b>                | Appelle la commande précédente. Il s'agit d'une opération chronologique. Revient à la ligne précédente dans une commande de plusieurs lignes (si le curseur ne se trouve pas sur la première ligne).                                                                                                                                                                                                                                                                                                                          |
| <b>Echap-&lt;</b>            | Appelle la ligne d'historique la moins récente.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>Echap-&gt;</b>            | Appelle la ligne d'historique la plus récente.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Ctrl-N</b>                | Appelle la commande suivante. Il s'agit d'une opération chronologique.                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>Ctrl-R <i>Chaîne</i></b>  | Inverse l'historique de recherche d'une commande antérieure contenant la <i>Chaîne</i> . Si la valeur spécifiée est zéro, la recherche s'effectue vers l'avant. La chaîne se termine par Entrée ou un caractère de nouvelle ligne. Si elle est précédée d'un caret (^), la ligne correspondante doit commencer par <i>Chaîne</i> . Si <i>Chaîne</i> est omis, la recherche porte sur la ligne de commande contenant le paramètre <i>Chaîne</i> le plus récent. Dans ce cas, la valeur 0 inverse la direction de la recherche. |
| <b>Ctrl-O</b>                | Exécute la ligne courante et extrait du fichier d'historique la ligne suivant la ligne courante.                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Echap <i>Chiffres</i></b> | Définit un paramètre numérique. Les chiffres sont considérés comme paramètre de la commande suivante. Les commandes admettant un paramètre sont <b>Ctrl-F</b> , <b>Ctrl-B</b> , <b>ERASE</b> , <b>Ctrl-C</b> , <b>Ctrl-D</b> , <b>Ctrl-K</b> , <b>Ctrl-R</b> , <b>Ctrl-P</b> , <b>Ctrl-N</b> , <b>Ctrl-]</b> , <b>Echap-</b> , <b>Echap-Ctrl-]</b> , <b>Echap-</b> , <b>Echap-B</b> , <b>Echap-C</b> , <b>Echap-D</b> , <b>Echap-F</b> , <b>Echap-H</b> , <b>Echap-L</b> et <b>Echap-Ctrl-H</b> .                             |
| <b>Echap <i>Lettre</i></b>   | (Touche de fonction) Recherche l'alias <i>_Lettre</i> dans la liste des alias. Si la recherche aboutit, la valeur de l'alias trouvé est placée dans la file d'entrée. Le paramètre <i>Lettre</i> ne doit pas spécifier de fonctions d'échappement (escape).                                                                                                                                                                                                                                                                   |
| <b>Echap-[ <i>Lettre</i></b> | (Touche de fonction) Recherche l'alias <i>_Lettre</i> (double trait de soulignement) dans la liste des alias. Si la recherche aboutit, la valeur de l'alias trouvé est placée dans la file d'entrée. Cette commande permet de programmer des touches de fonction sur plusieurs terminaux.                                                                                                                                                                                                                                     |
| <b>Echap-</b>                | Insère dans la ligne le dernier mot de la commande précédente. Précédée d'un paramètre numérique, la valeur de ce paramètre détermine quel mot insérer à la place du dernier mot.                                                                                                                                                                                                                                                                                                                                             |
| <b>Echap-._</b>              | Identique à Echap-. (Echap, tiret, point).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Echap-*</b>               | Tente d'effectuer la substitution de nom sur le mot courant. Lorsque le mot ne correspond à aucun fichier ou contient des caractères spéciaux, un astérisque est ajouté.                                                                                                                                                                                                                                                                                                                                                      |
| <b>Echap-Echap</b>           | Traitement de nom de fichier. Remplace le mot courant par le préfixe commun le plus long de tous les noms de fichier correspondant au mot courant doté d'un astérisque. S'il n'y a qu'une seule correspondance, une barre oblique (/) est ajoutée s'il s'agit d'un répertoire, ou un espace sinon.                                                                                                                                                                                                                            |
| <b>Echap-=.</b>              | Liste les fichiers qui correspondent à la trame du mot courant comme si un astérisque avait été ajouté.                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Ctrl-U</b>                | Multiplie par 4 le paramètre de la commande suivante.                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

|                |                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| \              | Annule l'effet du caractère suivant. Les caractères d'édition et les caractères ERASE, KILL et INTERRUPT (habituellement la touche Suppr) peuvent être entrés sur une ligne de commande ou de recherche sous réserve d'être précédés d'une barre oblique inverse (\). Cette dernière annule les caractéristiques d'édition du caractère suivant (le cas échéant). |
| <b>Ctrl-V</b>  | Affiche la version du shell.                                                                                                                                                                                                                                                                                                                                      |
| <b>Echap-#</b> | Insère un signe # (dièse) au début de la ligne et exécute celle-ci. Cela entraîne l'insertion d'un commentaire dans le fichier historique.                                                                                                                                                                                                                        |

## Mode d'édition vi

L'éditeur vi offre deux modes de saisie : le mode Entrée (entrée des commandes) et le mode Contrôle (modification des commandes) appelé par la touche Echap.

La plupart des commandes de contrôle admettent un paramètre *Compte* de répétition, en option, avant la commande. En mode vi, sur la plupart des systèmes, le traitement normal est d'abord activé. La commande est réaffichée lorsque :

- le vitesse est supérieure ou égale à 1200 ;
- la commande contient des caractères de contrôle ;
- moins d'une seconde s'est écoulée depuis l'affichage de l'invite.

Le caractère Echap interrompt le traitement normal pour le reste de la commande, ce qui permet de modifier la ligne de commande. Ce schéma offre les avantages du traitement canonique avec écho du mode brut. Si l'option **viraw** est spécifiée, le traitement canonique est désactivé. Ce mode est le mode par défaut pour les systèmes qui n'admettent qu'un seul délimiteur de fin de ligne, et peut être pratique pour certains terminaux.

Les commandes vi sont regroupées par catégorie :

- entrée,
- déplacement,
- recherche,
- modification de texte,
- divers.

## Commandes d'entrée

**Remarque** : Par défaut, l'éditeur s'ouvre en mode Entrée.

|               |                                                                                                                                                                                                                                                                                                                 |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ERASE</b>  | (Caractère d'effacement défini par l'utilisateur, via la commande <b>stty</b> , généralement la séquence Ctrl-H ou #). Efface le caractère précédent.                                                                                                                                                           |
| <b>Ctrl-W</b> | Efface le mot précédent séparé par un blanc.                                                                                                                                                                                                                                                                    |
| <b>Ctrl-D</b> | Met fin au shell.                                                                                                                                                                                                                                                                                               |
| <b>Ctrl-V</b> | Annule l'effet du caractère suivant. Les caractères d'édition, ERASE ou KILL, par exemple, peuvent être entrés sur une ligne de commande ou de recherche sous réserve d'être précédés de la séquence Ctrl-V. La combinaison Ctrl-V annule les caractéristiques d'édition du caractère suivant (le cas échéant). |
| <b>\</b>      | Annule l'effet du caractère ERASE ou KILL suivant.                                                                                                                                                                                                                                                              |

## Commandes de déplacement

Ces commandes servent à déplacer le curseur :

|                  |                                                           |
|------------------|-----------------------------------------------------------|
| <b>[Compte]l</b> | Avance le curseur d'un caractère.                         |
| <b>[Compte]w</b> | Avance le curseur d'un mot (alphanumérique).              |
| <b>[Compte]W</b> | Avance le curseur au début du mot suivant après un blanc. |
| <b>[Compte]e</b> | Avance le curseur à la fin du mot courant.                |
| <b>[Compte]E</b> | Avance le curseur à la fin du mot courant.                |
| <b>[Compte]h</b> | Reculé le curseur d'un caractère.                         |
| <b>[Compte]b</b> | Ramène le curseur au mot précédent.                       |
| <b>[Compte]B</b> | Ramène le curseur au mot séparé par un blanc précédent.   |

|                     |                                                                                                                                                          |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| [ <i>Compte</i> ]]  | Amène le curseur à la colonne spécifiée par <i>Compte</i> .                                                                                              |
| [ <i>Compte</i> ]fc | Trouve le caractère <i>c</i> suivant de la ligne courante.                                                                                               |
| [ <i>Compte</i> ]Fc | Trouve le caractère <i>c</i> précédent de la ligne courante.                                                                                             |
| [ <i>Compte</i> ]tc | Equivalent à <b>f</b> suivi de <b>h</b> .                                                                                                                |
| [ <i>Compte</i> ]Tc | Equivalent à <b>F</b> suivi de <b>l</b> .                                                                                                                |
| [ <i>Compte</i> ];  | Répète, le nombre de fois spécifié par <i>Compte</i> , la dernière commande de recherche sur un caractère : <b>f</b> , <b>F</b> , <b>t</b> ou <b>T</b> . |
| [ <i>Compte</i> ],  | Inverse, le nombre de fois spécifié par <i>Count</i> , la dernière commande de recherche sur un caractère.                                               |
| <b>0</b>            | Ramène le curseur au début de la ligne.                                                                                                                  |
| <b>^</b>            | Amène le curseur au premier caractère non blanc de la ligne.                                                                                             |
| <b>\$</b>           | Amène le curseur à la fin de la ligne.                                                                                                                   |

## Commandes de recherche

Ces commandes donnent accès au fichier d'historique des commandes :

|                    |                                                                                                                                                                                                                                                                                                                                          |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [ <i>Compte</i> ]k | Extrait la commande précédente.                                                                                                                                                                                                                                                                                                          |
| [ <i>Compte</i> ]– | Equivalent à la commande <b>k</b> .                                                                                                                                                                                                                                                                                                      |
| [ <i>Compte</i> ]j | Extrait la commande suivante. Entrer la commande <b>j</b> donne accès à la commande suivante.                                                                                                                                                                                                                                            |
| [ <i>Compte</i> ]+ | Equivalent à la commande <b>j</b> .                                                                                                                                                                                                                                                                                                      |
| [ <i>Compte</i> ]G | Extrait la commande dont le numéro est spécifié par <i>Compte</i> . Il s'agit par défaut de la commande la moins récente de l'historique.                                                                                                                                                                                                |
| <i>/Chaîne</i>     | Recherche en amont dans l'historique d'une commande contenant <i>Chaîne</i> . La chaîne se termine par Entrée ou un caractère de nouvelle ligne. Si la chaîne spécifiée est précédée d'un ^ (caret ), la ligne correspondante doit commencer par <i>Chaîne</i> . Si <i>Chaîne</i> a une valeur nulle, la chaîne précédente est utilisée. |
| ? <i>Chaîne</i>    | Identique à <i>/Chaîne</i> , à ceci près que la recherche s'effectue vers l'aval.                                                                                                                                                                                                                                                        |
| <b>n</b>           | Recherche l'occurrence suivante de la dernière trame des commandes <i>/Chaîne</i> ou <b>?</b>                                                                                                                                                                                                                                            |
| <b>N</b>           | Recherche l'occurrence suivante de la dernière trame des commandes <i>/Chaîne</i> ou <b>?</b> , mais dans la direction opposée. Explore l'historique à la recherche de la chaîne entrée via la commande <i>/Chaîne</i> précédente.                                                                                                       |

## Commandes de modification de texte

Ces commandes permettent de modifier la ligne de commande :

|                                                            |                                                                                                                                                                                                                                        |
|------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>a</b>                                                   | Ouvre le mode Entrée et insère le texte après le caractère courant.                                                                                                                                                                    |
| <b>A</b>                                                   | Ajoute le texte à la fin de la ligne. Equivalent à la commande <b>\$a</b> .                                                                                                                                                            |
| [ <i>Compte</i> ]c<br><i>Déplacement</i>                   |                                                                                                                                                                                                                                        |
| <b>c</b> [ <i>Compte</i> ] <i>Déplacement</i><br><i>nt</i> | Efface la zone entre le caractère courant et la caractère spécifié par <i>Déplacement</i> , et ouvre le mode Entrée. Si <i>Déplacement</i> a la valeur <b>c</b> , l'intégralité de la ligne est supprimée (le mode Entrée est ouvert). |

|          |                                                                                                                                 |
|----------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>C</b> | Efface la zone entre le caractère courant et la fin de la ligne, et ouvre le mode Entrée. Equivalent à la commande <b>c\$</b> . |
| <b>S</b> | Equivalent à la commande <b>cc</b> .                                                                                            |
| <b>D</b> | Efface la zone entre le caractère courant et la fin de la ligne. Equivalent à la commande <b>d\$</b> .                          |

#### [[*Compte*]d*Déplacement*

|                                                  |                                                                                                                                                                                                                                                                                                                   |
|--------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>d</b> [ <i>Compte</i> ]<br><i>Déplacement</i> | Efface la zone entre le caractère courant et le caractère spécifié par <i>Déplacement</i> (ce caractère compris). Si <i>Déplacement</i> a la valeur <b>d</b> , l'intégralité de la ligne est supprimée.                                                                                                           |
| <b>i</b>                                         | Ouvre le mode Entrée et insère le texte avant le caractère courant.                                                                                                                                                                                                                                               |
| <b>I</b>                                         | Insère le texte au début de la ligne. Equivalent à la commande <b>Oi</b> .                                                                                                                                                                                                                                        |
| [ <i>Compte</i> ]P                               | Place la dernière modification de texte avant le curseur.                                                                                                                                                                                                                                                         |
| [ <i>Compte</i> ]p                               | Place la dernière modification de texte après le curseur.                                                                                                                                                                                                                                                         |
| <b>R</b>                                         | Ouvre le mode Entrée et remplace les caractères affichés à l'écran.                                                                                                                                                                                                                                               |
| [ <i>Compte</i> ]rc                              | Remplace le nombre de caractères spécifiés par <i>Compte</i> , à partir de la position du curseur, par les caractères spécifiés par <i>c</i> . Cette commande fait ensuite avancer le curseur.                                                                                                                    |
| [ <i>Compte</i> ]x                               | Efface le caractère courant.                                                                                                                                                                                                                                                                                      |
| [ <i>Compte</i> ]X                               | Efface le caractère précédent.                                                                                                                                                                                                                                                                                    |
| [ <i>Compte</i> ].                               | Répète la dernière modification de texte.                                                                                                                                                                                                                                                                         |
| [[ <i>Compte</i> ]~                              | Convertit les majuscules en minuscules (et réciproquement) le nombre de caractères spécifié par <i>Compte</i> , à partir de la position du curseur, et avance le curseur.                                                                                                                                         |
| [ <i>Compte</i> ]                                | Ajoute le mot spécifié par le paramètre <i>Compte</i> de la commande précédente et ouvre le mode Entrée. Il s'agit du dernier mot si <i>Compte</i> est omis.                                                                                                                                                      |
| *                                                | Ajoute un * (astérisque) au mot courant et tente une substitution de nom de fichier. Si la recherche n'aboutit pas, le système le signale. Sinon, le mot est remplacé par la trame correspondante et le mode Entrée s'ouvre.                                                                                      |
| \                                                | Traitement de nom de fichier. Remplace le mot courant par le préfixe commun le plus long de tous les noms de fichier correspondant au mot courant doté d'un astérisque. S'il n'y a qu'une occurrence, une barre oblique (/) est ajoutée s'il s'agit d'un répertoire. Dans le cas contraire, un espace est ajouté. |

## Commandes diverses

Cette catégorie regroupe les commandes :

[*Compte*]y  
*Déplacement*

y[*Compte*]  
*Déplacement*

**Y**

Sélectionne la zone entre le caractère courant et le caractère dont la position est spécifiée par *Déplacement* (ce caractère compris), et place le tout dans le tampon de suppression. Texte et curseur restent inchangés.

Sélectionne la zone entre le caractère courant et la fin de la ligne. Equivalent à la commande **y\$**.

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>u</b>         | Annule la dernière commande de modification de texte.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>U</b>         | Annule toutes les commandes de modification de texte exécutées sur la ligne.                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>[Compte]v</b> | Renvoie la commande <code>fc -e \${VISUAL:-\${EDITOR:-vi}}</code> <code>Compte</code> dans le tampon d'entrée. Si <code>Compte</code> est omis, la ligne courante est utilisée.                                                                                                                                                                                                                                                                                       |
| <b>Ctrl-L</b>    | Effectue un saut de ligne et affiche la ligne courante. Cette commande ne s'applique qu'en mode Contrôle.                                                                                                                                                                                                                                                                                                                                                             |
| <b>Ctrl-J</b>    | (Nouvelle ligne) Exécute la ligne courante, quel que soit le mode en vigueur.                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Ctrl-M</b>    | (Retour) Exécute la ligne courante, quel que soit le mode en vigueur.                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>#</b>         | Envoie la ligne en la faisant précéder d'un <b>#</b> (dièse). Utile pour insérer la ligne courante dans l'historique sans l'exécuter.<br><br>Si la ligne de commande comporte une barre verticale ( <code> </code> ), un point-virgule ou un caractère nouvelle ligne, autant de <b>#</b> (dièse) complémentaires sont insérés avant ces symboles. Pour effacer tous les dièses, rappelez la ligne de commande à partir de l'historique et entrez un autre <b>#</b> . |
| <b>=</b>         | Liste les noms de fichier correspondant au mot courant comme s'il était doté d'un astérisque.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>@Lettre</b>   | Recherche l'alias <code>_Lettre</code> dans la liste des alias. Si la recherche aboutit, la valeur de l'alias trouvé est placée dans la file d'entrée, en vue de son traitement.                                                                                                                                                                                                                                                                                      |

---

# Liste des commandes intégrées du shell Korn ou POSIX

## Commandes intégrées spéciales

|                        |                                                                                                                                        |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>:</b> (deux points) | Ne développe que les arguments.                                                                                                        |
| <b>.</b> (point)       | Lit un fichier et en exécute les commandes.                                                                                            |
| <b>break</b>           | Quitte, le cas échéant, la boucle <b>for</b> , <b>while</b> , <b>until</b> ou <b>select</b> .                                          |
| <b>continue</b>        | Reprend à l'itération suivante de la boucle <b>for</b> , <b>while</b> , <b>until</b> ou <b>select</b> .                                |
| <b>eval</b>            | Lit les arguments comme entrée et exécute la ou les commandes qui en résultent.                                                        |
| <b>exec</b>            | Exécute, à la place du shell, la commande spécifiée par <i>Argument</i> , sans créer de nouveau process.                               |
| <b>exit</b>            | Quitte le shell dont l'état de sortie est spécifié par le paramètre <i>n</i> .                                                         |
| <b>export</b>          | Marque les noms spécifiés, pour exportation automatique vers l'environnement des commandes ultérieures.                                |
| <b>newgrp</b>          | Equivalent à la commande <b>exec /usr/bin/newgrp [Groupe ...]</b> .                                                                    |
| <b>readonly</b>        | Marque en lecture seule les noms spécifiés.                                                                                            |
| <b>return</b>          | Provoque le renvoi d'une fonction shell au script appelant.                                                                            |
| <b>set</b>             | Sauf options ou arguments contraires, écrit les noms et les valeurs des variables shell dans l'ordre défini par l'environnement local. |
| <b>shift</b>           | Renomme les paramètres positionnels.                                                                                                   |
| <b>times</b>           | Affiche les temps utilisateur et système cumulés pour le shell et les process exécutés à partir du shell.                              |
| <b>trap</b>            | Exécute une commande donnée lorsque le shell reçoit un ou plusieurs signaux déterminés.                                                |
| <b>typeset</b>         | Définit les attributs et les valeurs des paramètres shell.                                                                             |
| <b>unset</b>           | Annule les définitions des paramètres et les valeurs spécifiées.                                                                       |

## Commandes intégrées standard

|                |                                                                                                                                                             |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>alias</b>   | Imprime une liste d'alias sur la sortie standard.                                                                                                           |
| <b>bg</b>      | Place en arrière-plan les travaux spécifiés.                                                                                                                |
| <b>cd</b>      | Passe au répertoire spécifié ou remplace la chaîne courante par la chaîne spécifiée.                                                                        |
| <b>echo</b>    | Ecrit les chaînes de caractères sur la sortie standard.                                                                                                     |
| <b>fc</b>      | Sélectionne un ensemble de commandes à partir de la dernière commande HISTSIZE lancée au terminal. Réexécute la commande après substitution ancien-nouveau. |
| <b>fg</b>      | Amène à l'avant-plan le travail spécifié.                                                                                                                   |
| <b>getopts</b> | Contrôle les options légales du paramètres <i>Argument</i> .                                                                                                |
| <b>jobs</b>    | Affiche des informations sur les travaux spécifiés.                                                                                                         |
| <b>kill</b>    | Envoie le signal <b>TERM</b> (terminate) aux travaux ou aux process spécifiés.                                                                              |
| <b>let</b>     | Evalue les expressions arithmétiques spécifiées.                                                                                                            |
| <b>print</b>   | Imprime la sortie shell.                                                                                                                                    |

|                |                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>pwd</b>     | Equivalent à la commande <b>print -r -r \$PWD</b> .                                                                          |
| <b>read</b>    | Lit l'entrée shell.                                                                                                          |
| <b>ulimit</b>  | Définit ou affiche les limites des ressources utilisateur, telles que définies dans le fichier <b>/etc/security/limits</b> . |
| <b>umask</b>   | Détermine les droits d'accès au fichier.                                                                                     |
| <b>unalias</b> | Supprime de la liste des alias les paramètres correspondant à la liste de noms spécifiée.                                    |
| <b>wait</b>    | Attend le travail spécifié et quitte.                                                                                        |
| <b>whence</b>  | Indique l'interprétation à donner à chaque nom spécifié, lorsqu'utilisé comme nom de commande.                               |

Pour en savoir plus, reportez-vous à Commandes intégrées (shell Korn ou POSIX).

---

## Liste des commandes intégrées du shell Bourne

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| <b>:</b>        | Renvoie un état de sortie nul.                                                                                |
| <b>.</b>        | Lit et exécute les commandes d'un fichier et revient au shell.                                                |
| <b>break</b>    | Sort, le cas échéant, de la boucle <b>for</b> , <b>while</b> ou <b>until</b> .                                |
| <b>cd</b>       | Passe au répertoire spécifié.                                                                                 |
| <b>continue</b> | Reprend à l'itération suivante de la boucle <b>for</b> , <b>while</b> ou <b>until</b> .                       |
| <b>echo</b>     | Ecrit les chaînes de caractères sur la sortie standard.                                                       |
| <b>eval</b>     | Lit les arguments comme entrée et exécute la ou les commandes qui en résultent.                               |
| <b>exec</b>     | Exécute, à la place du shell, la commande spécifiée par <i>Argument</i> , sans créer de nouveau process.      |
| <b>exit</b>     | Quitte le shell dont l'état de sortie est spécifié par le paramètre <i>n</i> .                                |
| <b>export</b>   | Marque les noms spécifiés, pour exportation automatique vers l'environnement des commande ultérieures.        |
| <b>hash</b>     | Recherche et mémorise l'emplacement du chemin d'accès aux commandes spécifiées.                               |
| <b>pwd</b>      | Affiche le répertoire courant.                                                                                |
| <b>read</b>     | Lit une ligne de l'entrée standard.                                                                           |
| <b>readonly</b> | Marque en lecture seule le <i>Nom</i> spécifié.                                                               |
| <b>return</b>   | Provoque la fin d'une fonction, avec une valeur de retour donnée.                                             |
| <b>set</b>      | Contrôle l'affichage de divers paramètres sur la sortie standard.                                             |
| <b>shift</b>    | Décale vers la gauche les arguments de la ligne de commande.                                                  |
| <b>test</b>     | Evalue des expressions conditionnelles.                                                                       |
| <b>times</b>    | Affiche les temps cumulés utilisateur et système, pour le shell et les process exécutés à partir de celui-ci. |
| <b>trap</b>     | Exécute une commande donnée lorsque le shell reçoit un ou plusieurs signaux déterminés.                       |
| <b>type</b>     | Indique l'interprétation à donner à chaque nom spécifié, lorsqu'utilisé comme nom de commande.                |
| <b>ulimit</b>   | Affiche ou ajuste les ressources shell affectées.                                                             |
| <b>umask</b>    | Détermine les droits d'accès au fichier.                                                                      |
| <b>unset</b>    | Supprime la variable ou la fonction correspondant au nom spécifié.                                            |
| <b>wait</b>     | Attend la fin de l'exécution du process enfant et indique son état à la sortie.                               |

Pour en savoir plus, reportez-vous à Commandes intégrées (shell Bourne).

---

## Liste des commandes intégrées du shell C

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>alias</b>    | Affiche les alias spécifiés (tous, par défaut).                                                                                                                                                                                                                                                                                                                                                                       |
| <b>bg</b>       | Place à l'arrière-plan le travail courant ou les travaux spécifiés.                                                                                                                                                                                                                                                                                                                                                   |
| <b>break</b>    | Reprend l'exécution après la fin de la commande <b>foreach</b> ou <b>while</b> la plus proche.                                                                                                                                                                                                                                                                                                                        |
| <b>breaksw</b>  | Interrompt une commande <b>switch</b> .                                                                                                                                                                                                                                                                                                                                                                               |
| <b>case</b>     | Définit un label dans une commande <b>switch</b> .                                                                                                                                                                                                                                                                                                                                                                    |
| <b>cd</b>       | Passe au répertoire spécifié.                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>chdir</b>    | Passe au répertoire spécifié.                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>continue</b> | Poursuit l'exécution de la commande <b>foreach</b> ou <b>while</b> la plus proche.                                                                                                                                                                                                                                                                                                                                    |
| <b>default</b>  | Étiquette le cas par défaut d'une instruction <b>switch</b> .                                                                                                                                                                                                                                                                                                                                                         |
| <b>dirs</b>     | Affiche la pile de répertoires.                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>echo</b>     | Écrit les chaînes de caractères sur la sortie standard du shell.                                                                                                                                                                                                                                                                                                                                                      |
| <b>else</b>     | Exécute les commandes qui suivent le deuxième <b>else</b> dans une séquence de commande <b>if (Expression) then ...else if (Expression2) then ... else ... endif</b> .                                                                                                                                                                                                                                                |
| <b>end</b>      | Indique la fin d'une séquence de commandes précédées de la commande <b>foreach</b> . Voir aussi la commande <b>while</b> .                                                                                                                                                                                                                                                                                            |
| <b>endif</b>    | Exécute les commandes qui suivent la seconde instruction <b>then</b> dans une séquence de commande <b>if (Expression) then ... else if (Expression2) then ... else ... endif</b> .                                                                                                                                                                                                                                    |
| <b>endsw</b>    | Marque la fin d'une séquence de commande <b>switch (String) case String : ... .. breaksw default: ... .. breaksw endsw</b> . Cette séquence de commande compare successivement chaque label "case" à la valeur de la variable <i>Chaîne</i> . L'exécution se poursuit après un <b>endsw</b> si une commande <b>breaksw</b> est exécutée ou qu'aucun label ne correspond et qu'aucune valeur par défaut n'est définie. |
| <b>eval</b>     | Lit les arguments comme entrée pour le shell et exécute le(s) commande(s) résultante(s), dans le contexte du shell courant.                                                                                                                                                                                                                                                                                           |
| <b>exec</b>     | Exécute la commande spécifiée, à la place du shell courant.                                                                                                                                                                                                                                                                                                                                                           |
| <b>exit</b>     | Quitte le shell avec la valeur de variable d'état du shell, ou celle de l'expression spécifiée.                                                                                                                                                                                                                                                                                                                       |
| <b>fg</b>       | Amène à l'avant-plan le travail courant ou les travaux spécifiés, et poursuit leur exécution même s'ils ont été arrêtés.                                                                                                                                                                                                                                                                                              |
| <b>foreach</b>  | Définit tour à tour une variable <i>Nom</i> pour chaque membre spécifié par la variable <i>Liste</i> et une séquence de commandes, jusqu'à atteindre une commande <b>end</b> .                                                                                                                                                                                                                                        |
| <b>glob</b>     | Affiche une liste avec développement de l'historique, des variables et des noms de fichiers.                                                                                                                                                                                                                                                                                                                          |
| <b>goto</b>     | Poursuit l'exécution à partir d'une ligne déterminée.                                                                                                                                                                                                                                                                                                                                                                 |
| <b>hashstat</b> |                                                                                                                                                                                                                                                                                                                                                                                                                       |
|                 | Affiche des statistiques indiquant le pourcentage de recherches abouties de commandes, par la table de hachage.                                                                                                                                                                                                                                                                                                       |
| <b>history</b>  | Affiche les listes des événements de l'historique.                                                                                                                                                                                                                                                                                                                                                                    |

|                 |                                                                                                                                                                                                                                                                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>if</b>       | Exécute une commande donnée si l'expression spécifiée est vraie.                                                                                                                                                                                                                                                                          |
| <b>jobs</b>     | Liste les travaux actifs.                                                                                                                                                                                                                                                                                                                 |
| <b>kill</b>     | Envoie le signal <b>TERM</b> (terminate), ou celui spécifié par <i>Signal</i> , aux travaux ou aux process spécifiés.                                                                                                                                                                                                                     |
| <b>limit</b>    | Limite l'usage de la ressource par le process courant et les process qu'il crée.                                                                                                                                                                                                                                                          |
| <b>login</b>    | Met fin à un shell de connexion et le remplace par une occurrence de la commande <b>/usr/sbin/login</b> .                                                                                                                                                                                                                                 |
| <b>logout</b>   | Termine un shell de connexion.                                                                                                                                                                                                                                                                                                            |
| <b>nice</b>     | Définit la priorité des commandes exécutées dans le shell.                                                                                                                                                                                                                                                                                |
| <b>nohup</b>    | Provoque la non prise en compte des arrêts dans la suite de la procédure.                                                                                                                                                                                                                                                                 |
| <b>notify</b>   | Le shell vous prévient (de façon asynchrone) lorsque l'état du travail courant ou du travail spécifié change.                                                                                                                                                                                                                             |
| <b>onintr</b>   | Contrôle l'action du shell au niveau des interruptions.                                                                                                                                                                                                                                                                                   |
| <b>popd</b>     | Détruit la pile de répertoires et revient au nouveau premier répertoire.                                                                                                                                                                                                                                                                  |
| <b>pushd</b>    | Intervertit des éléments de la pile des répertoires.                                                                                                                                                                                                                                                                                      |
| <b>rehash</b>   | Recalcule, dans la variable shell path, la table d'adressage contenant les répertoires.                                                                                                                                                                                                                                                   |
| <b>repeat</b>   | Exécute la commande spécifiée, soumise aux mêmes restrictions que l'instruction <b>if</b> , autant de fois que spécifié.                                                                                                                                                                                                                  |
| <b>set</b>      | Affiche la valeur de toutes les variables shell.                                                                                                                                                                                                                                                                                          |
| <b>setenv</b>   | Modifie la valeur de la variable d'environnement spécifiée.                                                                                                                                                                                                                                                                               |
| <b>shift</b>    | Décale vers la gauche la variable spécifiée.                                                                                                                                                                                                                                                                                              |
| <b>source</b>   | Lit la commande spécifiée par la variable <i>Nom</i> .                                                                                                                                                                                                                                                                                    |
| <b>stop</b>     | Arrête les travaux (courants ou spécifiés) exécutés en arrière-plan.                                                                                                                                                                                                                                                                      |
| <b>suspend</b>  | Met fin au shell (équivalent à un signal <b>STOP</b> ).                                                                                                                                                                                                                                                                                   |
| <b>switch</b>   | Marque la fin d'une séquence de commande <b>switch (Chaîne) case Chaîne : ... breaksw default: ... breaksw endsw</b> . Cette séquence de commande compare successivement chaque étiquette à la valeur de la variable <i>Chaîne</i> . Cette séquence compare successivement chaque label "case" à la valeur de la variable <i>Chaîne</i> . |
| <b>time</b>     | Affiche un récapitulatif des temps consommés par le shell et son process fils.                                                                                                                                                                                                                                                            |
| <b>umask</b>    | Détermine les droits d'accès au fichier.                                                                                                                                                                                                                                                                                                  |
| <b>unalias</b>  | Désactive les alias dont le nom correspond à la <i>Trame</i> .                                                                                                                                                                                                                                                                            |
| <b>unhash</b>   | Désactive l'usage de la table de hachage pour la localisation des programmes en cours.                                                                                                                                                                                                                                                    |
| <b>unlimit</b>  | Supprime toute limitation de ressources.                                                                                                                                                                                                                                                                                                  |
| <b>unset</b>    | Supprime toutes les variables dont le nom correspond à la <i>Trame</i> .                                                                                                                                                                                                                                                                  |
| <b>unsetenv</b> | Supprime de l'environnement toutes les variables dont le nom correspond à la <i>Trame</i> .                                                                                                                                                                                                                                               |
| <b>wait</b>     | Attend tous les travaux en arrière-plan.                                                                                                                                                                                                                                                                                                  |

**while** Evaluate les commandes de la séquence qui va du **while** au **end** correspondant, tant que la variable *Expression* a une valeur non nulle.

**@** Affiche la valeur des variables shell spécifiées.

Pour en savoir plus, reportez-vous à Commandes intégrées (shell C).

---

## Voir aussi Shell Korn

Shells.

Shell Korn ou POSIX

Commandes du shell Korn ou POSIX.

Alias de commandes (shell Korn ou POSIX)

Substitution de paramètres (shell Korn ou POSIX), page 11-17.

Substitution de commandes (shell Korn ou POSIX)

Evaluation arithmétique (shell Korn ou POSIX).

Séparation de zones

Substitution de noms de fichiers (shell Korn ou POSIX)

Suppression des caractères de déclaration

Réacheminement des entrées/sorties (shell Korn ou POSIX).

Contrôle des travaux (shell Korn ou POSIX).

Edition en ligne (shell Korn ou POSIX).

Commandes **ksh** et **stty**.

Commandes **alias**, **cd**, **export**, **fc**, **getopts**, **read**, **set** et **typeset** du shell Korn.

Fichier **/etc/passwd**.

Shell Bourne

Shell C

---

## Voir aussi Shell Bourne

Shells.

Shell Bourne

Shell restreint.

Commandes du shell Bourne.

Substitution de variables et de noms de fichiers (shell Bourne) , page 11-83.

Réacheminement des entrées/sorties (shell Bourne).

Commandes **bsh** ou **Rsh**, et **login**.

Commande shell Bourne spéciale AUCUN LIEN **read**.

Sous-routines **setuid** et **setgid**.

Fichier spécial **null**.

Fichiers **environment** et **profile**.

Shell Korn ou POSIX

Shell C

---

## Voir aussi Shell C

Shells.

Commandes du shell C.

Substitution d'historique (shell C).

Substitution d'alias (shell C).

Substitution de variables et de noms de fichiers (shell C)

Variables d'environnement (shell C).

Réacheminement des entrées/sorties (shell C).

Contrôle des travaux (shell C).

Commandes **cs**h et **ed**.

Commandes intégrées du shell C **alias**, **unalias**, **jobs**, **notify** et **set**.

---

## Shell Bourne

Le shell Bourne est un interpréteur de commandes interactif et un langage de programmation. Il est appelé par la commande **bsh**.

Le shell Bourne peut être exécuté comme shell de connexion ou comme un sous-shell sous le shell de connexion. C'est la commande **login** uniquement qui permet d'exécuter le shell Bourne comme shell de connexion. Pour cela, il faut faire précéder la commande **bsh** d'un - (tiret) :`-bsh`. Sous cette forme, le shell commence par lire et exécuter les commandes du fichier système **/etc/profile** et, le cas échéant, celles de votre fichier **\$HOME/.profile**. Le fichier **/etc/profile** définit les variables requises par tous les utilisateurs. Le shell est prêt à lire les commandes à partir de votre entrée standard.

Si le paramètre *Fichier [Paramètre]* est spécifié au lancement du shell Bourne, le shell exécute le fichier script identifié par *Fichier*, avec ses paramètres éventuels. Le fichier script doit être accessible en lecture. Les éventuels paramètres **setuid** et **setgid** sont ignorés. Le shell lit ensuite les commandes. Avec un indicateur **-c** ou **-s**, aucun fichier script ne doit être spécifié.

Ce chapitre traite des points suivants :

- Environnement du shell Bourne, page 11-70.
- Shell restreint, page 11-72.
- Commandes du shell Bourne, page 11-73.
  - Déclaration de caractères.
  - Traitement des signaux
  - Commandes composées (shell Bourne).
  - Substitution de commandes (shell Bourne).
- Substitution de variables et de noms de fichiers (shell Bourne), page 11-83.
  - Substitution de variables (shell Bourne).
  - Variables définies par l'utilisateur.
  - Substitution conditionnelle.
  - Paramètres positionnels.
  - Substitution de noms de fichiers (shell Bourne), page 11-84.
  - Classes de caractères.
- Réacheminement des entrées/sorties (shell Bourne), page 11-91.

## Environnement du shell Bourne

Toutes les variables (avec leurs valeurs) connues d'une commande lors de son lancement constituent son *environnement* : variables héritées du process parent et variables spécifiées comme paramètres-clés sur la ligne de commande.

Le shell passe à ses process fils les variables nommées comme arguments de la commande intégrée **export**. Cette commande place les variables nommées à la fois dans l'environnement du shell et dans celui de ses futurs process fils.

Les mots-clés sont des paires nom-valeur spécifiées généralement avant le nom de procédure dans la ligne de commande (voir également l'indicateur de la commande **set**). Ces variables sont placées dans l'environnement de la procédure appelée.

Prenons pour exemple la procédure simple suivante, qui affiche les valeurs de deux variables (enregistrées dans le fichier de commande `key_command`) :

```
key_command
echo $a $b
```

Voici les commandes et les sorties générées :

| Entrée |        |             | Sortie    |
|--------|--------|-------------|-----------|
| a=key1 | b=key2 | key_command | key1 key2 |
| a=tom  | b=john | key_command | tom john  |

Les mots-clés d'une procédure ne sont pas décomptés dans le chiffre enregistré dans  `$#` .

Une procédure a accès aux valeurs de toutes les variables de son environnement. Mais si elle modifie l'une de ses valeurs, ces modifications ne sont pas répercutées dans l'environnement du shell. Elles ne s'appliquent que localement à la procédure concernée. Pour que les modifications s'appliquent à l'environnement transmis aux processus fils, exportez les nouvelles valeurs dans cette procédure.

Pour afficher la liste des variables exportables du shell courant, entrez :

```
export
```

Pour afficher la liste des variables en lecture seule du shell courant, entrez :

```
readonly
```

Pour afficher la liste des paires variable-valeur de l'environnement courant, entrez :

```
env
```

Pour en savoir plus sur les environnements utilisateur, reportez-vous à "Fichier `/etc/environment`", page 10-4.

---

## Shell restreint

Le shell restreint permet de définir des noms de connexion et des environnements d'exécution dont les pouvoirs sont plus contrôlés que ceux du shell Bourne normal. Il est appelé par la commande **Rsh** ou **bsh -r**. Le comportement de ces commandes est identique à celui de la commande **bsh**, à ceci près qu'elles n'autorisent pas :

- changer de répertoire (commande **cd**),
- définir la valeur des variables **PATH** ou **SHELL**,
- spécifier des chemins ou des noms de commandes comportant une barre oblique (/),
- réacheminer les sorties.

Si le shell restreint détermine qu'une commande est une procédure shell, il appelle le shell Bourne pour l'exécuter. L'utilisateur peut ainsi accéder à toutes les fonctionnalités du shell Bourne par le biais d'un menu de commandes limité. Ce schéma suppose que l'utilisateur n'a pas accès en écriture et en exécution au même répertoire.

Si le paramètre *Fichier* [*Paramètre*] est spécifié au lancement du shell Bourne, le shell exécute le fichier script identifié par *Fichier*, avec ses paramètres éventuels. Le fichier script doit être accessible en lecture. Les éventuels paramètres **setuid** et **setgid** sont ignorés. Le shell lit ensuite les commandes.

**Remarque :** Avec un indicateur **-c** ou **-s**, ne spécifiez pas de fichier script.

Lancé via la commande **Rsh**, le shell renforce les restrictions après interprétation des fichiers **.profile** et **/etc/environment**. C'est donc à celui qui écrit le fichier **.profile** qu'incombe le contrôle des actions autorisées à l'utilisateur et du répertoire auquel il accède (de préférence pas le répertoire de connexion). Un administrateur peut créer un répertoire de commandes dans le répertoire **/usr/rbin**, utilisable par la commande **Rsh**, en modifiant en conséquence la variable **PATH**. Lancé via la commande **bsh -r**, le shell applique les restrictions au moment où il interprète les fichiers **.profile**.

Appelé par **-Rsh**, le shell restreint lit le fichier **.profile** de l'utilisateur (**\$HOME/.profile**). Il agit comme le shell Bourne normal, à ceci près que toute interruption provoque la sortie immédiate et non un retour au niveau commande.

- Voir aussi Shell Bourne

---

## Commandes du shell Bourne

Lorsque vous lancez une commande dans le shell Bourne, il commence par l'analyser et effectuer toutes les substitutions indiquées. Puis il exécute la commande, à condition que :

- le nom de commande corresponde à une commande spéciale du shell Bourne ;
- OU
- le nom de commande corresponde au nom d'une fonction définie. Dans ce cas, le shell définit les paramètres positionnels par ceux de la fonction.

Si le nom de commande ne correspond ni à une commande intégrée ni à une fonction définie et que la commande nomme un fichier exécutable qui est un programme (binaire) compilé, le shell (en tant que *parent*) génère dynamiquement un nouveau process (*fil*s) qui exécute immédiatement le programme. Si le fichier est déclaré exécutable mais n'est pas un programme compilé, le shell suppose qu'il s'agit d'une procédure shell. Dans ce cas, le shell génère dynamiquement une autre instance de lui-même (un *sous-shell*) qui lit le fichier et en exécute les commandes. Le shell exécute également les commandes mises entre parenthèses dans le sous-shell. Pour l'utilisateur, l'exécution d'un programme compilé se fait exactement de la même manière qu'une procédure shell. Le shell recherche normalement les commandes dans les répertoires du système de fichiers, dans l'ordre suivant :

1. `/usr/bin`
2. `/etc`
3. `/usr/sbin`
4. `/usr/ucb`
5. `$HOME/bin`
6. `/usr/bin/X11`
7. `/sbin`
8. Répertoire courant.

Le shell explore successivement chaque répertoire, passant au suivant lorsqu'il ne parvient pas à trouver la commande.

**Remarque :** L'ordre de recherche est déterminé par la variable **PATH** : pour modifier l'ordre, modifiez donc cette variable.

Si vous indiquez un chemin d'accès en lançant une commande (par exemple, `/usr/bin/sort`), le shell n'explore que le répertoire correspondant. Si le nom de commande contient une barre oblique (`/`), le shell ne tient pas compte du chemin d'accès.

Vous pouvez spécifier un chemin d'accès complet commençant par le répertoire racine (par exemple `/usr/bin/sort`). Vous pouvez également indiquer un chemin d'accès relatif au répertoire courant. Si vous spécifiez :

```
bin/myfile
```

le shell cherche le répertoire `bin` dans le répertoire courant, puis, dans ce répertoire, le fichier `myfile`.

**Remarque :** Le shell restreint ne traite pas les commandes contenant une barre oblique (`/`).

Le shell mémorise le chemin d'accès de chaque commande exécutée (pour éviter des recours répétés à la commande **exec**). Si la commande se trouve dans un répertoire relatif (ne commençant pas par `/`), le shell doit la rechercher à nouveau à chaque changement de répertoire courant. Si vous modifiez la valeur de la variable **PATH** ou que vous exécutez la commande **hash -r**, le shell efface tous les chemins mémorisés.

Ce chapitre traite des points suivants :

- Déclaration de caractères, page 11-74.
- Traitement des signaux, page 11-74.
- Commandes intégrées (shell Bourne), page 11-76.
- Substitution de commandes (shell Bourne), page 11-82.

## Déclaration de caractères.

De nombreux caractères ont une spécification spéciale pour le shell. Vous souhaiterez quelquefois passer outre. Pour cela, vous n'avez qu'à encadrer les caractères concernés par des apostrophes (') et des guillemets ("), ou en les faisant précéder d'une barre oblique inverse (\).

Tous ces caractères, à l'exception de l'apostrophe, sont alors interprétés littéralement. Ainsi, la commande :

```
stuff='echo $? $*; ls * | wc'
```

affecte la chaîne littérale `echo $? $*; ls * | wc` à la variable `stuff`. Le shell n'exécute pas les commandes **echo**, **ls** et **wc** et ne développe ni les variables  `$?`  et  `$*` , ni le caractère astérisque (\*).

Placés entre guillemets, les caractères \$ (dollar), ` (apostrophe inverse) et " (guillemets) conservent leur signification spéciale, tous les autres caractères étant interprétés littéralement. La substitution des variables et des commandes encadrées par des guillemets a donc lieu. En outre, les guillemets n'affectent pas les commandes dans une substitution de commande faisant partie de la chaîne déclarée, aussi les caractères conservent-ils dans ce cas leur signification spéciale.

Soit la séquence :

```
ls *
file1 file2 file3
message="Ce répertoire contient `ls * ` "
echo $message
Ce répertoire contient file1 file2 file3
```

le caractère spécial \* figurant dans la chaîne à substituer a été développé.

Pour annuler la signification spéciale des caractères \$ (dollar), ` (apostrophe inverse) et " (double guillemets), placés entre guillemets, faites-les précéder d'une barre oblique inverse. Dans une expression non encadrée de guillemets, faire précéder un caractère d'une barre oblique inverse équivaut à le placer entre apostrophes. Par conséquent, une barre oblique inverse précédant immédiatement un caractère nouvelle ligne (c'est-à-dire une barre oblique inverse en fin de ligne) annule le caractère nouvelle ligne, ce qui vous permet de continuer la ligne de commande sur la ligne suivante (physiquement).

## Traitement des signaux

Les signaux **INTERRUPT** et **QUIT** sont ignorés lorsque la commande est suivie d'un & (perluète) ; c'est-à-dire exécutée en arrière-plan. Sinon, les valeurs des signaux sont celles transmises par le shell parent, à l'exception du signal VIOLATION DE SEGMENTATION. Pour en savoir plus, reportez-vous à la commande intégrée **trap**.

## Commandes composées (shell Bourne)

Une commande composée est l'un des éléments suivants :

- pipeline (séquence de commandes séparées par le symbole |),
- liste de commandes simples,
- commande commençant par un mot réservé,
- commande commençant par l'opérateur ( (parenthèse ouvrante).

Sauf indication contraire, la valeur renvoyée par une commande composée est celle de la dernière commande simple exécutée.

## Mots réservés

Les mots réservés ne sont identifiés que lorsqu'ils apparaissent sans guillemets et comme premier mot d'une commande.

|              |              |             |
|--------------|--------------|-------------|
| <b>for</b>   | <b>do</b>    | <b>done</b> |
| <b>case</b>  | <b>esac</b>  |             |
| <b>if</b>    | <b>then</b>  | <b>fi</b>   |
| <b>elif</b>  | <b>else</b>  |             |
| <b>while</b> | <b>until</b> |             |
| {            | }            |             |
| (            | )            |             |

**for** *Identificateur*[**in** *Mot* . . . ] **do** *Liste* **done**

Affecte à *Identificateur* la valeur du (des) paramètre(s) *Mot* (un à la fois) et exécute les commandes spécifiées par *Liste*. Si **in** *Mot* est omis . . . , la commande **for** exécute le contenu du paramètre *Liste* pour chaque paramètre positionnel défini, la commande s'achevant lorsque tous ces paramètres ont été traités.

**case** *Mot* **in** *Trame* [|*Trame*] . . . ) *Liste*;; [*Trame* [|*Trame*] . . . ) *Liste*;;] . . . **esac**

Exécute les commandes spécifiées par *Liste*, associées à la première *Trame* correspondant à la valeur de *Mot*. Les conventions de notation sont les mêmes que celles utilisées pour la substitution de noms de fichiers, à ceci près qu'une barre oblique (/), un point en tête (.) ou une séquence barre oblique/point ne doivent pas obligatoirement correspondre explicitement.

**if** *Liste* **then** *Liste* [**elif** *Liste* **then** *Liste*] . . . [**else** *Liste*] **fi**

Exécute les commandes spécifiées par le paramètre *Liste* qui suit la commande **if**. Si la commande renvoie une valeur nulle, le shell exécute les commandes spécifiées par la première commande **then**. Sinon, il exécute celles définies par la commande **elif** (si elle est précisée). Si la valeur de sortie est encore nulle, le shell passe aux commandes définies par le **then** suivant. Si la valeur renvoyée est non nulle, le shell exécute les commandes définies par la *Liste* suivant la commande **else** (si elle est précisée). Si aucune commande **else** *Liste* ou **then** *Liste* n'est exécutée, la commande **if** renvoie une valeur nulle.

### **while** *Liste* **do** *Liste* **done**

Exécute les commandes spécifiées par le paramètre *Liste* qui suit la commande **while**. Si la commande renvoie une valeur nulle, le shell exécute le paramètre *Liste* suivant la commande **do**. Il boucle ensuite sur les listes jusqu'à obtenir une valeur de sortie non nulle. Si aucune des commandes de **do** *Liste* n'est exécutée, la commande **while** renvoie une valeur nulle.

### **until** *Liste* **do** *Liste* **done**

Exécute les commandes spécifiées par le paramètre *Liste* qui suit la commande **until**. Si la commande renvoie une valeur nulle, le shell exécute le paramètre *Liste* suivant la commande **do**. Il boucle ensuite sur les listes jusqu'à obtenir une valeur de sortie nulle. Si aucune des commandes de **do** *Liste* n'est exécutée, la commande **until** renvoie une valeur nulle.

( *Liste* )

Exécute, dans un sous-shell, les commandes spécifiées par le paramètre *Liste*.

{ *Liste*; }

Exécute, dans le shell courant, les commandes spécifiées par le paramètre *Liste*, sans créer de sous-shell.

*Nom* () { *Liste* }

Définit une fonction référencée par *Nom*. Le corps de cette fonction est constitué des commandes entre accolades spécifiées par *Liste*.

## Commandes intégrées (shell Bourne)

Les commandes spéciales, intégrées au shell Bourne, sont exécutées dans le processus shell. Sauf indication contraire, la sortie est envoyée au descripteur de fichier 1 avec un état de sortie nul lorsque la commande est exempte d'erreur de syntaxe. Le réacheminement des entrées/sorties est autorisé.

La Liste des commandes intégrées du shell Bourne répertorie ces commandes par ordre alphabétique.

Les commandes ci-dessous sont traitées quelque peu différemment des autres commandes intégrées :

|                  |                 |              |
|------------------|-----------------|--------------|
| <b>:</b> (colon) | <b>exec</b>     | <b>shift</b> |
| <b>.</b> (dot)   | <b>exit</b>     | <b>times</b> |
| <b>break</b>     | <b>export</b>   | <b>trap</b>  |
| <b>continue</b>  | <b>readonly</b> | <b>wait</b>  |
| <b>eval</b>      | <b>return</b>   |              |

Le shell Bourne traite ces commandes comme suit :

- Les listes d'affectation de paramètres précédant la commande restent en vigueur après l'exécution de la commande.
- Les réacheminements d'E/S sont traités après les affectations de paramètres.
- Toute erreur détectée dans un script shell provoque l'arrêt du script.

## Description des commandes spéciales

Voici les commandes intégrées spéciales du shell Bourne :

### Commandes intégrées

- :** Renvoie un état de sortie nul.
- . *Fichier*** Lit et exécute les commandes de *Fichier* et revient au shell. Ne lance pas de sous-shell. Le shell recherche le fichier dans le répertoire défini par la variable **PATH**.
- break [ *n* ]** Sort, le cas échéant, de la boucle **for**, **while** ou **until**. Si *n* est spécifié, la commande **break** sort à la *nième* boucle.
- continue [ *n* ]** Reprend à l'itération suivante de la boucle **for**, **while** ou **until**. Si vous spécifiez *n*, la commande reprend à la *nième* boucle.
- cd [ *Répertoire* ]** Passe au répertoire défini par *Répertoire*. En l'absence de spécification de *Répertoire*, c'est la valeur de la variable shell **HOME** qui est utilisée. La variable shell **CDPATH** définit le chemin d'accès pour *Répertoire*. **CDPATH** est constitué d'une liste de répertoires séparés par des deux-points (:). Un chemin d'accès vide (qui est le chemin par défaut) désigne le répertoire courant. Ce répertoire est affiché immédiatement après le signe égal ou entre les deux-points dans la liste des chemins. Si *Répertoire* commence par une barre oblique (/), le shell ne tient pas compte du chemin d'accès. Sinon, tous les répertoires de la variable shell **CDPATH** sont explorés.
- Remarque :** Le shell restreint ne traite pas la commande **cd**.
- echo [ *Chaîne . . .* ]** Ecrit les chaînes de caractères sur la sortie standard. Pour en savoir plus, reportez-vous à la commande **echo**. L'indicateur **-n** n'est pas pris en charge.
- eval [ *Argument . . .* ]** Lit les arguments comme entrée du shell et exécute le(s) commande(s) résultante(s).
- exec [ *Argument . . .* ]** Exécute, à la place du shell, la commande spécifiée par *Argument*, sans créer de nouveau process. Des arguments d'entrée/sortie peuvent apparaître et, si d'autres arguments n'apparaissent pas, engendrer des modifications de l'entrée ou de la sortie shell. Ce qui est fortement déconseillé s'il s'agit de votre shell de connexion.
- exit [ *n* ]** Quitte le shell dont l'état de sortie est spécifié par le paramètre *n*. Si *n* est omis, la valeur de sortie est celle de la dernière commande exécutée (CtrlD provoque également la sortie du shell). La valeur du paramètre *n* est un entier compris entre 0 et 255.
- export [ *Nom . . .* ]** Marque les noms spécifiés, pour exportation automatique vers l'environnement des commande ultérieures. Si *Nom* est omis, affiche la liste des noms exportés dans ce shell. Vous ne pouvez exporter des noms de fonctions.

**hash** [ **-r** ] [ *Commande . . .* ]

Recherche et mémorise l'emplacement du chemin d'accès à chaque *Command* spécifiée. A l'inverse, l'indicateur **-r** efface tous les chemins mémorisés. En l'absence de cet indicateur ou de commandes, le shell affiche des informations sur les commandes mémorisées, au format :

```
 Hits Cost Command
```

Hits indique le nombre d'exécution d'une commande par un process shell. Cost mesure le travail requis pour localiser une commande dans le chemin. Command précise le chemin d'accès à la commande. Dans certains cas, l'emplacement mémorisé doit être recalculé – s'il s'agit d'un chemin relatif et que vous changez de répertoire courant, par exemple. Les commandes concernées sont indiquées par un astérisque (\*), affiché en regard de Hits. Cost est incrémenté en conséquence.

**pwd**

Affiche le répertoire courant. Reportez-vous à la description de la commande **pwd**.

**read** [ *Nom . . .* ]

Lit une ligne de l'entrée standard. Affecte le premier mot de la ligne au premier paramètre *Nom*, le deuxième mot au deuxième paramètre, etc., les mots résiduels étant affectés au dernier paramètre *Nom*. Cette commande renvoie une valeur nulle, sauf si elle détecte un caractère fin de fichier.

**readonly** [ *Nom . . .* ]

Marque en lecture seule le *Nom* spécifié. La valeur de ce nom ne peut être redéfinie. Si *Nom* est omis, affiche la liste des noms en lecture seule.

**return** [ *n* ]

Provoque la fin d'une fonction, avec la valeur de retour *n*. Si *n* est omis, la fonction renvoie l'état de la dernière commande exécutée de cette fonction. Cette commande n'est valide que lorsqu'exécutée sous la fonction shell.

**set** [ *Indicateur* [ *Argument* ] . . . ]

Définit un ou plusieurs des indicateurs suivants :

**-a** Marque pour exportation toutes les variables affectées. Si l'affectation précède un nom de commande, l'attribut d'exportation n'est valable que pour l'environnement d'exécution de cette commande - sauf s'il s'agit d'une commande intégrée spéciale. Dans ce cas, l'attribut reste en vigueur même après exécution de la commande intégrée. Si l'affectation ne précède pas un nom de commande, ou qu'elle résulte d'une commande **getopts** ou **read**, l'attribut reste en vigueur jusqu'à ce que l'affectation soit annulée.

**-e** Provoque la sortie immédiate d'une commande si toutes les conditions suivantes sont réunies :

- elle renvoie une valeur supérieure à zéro,
- elle n'est pas un élément de la liste d'une commande **while**, **until** ou **if**,
- elle n'est pas en cours de test via des listes AND ou OR,
- elle n'est pas un pipeline précédé du mot réservé ! (point d'exclamation).

- f** Désactive la substitution de nom de fichier.
- h** Localise et mémorise les commandes appelées par les fonctions au moment de leur définition. (Normalement, ces commandes sont localisées au moment de l'exécution de la fonction ; voir la commande **hash**.)
- k** Place tous les mots-clés dans l'environnement d'une commande (et pas seulement ceux qui précèdent le nom de commande).
- n** Lit les commandes sans les exécuter. L'indicateur **-n** permet de vérifier les erreurs de syntaxe des scripts shell.
- t** Quitte après lecture et exécution d'une seule commande.
- u** Traite toute variable non définie comme une erreur et sort immédiatement après avoir opéré la substitution de variable. Un shell interactif ne permet pas de sortie.
- v** Affiche les lignes d'entrée au fur et à mesure de leur lecture.
- x** Affiche les commandes et leurs arguments avant leur exécution.
- Ne modifie pas les indicateurs. Utile pour définir le paramètre positionnel **\$1** par une chaîne commençant par un signe **-** (moins).

Un signe **+** (plus) à la place d'un signe **-** (moins) désactive les indicateurs. Vous pouvez aussi spécifier ces indicateurs sur la ligne de commande du shell. La variable spéciale **\$-** contient l'ensemble des indicateurs courants.

Tout *Argument* de la commande **set** devient un paramètre positionnel affecté, dans l'ordre, à **\$1**, **\$2**, etc. A défaut d'*indicateur* ou d'*Argument*, la commande **set** affiche les noms et valeurs de toutes les variables shell courantes.

### **shift** [*n*]

Décale les arguments de la ligne de commande vers la gauche ; autrement dit, réaffecte les paramètres positionnels en rejetant la valeur courante de **\$1** et en réaffectant la valeur de **\$2** à **\$1**, de **\$3** à **\$2**, etc. S'il y a plus de 9 arguments, le 10ème est affecté à **\$9** et s'il en reste d'autres, ils ne sont pas affectés (jusqu'au prochain **shift**). S'il y a 9 arguments (ou moins), la commande **shift** annule l'affectation du paramètre positionnel doté du numéro le plus élevé.

Le paramètre positionnel **\$0** n'est jamais décalé. La commande **shift** *n* est une notation abrégée qui permet de spécifier *n* décalages consécutifs. La valeur par défaut de *n* est 1.

**test** *Expression* | [ *Expression* ]

Evalue des expressions conditionnelles. Reportez-vous à la description de la commande **test**. L'indicateur **-h** n'est pas accepté par la commande de test intégrée dans **bsh**.

**times**

Affiche les temps cumulés utilisateur et système, pour le shell et les process exécutés à partir de celui-ci.

**trap** [ *Commande* ] [ *n* ] . . .

Exécute la commande spécifiée par le paramètre correspondant lorsque le shell reçoit le ou les signaux déterminés par *n*. Les commandes **trap** sont exécutées dans l'ordre des numéros des signaux. Toute tentative de définir un traitement d'interruption sur un signal qui a été ignoré en entrée du shell courant est sans effet.

**Remarque :** Le shell balaye le paramètre *Commande* une première fois lorsque le traitement d'interruption est défini, puis de nouveau à l'exécution du traitement.

Si vous ne spécifiez pas de commande, tous les traitements d'interruption spécifiés par *n* sont restaurés à leurs valeurs courantes. Si la chaîne spécifiée est vide, le shell et les commandes appelées ignorent le signal. Si *n* vaut 0 (zéro), la commande est exécutée lorsque vous quittez le shell. Si vous n'avez spécifié ni commande ni signal, la commande **trap** affiche la liste des commandes associées à un numéro de signal.

**type** [*Nom* . . . ]

Indique l'interprétation à donner à chaque *Nom* spécifié, lorsqu'utilisé comme nom de commande.

**ulimit** [-HS] [-c | -d | -f | -m | -s | -t] [*limite*]

Affiche ou ajuste les ressources shell affectées. Il existe deux modes d'affichage des ressources, individuel et par groupe. Par défaut, les ressources sont affichées avec leur limite logicielle, ou leur borne inférieure, en tant que groupe.

L'affectation des ressources shell dépend de l'ID utilisateur effectif du shell courant. Le niveau matériel d'une ressource ne peut être défini si cet ID n'est pas celui de l'utilisateur racine. Une erreur est alors générée. Par défaut, l'utilisateur racine définit les limites à la fois logicielles et matérielles des ressources. Il doit donc être particulièrement vigilant lors de l'emploi des indicateurs **-S**, **-H** ou de l'indicateur par défaut. Si vous n'êtes pas utilisateur racine, vous ne pouvez définir que la limite logicielle d'une ressource et, si vous la modifiez à la baisse, vous ne pourrez l'augmenter à nouveau, pour la ramener à son niveau antérieur.

Pour définir les limites d'une ressource, sélectionnez l'indicateur adhoc et la valeur souhaitée (obligatoirement un entier). Vous ne pouvez définir qu'une ressource à la fois : si vous spécifiez plusieurs indicateurs de ressources, vous risquez d'obtenir des résultats indéfinis. Par défaut, **ulimit** suivi d'une simple valeur définit la taille des fichiers du shell. L'indicateur **-f** est facultatif.

La commande **ulimit** accepte les indicateurs suivants :

- c** Définit ou affiche un segment central du shell.
- d** Définit ou affiche un segment de données du shell.
- f** Définit ou affiche la taille des fichiers shell.
- H** Définit ou affiche la limite matérielle d'une ressource (utilisateur racine exclusivement)
- m** Définit ou affiche la mémoire shell.
- s** Définit ou affiche le segment de pile shell.
- S** Définit ou affiche la limite logicielle d'une ressource.
- t** Définit ou affiche le temps CPU maximal pour le shell.

**umask** [*nnn*]

Détermine les droits d'accès au fichier. Cette valeur, associée aux droits du process créateur, détermine les droits sur un fichier au moment de sa création. La valeur par défaut est 022. Si aucune valeur n'est entrée, **umask** affiche la valeur courante.

**unset** [*Nom . . .*]

Supprime la variable ou la fonction correspondant à chaque *Nom* spécifié. La définition des variables **PATH**, **PS1**, **PS2**, **MAILCHECK** et **IFS** ne peut être supprimée.

**wait** [*n*]

Attend la fin de l'exécution du process enfant spécifié par *n* pour sortir, puis renvoie l'état de sortie de ce process. Si *n* est omis, le shell attend la fin de tous les process enfants actifs et renvoie la valeur 0.

## Substitution de commandes (shell Bourne).

La substitution de commande permet de définir comme argument d'une commande le résultat d'une autre commande. Si vous encadrez une ligne de commande d'apostrophes inverses `` , le shell exécute d'abord la (les) commande(s), puis remplace l'intégralité de l'expression (apostrophes inverses comprises) par le résultat. Cette fonction est souvent utilisée pour affecter des variables shell. Par exemple, l'instruction :

```
today=`date`
```

affecte à la variable `today` la chaîne représentant la date courante. L'affectation suivante enregistre, dans la variable `files`, le nombre de fichiers du répertoire courant:

```
files=`ls | wc -l`
```

La substitution de commande peut être exécutée sur n'importe quelle commande dont le résultat est envoyé en sortie standard.

Pour imbriquer des substitutions, faites précéder chaque apostrophe inverse imbriquée d'une barre oblique inverse (\), comme suit :

```
logmsg=`echo Your login directory is \ `pwd```
```

Vous pouvez également attribuer indirectement des valeurs aux variables shell via la commande intégrée **read**. Cette commande lit une ligne à partir de l'entrée standard (le clavier généralement) et affecte successivement les mots de cette ligne aux variables nommées. Par exemple :

```
read first init last
```

lit une ligne de la forme :

```
J. Q. Public
```

avec le même effet que si vous aviez tapé :

```
first=J. init=Q. last=Public
```

La commande spéciale **read** affecte les mots résiduels à la dernière variable.

---

## Substitution de variables et de noms de fichiers (shell Bourne)

Le shell Bourne autorise la substitution de variables et de noms de fichiers.

Cette section traite des points suivants :

- Substitution de variables (shell Bourne), page 11-83.
- Variables définies par l'utilisateur, page 11-83.
- Substitution conditionnelle, page 11-88
- Paramètres positionnels, page 11-87.
- Substitution de noms de fichiers (shell Bourne), page 11-84.
- Classes de caractères, page 11-90.
- Voir aussi Shell Bourne

### Substitution de variables (shell Bourne)

Différents mécanismes sont disponibles pour créer des variables (affecter une valeur de chaîne à un nom). Certaines variables, paramètres positionnels et mots-clés, ne peuvent normalement être spécifiés qu'à partir d'une ligne de commande. D'autres variables sont des noms auxquels peuvent être attribuées (par vous ou par le shell) des valeurs de chaîne.

### Variables définies par l'utilisateur

Le shell identifie les variables alphanumériques auxquelles des valeurs de chaîne peuvent être affectées. Pour affecter une valeur de chaîne à un nom, entrez :

```
Name=Chaîne
```

Un nom est une série de lettres, de chiffres et de traits de soulignement commençant par un trait de soulignement ou une lettre. Pour utiliser la valeur assignée à une variable, faites précéder le nom d'un signe \$ (dollar) : la variable *\$Name*, par exemple, utilise la valeur de *Chaîne*. Notez l'absence d'espaces de part et d'autre du signe = (égal) dans une instruction d'affectation. Les paramètres positionnels ne peuvent apparaître dans une instruction d'affectation : ils ne peuvent être définis que comme décrit à la section "Paramètres positionnels", page 11-87. Une ligne de commande peut comporter plusieurs instructions d'affectation, mais n'oubliez pas que le shell les exécute de la droite vers la gauche.

Si vous encadrez *Chaîne* de guillemets ou d'apostrophes (" ou '), le shell n'interprète pas les caractères blancs, tabulations, deux-points et nouvelle ligne de la chaîne comme des délimiteurs, mais les imbrique littéralement dans la chaîne.

Si vous encadrez *Chaîne* de guillemets ("), le shell reconnaît les variables de la chaîne et effectue les substitutions correspondantes, c'est-à-dire, remplace les références à des paramètres positionnels et autres noms de variable précédés du signe (\$) par leurs valeurs correspondantes. Le shell effectue également la substitution de commande dans les chaînes entre guillemets.

Si vous encadrez *Chaîne* d'apostrophes ('), le shell n'effectue pas de substitution de variable ou de commande dans la chaîne. L'exemple suivant en est une illustration :

```
You: num=875
 number1="Add $num"
 number2='Add $num'
 echo $number1
System: Add 875
You: echo $number2
System: Add $num
```

Le shell ne réinterprète pas les blancs dans les affectations après la substitution de variable. `$first` et `$second` ont donc la même valeur :

```
first='a string with embedded blanks'
second=$first
```

Pour référencer une variable, vous pouvez mettre son nom (ou le chiffre désignant le paramètre positionnel) entre `{ }` (accolades) pour le séparer de la chaîne suivante. En particulier, lorsque le caractère qui suit immédiatement le nom est une lettre, un chiffre ou un trait de soulignement, et que la variable n'est pas un paramètre positionnel, les accolades sont obligatoires :

```
You: a='This is a'
 echo "${a}n example"
System: This is an example
You: echo "$a test"
System: This is a test
```

Pour en savoir plus sur l'usage des accolades, reportez-vous à Substitution conditionnelle, page 11-88.

## Variables utilisées par le shell

Le shell utilise les variables suivantes. Certaines sont définies par le shell, mais pouvez toutes les définir ou les redéfinir.

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CDPATH</b>     | Chemin d'accès utilisé par la commande <b>cd</b> (changement de répertoire).                                                                                                                                                                                                                                                                                                                                                              |
| <b>HOME</b>       | Nom de votre <i>répertoire de connexion</i> , qui est le répertoire courant dès que la connexion est établie. Cette variable est initialisée par le programme <b>login</b> . La commande <b>cd</b> utilise par défaut la valeur de la variable <b>\$HOME</b> . Utiliser cette variable plutôt qu'un nom de chemin explicite dans une procédure shell permet d'exécuter la procédure à partir d'un répertoire différent sans modification. |
| <b>IFS</b>        | Séparateurs de zones internes, utilisés par le shell pour l'interprétation des blancs (voir Interprétation des blancs, page 11-85). La variable <b>IFS</b> comprend initialement les caractères blanc, tabulation et nouvelle ligne.                                                                                                                                                                                                      |
| <b>LANG</b>       | Environnement local à utiliser pour les catégories locales, lorsque ni la variable <b>LC_ALL</b> , ni la variable d'environnement correspondante (commençant par <b>LC_</b> ) n'en spécifient. Pour en savoir plus sur l'environnement local, reportez-vous à "Environnement local : généralités" dans <i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i> .                                                         |
| <b>LC_ALL</b>     | Environnement local à utiliser pour remplacer les valeurs des catégories locales spécifiées par la variable <b>LANG</b> ou par d'autres variables d'environnement commençant par <b>LC_</b> . Pour en savoir plus sur l'environnement local et sur la variable, reportez-vous à "Environnement local : généralités" dans <i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i> .                                       |
| <b>LC_COLLATE</b> | Ordre pour le tri des noms et pour la définition d'intervalles de caractères dans les trames. Pour en savoir plus sur l'environnement local et sur la variable <b>LANG</b> , reportez-vous à "Environnement local : généralités" dans <i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i> .                                                                                                                          |

|                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>LC_CTYPE</b>    | Environnement local pour interpréter des séquences d'octets de données comme des caractères (c'est-à-dire, différencier les caractères mono-octet des caractères multi-octets dans les arguments et les fichiers entrés), ces caractères étant définis comme des lettres (classe <b>alpha</b> ), ainsi que le comportement des classes de caractères dans les recherches génériques. Pour en savoir plus sur l'environnement local, reportez-vous à "Environnement local : généralités" dans <i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i> .                                                                                                                                                                                                  |
| <b>LC_MESSAGES</b> | Langue dans laquelle sont écrits les messages. Pour en savoir plus sur l'environnement local et sur la variable <b>LANG</b> , reportez-vous à "Environnement local : généralités" dans <i>AIX 4.3 Guide d'administration : système d'exploitation et unités</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>LIBPATH</b>     | Chemin d'accès aux bibliothèques partagées.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>LOGNAME</b>     | Nom de connexion, marqué <b>readonly</b> dans le fichier <b>etc/profile</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>MAIL</b>        | Chemin d'accès au fichier utilisé par la messagerie pour détecter l'arrivée de courrier. Si cette variable est définie, le shell vérifie périodiquement l'heure à laquelle le fichier a été modifié, et affiche la valeur de <b>\$MAILMSG</b> si cette heure a changé et que la longueur du fichier est positive. Déclarez la variable <b>MAIL</b> dans le fichier <b>.profile</b> . Les utilisateurs de la commande lui affectent généralement la valeur <b>/usr/spool/mail/\$LOGNAME</b> .                                                                                                                                                                                                                                                                             |
| <b>MAILCHECK</b>   | Délai (en secondes) entre deux "levées" de courrier par le shell dans les fichiers spécifiés par <b>MAILPATH</b> ou <b>MAIL</b> . La valeur par défaut est 600 secondes (10 minutes). Si vous attribuez à <b>MAILCHECK</b> la valeur 0, le shell effectue une vérification avant chaque invite.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| <b>MAILMSG</b>     | Message de notification de courrier. Si vous définissez explicitement <b>MAILMSG</b> par une chaîne vide ( <b>MAILMSG=""</b> ), aucun message ne s'affiche.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>MAILPATH</b>    | Liste de noms de fichier, séparés par deux points (:). Si cette variable est définie, le shell vous informe de l'arrivée de courrier dans l'un des fichiers de la liste. Vous pouvez faire suivre chaque nom de fichier d'un signe % (pourcentage) et d'un message qui s'affichera à l'arrivée de courrier. Sinon, le shell utilise la valeur de la variable <b>MAILMSG</b> ou, par défaut, le message [YOU HAVE NEW MAIL].<br><br><b>Remarque :</b> Lorsque la variable <b>MAILPATH</b> est définie, ce sont les fichiers correspondants qui sont vérifiés et non le fichier défini par <b>MAIL</b> . Pour contrôler à la fois les fichiers définis par <b>MAILPATH</b> et par <b>MAIL</b> , incluez le fichier <b>MAIL</b> dans la liste de fichiers <b>MAILPATH</b> . |
| <b>PATH</b>        | Chemin d'accès aux commandes, sous forme de liste de chemins à des répertoires, séparés par des deux points. Lorsqu'il recherche une commande, le shell explore ces répertoires dans l'ordre spécifié. Le répertoire courant est représenté par une chaîne nulle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

La variable **PATH** est habituellement initialisée dans le fichier **/etc/environment**, en général par : **/usr/bin:/etc:/usr/sbin:/usr/ucb:/usr/bin/X11:/sbin**. Vous pouvez redéfinir cette variable en fonction de vos besoins. La variable **PATH** de votre fichier **.profile** comporte également **\$HOME/bin** et votre répertoire courant.

Si vous avez défini un répertoire spécial pour vos projets, **/projet/bin** par exemple, que vous souhaitez explorer en premier, définissez la variable **PATH** comme suit :

```
PATH=/project/bin:$PATH
```

Pour affecter à **PATH** une autre valeur, insérez-la de préférence dans le fichier **\$HOME/.profile**. Vous ne pouvez pas redéfinir la variable **PATH** si vous exécutez des commandes dans le shell restreint.

|                |                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PS1</b>     | Chaîne constituant l'invite système principale. Cette chaîne est affichée par un shell interactif pour signifier qu'il attend une entrée. La valeur par défaut de <b>PS1</b> est \$ (dollar) suivi d'un espace, pour les utilisateurs non racine.                                                                                                                                   |
| <b>PS2</b>     | Chaîne constituant l'invite secondaire. Elle est affichée par le shell pour signifier qu'il attend une suite lorsqu'il rencontre un caractère nouvelle ligne. La valeur par défaut de <b>PS2</b> est > (supérieur à), suivi d'un blanc.                                                                                                                                             |
| <b>SHACCT</b>  | Nom d'un de vos fichiers. Si cette variable est définie, le shell inscrit un enregistrement comptable à chaque exécution d'un script shell dans ce fichier. Vous disposez ensuite de programmes de comptabilité, tels que <b>acctcom</b> et <b>acctcms</b> pour analyser les données collectées.                                                                                    |
| <b>SHELL</b>   | Chemin d'accès au shell, enregistré avec l'environnement. Cette variable doit être définie et exportée par le fichier <b>\$HOME/.profile</b> de chaque connexion restreinte.                                                                                                                                                                                                        |
| <b>TIMEOUT</b> | Délai d'inactivité (en minutes) au bout duquel le shell quitte. Si cette variable est affectée d'une valeur positive, le shell quitte si aucune commande n'est lancée dans le délai prescrit, après l'affichage de l'invite <b>PS1</b> . (Notez que le shell peut être compilé avec l'indication d'une limite maximale pour le délai.) Les paramètres sont séparés par des espaces. |

## Variables prédéfinies

Plusieurs variables ont des significations spéciales. Voici les variables qui ne peuvent être définies que par le shell :

|             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>\$@</b>  | <p>Développe les paramètres positionnels, en commençant par <b>\$1</b>. Les paramètres sont séparés par des espaces.</p> <p>Si vous encadrez <b>\$@</b> de guillemets (""), le shell interprète chaque paramètre positionnel comme une chaîne distincte. En l'absence de paramètre positionnel, le shell développe l'instruction en chaîne nulle non déclarée.</p>                                                                                                                       |
| <b>\$*</b>  | <p>Développe les paramètres positionnels, en commençant par <b>\$1</b>. Le shell sépare chaque paramètre par le premier caractère de la variable <b>IFS</b>.</p> <p>Si vous encadrez <b>\$*</b> de guillemets (""), le shell encadre de guillemets les valeurs du paramètre positionnel. Les valeurs sont séparées par le premier caractère de la valeur de la variable <b>IFS</b>.</p>                                                                                                  |
| <b>\$#</b>  | <p>Spécifie le nombre (décimal) de paramètres positionnels transmis au shell, nom de la procédure shell elle-même exclu. Le paramètre <b>\$#</b> génère donc le paramètre positionnel ayant la valeur la plus élevée. Ce paramètre sert principalement à vérifier que le nombre d'arguments requis est présent. Seuls les paramètres positionnels <b>\$0</b> à <b>\$9</b> sont accessibles via le shell. Pour en savoir plus, reportez-vous à "Paramètres positionnels", page 11-87.</p> |
| <b>\$?</b>  | <p>Valeur de sortie de la dernière commande exécutée (chaîne décimale). La plupart des commandes renvoient 0 pour indiquer qu'elles ont été correctement exécutées. Le shell renvoie la valeur courante de la variable <b>\$?</b> comme valeur de sortie.</p>                                                                                                                                                                                                                            |
| <b>\$\$</b> | <p>Numéro de process du process courant. Un numéro de process étant unique, cette chaîne est souvent utilisée pour générer des noms uniques pour les fichiers temporaires.</p> <p>L'exemple suivant illustre comment créer des fichiers temporaires dans un répertoire réservé à cet effet :</p> <pre>temp=/tmp/\$\$ ls &gt;\$temp . . . rm \$temp</pre>                                                                                                                                 |
| <b>\$!</b>  | <p>Numéro de process de la dernière commande d'arrière-plan invoquée via la terminaison <b>&amp;</b> (perluète).</p>                                                                                                                                                                                                                                                                                                                                                                     |
| <b>\$-</b>  | <p>Chaîne constituée des noms des indicateurs d'exécution, actuellement définis dans le shell.</p>                                                                                                                                                                                                                                                                                                                                                                                       |

## Interprétation des blancs

Après exécution des substitutions de variables et de commandes, le shell balaye les résultats, à la recherche des séparateurs de zones (définis dans la variable **IFS**). A chaque occurrence d'un séparateur, il partage la ligne en mots distincts, séparés par des espaces. Il conserve les arguments nuls explicites (" " ou " "), mais élimine les implicites (résultant de paramètres non renseignés).

## Substitution conditionnelle

En général, le shell remplace l'expression `$Variable` par la chaîne affectée, le cas échéant à *Variable*. Une notation spéciale permet cependant d'exécuter une *substitution conditionnelle*, selon que la variable est spécifiée ou non nulle, ou les deux à la fois. Par définition, une variable est définie lorsqu'une valeur lui a été affectée. La valeur d'une variable peut être la chaîne nulle, que vous pouvez affecter d'une des façons suivantes :

```
A=
```

```
bcd=""
```

```
Efg=' '
```

```
set ' ' ""
```

Affecte la chaîne vide à A, bcd et Efg.

Affecte la chaîne nulle aux premier et deuxième paramètres positionnels, et annule la définition des autres.

Voici la liste des expressions utilisables dans une substitution conditionnelle.

**`${Variable- Chaîne}`**

Si la variable est définie, remplace l'expression par la valeur de *Variable*. Sinon, la remplace par la valeur de *Chaîne*.

**`${Variable:-Chaîne}`**

Si la variable est définie et non nulle, remplace l'expression par la valeur de *Variable*. Sinon, la remplace par la valeur de *Chaîne*.

**`${Variable=Chaîne}`**

Si la variable est définie, remplace l'expression par la valeur de *Variable*. Sinon, remplace la valeur de *Variable* par la valeur de *Chaîne*, puis l'expression par la valeur de *Variable*. Vous ne pouvez pas affecter ainsi des valeurs aux paramètres positionnels.

**`${Variable:=Chaîne}`**

Si la variable est définie et non nulle, remplace l'expression par la valeur de *Variable*. Sinon, remplace la valeur de *Variable* par la valeur de *Chaîne*, puis l'expression par la valeur de *Variable*. Vous ne pouvez pas affecter ainsi des valeurs aux paramètres positionnels.

**`${Variable?Chaîne}`**

Si la variable est définie, remplace l'expression par la valeur de *Variable*. Sinon, affiche un message de la forme :

```
Variable: Chaîne
```

et sort du shell courant (sauf s'il s'agit du shell de connexion). Si *Chaîne* est omis, le shell affiche le message :

```
Variable: parameter null or not set
```

`${Variable:?Chaîne}`

Si la variable est définie et non nulle, remplace l'expression par la valeur de *Variable*. Sinon, affiche un message de la forme :

```
Variable:: Chaîne
```

et sort du shell courant (sauf s'il s'agit du shell de connexion). Si *Chaîne* est omis, le shell affiche le message :

```
Variable: parameter null or not set
```

`${Variable+Chaîne}`

Si la variable est définie, remplace l'expression par la valeur de *Chaîne*. Sinon, lui substitue la chaîne nulle.

`${Variable:+Chaîne}`

Si la variable est définie et non nulle, remplace l'expression par la valeur de *Chaîne*. Sinon, lui substitue la chaîne nulle.

Lorsqu'il exécute une substitution conditionnelle, le shell n'évalue la *Chaîne* qu'au moment où il l'utilise comme chaîne de substitution. Ainsi, dans l'exemple suivant, le shell n'exécute la commande **pwd** que lorsque *d* est nul ou non spécifié :

```
echo ${d:-`pwd`}
```

## Paramètres positionnels

Lorsque vous exécutez une procédure shell, le shell crée implicitement des paramètres positionnels qui référencent les mots de la ligne de commande par leur position sur la ligne. Le mot en position 0 (nom de la procédure) est appelé **\$0**, le mot suivant (premier paramètre) **\$1**, etc, jusqu'à **\$9**. Pour référencer des paramètres au-delà de 9, passez par la commande intégrée AUCUN LIEN **shift**.

Pour redéfinir explicitement des paramètres positionnels, vous disposez de la commande intégrée AUCUN LIEN **set**.

**Remarque** : Lorsqu'une position n'est pas spécifiée, son paramètre positionnel est nul. Les paramètres positionnels sont globaux et peuvent donc être transmis aux procédures shell imbriquées.

## Substitution de noms de fichiers (shell Bourne)

Les paramètres de commande sont souvent des noms de fichiers. Vous pouvez générer automatiquement une liste de noms de fichier comme paramètres d'une ligne de commande. Pour ce faire, spécifiez un caractère que le shell identifie comme métacaractère. Si une commande contient ce caractère, le shell remplace le caractère par le nom de fichier enregistré dans un répertoire.

**Remarque** : Le shell Bourne n'autorise pas le développement de nom de fichier basé sur la classification des caractères par équivalence.

Il s'agit des caractères suivants :

- |        |                                                                                                                         |
|--------|-------------------------------------------------------------------------------------------------------------------------|
| *      | Correspond à n'importe quelle chaîne, chaîne nulle comprise.                                                            |
| ?      | Correspond à un seul caractère (quelconque).                                                                            |
| [...]  | Correspond à n'importe quel caractère entre crochets.                                                                   |
| [!...] | Correspond à n'importe quel caractère <i>autre que</i> ceux spécifiés entre crochets à la suite du point d'exclamation. |

Mise entre crochets, une paire de caractères séparés par un – (tiret), correspond à n'importe quel jeu de caractères compris lexicalement entre ces deux valeurs (selon l'ordre de classement binaire des caractères).

L'utilisation de trames souffre quelques restrictions. Si le premier caractère d'un nom de fichier est un . (point), il ne peut être mis en correspondance qu'avec une trame commençant aussi par un point. Par exemple, \* (astérisque) correspond aux fichiers `myfile` et `yourfile`, mais pas aux fichiers `.myfile` et `.yourfile`. Pour ces fichiers, utilisez, par exemple :

```
. *file
```

Si la recherche n'aboutit pas, c'est la trame elle-même qui est renvoyée en sortie.

Excluez des noms de fichiers et de répertoires les caractères \* (astérisque), ? point d'interrogation), [ (crochet gauche) et ] (crochet droit) : ils risquent de provoquer une boucle infinie lors des recherches sur la base de trames.

## Classes de caractères

Pour faire correspondre des noms de fichiers, vous disposez également des classes de caractères :

```
[[:charclass:]]
```

Avec ce format, le système établit une correspondance avec chacun des caractères appartenant à la classe spécifiée. Les classes définies correspondent aux sous-routines **ctype**. Il s'agit des classes suivantes :

| <b>Classe</b> | <b>Définition</b>                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------|
| <b>alnum</b>  | Caractères alphanumériques                                                                            |
| <b>alpha</b>  | Majuscules et minuscules                                                                              |
| <b>blank</b>  | Espace ou tabulation horizontale                                                                      |
| <b>cntrl</b>  | Caractères de contrôle                                                                                |
| <b>digit</b>  | Chiffres                                                                                              |
| <b>graph</b>  | Caractères graphiques                                                                                 |
| <b>lower</b>  | Minuscules                                                                                            |
| <b>print</b>  | Caractères imprimables                                                                                |
| <b>punct</b>  | Caractères de ponctuation                                                                             |
| <b>space</b>  | Espace, tabulation horizontale, retour chariot, nouvelle ligne, tabulation verticale ou page suivante |
| <b>upper</b>  | Lettres majuscules                                                                                    |
| <b>xdigit</b> | Chiffres hexadécimaux                                                                                 |

---

## Réacheminement des entrées/sorties (shell Bourne)

La plupart des commandes ignorent l'origine de leurs entrées et la destination de leurs sorties (clavier, l'écran ou fichier). Une commande peut donc être lancée aussi bien via le clavier que via un pipeline.

Les options ci-après peuvent apparaître n'importe où dans une commande simple. Elles peuvent aussi la précéder ou la suivre, mais ne peuvent lui être transmises.

|                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| < <i>Fichier</i>    | Ouvre le fichier comme entrée standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| > <i>Fichier</i>    | Ouvre le fichier comme sortie standard. Crée le fichier s'il n'existe pas, le tronque à une longueur nulle sinon.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| > > <i>Fichier</i>  | Ouvre le fichier comme sortie standard. Créer le fichier s'il n'existe pas, y ajoute les sorties sinon.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <<[-] <i>eofstr</i> | Lit toutes les lignes à partir de la variable <i>eofstr</i> jusqu'à la ligne ne contenant que <i>eofstr</i> ou un caractère de fin de fichier. Si un des caractères de <i>eofstr</i> est déclaré, le shell n'interprète ni ne développe aucun caractère des lignes d'entrée. Sinon, il exécute la substitution de commande et de variable et ignore le caractère nouvelle ligne déclaré ( <b>\new-line</b> ). Pour déclarer des caractères de la variable <i>eofstr</i> ou dans les lignes d'entrée, précédez-les d'une barre oblique inverse (\).<br><br>Si vous ajoutez un signe – (moins) à l'option de réacheminement <<, toutes les tabulations en tête de <i>eofstr</i> et des lignes d'entrée sont supprimées. |
| <& <i>Chiffre</i>   | Associe l'entrée standard au descripteur de fichier spécifié par <i>Chiffre</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| >& <i>Chiffre</i>   | Associe l'entrée standard au descripteur de fichier spécifié par <i>Chiffre</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <&-                 | Ferme l'entrée standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| >&-                 | Ferme la sortie standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

**Remarque :** Le shell restreint n'autorise pas la réacheminement des sorties.

Pour en savoir plus, reportez-vous à "Réacheminement des entrées/sorties", page 4-1 et Shell Bourne.

---

## Shell C

Le shell C est un interpréteur de commandes interactif et un langage de programmation. Il utilise une syntaxe proche du langage C. La commande **cs** démarre le shell C.

Lorsque vous vous connectez, le shell commence par rechercher le fichier système de configuration **/etc/csh.cshrc** et, s'il le trouve, exécute les commandes qu'il contient. Il exécute ensuite le fichier **/etc/csh.login** (si celui-ci est disponible). Il explore enfin votre répertoire personnel à la recherche des fichiers **.cshrc** et **.login** : ces fichiers (s'ils existent) contiennent des informations personnalisées sur l'exécution du shell Cfiles. Toutes les variables éventuellement définies dans les fichiers **.cshrc** et **.login** (de votre répertoire **\$HOME**) priment sur celles des fichiers **/etc/csh.cshrc** et **/etc/csh.login**. Ces derniers fichiers ne sont modifiables que par un utilisateur racine.

Les fichiers **/etc/csh.login** et **\$HOME/.login** ne sont exécutés qu'une fois, au moment de la connexion. Ils servent généralement à enregistrer les définitions des variables d'environnement, les commandes à exécuter à la connexion et les commandes définissant les caractéristiques du terminal.

Les fichiers **/etc/csh.cshrc** et **\$HOME/.cshrc** sont exécutés à la connexion et à chaque invocation de la commande **cs** ou d'un script shell C. Ils servent généralement à définir les caractéristiques du shell C, telles que les alias et les variables (history, noclobber, ignoreeof, etc.). Nous vous conseillons de n'utiliser que des commandes intégrées du shell C dans les fichiers **/etc/csh.cshrc** et **\$HOME/.cshrc**, pour réduire au minimum le délai de démarrage des scripts shell.

Cette section traite des points suivants :

- Règles d'utilisation, page 11-93.
- Traitement des signaux, page 11-93.
- Commandes du shell C, page 11-94.
  - Commandes intégrées (shell C), page 11-94.
  - Expressions et opérateurs (shell C), page 11-102.
  - Substitution de commandes (shell C), page 11-103.
  - Exécution des commandes non intégrées, page 11-104.
- Substitution d'historique (shell C), page 11-105.
  - Liste d'historique, page 11-105.
  - Spécification d'événement, page 11-106.
  - Guillemets et apostrophes, page 11-107.
- Substitution d'alias (shell C), page 11-108.
- Substitution de variables et de noms de fichiers (shell C), page 11-109.
  - Substitution de variable (shell C), page 11-109.
  - Substitution de nom de fichier (shell C), page 11-111.
  - Développement de nom de fichier, page 11-111.
  - Abréviations de nom de fichier, page 11-111.
  - Classes de caractères, page 11-112.
- Variables d'environnement (shell C), page 11-113.
- Réacheminement des entrées/sorties (shell C), page 11-116.
  - Flux de contrôle

- Contrôle des travaux (shell C), page 11-118.
- Voir aussi Shell C

## Règles d'utilisation

Voici les règles applicables au shell C :

- Les mots ne peuvent dépasser 1024 octets.
- Les listes d'arguments sont limitées à ARG\_MAX octets. La variable ARG\_MAX est définie dans le fichier **/usr/include/sys/limits.h**.
- Le nombre d'arguments d'une commande impliquant le développement de noms de fichiers est limité à 1/6ème du nombre de caractères autorisés dans une liste d'arguments.
- Les substitutions de commandes ne peuvent porter sur un nombre de caractères supérieur à celui autorisé dans une liste d'arguments.
- Pour détecter les boucles, le shell limite à 20 le nombre de substitutions d'alias sur une seule ligne.
- La commande **cs** ne prend pas en charge le développement de noms de fichiers basés sur la classification des caractères par équivalence.

## Traitement des signaux

Le shell C ignore normalement les signaux quit. Les travaux exécutés distinctement ne sont pas affectés par les signaux générés par le clavier (**INTERRUPT**, **QUIT** et **HANGUP**). Les autres signaux ont la valeur que le shell hérite de son parent. Vous pouvez contrôler le mode de gestion des signaux d'INTERRUPTION et de TERMINAISON dans les procédures shell au moyen de **onintr**. Les shells de connexion interceptent ou ignorent les signaux **TERMINATE** suivant leur configuration. Les autres shells les passent aux processus fils. En aucun cas, les interruptions (signaux **INTERRUPT**) ne sont autorisées lorsqu'un shell de connexion est en train de lire le fichier **.logout**.

---

## Commandes du shell C

Une commande simple est une séquence de mots, séparés par des espaces ou des tabulations.

Un *mot* est une séquence de caractères et/ou de chiffres, sans espaces non délimités. Sont également considérés comme des mots les caractères suivants, lorsqu'ils sont utilisés comme séparateurs de commandes ou caractères de terminaison :

```
& | ;
&& || << > >
< > ()
```

Ces caractères peuvent faire partie d'un autre mot. Faites-les précéder d'une barre oblique inverse ( pour que le shell ne les interprète pas comme des caractères spéciaux. Les chaînes encadrées d'apostrophes ( ' ' ), de guillemets ( " " ) ou d'apostrophes inverses ( ` ` ) peuvent également faire partie d'un mot. Les espaces, tabulations et caractères spéciaux ne constituent pas des mots distincts lorsqu'ils sont encadrés de ces signes. Vous pouvez également encadrer le caractère nouvelle ligne sous réserve de le faire précéder d'une barre oblique inverse ( \ ).

Le premier mot d'une commande simple (numéroté 0) précise généralement le nom de la commande. Les autres mots, à quelques exceptions près, sont passés à cette commande. Si la commande spécifie un fichier exécutable qui est un programme compilé, le shell exécute immédiatement le programme. Si le fichier exécutable n'est pas un programme compilé, le shell suppose qu'il s'agit d'une procédure shell. Le shell lance alors une autre instance de lui-même (un sous-shell) pour lire le fichier et exécuter les commandes qu'il contient.

Ce chapitre traite des points suivants :

- Commandes intégrées (shell C), page 11-94.
- Expressions et opérateurs (shell C), page 11-102.
- Substitution de commande (shell C), page 11-103.
- Exécution des commandes non intégrées, page 11-104.
- Voir aussi Shell C

## Commandes intégrées (shell C)

Les commandes intégrées sont exécutées à l'intérieur du shell. Si une commande de ce type constitue un élément d'un pipeline (excepté la dernière), elle est exécutée dans un sous-shell.

**Remarque :** Si vous entrez une commande à l'invite du shell C, le système recherche d'abord une commande intégrée. S'il n'en trouve pas, il explore les répertoires spécifiés par la variable shell **path**, à la recherche d'une commande système. Notez que certaines commandes shell C intégrées portent le même nom que des commandes système : ces commandes n'ont toutefois pas nécessairement la même fonction. Reportez-vous à la description de la commande concernée.

Si vous lancez, à partir du shell, une procédure dont les premiers caractères sont `#!/ShellPathname`, le shell C lance le shell spécifié dans le commentaire pour exécuter la procédure. Sinon, il exécute le shell par défaut (le shell lié à `/usr/bin/sh`). Dans ce cas, les commandes intégrées du shell C ne sont pas reconnues. Pour que le système exécute ces commandes, commencez la procédure par la ligne `#!/usr/bin/csh`.

Reportez-vous à la Liste des commandes intégrées (shell C).

## Description des commandes

Voici la liste des commandes intégrées du shell C :

|                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>alias</b> [ <i>Nom</i> [ <i>ListeMots</i> ]] | Affiche la liste de tous les alias si aucun paramètre n'est spécifié. Sinon, affiche l'alias du nom indiqué. Si <i>ListeMots</i> est une liste de mots, elle est affectée comme alias de la variable <i>Nom</i> . Ce nom ne peut être <b>alias</b> ou <b>unalias</b> .                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>bg</b> [% <i>Travail</i> ...]                | Place le <i>Travail</i> spécifié ou le travail courant en arrière-plan, et poursuit, le cas échéant, son exécution.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>break</b>                                    | Reprend l'exécution après la fin de la commande <b>foreach</b> ou <b>while</b> la plus proche.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>breaksw</b>                                  | Effectue une interruption (break) à partir d'une commande <b>switch</b> . Reprend après la commande <b>endsw</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>case</b> <i>Label</i> :                      | Définit un <i>Label</i> dans une commande <b>switch</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>cd</b> [ <i>Nom</i> ]                        | Equivalent à la commande <b>chdir</b> (voir description ci-dessous).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>chdir</b> [ <i>Nom</i> ]                     | Passe au répertoire spécifié par <i>Nom</i> (ou, à défaut, à votre répertoire personnel). Si <i>Nom</i> n'est pas un sous-répertoire du répertoire courant et ne commence pas par une barre oblique (/), un point suivi d'une barre oblique (./) ou deux points consécutifs suivis d'une barre oblique (.../), le shell vérifie chaque élément de la variable <b>cdpath</b> pour déterminer s'il contient un sous-répertoire correspondant à <i>Nom</i> . Dans la négative, si <i>Nom</i> est une variable shell commençant par une barre oblique (/), le shell tente cette action pour déterminer s'il s'agit d'un répertoire. Les commandes <b>chdir</b> et <b>cd</b> sont équivalentes. |
| <b>continue</b>                                 | Reprend l'exécution à l'instruction <b>end</b> de la commande <b>foreach</b> ou <b>while</b> la plus proche.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <b>default</b> :                                | Libelle le cas <b>default</b> d'une instruction <b>switch</b> . Le cas <b>default</b> doit suivre tous les autres labels <b>case</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>dirs</b>                                     | Affiche la pile de répertoires.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>echo</b>                                     | Ecrit les chaînes de caractères sur la sortie standard du shell.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>else</b>                                     | Exécute les commandes qui suivent le second <b>else</b> dans une séquence de commandes <b>if</b> ( <i>Expression</i> ) <b>then</b> ... <b>else</b> <b>if</b> ( <i>Expression2</i> ) <b>then</b> ... <b>else</b> ... <b>endif</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>end</b>                                      | Donne successivement à <i>Nom</i> la valeur de chaque élément de la <i>Liste</i> et exécute la séquence de commandes entre les instructions <b>foreach</b> et <b>end</b> correspondantes. Les instructions <b>foreach</b> et <b>end</b> doivent se trouver seules sur des lignes distinctes.<br><br>L'instruction <b>continue</b> permet de reprendre la boucle, <b>break</b> de l'interrompre. Lorsque la commande est lue à partir du terminal, la boucle est d'abord lue une fois et un point d'interrogation ? s'affiche pour vous inviter à entrer les commandes. Les commandes à l'intérieur des boucles, entrées à l'invite ?, ne sont pas placées dans la liste d'historique.      |

|                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>endif</b>                        | Si <i>Expression</i> est vraie, exécute les commandes qui suivent le premier <b>then</b> . Si <b>else if</b> <i>Expression2</i> est vraie, exécute les commandes qui suivent le deuxième <b>then</b> . Si <b>else if</b> <i>Expression2</i> est fausse, exécute les <i>Commandes</i> qui suivent le <b>else</b> . Vous pouvez spécifier autant de paires <b>else-if</b> que vous le souhaitez. Une seule instruction <b>endif</b> est requise. Le segment <b>else</b> est facultatif. Les mots <b>else</b> et <b>endif</b> doivent se trouver en tête de ligne. Le segment <b>if</b> doit se trouver seul sur une ligne ou suivre une commande <b>else</b> .                                                                                                                                                                        |
| <b>endsw</b>                        | Compare les labels <b>case</b> avec les valeurs de <i>Chaîne</i> . <i>Chaîne</i> est d'abord développée (commandes et noms de fichiers). Utilisez les caractères joker * (astérisque), ? (point d'interrogation), et [ . . . ] (points de suspension entre crochets) dans les labels <b>case</b> , dont les variables sont développées. Si aucun des labels ne correspond, l'exécution démarre après le label <b>default</b> . Les labels <b>case</b> et <b>default</b> doivent apparaître en début de ligne. La commande <b>breaksw</b> entraîne la poursuite de l'exécution après la commande <b>endsw</b> . Sinon, le contrôle peut passer aux labels <b>case</b> et <b>default</b> , comme dans le langage C. Si aucun label ne correspond et qu'il n'existe pas de <b>default</b> , l'exécution se poursuit après la commande. |
| <b>eval</b> <i>Paramètre . . .</i>  | Lit la valeur de la variable <i>Paramètre</i> comme entrée du shell et exécute la commande résultante dans le contexte shell courant. Cette commande permet d'exécuter des commandes générées par le résultat de substitution de commandes ou de variables, dans la mesure où l'analyse intervient avant ces substitutions.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>exec</b> <i>Commande</i>         | Exécute la <i>Commande</i> spécifiée, à la place du shell courant.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>exit</b> [( <i>Expression</i> )] | Quitte le shell avec la valeur de la variable shell <b>status</b> (si aucune <i>Expression</i> n'est spécifiée) ou avec la valeur <i>Expression</i> spécifiée.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>fg</b> [% <i>Travail</i> ...]    | Amène au premier plan le travail courant ou le <i>travail</i> spécifié et, le cas échéant, poursuit son exécution.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>glob</b> <i>Liste</i>            | Affiche une liste avec développement de l'historique, des variables et des noms de fichiers. Les mots sont séparés par un caractère nul et aucun retour chariot n'est inséré à la fin.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>goto</b> <i>Mot</i>              | Poursuit l'exécution après la ligne spécifiée par <i>Mot</i> . <i>Mot</i> est développé (nom de fichier et commande) pour devenir une chaîne de la forme définie par la variable <i>Label</i> . Le shell repasse sur l'entrée autant que possible à la recherche d'une ligne de la forme <i>Label:</i> , éventuellement précédée d'espaces ou de tabulations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>hashstat</b>                     | Affiche des statistiques indiquant le pourcentage de recherches abouties de commandes, par la table de hachage.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>history</b> [-r   -h] [n]                                                       | Affiche les listes des événements de l'historique. Cet affichage s'effectue par ordre chronologique. Si <i>n</i> est spécifié, seuls les <i>n</i> événements les plus récents sont affichés. <b>-r</b> inverse l'ordre d'affichage (événement le plus récent le premier). <b>-h</b> affiche la liste sans chiffres en tête. Servez-vous en pour générer des fichiers exploitables avec l'indicateur <b>-h</b> de la commande <b>source</b> .                                                                                  |
| <b>if</b> ( <i>Expression</i> ) <i>Commande</i>                                    | Exécute la <i>Commande</i> spécifiée (arguments compris) si <i>Expression</i> est vraie. Les substitutions sur la variable <i>Commande</i> ont lieu plus tôt, en même temps que pour le reste de l'instruction <b>if</b> . <i>Commande</i> doit être une commande simple (et non un pipeline, une liste de commandes ou une liste de commandes entre parenthèses).<br><br><b>Remarque</b> : Les réacheminements d'entrée/sortie interviennent même si <i>Expression</i> est fausse et que <i>Commande</i> n'est pas exécutée. |
| <b>jobs</b> [-l]                                                                   | Liste les travaux actifs. Avec <b>-l</b> (l minuscule), indique, outre les numéros et les ID de <b>travaux</b> , les ID processus.                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>kill</b> -l   [[- <i>Signal</i> ]<br>% <i>Travail</i> . . .   <i>PID</i> . . .] | Envoie le signal <b>TERM</b> (terminate), ou celui spécifié par <i>Signal</i> , au <i>Travail</i> ou au <i>PID</i> (process) spécifié. Ces signaux peuvent être un nom ou un nombre, définis dans le fichier <b>/usr/include/sys/signal.h</b> (amputés de leur préfixe <b>SIG</b> ). L'indicateur <b>-l</b> (L minuscule) liste les noms de signaux.                                                                                                                                                                          |

**limit** [-h] [*Ressource*  
[*Usage-max*]]

Limite l'usage de la ressource par le processus courant et les processus qu'il crée. Ces limites sont définies dans le fichier **/etc/security/limits**. Les ressources contrôlables sont : temps d'utilisation CPU, la taille des fichiers, la taille des données, la taille du vidage de la mémoire centrale et l'utilisation de la mémoire. Les valeurs maximales pour l'affectation de ces ressources sont définies via la commande **mkuser**, au moment de l'ajout de l'utilisateur au système. Elles peuvent être modifiées à l'aide de la commande **chuser**.

Les limites sont d'ordre logiciel ou matériel. Les utilisateurs peuvent définir des limites logicielles, jusqu'aux plafonds imposés par le matériel. Cependant, seul un utilisateur racine est habilité à modifier les limites matérielles. L'indicateur **-h** affiche les limites matérielles.

Si *Usage-Max* n'est pas spécifié, la commande **limit** affiche la limite courante de la ressource spécifiée. Si *Ressource* n'est pas spécifiée, elle affiche les limites courantes de toutes les ressources. Pour en savoir plus sur les ressources contrôlées par la commande **limit**, reportez-vous aux sous-routines **getrlimit**, **setrlimit** et **vlimit** dans le manuel *AIX Technical Reference, Volume 1: Base Operating System and Extensions*.

Le paramètre *MaxUse* relatif au temps d'utilisation CPU est au format hh:mm:ss. Pour les autres ressources, il est spécifié par un nombre en virgule flottante ou par un entier éventuellement suivi d'une unité de mesure. Cette unité de mesure peut être : k ou kilo-octets (1 024 octets), m ou mega-octets, ou b ou blocs (unités utilisées par la sous-routine **ulimit** comme expliqué dans le manuel *AIX Technical Reference, Volume 2: Base Operating System and Extensions*). Par défaut, la taille des ressources est indiquée en unités k. Pour les noms de ressources comme pour les unités de mesure, des préfixes non ambigus suffisent.

**Remarque** : Cette commande ne limite la mémoire physique disponible pour un process que si d'autres process actifs ont besoin de mémoire système.

**login**

Met fin à un shell de connexion et le remplace par une occurrence de la commande **/usr/bin/login**. C'est un moyen de se déconnecter (cette commande a été incluse pour des raisons de compatibilité avec les commandes **ksh** et **bsh**).

**logout**

Termine un shell de connexion. Particulièrement utile si l'option **ignoreeof** est active.

**nice** [+n] [*Commande*]

Si n n'est pas spécifié, définit à 24 la priorité d'exécution des commandes dans le shell. Si +n est spécifié, définit la priorité au nombre spécifié. Si l'indicateur +n et *Commande* sont spécifiés, exécute *Commande* avec une priorité de 24 plus n. Si vous détenez les droits de l'utilisateur racine, vous pouvez indiquer un nombre négatif avec **nice**. *Commande* est toujours exécutée dans un sous-shell, et les restrictions applicables aux commandes d'une instruction **if** simple valent pour cette commande.

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>nohup</b> [ <i>Commande</i> ]            | Si <i>Commande</i> n'est pas spécifiée, les arrêts ( <b>hangups</b> ) sont ignorés pour toute la suite du script. Si <i>Commande</i> est spécifiée, ils ne sont ignorés que pour cette <i>Commande</i> . Pour exécuter un pipeline ou une liste de commandes, placez-le (–la) dans un script shell, accordez à celui-ci des droits d'exécution et définissez le shell comme valeur de <i>Commande</i> . Tous les process exécutés en arrière-plan avec & (perluète) sont protégés : ils ne recevront pas de signal <b>hangup</b> lorsque vous vous déconnecterez. Ils restent néanmoins susceptibles de recevoir des signaux <b>hangup</b> explicites, sauf si vous utilisez l'instruction <b>nohup</b> . |
| <b>notify</b> [% <i>Travail</i> . . . ]     | Indique au shell de vous prévenir (de façon asynchrone) lorsque l'état du travail courant ou du travail spécifié change. Normalement, le shell émet un avertissement juste avant d'afficher l'invite. Cette fonction est automatiquement activée si la variable shell <b>notify</b> est spécifiée.                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>onintr</b> [–   <i>Label</i> ]           | Contrôle l'action du shell au niveau des interruptions. En l'absence d'argument, restaure l'action par défaut (arrêt du script ou retour au niveau entrée des commandes). Avec l'indicateur – (tiret), ignore toutes les interruptions. Si <i>Label</i> est spécifié, le shell exécute l'instruction <b>goto</b> <i>Label</i> à réception d'une interruption ou lorsqu'un process enfant se termine suite à une interruption. Dans tous les cas, si le shell est exécuté séparément et que les interruptions sont ignorées, l'instruction <b>onintr</b> n'a pas de sens : les interruptions sont toujours ignorées du shell et de toutes les commandes appelées.                                          |
| <b>popd</b> [+ <i>n</i> ]                   | Détruit la pile de répertoires et revient au nouveau premier répertoire. Si vous spécifiez + <i>n</i> , la commande rejette la <i>nième</i> entrée de la pile. Les éléments de la pile sont numérotés de haut en bas, à partir de 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| <b>pushd</b> [+ <i>n</i>   <i>Nom</i> ]     | Sans arguments, intervertit les deux premiers éléments de la pile des répertoires. Avec la variable <i>Nom</i> , passe au nouveau répertoire en décalant l'ancien répertoire courant (indiqué dans la variable shell <b>pwd</b> ) dans la pile. Avec la variable + <i>n</i> , place le <i>nième</i> élément de la pile en tête, et en fait le répertoire courant. Les éléments de la pile sont numérotés de haut en bas, à partir de 0.                                                                                                                                                                                                                                                                   |
| <b>rehash</b>                               | Recalcule la table d'adressage du contenu des répertoires dans la variable shell <b>path</b> . Cette action est requise si de nouvelles commandes sont ajoutées aux répertoires dans la variable <b>path</b> pendant que vous êtes connecté. La commande <b>rehash</b> n'est obligatoire que si des commandes sont ajoutées dans un répertoire utilisateur ou si un changement intervient au niveau du contenu d'un répertoire système.                                                                                                                                                                                                                                                                   |
| <b>repeat</b> <i>Compte</i> <i>Commande</i> | Exécute la <i>Commande</i> spécifiée, soumise aux mêmes restrictions que l'instruction <b>if</b> , autant de fois que spécifié par <i>Compte</i> .<br><br><b>Remarque</b> : Les réacheminements d'E/S interviennent une et une seule fois, même si <i>Compte</i> vaut 0 (zéro).                                                                                                                                                                                                                                                                                                                                                                                                                           |

|                                                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>set</b> [[ <i>Nom</i> [ <i>n</i> ]] [= <i>Mot</i> ]]  <br>[ <i>Nom</i> = ( <i>Liste</i> )] | Sans argument, affiche la valeur de toutes les variables shell (sous forme de liste entre parenthèses pour les variables composées de plusieurs mots). Si <i>Nom</i> seul est spécifié, affecte à <i>Nom</i> la chaîne nulle. Sinon, affecte à <i>Nom</i> la valeur de <i>Mot</i> ou de la liste de mots spécifiés dans <i>Liste</i> . Si <i>n</i> est précisé, affecte au <i>n</i> ème élément de <i>Nom</i> (s'il existe) la valeur de <i>Mot</i> . Dans tous les cas, la valeur est développée (nom de fichier et commande). Les arguments peuvent être répétés pour affecter plusieurs valeurs via une seule commande <b>set</b> . Le développement de variables intervient pour tous les arguments, avant toute affectation. |
| <b>setenv</b> <i>Nom</i> <i>Valeur</i>                                                        | Affecte à la variable d'environnement spécifiée par <i>Nom</i> la valeur <i>Valeur</i> (chaîne unique). Les variables d'environnement les plus courantes ( <b>USER</b> , <b>TERM</b> , <b>HOME</b> et <b>PATH</b> ) sont automatiquement importées et exportées à partir des variables shell C <b>user</b> , <b>term</b> , <b>home</b> et <b>path</b> . Il est inutile d'exécuter l'instruction <b>setenv</b> pour ces variables.                                                                                                                                                                                                                                                                                                 |
| <b>shift</b> [ <i>Variable</i> ]                                                              | Décale vers la gauche les éléments de la variable shell <b>argv</b> ou de la <i>Variable</i> spécifiée. Une erreur se produit si <b>argv</b> n'est pas définie ou contient moins d'un mot.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>source</b> [-h] <i>Nom</i>                                                                 | Lit les commandes spécifiées par <i>Nom</i> . Vous pouvez imbriquer des commandes <b>source</b> , mais si les imbrications sont trop poussées, vous risquez de déborder des descripteurs de fichiers. Une erreur dans une commande <b>source</b> met fin à toutes les autres commandes <b>source</b> imbriquées. Normalement, les entrées effectuées au cours de l'exécution de <b>source</b> ne sont pas enregistrées dans la liste d'historique. L'indicateur <b>-h</b> place la commande dans l'historique sans l'exécuter.                                                                                                                                                                                                    |
| <b>stop</b> [% <i>Travail</i> ...]                                                            | Arrête le travail courant ou le <i>Travail</i> spécifié, exécuté en arrière-plan.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>suspend</b>                                                                                | Met fin au shell (équivalent à un signal <b>STOP</b> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>switch</b> ( <i>chaîne</i> )                                                               | Commence une <i>Chaîne case</i> de <b>switch</b> ( <i>chaîne</i> ) : ...<br>séquence de commande <b>breaksw default</b> : ... <b>breaksw endsw</b> . Cette séquence compare chaque label avec les valeurs de <i>Chaîne</i> . Si aucun des labels ne correspond, l'exécution démarre après le label default.                                                                                                                                                                                                                                                                                                                                                                                                                       |

**time** [*Commande*]

Contrôle le minutage automatique des commandes. Si *Commande* n'est pas spécifiée, affiche un récapitulatif du temps consommé par le shell et ses enfants. Si *Commande* est spécifiée, la commande est minutée. Le shell affiche ensuite un récapitulatif, comme décrit pour la variable shell **time**. Si nécessaire, un shell supplémentaire est créé, pour afficher les statistiques à la fin de la commande.

Voici un exemple combinant les commandes **time** et **sleep** :

```
time sleep
```

Le résultat de cette commande est du type :

```
0.0u 0.0s 0:00 100% 44+4k 0+0io 0pf+0w
```

Les sept zones de cette sortie sont les suivantes :

| Zone             | Description                                                                                                                  |
|------------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>Première</b>  | Temps CPU (en secondes) dédié au process utilisateur.                                                                        |
| <b>Deuxième</b>  | Temps CPU (en secondes) consommées par le noyau pour le process utilisateur.                                                 |
| <b>Troisième</b> | Temps écoulé pour la commande.                                                                                               |
| <b>Quatrième</b> | Temps CPU total (utilisateur + système) exprimé en pourcentage du temps écoulé.                                              |
| <b>Cinquième</b> | Somme de la moyenne de la taille de mémoire partagée et de la moyenne de la taille de mémoire non partagée (en kilo-octets). |
| <b>Sixième</b>   | Nombre d'opérations d'entrée/sortie de blocs.                                                                                |
| <b>Septième</b>  | Somme des erreurs de pagination et du nombre de permutations (swaps).                                                        |

**umask** [*Valeur*]

Détermine les droits d'accès au fichier. Cette *Valeur*, associée aux droits du process créateur, détermine les droits sur un fichier au moment de sa création (022, par défaut). A défaut de *Valeur* spécifiée, la valeur en cours est affichée.

**unalias** \*|*Trame*

Supprime les alias dont le nom correspond à la *Trame*. La commande **unalias** \* supprime tous les alias. L'absence d'alias ne provoque pas d'erreur.

**unhash**

Désactive l'usage de la table de hachage pour la localisation des programmes en cours.

**unlimit** [-h][*Ressource*]

Supprime la limitation imposée à la variable *Ressource* . Si *Ressource* n'est pas spécifiée, toutes les limitations sont supprimées. Reportez-vous à la description de la commande **limit**.

L'indicateur **-h** supprime les limites matérielles correspondantes. Seul un utilisateur racine est habilité à effectuer cette opération.

**unset** \*|*Trame*

Supprime toutes les variables dont le nom correspond à la *Trame*. **unset** \* supprime toutes les variables. L'absence de variables ne provoque pas d'erreur.

|                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>unsetenv</b> <i>Trame</i>                                   | Supprime de l'environnement toutes les variables dont le nom correspond à la <i>Trame</i> . (Reportez-vous à la commande <b>setenv</b> .)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>wait</b>                                                    | Attend tous les travaux en arrière-plan. Si le shell est interactif, un INTERRUPT (Ctrl-C généralement) met fin à l'attente. Le shell affiche ensuite les noms et numéros des travaux en suspens.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>while</b> ( <i>Expression</i> )<br><i>Commande. . . end</i> | Evalue les commandes entre les instructions <b>while</b> et <b>end</b> correspondantes, tant que la variable <i>Expression</i> est différente de zéro. L'instruction <b>break</b> permet d'interrompre la boucle, <b>continue</b> de la poursuivre. Les instructions <b>while</b> et <b>end</b> doivent se trouver chacune sur une ligne distincte. Si l'entrée provient d'un terminal, des invites s'affichent après <b>while</b> ( <i>Expression</i> ), comme pour l'instruction <b>foreach</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| @ [ <i>Nom</i> [ <i>n</i> ] = <i>Expression</i> ]              | Sans argument, affiche le valeurs de toutes les variables shell. Sinon, affecte <i>Nom</i> à la variable <i>Expression</i> . Si l'expression contient un caractère < (inférieur à), > (supérieur à), & (perluète) ou   (barre verticale), cette partie de l'expression doit être placée entre parenthèses. Si <i>n</i> est précisé, affecte au <i>nième</i> élément de <i>Nom</i> la valeur de <i>Expression</i> . La variable <i>Nom</i> et son <i>nième</i> élément doivent exister.<br><br>Les opérateurs du langage C, tels que *= (astérisque, signe égal) et += (signe plus, signe égal), sont disponibles. L'espace séparant la variable <i>Nom</i> de l'opérateur d'affectation est facultatif. Les espaces séparant les éléments de <i>Expression</i> sont en revanche obligatoires (faute de quoi, il seraient interprétés comme un seul mot). Les suffixes spéciaux, ++ (double plus) et -- (double tiret) augmentent et diminuent, respectivement, la valeur de la variable <i>Nom</i> . |

## Expressions et opérateurs du shell C

La commande intégrée @ et les instructions **exit**, **if**, et **while** comportent des opérateurs semblables à ceux du langage C, avec les mêmes règles de précedence. Voici les opérateurs disponibles :

| Opérateur   | Signification                                  |
|-------------|------------------------------------------------|
| ()          | change la précedence                           |
| ~           | complément                                     |
| !           | négation                                       |
| * / %       | multiplie, divise, modulo                      |
| + -         | ajoute, soustrait                              |
| << >>       | décale à gauche, décale à droite               |
| <= >= < >   | opérateurs relationnels                        |
| == != =~ !~ | comparaison de chaîne/correspondance de trames |
| &           | "et" au niveau bit                             |
| ^           | "ou exclusif" au niveau bit                    |
|             | "ou inclusif" au niveau bit                    |
| &&          | "et" logique                                   |
|             | "ou" logique                                   |

Les opérateurs sont classés par ordre de précedence décroissante ¶ de haut en bas, et de gauche à droite.

**Remarque** : Les opérateurs + et – sont associatifs, en partant de la droite. Ainsi,  $a + b - c$  est évalué :

$$a + (b - c)$$

et non :

$$(a + b) - c$$

Les opérateurs ==, !=, =~, et !~ comparent des arguments chaîne. Tous les autres opèrent sur des nombres. Les opérateurs =~ et !~ sont semblables à == et !=, excepté que l'expression de droite est une *trame* à laquelle est comparé l'opérande de gauche. Ce qui réduit la fréquence des recours à l'instruction **switch** dans les procédures shell, lorsque seule la correspondance des caractères génériques est réellement requise.

Les opérateurs logiques || (ou) et && (et) sont également disponibles. Ils permettent, par exemple, de tester l'appartenance d'une variable à un intervalle :

```
if ($#argv > 2 && $#argv < 7) then
```

Ici, le nombre d'arguments doit être supérieur à 2 et inférieur à 7.

Les chaînes commençant par 0 (zéro) sont interprétées comme des nombres octaux. Les arguments nuls ou manquants sont interprétés comme 0. Toutes les expressions ont pour résultat des chaînes représentant des nombres décimaux. Notez que les deux éléments d'une expression peuvent se trouver dans un même mot. Excepté lorsqu'ils se trouvent à côté d'éléments syntaxiquement significatifs pour l'analyseur (& | < > ( )), les éléments d'expression doivent être encadrés d'espaces.

On peut trouver dans des expressions, à titre d'opérandes, des exécutions de commandes entre accolades et des interrogations de fichier de la forme (**–opérateur** *Nomfichier*), où **opérateur** est l'un des caractères suivants :

|          |                    |
|----------|--------------------|
| <b>r</b> | accès en lecture   |
| <b>w</b> | accès en écriture  |
| <b>x</b> | accès en exécution |
| <b>e</b> | existence          |
| <b>o</b> | propriété          |
| <b>z</b> | taille zéro        |
| <b>f</b> | fichier ordinaire  |
| <b>d</b> | répertoire         |

*Nomfichier* est développé (commande et nom de fichier), puis testé pour vérifier s'il a la relation spécifiée à l'utilisateur réel. Si le fichier n'existe pas ou est inaccessible, toutes les requêtes retournent la valeur faux (0). Si la commande aboutit, l'interrogation retourne la valeur vrai (1). Sinon, elle renvoie la valeur faux (0). Si vous souhaitez davantage de précisions sur l'état, lancez la commande en dehors d'une expression et examinez la variable shell **status**.

## Substitution de commandes (shell C)

Lors d'une *substitution de commandes*, le shell exécute une commande et remplace cette commande par son résultat. Pour exécuter des substitutions de commandes dans le shell C, encadrez la commande ou la chaîne de commandes par ` ` (apostrophes inverses). Le shell décompose normalement le résultat de la commande en mots séparés au niveau des espaces, des tabulations et des caractères nouvelle ligne. Puis il remplace la commande d'origine par cette sortie.

Dans l'exemple suivant, les `` (apostrophes inverses) encadrant la commande **date** indiquent que le résultat de la commande sera remplacé :

```
echo The current date and time is: `date`
```

Le résultat de cette commande est du type :

```
The current date and time is: Wed Apr 8 13:52:14 CDT 1992
```

Le shell C exécute sélectivement les substitutions de commandes sur les arguments des commandes intégrées. Ce qui signifie qu'il ne développe pas les parties des expressions qui ne sont pas évaluées. Dans le cas de commandes non intégrées, le shell effectue la substitution du nom de la commande en dehors de la liste d'arguments. La substitution a lieu dans un shell fils du shell principal, après réacheminement des entrées/sorties.

Si une chaîne de commandes est encadrée de guillemets (" "), le shell interprète comme séparateurs de mots les seuls caractères nouvelle ligne, préservant les espaces et les tabulations à l'intérieur du mot. Dans tous les cas, l'unique caractère nouvelle ligne final ne génère pas de nouveau mot.

## Exécution des commandes Shell C non intégrées

Lorsque le shell C détermine qu'une commande n'est pas une commande intégrée, il tente de l'exécuter via l'appel système **execv**. Chaque mot de la variable shell **path** correspond à un répertoire à partir duquel le shell tente de lancer la commande. En l'absence des options **-c** et **-t**, le shell ventile les noms de ces répertoires dans une table interne. Le shell ne tente l'exécution de **exec** dans un répertoire que s'il y a une chance que la commande s'y trouve. Si vous désactivez ce mécanisme au moyen de la commande **unhash** ou que vous associez au shell l'option **-c** ou **-t**, le shell est concaténé avec le nom de la commande considérée pour former le chemin d'accès à un fichier. Le shell effectue la même opération pour chaque élément de répertoire de la variable **path** ne commençant pas par / (barre oblique). Il tente ensuite d'exécuter la commande.

Les commandes entre parenthèses sont toujours exécutées dans un sous-shell. Par exemple :

```
(cd ; pwd) ; pwd
```

affiche le répertoire personnel sans modifier l'emplacement du répertoire courant. En revanche, la commande :

```
cd ; pwd
```

définit le répertoire personnel comme répertoire courant. Les commandes entre parenthèses sont le plus souvent utilisées pour empêcher la commande **chdir** d'agir sur le shell courant.

Si le fichier est accessible en exécution, sans être un fichier exécutable binaire, le shell suppose qu'il s'agit d'un fichier de commandes shell et lance un nouveau shell pour le lire.

S'il existe un alias du shell, les mots de l'alias sont insérés en préfixe de la liste d'arguments, pour former la commande shell. Le premier mot de l'alias doit être le chemin d'accès complet au shell.

---

## Substitution d'historique (shell C)

La substitution d'historique permet de modifier des mots d'une commande précédente pour en créer de nouvelles. Cette fonction simplifie la reprise de commandes et d'arguments d'une commande précédente pour les intégrer à la commande courante, et permet également de corriger aisément les fautes de frappe dans une commande.

Les substitutions d'historique commencent par un point d'exclamation (!) et peuvent se trouver n'importe où sur la ligne de commande, sachant qu'elles ne peuvent être imbriquées (c'est-à-dire qu'une substitution d'historique ne peut en contenir une autre). Vous pouvez faire précéder le point d'exclamation (!) d'une barre oblique inversée (\) pour annuler la signification spéciale du point d'exclamation. En outre, si vous placez le point d'exclamation avant un espace, une tabulation, un caractère nouvelle ligne, un signe égal (=) ou une parenthèse ouvrante ((), la substitution n'a pas lieu.

Une substitution d'historique peut également être générée par un ^ (caret) en début de ligne. Le shell envoie toute ligne contenant des substitutions d'historique à la station de travail avant d'exécuter la ligne.

Cette section traite des points suivants :

- Liste d'historique, page 11-105.
- Spécification d'événement, page 11-106.
- Guillemets et apostrophes, page 11-107.

## Liste d'historique

La liste d'historique sauvegarde les commandes lues par le shell, constituées de un ou de plusieurs mots. La substitution d'historique réintroduit des séquences de mots issues de ces commandes sauvegardées, dans le flux des entrées.

La variable shell **history** contrôle la taille de la liste historique. Vous devez définir cette variable soit dans le fichier **.cshrc**, soit sur la ligne de commande avec la commande intégrée **set**. La commande précédente est retenue quelle que soit la valeur de la variable **history**. Les commandes de l'historique sont numérotées à partir de 1. La commande intégrée **history** génère une sortie du type :

```
9 write michael
10 ed write.c
11 cat oldwrite.c
12 diff *write.c
```

Le shell affiche les chaînes de commandes avec leur numéro d'événement. Il est généralement inutile de se servir des numéros pour faire référence aux événements, mais vous pouvez intégrer le numéro de l'événement en cours à l'invite système en plaçant un point d'exclamation (!) dans la chaîne d'invite affectée à la variable d'environnement **prompt**.

Une référence historique complète contient une spécification d'événement, un descripteur de mot, et un ou plusieurs modificateurs, selon le format général suivant :

```
Événement [.]Mot :Modificateur[:Modificateur] . . .
```

**Remarque** : Seul un mot peut être modifié. Les chaînes contenant des espaces ne sont pas admises.

Dans l'exemple de sortie précédent, l'événement courant porte le numéro 13. En conservant cet exemple, voici des références à des événements précédents :

|        |                                                                |
|--------|----------------------------------------------------------------|
| !10    | Événement numéro 10.                                           |
| !-2    | Événement numéro 11 (événement courant moins 2).               |
| !d     | Mot commençant par <code>d</code> (événement numéro 12).       |
| !?mic? | Mot contenant la chaîne <code>mic</code> (événement numéro 9). |

Ces formats, sans autre modification, réintroduisent simplement les mots des événements spécifiés, séparés par de simples espaces. Cas particuliers : !! (double point d'exclamation) fait référence à la commande précédente. La commande !! seule sur une ligne d'entrée relance la commande précédente.

## Spécification d'événement

Pour sélectionner des mots d'un événement, faites suivre sa spécification d'un : (deux points) et de l'un des descripteurs de mots suivants (les mots d'une ligne d'entrée sont numérotés à partir de 0) :

|            |                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------|
| <b>0</b>   | Premier mot (nom de la commande).                                                                       |
| <b>n</b>   | <i>n</i> ième argument.                                                                                 |
| <b>^</b>   | Premier argument.                                                                                       |
| <b>\$</b>  | Dernier argument.                                                                                       |
| <b>%</b>   | Mot correspondant à une chaîne <i>?Chaîne?</i> le précédant immédiatement.                              |
| <b>x-y</b> | Intervalle de mots entre le <i>x</i> ième et le <i>y</i> ième mot.                                      |
| <b>-y</b>  | Intervalle de mots entre le premier (0) et le <i>y</i> ième mot.                                        |
| <b>*</b>   | Premier mot du dernier argument, ou rien s'il y a un seul mot (le nom de la commande) dans l'événement. |
| <b>x*</b>  | <i>x</i> ième et le dernier argument.                                                                   |
| <b>x-</b>  | Equivalent à <i>x*</i> , avec omission du dernier argument.                                             |

Vous pouvez omettre les deux points entre la spécification d'événement et le descripteur de mot si celui-ci commence par **^** (caret), **\$** (dollar), **\*** (astérisque), **-** (tiret) ou **%** (pourcentage). Vous pouvez également placer une séquence de modificateurs parmi les suivants après le descripteur facultatif du mot, en les faisant précéder de deux points :

|          |                                                                            |
|----------|----------------------------------------------------------------------------|
| <b>h</b> | Supprime l'extension finale d'un chemin d'accès, en conservant le début.   |
| <b>r</b> | Supprime un élément final <code>.xxx</code> , en conservant le nom racine. |
| <b>e</b> | Supprime tout sauf le suffixe <code>.xxx</code> .                          |

**s/AncienMot/NouveauMot/**

Affecte à *AncienMot* la valeur de *NouveauMot*.

La partie gauche d'une substitution ne constitue pas une trame, au sens d'une chaîne reconnue par un éditeur. Il s'agit en fait d'un mot, isolé, sans espace. Normalement, une barre oblique (/) délimite le mot d'origine (*AncienMot*) et le mot de remplacement (*NouveauMot*). Toutefois, vous pouvez utiliser un caractère quelconque comme délimiteur. Dans l'exemple suivant, le signe % (pourcentage) permet d'inclure une barre oblique (/) dans les mots :

```
s%/home/myfile%/home/yourfile%
```

Le shell remplace une & (perluète) par le texte *AncienMot* dans la variable *NouveauMot*. Dans l'exemple suivant, `/home/myfile` devient `/temp/home/myfile`.

```
s%/home/myfile%/temp&%
```

Le shell remplace un mot nul dans une substitution soit par la dernière substitution, soit par la dernière chaîne utilisée dans la chaîne de balayage contextuelle `!?Chaîne?`. Vous pouvez omettre le dernier délimiteur (`/`) si un caractère nouvelle ligne suit immédiatement. Pour délimiter une liste d'historique, vous disposez des modificateurs suivants :

|              |                                                                                                                                        |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <b>t</b>     | Supprime tous les chemins d'accès en tête, en conservant la fin.                                                                       |
| <b>&amp;</b> | Répète la substitution précédente.                                                                                                     |
| <b>g</b>     | Applique la modification globalement.                                                                                                  |
| <b>p</b>     | Affiche la nouvelle commande, sans l'exécuter.                                                                                         |
| <b>q</b>     | Déclare les mots remplacés, prévenant ainsi toute substitution ultérieure.                                                             |
| <b>x</b>     | Equivalent du modificateur <b>q</b> , mais décompose les mots au niveau des espaces, des tabulations et des caractères nouvelle ligne. |

Sauf si le modificateur **g** précède les modificateurs ci-dessus, le changement ne s'applique qu'au premier mot modifiable.

Si vous indiquez une référence historique sans spécification d'événement (par exemple, `!$`), le shell utilise comme événement la commande précédente. Si une référence historique antérieure intervient sur la même ligne, le shell répète la référence précédente. Ainsi, la séquence suivante génère les premier et dernier arguments de la commande correspondant à `?foo?`.

```
!?foo?^ !$
```

Une abréviation spéciale remplace une référence historique lorsque le premier caractère non blanc d'une ligne d'entrée est un `^` (caret). Ce format équivaut à `! :s^`, et fournit ainsi un raccourci commode pour les substitutions sur le texte de la ligne précédente. La commande `^ lb^ lib` corrige l'orthographe de `lib` dans la commande précédente.

Si nécessaire, vous pouvez encadrer une substitution d'historique par `{ }` (accolades) pour l'isoler des caractères qui la suivent. Par exemple, si vous souhaitez utiliser une référence à la commande :

```
ls -ld ~paul
```

pour exécuter la commande :

```
ls -ld ~paula
```

utilisez la construction :

```
!{l}a
```

Dans cet exemple, `!{l}a` recherche une commande commençant par `l` et lui ajoute un `a` final.

## Guillemets et apostrophes

Encadrer une chaîne d'apostrophes ou de guillemets empêche toute substitution ultérieure. De simples apostrophes (`' '`) empêchent que la chaîne soit interprétée, tandis que des guillemets (`" "`) permettent des développements ultérieurs. Dans les deux cas, le texte résultant devient (en totalité ou en partie) un mot simple.

---

## Substitution d'alias (shell C)

Un *alias* est un nom affecté à une commande ou à une chaîne de commandes, que vous pouvez utiliser exactement comme une commande. Le shell maintient une liste des alias définis.

Lorsque le shell a examiné la ligne de commande, il la décompose en commandes distinctes et recherche le premier mot de chaque commande, de gauche à droite, pour déterminer s'il s'agit d'un alias. Dans l'affirmative, le shell remplace, au moyen du mécanisme d'historique, le texte de l'alias par celui de la commande qu'il référence. Les mots résultants remplacent la commande et la liste d'arguments. En l'absence de référence à la liste d'historique, la liste d'arguments reste inchangée.

Pour en savoir plus, reportez-vous à Substitution d'historique (shell C)

Les commandes intégrées **alias** et **unalias** établissent, affichent et modifient la liste des alias. Voici le format de la commande **alias** :

```
alias [Nom [ListeMots]]
```

La variable *Nom* (facultative) spécifie l'alias du nom indiqué. Si vous spécifiez une liste de mots avec la variable *ListeMots*, la commande l'affecte comme alias de la variable *Nom*. Si vous lancez la commande **alias** sans variable, elle affiche tous les alias du shell C.

Si l'alias de la commande **ls** est `ls -l`, la commande :

```
ls /usr
```

est remplacée par :

```
ls -l /usr
```

La liste d'arguments reste inchangée car il n'y a pas de référence à la liste d'historique dans la commande avec **alias**. De même, si l'alias de la commande **lookup** est :

```
grep \!^ /etc/passwd
```

le shell remplace `lookup bill` par :

```
grep bill /etc/passwd
```

Dans cet exemple, `!^` fait référence à la liste d'historique et le shell la remplace par le premier argument de la ligne d'entrée, `bill`.

Vous pouvez utiliser des caractères joker dans un alias. La commande :

```
alias lprint 'pr &bslash2.!* >
```

```
> print'
```

crée une commande qui formate ses arguments pour une imprimante ligne. Le caractère `!` est protégé du shell dans l'alias, de sorte qu'il n'est pas développé tant que la commande **pr** n'est pas exécutée.

Si le shell détecte un alias, il effectue la transformation de mot dans le texte d'entrée et relance le processus alias sur la nouvelle ligne. Si le premier mot du texte suivant est le même que l'ancien, un indicateur de fin (du processus alias) empêche le bouclage. Les autres boucles ultérieures sont détectées et génèrent une erreur.

---

## Substitution de variables et de noms de fichiers (shell C)

Le shell C autorise la substitution de variable et de nom de fichier.

Cette section traite des points suivants :

- Substitution de variables (shell C), page 11-109.
- Substitution de nom de fichier (shell C), page 11-111
- Développement de nom de fichier, page 11-111.
- Abréviations de nom de fichier, page 11-111.
- Classes de caractères, page 11-112.

### Substitution de variable (shell C)

Le shell C maintient un ensemble de variables, chacune d'elles ayant pour valeur une liste de zéro ou plusieurs mots. Certaines sont définies ou référencées par le shell. Par exemple, la variable **argv** est une image de la liste de variables shell, et les mots comportant la valeur de cette variable sont référencés de façon particulière.

Vous pouvez modifier et afficher la valeur des variables via les commandes **set** et **unset**. Certaines variables sont des "bascules", c'est-à-dire qu'elles activent ou désactivent une fonction: le shell ne fait aucun cas de leur valeur, seul importe le fait qu'elles soient ou non activées. Par exemple, la variable **verbose** provoque l'affichage (écho) des commandes entrées. La définition de cette variable est activée par l'indicateur **-v** sur la ligne de commande.

D'autres opérations exécutent un traitement numérique des variables. La commande **@** effectue des calculs, dont le résultat est affecté à une variable. Les valeurs des variables sont toutefois toujours représentées par des chaînes (zéro ou plusieurs). Pour les opérations numériques, la chaîne nulle est évaluée à zéro, et les mots suivants sont ignorés.

Lorsque vous émettez une commande, le shell analyse la ligne d'entrée et exécute les substitutions d'alias. Ensuite, avant d'exécuter la commande, il exécute les substitutions de variables. Le caractère \$ (dollar) code la substitution. Il est néanmoins transmis tel quel s'il est suivi d'un espace, d'une tabulation ou d'un caractère nouvelle ligne. Faire précéder le caractère \$ d'une barre oblique inverse (\) empêche ce développement, sauf dans les deux cas suivants :

- La commande est entre guillemets (" "). Le shell exécute alors la substitution.
- La commande est entre apostrophes (') : le shell n'exécute alors jamais la substitution. Les chaînes entre ' ' sont interprétées au niveau de la substitution de commande. (Reportez-vous à Substitution de commande (shell C), page 11-103.

Le shell détecte les réacheminements d'E/S avant de développer les variables, et les développe séparément. Sinon, le nom de la commande et la liste complète d'arguments sont développés simultanément. Il est ainsi possible que le premier mot (commande) génère plusieurs mots, dont le premier devient le nom de la commande, les suivants étant des paramètres.

Sauf s'il est entre guillemets (" ") ou associé au modificateur **:q**, le résultat d'une substitution de variable peut faire lui-même l'objet d'une substitution de commande et de nom de fichier. Une variable entre guillemets, composée de plusieurs mots, est développée sous la forme d'un seul mot (ou d'une portion de mot), les mots initiaux étant séparés par des espaces. Si vous spécifiez le modificateur **:q**, la variable est développée en plusieurs mots, entre guillemets, séparés par des espaces, pour empêcher toute substitution ultérieure (de commande ou de nom de fichier).

Voici la syntaxe d'entrée des valeurs de variable dans le shell. Sauf indication contraire, référencer une variable non définie par la commande **set** génère une erreur.

Les modificateurs **:gh**, **:gt**, **:gr**, **:h**, **:r**, **:q** et **:x** sont applicables aux substitutions suivantes. Si des accolades ( { } ) apparaissent dans le format de la commande, les modificateurs doivent se trouver entre accolades. L'implémentation courante autorise un seul modificateur deux-points : (deux points) pour chaque développement de variable.

**\$Nom**

**\${Nom}**

Remplacé par les mots affectés à la variable *Nom*, séparés par des espaces. Les accolades isolent la variable *Nom* des caractères suivants (qui lui seraient sinon intégrés). Les noms des variables shell commencent par une lettre et comportent au maximum 20 lettres et chiffres, ou caractères de soulignement ( \_ ). Si *Nom* ne spécifie pas une variable shell, mais est défini dans l'environnement, sa valeur est alors retournée. Les modificateurs précédés de deux points, de même que les autres formats ici décrits, ne sont pas disponibles dans ce cas.

**\$Nom[numéro]**

**\${Nom[numéro]}**

Ne sélectionne que quelques mots de la variable *Nom*. Le numéro, qui peut faire l'objet de substitution de variable, est soit un chiffre seul, soit deux chiffres séparés par un tiret ( - ). Le premier mot de la chaîne représentant la valeur de la variable est numéroté 1. Par défaut, le premier chiffre d'un intervalle vaut 1 et le dernier, **\$#Nom**. Le symbole\* (astérisque) sélectionne tous les mots. Si le second argument est omis ou qu'il se trouve dans un intervalle, un premier intervalle vide ne génère pas d'erreur.

**\$#Nom**

**\${#Nom}**

Indique le nombre de mots de la variable. Cette fonction est très utile pour un usage ultérieur dans *[numéro]* comme indiqué ci-dessus. Par exemple, **\$Nom[ \$#Nom ]**.

**\$0**

Remplace le nom du fichier à partir duquel sont lues les entrées de commandes. Si ce nom est inconnu, une erreur est générée.

**\$numéro**

**\${numéro}**

Equivalent à **\$argv[numéro]**.

**\$\***

Equivalent à **\$argv[\*]**.

Les modificateurs : ne sont pas applicables aux substitutions suivantes :

**\$?nom**

**\${?nom}**

Remplace par la chaîne 1 si *nom* est définie, par 0 (zéro) sinon.

**\$?0**

Remplace par 1 si le nom du fichier d'entrée courant est reconnu, par 0 (zéro) sinon.

**\$\$**

Remplace par le numéro (décimal) du process du shell parent.

**\$<**

Remplace par une ligne de l'entrée standard, sans autre interprétation. Cette substitution permet de lire à partir du clavier dans une procédure shell.

## Substitution de nom de fichier (shell C)

Le shell C offre plusieurs fonctions de raccourci permettant d'économiser temps et frappe de touches. Si un mot contient un des caractères\* (astérisque), ? point d'interrogation), [ ] (crochets) ou (accolades), ou commence par ~ (tilde), ce mot fait l'objet de substitutions de noms de fichiers. Le shell C considère le mot comme un terme générique et le remplace selon une liste alphabétique de noms de fichiers.

La séquence de classement courante est spécifiée par les variables d'environnement **LC\_COLLATE** ou **LANG**. Dans une liste de mots spécifiant des substitutions de noms de fichiers, une erreur est générée si aucun terme générique ne correspond à un fichier existant. Il n'est toutefois pas nécessaire que tous les termes génériques aient un correspondant. Seuls les caractères \* (astérisque), ? (point d'interrogation) et [ ] (crochets) indiquent des correspondances ou des développements de noms de fichiers. Les caractères ~ (tilde) et {} (accolades) indiquent des abréviations de noms de fichiers.

## Développement de nom de fichier

Le caractère \* (astérisque) remplace n'importe quelle chaîne de caractères, chaîne nulle comprise. Par exemple, dans un répertoire contenant les fichiers :

```
a aa aax alice b bb c cc
```

La commande `echo a*` affiche tous les fichiers commençant par a :

```
a aa aax alice
```

**Remarque :** Lors de la comparaison, les caractères . (point) et / (barre oblique) doivent également correspondre.

Le caractère ? (point d'interrogation) remplace un seul caractère. La commande :

```
ls a?x
```

affiche la liste des fichiers commençant par a, suivis d'un seul caractère et terminés par x :

```
aax
```

Pour rechercher un caractère ou un ensemble de caractères, encadrez le(s) caractère(s) de crochets ([ ]). La commande :

```
ls [abc]
```

affiche la liste des fichiers correspondant exactement à l'un des caractères entre crochets :

```
a b c
```

Entre les crochets, un intervalle de caractères est indiqué par [a-z]. Les caractères correspondants sont définis par la séquence de classement courante.

## Abréviation de nom de fichier

Les caractères ~ (tilde) et { (accolade ouvrante) indiquent une abréviation de nom de fichier. Un ~ en début de fichier représente les répertoires personnels. Isolé, le caractère ~ prend la valeur de votre répertoire personnel, défini par la variable **home**. Par exemple, la commande :

```
ls ~
```

affiche tous les fichiers et répertoires de votre répertoire **\$HOME**.

Lorsqu'il est suivi d'un nom composé de lettres, de chiffres et de tirets (-), le shell recherche un utilisateur portant ce nom et remplace le répertoire \$HOME de l'utilisateur.

**Remarque :** Si le ~ est suivi d'un caractère autre qu'une lettre ou une barre oblique (/), ou qu'il apparaît ailleurs qu'au début d'un mot, il n'est pas développé.

Pour faire correspondre des caractères dans des noms de fichiers sans taper l'intégralité du nom, encadrez les noms de fichiers d'accolades ({}). Ainsi, a{b, c, d}e est un raccourci de `abe ace ade`. Le shell préserve l'ordre (de gauche à droite) et, à cet effet, enregistre

séparément le résultat des correspondances à un niveau inférieur. Cette construction peut être imbriquée. Ainsi :

```
~source/s1/{oldls,ls}.c
```

devient :

```
/usr/source/s1/oldls.c /usr/source/s1/ls.c
```

si le répertoire personnel `source` est `/usr/source`. De même :

```
../{memo,*box}
```

peut devenir :

```
../memo ../box ../mbox
```

**Remarque :** `memo` n'est pas trié avec le résultat de la comparaison `*box`. Dans ce cas particulier, les accolades { (ouvrante), } (fermante) et } (par paire) sont passées sans changement.

## Classes de caractères

Pour effectuer des comparaisons, vous pouvez également vous servir des classes de caractères :

```
[:charclass:]
```

Avec ce format, le système établit une correspondance avec chacun des caractères appartenant à la classe spécifiée. Les classes définies correspondent aux sous-routines **ctype**.

| <b>Classe</b> | <b>Définition</b>                                                                                     |
|---------------|-------------------------------------------------------------------------------------------------------|
| <b>alnum</b>  | Caractères alphanumériques                                                                            |
| <b>alpha</b>  | Majuscules et minuscules                                                                              |
| <b>cntrl</b>  | Caractères de contrôle                                                                                |
| <b>digit</b>  | Chiffres                                                                                              |
| <b>graph</b>  | Caractères graphiques                                                                                 |
| <b>lower</b>  | Minuscules                                                                                            |
| <b>print</b>  | Caractères imprimables                                                                                |
| <b>punct</b>  | Caractères de ponctuation                                                                             |
| <b>space</b>  | Espace, tabulation horizontale, retour chariot, nouvelle ligne, tabulation verticale ou page suivante |
| <b>upper</b>  | Lettres majuscules                                                                                    |
| <b>xdigit</b> | Chiffres hexadécimaux                                                                                 |

Supposons que vous soyez dans un répertoire contenant les fichiers :

```
a aa aax Alice b bb c cc
```

A l'invite shell, tapez :

Le shell C affiche la liste de tous les fichiers dont le nom commence par une minuscule:

```
a aa aax b bb c cc
```

Pour en savoir plus, reportez-vous à la commande **ed**.

---

## Variables d'environnement (shell C)

Certaines variables ont une signification spéciale pour le shell. Par exemple, **argv**, **cwd**, **home**, **path**, **prompt**, **shell** et **status** sont toujours définies par le shell. Sauf pour **cwd** et **status**, cette action n'a lieu qu'au moment de l'initialisation. Ces variables conservent leur valeur tant que vous ne les modifiez pas explicitement.

La commande **csch** copie les variables d'environnement **USER**, **TERM**, **HOME** et **PATH** dans les variables **csch**, **user**, **term**, **home** et **path**, respectivement. Les valeurs sont recopiées dans l'environnement dès que les variables shell normales sont redéfinies. Il n'est pas nécessaire de se soucier de la variable **path** ailleurs que dans le fichier **.cschrc** : les sous-process **csch** importent la définition du chemin d'accès, à partir de l'environnement, et la réexportent si elle est modifiée.

Voici les variables spéciales du shell :

|                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>argv</b>       | Arguments passés aux scripts shell. Les paramètres positionnels sont remplacés à partir de cette variable.                                                                                                                                                                                                                                                                                                                                                                   |
| <b>cdpath</b>     | Liste de répertoires, à explorer par la commande <b>chdir</b> ou <b>cd</b> lors de recherche de sous-répertoires.                                                                                                                                                                                                                                                                                                                                                            |
| <b>cwd</b>        | Chemin d'accès complet au répertoire courant.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>echo</b>       | Activé par l'indicateur <b>-x</b> sur la ligne de commande, affiche (écho) chaque commande et ses arguments avant qu'ils soient exécutés. Les commandes non intégrées sont développées avant d'être affichées. Les commandes intégrées le sont avant toute substitution de commande ou de nom de fichier, dans la mesure où ces substitutions sont ensuite effectuées de façon sélective.                                                                                    |
| <b>histchars</b>  | Chaîne modifiant les caractères utilisés dans une substitution d'historique. Le premier caractère devient le caractère de substitution d'historique (remplaçant le caractère par défaut !). (point d'exclamation). Le second remplace le caractère ^ (caret) dans les substitutions rapides.<br><b>Remarque :</b> Affecter à <b>histchars</b> un caractère utilisé dans une commande ou dans un nom de fichier peut provoquer des substitutions d'historique non souhaitées. |
| <b>historique</b> | Valeur numérique contrôlant la taille de la liste d'historique. Toute commande référencée dans ces événements n'est pas éliminée. Indiquer une valeur trop importante pour la variable <b>history</b> peut provoquer un dépassement mémoire. Qu'elle soit définie ou non, elle sauvegarde la dernière commande exécutée sur la liste d'historique.                                                                                                                           |
| <b>home</b>       | Répertoire personnel, initialisé par l'environnement. Le développement de nom de fichier du caractère ~ (tilde) fait référence à cette variable.                                                                                                                                                                                                                                                                                                                             |
| <b>ignoreeof</b>  | Spécifie au shell d'ignorer les caractères EOF (fin de fichier) issus des stations de travail. Ceci pour empêcher les shells d'être accidentellement tués à la lecture d'un caractère de fin de fichier (Ctrl-D).                                                                                                                                                                                                                                                            |

|                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>mail</b>      | <p>Fichiers à examiner par le shell pour le courrier. Ce que le shell effectue après chaque commande dont le résultat est une invite, à l'expiration du délai imparti. Le shell affiche le message <code>Mail in fichier</code>. si le fichier existe et que l'heure de son dernier accès est antérieure à l'heure de sa dernière modification.</p> <p>Si le premier mot de la variable <b>mail</b> est numérique, il indique un nouvel intervalle de vérification du courrier (en secondes – valeur par défaut 600, soit : 10 minutes). Si vous spécifiez plusieurs fichiers courrier, le shell affiche le message <code>New mail in fichier</code>, quand du courrier arrive dans le fichier concerné.</p>                                                                                                                          |
| <b>noclobber</b> | <p>Règleme les réacheminements en sortie pour garantir qu'aucun fichier ne sera accidentellement détruit et que les réacheminements effectuent des ajouts aux fichiers antérieurs.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>noglob</b>    | <p>Désactive le développement de nom de fichier. Cette fonction est très utile pour les procédures ne traitant pas de noms de fichiers, ou lorsqu'une liste de noms de fichiers a été générée et qu'aucun autre développement n'est souhaitable.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>nomatch</b>   | <p>Spécifie de ne pas générer d'erreur si le développement d'un nom de fichier ne correspond à aucun fichier existant, mais de renvoyer la trame initiale. Un format incorrect de la trame provoquera toujours une erreur.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <b>notify</b>    | <p>Spécifie au shell de vous avertir en mode asynchrone des modifications intervenues au niveau de l'état des travaux. Par défaut, ces modifications sont affichées juste avant l'affichage de l'invite shell.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>path</b>      | <p>Spécifie les répertoires à explorer à la recherche des commandes à exécuter. Un mot nul spécifie le répertoire courant. Si aucune variable <b>path</b> n'est définie, seuls les chemins d'accès complets sont pris en compte. Le chemin par défaut (extrait du fichier <b>/etc/environment</b> utilisé pour la connexion) est :</p> <pre>/usr/bin /etc /usr/sbin /usr/ucb /usr/bin/X11 /sbin</pre> <p>Un shell non associé à l'indicateur <b>-c</b> ni <b>-t</b> répartit normalement le contenu des répertoires dans la variable <b>path</b>, après lecture du fichier <b>.cshrc</b> et chaque fois que <b>path</b> est redéfini. Si des commandes sont ajoutées à ces répertoires pendant que le shell est actif, vous devez lancer la commande <b>rehash</b>, faute de quoi les commandes risquent de ne pas être trouvées.</p> |
| <b>prompt</b>    | <p>Chaîne affichée avant chaque lecture de commande à partir d'une station de travail interactive. Si un ! (point d'exclamation) apparaît dans la chaîne, il est remplacé par le numéro de l'événement courant. Si le caractère ! se trouve dans une chaîne entre guillemets ou entre apostrophes, il doit être précédé d'une barre oblique inversée (\). L'invite par défaut est %, pour les utilisateurs non racine. Pour les utilisateurs racine, elle est #.</p>                                                                                                                                                                                                                                                                                                                                                                  |

|                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>savehist</b> | Valeur numérique contrôlant le nombre d'entrées de la liste d'historique, sauvegardées dans le fichier ~/. <b>history</b> lorsque vous vous déconnectez. Toute commande référencée dans ce nombre d'événements est sauvegardée. Au cours du lancement, le shell lit ~/. <b>history</b> dans la liste d'historique, assurant la sauvegarde de l'historique à travers les multiples connexions. Donner à la variable <b>savehist</b> une valeur trop importante ralentit le lancement du shell. |
| <b>shell</b>    | Fichier où réside le shell C. Ce qui permet aux shells parallèles d'interpréter les fichiers dotés de bits d'exécution, mais non exécutables par le système. Cette valeur prend initialement la valeur du fichier personnel du shell C.                                                                                                                                                                                                                                                       |
| <b>status</b>   | Etat retourné par la dernière commande. Si la commande se termine anormalement, 0200 est ajouté à l'état. Les commandes intégrées qui n'aboutissent pas retournent un état de sortie de 1 ; celles qui aboutissent renvoie un état de 0 (zéro).                                                                                                                                                                                                                                               |
| <b>time</b>     | Contrôle le minutage automatique des commandes. Lorsque cette variable est définie, toute commande dépassant le délai imparti (en nombre de secondes CPU) affichera, en fin d'exécution, une ligne relative aux ressources utilisées. Pour en savoir plus, reportez-vous à la commande intégrée <b>time</b> .                                                                                                                                                                                 |
| <b>verbose</b>  | Activée par l'option <b>-v</b> sur la ligne de commande, affiche les mots de chaque commande, après exécution des substitutions d'historique.                                                                                                                                                                                                                                                                                                                                                 |

---

## Réacheminement des entrées/sorties (shell C)

Avant d'exécuter une commande, le shell C balaye la ligne d'entrée à la recherche de caractères de réacheminement. Ces derniers lui indiquent de réacheminer les entrées et les sorties.

Pour réacheminer les entrées et sorties standard d'une commande, vous disposez des instructions suivantes :

|                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>&lt;Fichier</code>             | Ouvre le <i>Fichier</i> spécifié (qui est d'abord développé – variable, commande et nom de fichier) comme fichier d'entrée standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <code>&lt;&lt;Mot</code>             | Lit l'entrée shell jusqu'à la ligne correspondant à la valeur de <i>Mot</i> . La variable <i>Mot</i> ne peut faire l'objet de substitution de variable, de nom de fichier ou de commande. Chaque ligne entrée lui est comparée avant toute substitution sur la ligne. Sauf si un caractère de déclaration ( <code>\</code> , <code>"</code> , <code>'</code> ou <code>`</code> ) figure dans la variable <i>Mot</i> , le shell exécute les substitutions de variable et de commande sur les lignes concernées, la barre oblique inversée ( <code>\</code> ) pouvant servir à déclarer les caractères <code>\$</code> (dollar), <code>\</code> (barre oblique inversée) et <code>'</code> (apostrophe). Dans les commandes remplacées, les espaces, les tabulations et les caractères ligne suivante sont préservés, sauf le dernier caractère nouvelle ligne, qui n'est pas pris en compte. Le texte résultant est placé dans un fichier temporaire anonyme, qui devient l'entrée standard de la commande. |
| <code>&gt; Fichier</code>            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt;!Fichier</code>            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt;&amp; Fichier</code>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt;&amp;! Fichier</code>      | Ouvre le <i>Fichier</i> spécifié comme sortie standard. Si le fichier n'existe pas, il est créé. S'il existe, il est tronqué et son contenu antérieur est perdu. Si la variable <b>noclobber</b> est définie, le fichier doit ne pas exister ou être un fichier spécial caractère, faute de quoi une erreur est générée. Ceci pour contribuer à prévenir toute destruction accidentelle de fichiers. Dans ce cas, recourez au format comportant un <code>!</code> (point d'exclamation) pour éliminer ce contrôle. <i>Fichier</i> est développé de la même façon que pour les fichiers d'entrée <code>&lt;</code> . Le format <code>&gt;&amp;</code> (perluète) achemine la sortie standard et la sortie d'erreur vers <i>Fichier</i> . L'exemple suivant illustre comment acheminer séparément la sortie standard vers <b>/dev/tty</b> et la sortie d'erreur standard vers <b>/dev/null</b> . Les parenthèses sont obligatoires pour la séparation :                                                      |
|                                      | <pre>% (find / -name vi -print &gt; /dev/tty) &gt;&amp; /dev/null</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>&gt; &gt;Fichier</code>        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt;&gt;! Fichier</code>       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt; &gt;&amp; Fichier</code>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>&gt; &gt;&amp;! Fichier</code> | Utilise <i>Fichier</i> comme sortie standard (comme <code>&gt;</code> ), mais <i>ajoute</i> la sortie à la fin du fichier. Si la variable <b>noclobber</b> est définie, une erreur se produit si le fichier n'existe pas, sauf si vous employez un format avec <code>!</code> (point d'exclamation). Sinon, l'instruction est semblable à <code>&gt;</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Une commande reçoit l'environnement dans lequel le shell a été appelé, modifié le cas échéant par les options d'entrée/sortie et la présence de la commande en tant que pipeline. Ainsi, contrairement à certains shells précédents, les commandes exécutées à partir d'un fichier de commandes shell n'ont pas accès au texte des commandes par défaut. A l'inverse, elles reçoivent les entrées standard d'origine, directement du shell. Le mécanisme << permet de présenter les données en ligne : les fichiers de commandes shell fonctionnent alors comme des éléments de pipelines et le bloc shell lit les entrées. Notez que l'entrée standard par défaut pour une commande exécutée isolément n'est pas remplacée par le fichier vide **/dev/null** : l'entrée standard reste l'entrée standard d'origine du shell.

Pour réacheminer les sorties d'erreur via un tube vers la sortie standard, utilisez le format **|&** (barre verticale, perluète) et non la seule **|** (barre verticale).

## Flux de contrôle

Le shell contient des commandes qui peuvent servir à réguler le flux de contrôle dans les fichiers de commandes (scripts shell) et, de façon limitée mais fort utile, dans les entrées ligne de commande. Ces commandes forcent le shell à relire ou à sauter ses entrées.

Les instructions **foreach**, **switch** et **while** et le format **if-then-else** de l'instruction **if**, requièrent que les mots-clés principaux apparaissent dans une seule commande simple sur une ligne d'entrée.

Si l'entrée shell ne peut être explorée, le shell place dans des tampons les entrées chaque fois qu'une boucle est lue et se sert du tampon interne pour effectuer les relectures requises par la boucle. Des **goto** arrière permettent ensuite de localiser les entrées que vous n'avez pu rechercher.

- Voir aussi C Shell

---

## Contrôle des travaux (shell C)

Le shell associe à chaque process un numéro de travail. Il maintient un tableau des travaux en cours et leur affecte des numéros (petits nombres entiers). Lorsque vous lancez un travail en arrière-plan, via un signe **&** (perluète), le shell affiche une ligne semblable à :

```
[1] 1234
```

Cette ligne indique que le travail porte le numéro 1 et qu'il se compose d'un seul process dont l'ID est 1234. Pour afficher la table des travaux en cours, lancez la commande intégrée **jobs**.

Un travail en arrière-plan doit affronter la concurrence pour obtenir des données en entrée à partir de la station de travail. Il en est de même pour les données en sortie, qui sont imbriquées avec les sorties des autres travaux.

Il existe plusieurs manières de référencer les travaux dans le shell. Le signe % (pourcentage) introduit un nom de travail : il peut s'agir du numéro de travail ou du nom de la commande qui a lancé le travail s'il s'agit d'un nom unique. Ainsi, si un processus **make** est exécuté en tant que travail 1, vous pouvez y faire référence par %1, ou par %make, s'il n'existe qu'un travail suspendu dont le nom commence par la chaîne "make". Vous pouvez également utiliser :

```
 $?Chaîne
```

pour spécifier un travail dont le nom contient la variable `Chaîne`, s'il n'existe qu'un seul travail de ce type.

Le shell détecte immédiatement les changements d'état des process. Si un travail bloque, rendant impossible la poursuite du traitement, le shell envoie un message à la station de travail. Ce message ne s'affiche que lorsque vous appuyez sur Entrée. Si, toutefois, la variable **notify** est définie, le shell émet immédiatement un message indiquant un changement au niveau de l'état des travaux en arrière-plan. Au moyen de la commande intégrée **notify**, marquez les process pour lesquels vous souhaitez que les changements d'état vous soient notifiés. Par défaut, la commande **notify** marque le process courant.

- Voir aussi C Shell

---

## Chapitre 12. Outils et utilitaires

Voici quelques unes des tâches que vous pouvez exécuter par le biais des commandes du système d'exploitation. Les commandes correspondantes sont décrites ensuite.

- Recherche d'une commande par mot-clé (commande **apropos**), page 12-1.
- Affichage d'un calendrier (commande **cal**), page 12-2.
- Affichage des messages de l'agenda (commande **calendar**), page 12-2.
- Affichage de l'aide (commande **help**), page 12-3.
- Appel du didacticiel (commande **learn**), page 12-3.
- Heure de sortie (commande **leave**), page 12-4.
- Décomposition d'un nombre en facteurs premiers (commande **factor**), page 12-4.
- Conversion des unités de mesure (commande **units**), page 12-5.
- Envoi de messages à un autre utilisateur connecté (commande **write**), page 12-7.
- Récapitulatif des commandes, page 12-9.

---

### Recherche d'une commande par mot-clé (commande **apropos**)

La commande **apropos** permet d'afficher les sections des manuels dont le titre contient le *Mot-clé* indiqué. Cette commande traite chaque mot séparément et ne différencie pas les majuscules des minuscules. Les mots qui font partie d'un autre mot sont également affichés. Par exemple, si vous recherchez le mot `compile`, la commande **apropos** affiche également les occurrences du mot `compiler`.

La commande **apropos** équivaut à la commande **man** assortie de l'option **-k**.

Par exemple, pour retrouver les pages de manuel dont le titre contient le mot `password`, entrez :

```
apropos password
```

Reportez-vous à la commande **apropos** dans le manuel *AIX Commands Reference*.

---

## Affichage d'un calendrier (commande cal)

La commande **cal** écrit (sur l'unité de sortie standard) le calendrier du mois ou de l'année spécifié.

Le paramètre *Mois* indique le mois souhaité : de 1 à 12, correspondant à janvier, février, etc. A défaut, la commande **cal** prend en compte le mois courant.

Le paramètre *Année* indique l'année souhaitée : la commande **cal** pouvant afficher le calendrier de n'importe quelle année entre 1 et 9999, précisez l'année en entier, et non simplement les deux derniers chiffres. A défaut, la commande **cal** prend en compte l'année courante.

Par exemple, pour afficher le calendrier de février 1984, entrez :

```
cal 2 1984
```

Par exemple, pour imprimer le calendrier de l'année 1984, entrez :

```
cal 1984 | qprt
```

Par exemple, pour afficher le calendrier de l'année 84 après J.C., entrez :

```
cal 84
```

Reportez-vous à la commande **cal** dans le manuel *AIX Commands Reference*.

---

## Affichage des messages de l'agenda (commande calendar)

Vous pouvez lire le fichier **calendar**, préalablement créé dans votre répertoire personnel, via la commande **calendar**. Cette commande inscrit, sur la sortie standard, les lignes du fichier contenant la date du jour ou de lendemain.

La commande **calendar** reconnaît les formats de date tels que Dec. 7 ou 12/7. Elle interprète également le caractère \* (astérisque) suivi d'une barre oblique (/). \*/7, par exemple, est interprété comme le septième jour de chaque mois.

Tous les vendredis, la commande **calendar** écrit toutes les lignes contenant les dates correspondant à vendredi, samedi, dimanche et lundi. Elle ne reconnaît en revanche pas les jours de congé, aussi inscrira-t-elle pour ces jours les seuls rendez-vous du lendemain.

Voici un exemple de fichier **calendar** :

```
*/25 - Prepare monthly report
Aug. 12 - Fly to Denver
aug 23 - board meeting
Martha out of town - 8/23, 8/24, 8/25
8/24 - Mail car payment
sat aug/25 - beach trip
August 27 - Meet with Simmons
August 28 - Meet with Simmons
```

Si nous sommes vendredi 24 août, la commande **calendar** affiche :

```
*/25 - Prepare monthly report
Martha out of town - 8/23, 8/24, 8/25
8/24 - Mail car payment
sat aug/25 - beach trip
August 27 - Meet with Simmons
```

Reportez-vous à la commande **calendar** dans le manuel *AIX Commands Reference*.

---

## Affichage de l'aide (commande help)

La commande **help** affiche un écran d'informations.

A l'invite, entrez :

```
help
```

Le système affiche un écran semblable à celui-ci :

```
Look in a printed manual for general help if you can. You should have
someone show you some things and then read "Using and Managing
AIX" manual.
```

```
The commands:
```

```
man -k keyword - lists commands relevant to a keyword
man command - prints out the manual pages for a command
 are helpful; other basic commands are:
cat - concatenates files (and just prints them out)
ex - text editor
ls - lists contents of directory
mail - sends and receives mail
msgs - system messages and junk mail
passwd - changes login password
sccshelp - views information on the Source Code Control System
smit - system management interface tool
tset - sets terminal modes
who - who is on the system
write - writes to another user
```

```
You could find programs about mail by the command: man -k mail
and print out the man command documentation via: man mail
You can log out by typing "exit".
```

Reportez-vous à la commande **help** dans le manuel *AIX Commands Reference*.

---

## Appel du didacticiel (commande learn)

La commande **learn** vous permet de vous entraîner à travailler sur des fichiers, des éditeurs, des macros, etc. Lors de votre premier appel à **learn**, vous serez invité à préciser le type de cours souhaité. Lors des sessions suivantes, vous serez positionné à l'endroit où vous avez arrêté la leçon précédente.

La commande **learn** recherche la première leçon concernant le sujet choisi (parmi les sujets suivants) :

- Files
- Editor
- More files
- Macros
- EQN
- C.

Par exemple, pour appeler la leçon sur les fichiers, entrez :

```
learn files
```

Vous serez ensuite invité à répondre à des questions.

Reportez-vous à la commande **learn** dans le manuel *AIX Commands Reference*.

---

## Heure de sortie (commande **leave**)

Vous pouvez demander au système de vous rappeler l'heure de sortie via la commande **leave**. Le rappel est effectué 5 minutes, puis 1 minute avant l'heure, à l'heure dite, puis toutes les minutes ensuite. Lorsque vous vous déconnectez, la commande **leave** quitte juste avant d'afficher le message suivant.

Si vous n'indiquez pas d'heure, la commande affiche `When do you have to leave?` Si vous vous contentez de passer à la ligne (réponse `:newline`), la commande est abandonnée. Sinon, vous devez indiquer une heure. Ce format peut être intégré dans un fichier **.login** ou **.profile**.

Entrez l'heure au format *hhmm*. Toutes les indications d'heure sont converties au format 12 heures, et supposées s'appliquer aux 12 heures suivantes. Vous disposez également de l'indicateur **+** pour indiquer une durée à partir de l'heure actuelle.

**Remarque :** La commande **leave** ignore les interruptions, les sorties et les opérations de fin. Pour annuler la commande **leave**, vous devez soit vous déconnecter, soit lancer la commande **kill -9** et fournir l'ID process.

Par exemple, pour programmer un rappel à 3:45, entrez :

```
leave 345
```

Par exemple, pour programmer un rappel dans 20 minutes, entrez :

```
leave +0020
```

Reportez-vous à la commande **leave** dans le manuel *AIX Commands Reference*.

---

## Décomposition d'un nombre en facteurs premiers (commande **factor**)

La commande **factor** décompose un nombre en facteurs premiers. En l'absence de paramètre *Nombre*, la commande attend que vous entriez un nombre positif inférieur à 1E14 (100 000 000 000 000), puis inscrit les facteurs premiers de ce nombre sur la sortie standard (en répétant les facteurs autant de fois que nécessaire). Pour quitter, tapez 0 (zéro) ou un caractère non numérique quelconque.

Appelée avec un argument, la commande **factor** détermine les facteurs premiers du paramètre *Nombre*, inscrit le résultat sur la sortie standard et quitte.

Par exemple, pour décomposer 123, entrez :

```
factor 123
```

Le résultat s'affiche :

```
123
3
41
```

Reportez-vous à la commande **factor** dans le manuel *AIX Commands Reference*.

---

## Conversion des unités de mesure (commande units)

La commande **units** permet de convertir une quantité exprimé dans une unité de mesure en son équivalent dans une autre unité. La commande **units** est une commande interactive. Vous êtes invité à préciser l'unité d'origine et l'unité dans laquelle opérer la conversion.

**Remarque :** Cette commande n'effectue que des changements n'impliquant que des multiplications. Elle ne peut, par exemple, convertir des degrés Fahrenheit en degrés Celsius - conversion qui suppose d'ajouter et de retrancher 32.

La commande **units** reconnaît **lb** comme unité de masse, mais interprète **pound** comme l'unité monétaire britannique (livre sterling). Les noms composés (tel **lightyear**) sont traités ensemble. Faites précéder de **br** les unités britanniques pour les différencier de leurs homologues américains (**brgallon**, par exemple). Le fichier **/usr/share/lib/unittab** contient la liste complète des unités acceptées par la commande **units**. Vous pouvez y définir d'autres unités, ou même créer votre propre *Fichier*, dont les données primeront sur celles du fichier **unittab**.

Les unités, abréviations et préfixes métriques courants sont reconnus par la commande **units**, ainsi que :

|              |                                                          |
|--------------|----------------------------------------------------------|
| <b>pi</b>    | Rapport de la circonférence d'un cercle et son diamètre. |
| <b>c</b>     | Vitesse de la lumière.                                   |
| <b>e</b>     | Charge d'un électron.                                    |
| <b>g</b>     | Constante de gravité.                                    |
| <b>force</b> | Identique à <b>g</b> .                                   |
| <b>mole</b>  | Nombre d'Avogadro.                                       |
| <b>water</b> | Pression par unité de hauteur d'eau.                     |
| <b>au</b>    | Unité d'astronomie.                                      |

Par exemple, pour afficher les facteurs de conversion de pouce à centimètre, entrez :

```
units
you have: in
you want: cm
```

La commande **units** affiche :

```
* 2.540000e+00
/ 3.937008e-01
```

indiquant qu'il convient de multiplier les pouces par  $2.540000e+00$  pour obtenir des centimètres, et de multiplier les centimètres par  $3.937008e-01$  pour obtenir des pouces.

Ces nombres sont donnés en notation exponentielle standard :  $3.937008e-01$  représente  $3.937008 \times 10^{-1}$ , soit encore  $0,3937008$ .

**Remarque :** Le deuxième nombre est l'inverse du premier :  $2.54$  est égal à  $1/0.3937008$ .

Par exemple, pour convertir une mesure en une autre unité, entrez :

```
units
you have: 5 years
you want: microsec
```

La commande **units** affiche :

```
* 1.577846e+14
/ 6.337753e-15
```

indiquant que 5 années équivalent à  $1.577846 \times 10^{14}$  microsecondes, et qu'une microseconde vaut  $6.337753 \times 10^{-15}$  années.

Par exemple, pour indiquer des mesures fractionnaires, entrez :

```
units
you have: 1|3 mi
you want: km
```

La commande **units** affiche :

```
* 5.364480e-01
/ 1.864114e+00
```

La barre verticale (|) indique la division : 1|3 signifie un tiers. Un tiers de mile vaut donc 0.536448 kilomètres.

Par exemple, pour indiquer des exposants, entrez :

```
units
you have: 1.2-5 gal
you want: floz
```

La commande **units** affiche :

```
* 1.536000e-03
/ 6.510417e+02
```

L'expression 1.2-5 gal équivaut à  $1.2 \times 10^{-5}$ . *Ne tapez pas de e* avant l'exposant (1.2e-5 gal est incorrect). Cet exemple indique que  $1.2 \times 10^{-5}$  (0.000012) gallons équivalent à  $1.536 \times 10^{-3}$  (0.001536) onces liquides.

Si les unités que vous spécifiez après `you have:` et `you want:` sont incompatibles :

```
you have: ft
you want: lb
```

La commande **units** affiche :

```
conformability
3.048000e-01 m
4.535924e-01 kg
```

Le message `conformability` signifie que les unités spécifiées ne peuvent être converties. Par exemple, transformer des pieds (longueur) en livres (poids) n'a aucun sens. La commande **units** affiche donc l'équivalent de chaque valeur en unités standard.

En d'autres termes, cet exemple indique que 1 pied égale 0.3048 mètre et que 1 livre égale 0.4535924 kilogrammes. Mètre et kilogramme sont les unités de mesure standard de longueur et de masse.

Reportez-vous à la commande **units** dans le manuel *AIX Commands Reference*.

---

## Envoi de messages à un autre utilisateur (commande write)

La commande **write** permet d'envoyer en temps réel des messages aux autres utilisateurs du système. Elle assure un échange proche d'une vraie conversation avec un autre utilisateur connecté. Chaque utilisateur envoie et reçoit à son tour de courts messages de l'autre station de travail. Pour envoyer des messages plus longs, il suffit de les placer dans un fichier et de les réacheminer comme entrée de la commande **write**.

A l'émission de la commande **write**, le message suivant, accompagné du bip défini par le caractère ASCII BEL, est envoyé vers le destinataire (ou cible) du message :

```
Message from IDEmetteur on NomHôteEmetteur (tty n) [Date] ...
```

Si la connexion est établie, la commande **write** envoie deux caractères ASCII BEL aux deux stations de travail. Le bip avertit l'émetteur qu'il peut commencer à entrer son message, et le destinataire qu'un message est sur le point de lui être adressé.

Par exemple, pour envoyer un message à l'utilisateur connecté `june`, entrez :

```
write june
```

Appuyez sur Entrée et tapez :

```
I need to see you! Meet me in the computer room at 12:30.
```

Appuyez ensuite sur Ctrl-D pour quitter le mode commande **write**.

Si votre ID utilisateur est `karen` et que vous travaillez sur la station de travail `tty3`, la station de travail de `june` affiche :

```
Message from karen on trek tty3 Aug 17 11:55:24 ...
I need to see you! Meet me in the computer room at 12:30.
```

```
<EOT>
```

Par exemple, pour poursuivre la conversation avec `june`, entrez :

```
write june
```

Appuyez sur Entrée et tapez :

```
Meet me in the computer room at 12:30.
o
```

La conversation démarre. Le `o` au début de la ligne suivante indique que le message est terminé (over). Il indique à June que vous attendez une réponse. N'appuyez pas sur Ctrl-D si vous souhaitez poursuivre.

June répond par :

```
write karen
```

Elle appuie sur Entrée et tape :

```
I'm running tests at 12:30. Can we meet at 3?
o
```

Vous pouvez répondre :

```
OK--the computer room at 3.
oo
```

La mention `oo` signifie "over and out," informant June que vous n'avez plus rien à dire. Si June indique également `oo`, vous n'avez plus qu'à appuyer tous deux sur Ctrl-D pour mettre fin à la conversation.

Par exemple, pour adresser à `june` un message préalablement inclus dans un fichier, entrez :

```
write june < message.text
```

Le contenu du fichier **message.text** s'affiche sur la station de travail de `june`.

Par exemple, pour écrire à la personne travaillant sur la station `console`, entrez :

```
write console
```

Appuyez sur Entrée et tapez :

```
The printer in building 998 has jammed.
Please send help.
```

Appuyez ensuite sur Ctrl-D.

Le message est envoyé à la personne connectée sur la station `/dev/console`.

La commande **write** permet également de converser avec des utilisateurs connectés à d'autres hôtes. Il suffit d'identifier l'utilisateur concerné via l'indicateur `-nNomHôte` ou le paramètre `Utilisateur@Hôte`. Il est également indispensable que le démon **writesrv** soit actif à la fois sur l'hôte courant et sur l'hôte distant.

Par exemple, pour envoyer un message à l'utilisateur `spuds` sur l'hôte distant `party`, entrez :

```
write -n party spuds
```

Appuyez sur Entrée et tapez :

```
Your new tape has just arrived,
come see me to pick it up.
Thanks!
```

Appuyez ensuite sur Ctrl-D.

OU

```
write spuds@party
```

Appuyez sur Entrée et tapez :

```
Your new tape has just arrived,
come see me to pick it up.
Thanks!
```

Appuyez ensuite sur Ctrl-D.

Reportez-vous à la commande **write** dans le manuel *AIX Commands Reference*.

---

## Récapitulatif des commandes

<b>apropos</b>	Recherche des commandes par mot-clé.
<b>cal</b>	Affiche un calendrier.
<b>calendar</b>	Ecrit les messages de l'agenda sur la sortie standard.
<b>factor</b>	Décompose un nombre en facteurs premiers.
<b>help</b>	Affiche l'aide.
<b>learn</b>	Appelle le didacticiel, qui propose cours et exercices concernant l'utilisation des fichiers, des macros, etc.
<b>leave</b>	Affiche les messages de l'agenda.
<b>units</b>	Convertit une quantité en une autre unité de mesure.
<b>write</b>	Envoie des messages à d'autres utilisateurs du système.



---

# Chapitre 13. Service de bibliothèque documentaire

---

## Utilisation de la documentation AIX en ligne

La documentation AIX en ligne est fournie sur les deux CD-ROM suivants :

- 86 A2 72JX: Hypertext Library. Basic Subset for AIX 4.3
- 86 X2 73JX: Hypertext Library. Full Set for AIX 4.3

Les consignes d'installation de *Hypertext Library* sont données dans le livret du CD-ROM et doivent être suivies scrupuleusement.

*Hypertext Library* est fournie avec des utilitaires appelés *Hypertext Library Utilities*. Ces utilitaires comprennent une fonction **Recherche** permettant de rechercher des informations dans l'ensemble de la bibliothèque et une capacité **Multi-impression** permettant d'imprimer plusieurs documents par un simple clic dans la fenêtre Résultats de la recherche.

*Hypertext Library* et *Hypertext Library Utilities* disposent tous deux d'une interface graphique et de caractères.

Le contenu de *Hypertext Library* et *Hypertext Library Utilities* est décrit dans la page d'accueil de *Hypertext Library*.

Pour de plus amples informations, consulter la brochure : "*About the Documentation CD-ROM*".

Les informations données dans ce chapitre **ne** concernent **pas** *Hypertext Library* et il est **fortement déconseillé** d'utiliser le *Service Bibliothèque* décrit ci-après avec *Hypertext Library*.



---

# Annexe A. Accès aux informations avec InfoExplorer

Cette section décrit les différents types de documentation disponibles dans la *bibliothèque de base de données en hypertexte* et la façon d'y accéder à l'aide d'InfoExplorer.

---

## A l'aide de l'interface InfoExplorer (ASCII)

L'interface ASCII du programme InfoExplorer propose une formation pour les utilisateurs novices et contient des procédures d'exécution d'autres tâches, comme indiqué dans la présentation suivante.

Le programme InfoExplorer est l'outil qui vous sert à faire connaissance avec le système d'exploitation AIX et d'autres logiciels. En tant qu'utilisateur d'une interface ASCII, vous avez accès à de nombreux volumes de documentation sur les logiciels et le matériel. Comme les informations d'InfoExplorer sont en *hypertexte*, vous n'avez aucune page à tourner pour en prendre connaissance. Vous n'avez qu'à visualiser les informations qui s'affichent et sélectionner des liens pour passer d'un article à un autre. Vous pouvez, par ce moyen, naviguer à votre guise dans la documentation en hypertexte.

## A l'aide des écrans InfoExplorer

Dans l'interface ASCII d'InfoExplorer, différents écrans correspondent à différents types d'informations. Par exemple, certains écrans servent à naviguer. Il s'agit d'écrans de *navigation* (par exemple, l'index et la bibliographie). InfoExplorer affiche chaque écran de navigation un par un. A chaque sélection d'une fonction, cet écran laisse la place aux informations voulues.

Un autre type d'écran contient le texte à lire. Il s'agit dans ce cas d'écrans de *lecture*. Ces derniers contiennent des informations relatives aux concepts, aux procédures ou de référence. Ils affichent des articles qui vous renseignent sur un sujet précis, vous expliquent une procédure ou vous informent sur des commandes, des sous-routines, des fichiers ou des formats de fichiers.

Vous pouvez, à partir d'écrans de navigation et de lecture, ouvrir d'autres écrans pour exécuter des tâches spécifiques. Le programme InfoExplorer fournit les écrans orientés tâches suivants :

<b>Recherche simple</b>	Gère les recherches à partir d'un titre ou d'informations à partir d'un seul ensemble de mots.
<b>Recherche composée</b>	Gère les recherches dans une ou plusieurs bases de données sélectionnées à partir de chaînes de recherche composées.
<b>Recherche de liste de correspondances</b>	Affiche l'emplacement des occurrences trouvées, leur nombre et le nombre d'articles dans lesquels elles ont été détectées.
<b>Glossaire</b>	Affiche un terme ou une expression du glossaire sélectionnée dans un écran de lecture.
<b>Valeurs par défaut d'éditeur</b>	Permet de définir des valeurs par défaut pour diverses fonctions d'InfoExplorer.
<b>Préférences d'éditeur</b>	Permet de personnaliser les options du programme InfoExplorer.

<b>Liste de liens</b>	Permet de sélectionner un lien précédent et d'accéder aux informations correspondantes, de supprimer un lien précédent de la liste, de vider toutes les listes ou de renommer un lien précédent.
<b>Liste de fichiers</b>	Permet de charger un fichier préalablement sauvegardé, de sauvegarder une liste dans un fichier ou de supprimer un fichier.

Lorsque vous démarrez une session, le programme InfoExplorer commence par afficher l'écran de navigation, puis l'écran de lecture. L'écran de lecture contient un article d'accueil ("Welcome to the InfoExplorer ASCII Interface") qui vous explique comment effectuer les opérations de base et accéder à l'aide, aux informations de copyright et de marques d'InfoExplorer. Pour revenir à l'écran de navigation, tapez Ctrl-w.

## A l'aide des menus

L'interface ASCII d'InfoExplorer affiche une barre de menus en haut de chaque écran. Vous n'avez qu'à sélectionner certaines options pour accéder à des fonctions spécifiques d'InfoExplorer.

## Obtention de l'aide

Pour avoir plus de détails sur l'utilisation du programme InfoExplorer, vous pouvez avoir recours à l'aide d' InfoExplorer. L'aide vous permet d'obtenir des informations sur l'utilisation de l'écran que vous regardez. Vous pouvez également aller consulter une liste d'articles décrivant le fonctionnement d'InfoExplorer.

Pour obtenir de l'aide, appuyez sur les touches Ctrl-O, mettez en surbrillance le mot **Help** à l'aide des flèches et appuyez sur Entrée. Vous pouvez alors choisir entre les options suivantes :

<b>On Screen</b>	Affiche les informations d'aide spécifiques pour l'écran en cours de visualisation, notamment les liens aux articles décrivant l'écran, les fonctions disponibles à partir de cet écran, ainsi que la liste des menus disponibles à partir de lui, avec des liens vers les descriptions des options du menu.  Vous pouvez sélectionner des liens pour afficher les informations qui vous intéressent. Pour revenir à l'écran d'information sur l'aide après sélection d'un lien, choisissez l'option <b>Path</b> ou <b>.History</b> .
<b>List of Helps</b>	Affiche la <b>liste d'aides</b> , liste de liens en hypertexte à des articles décrivant des aspects fondamentaux du programme et de la documentation InfoExplorer, ainsi que des liens aux aides correspondant à chaque écran.

---

## Initiation à InfoExplorer (ASCII)

InfoExplorer est un puissant outil de récupération de texte doté de plusieurs systèmes d'aide à la navigation pour vous aider à trouver et gérer des informations. Vous n'avez besoin de connaître que quelques uns de ces outils d'aide pour commencer à utiliser ce programme. Servez-vous des procédures de base présentées dans ce chapitre pour explorer la base de données InfoExplorer et y trouver des informations pratiques.

Suivez le didacticiel pas à pas à l'aide des touches de déplacement du curseur du clavier, ou utilisez les liens aux procédures individuelles.

Dans cet article, vous apprendrez à :

- Lancement d'InfoExplorer (ASCII), page A-3
- Reconnaissance des types d'écran (InfoExplorer ASCII), page A-3
- Sélection d'un lien hypertexte (InfoExplorer ASCII), page A-5
- Défilement des informations (InfoExplorer ASCII), page A-5
- Sélection d'une option de menu (InfoExplorer ASCII), page A-6
- Retour à un emplacement précédent (InfoExplorer ASCII), page A-6
- Recherche d'informations (InfoExplorer ASCII), page A-7
- Impression d'informations (InfoExplorer ASCII), page A-10
- Accès à l'aide (InfoExplorer ASCII), page A-11
- Touches et combinaisons de touches spéciales (InfoExplorer ASCII), page A-12
- Lancement d'une bibliothèque InfoExplorer secondaire (InfoExplorer ASCII), page A-14
- Arrêt d'InfoExplorer (ASCII), page A-14.

### Lancement d'InfoExplorer (ASCII)

Pour démarrer InfoExplorer, tapez `info` à une invite de commande et appuyez sur Entrée.

**Remarque :** Pour démarrer InfoExplorer (ASCII) dans une fenêtre AIX, entrez `info -a` et appuyez sur Entrée.

L'article d'accueil "Welcome to the InfoExplorer ASCII Interface" ou l'article de navigation par défaut s'affiche dans l'écran de lecture. L'écran de bienvenue explique comment effectuer des opérations de base sur les écrans et comment accéder aux informations relatives à InfoExplorer (aide, informations de copyright et sur les marques).

### Reconnaissance des types d'écran (InfoExplorer ASCII)

Il existe deux écrans principaux dans le programme InfoExplorer : l'*écran de navigation* et l'*écran de lecture*.

## Ecran de navigation

L'écran de navigation sert de point de départ à la recherche de documentation sur le système d'exploitation et sur les autres programmes disponibles sur votre station de travail, et contient des informations pour vous aider à trouver la documentation désirée. Les informations sont organisées par sujet et par tâche, par livre ou de façon alphabétique par commande ou par élément de référence de programmation. Les routes de navigation principales s'affichent sur l'écran de navigation :

<b>Topic &amp; Task Index</b>	Affiche les informations par tâche. Il s'agit de l'article de navigation par défaut.
<b>Commands</b>	Affiche la liste alphabétique des commandes disponibles.
<b>Books</b>	Affiche les articles dans l'ordre des manuels.
<b>Programming Reference</b>	Affiche les listes des fonctions de programmation dans l'ordre alphabétique, par catégorie fonctionnelle.

**Remarque :** Les informations de navigation indiquées dans ce chapitre s'appliquent au programme InfoExplorer. Elles peuvent ou non s'appliquer à d'autres bibliothèques.

La première ligne de l'écran contient la barre de menu, et la deuxième une barre de titre en vidéo inverse. Le titre de l'écran s'affiche à l'extrémité droite de la barre de titre. Cette dernière sépare également les options de menu du texte de l'article.

Pendant que vous visualisez le programme InfoExplorer par l'intermédiaire de l'interface ASCII, vous ne pouvez disposer que d'un écran de navigation à la fois. A chaque sélection d'un lien vers un autre article de navigation, le contenu de l'article sélectionné remplace celui de l'article original. Par exemple, si vous regardez l'index des sujets et des tâches (Topic & Task Index) et que vous décidez de visualiser les commandes (Commands), la liste des commandes s'affiche à la place de l'index sur l'écran de navigation.

## Ecran de lecture

L'écran de lecture contient des informations de référence, conceptuelles et de procédure. Les informations de procédure expliquent la façon d'exécuter une tâche, les informations conceptuelles traitent de différents sujets et les informations de référence donnent des informations sur les commandes, les sous-routines, etc.

Comme dans l'écran de navigation, la première ligne de l'écran contient la barre de menu, et la deuxième une barre de titre en vidéo inverse. "Info Document" s'affiche à l'extrémité droite de la barre de titre. Cette dernière sépare également les options de menu du texte de l'article.

Pendant que vous visualisez le programme InfoExplorer par l'intermédiaire de l'interface ASCII, vous ne pouvez disposer que d'un écran de lecture à la fois. A chaque sélection d'un lien vers un autre article de navigation, le contenu de l'article sélectionné remplace celui de l'article original. Par exemple, si vous lisez l'article "Starting the InfoExplorer Program (ASCII Interface)" et décidez d'afficher l'article sur la commande **info**, ce dernier s'affiche à la place du premier dans l'écran de lecture.

## Déplacement entre les écrans

Pour afficher l'écran de navigation à partir de l'écran de lecture (ou pour revenir à l'écran de lecture courant à partir de l'écran de navigation), appuyez sur la combinaison de touches Ctrl-w. Cette dernière permet de basculer entre écran de navigation et écran de lecture. Vous avez également la possibilité d'utiliser les options dans la barre de menu.

Avant de continuer, entraînez-vous à basculer d'un écran à l'autre. Observez les différences entre les deux types d'écrans.

Il existe d'autres types d'écran (*search*, *glossary*, *file* et *utility*), qui sont respectivement des écrans de recherche, de glossaire, de fichiers et d'utilitaires. Pour plus de détails à ce sujet, reportez-vous à l'article InfoExplorer en ligne Accessing Help on a Specific Screen.

## Sélection d'un lien hypertexte (InfoExplorer ASCII)

Imaginez que vous ayez besoin d'informations sur l'impression d'un fichier sur une imprimante ligne par ligne. Vous pouvez initier votre recherche à l'aide d'une route de navigation principale, par exemple la liste des tâches. A partir de là, vous pouvez suivre les liens hypertexte jusqu'à ce que vous atteigniez l'écran contenant les informations voulues.

Un lien hypertexte est une connexion entre deux segments d'informations. Ces liens apparaissent sous forme de texte souligné. Lorsque vous placez le curseur sur un lien, celui-ci passe en vidéo inverse. Pour afficher la cible ou le segment d'informations connecté, appuyez sur Entrée.

L'écran de bienvenue affiche cinq liens. Ils apparaissent sous la forme d'une liste de cinq éléments soulignés.

Pour passer au lien hypertexte suivant, appuyez sur la touche de tabulation ou sur les touches Ctrl-f. Si le lien suivant n'est pas visible, l'écran se rafraîchit pour l'afficher. Pour revenir au lien précédent, appuyez sur Ctrl-b.

Pour sélectionner un lien hypertexte, procédez comme suit :

1. Si le curseur ne se trouve pas dans la partie texte de l'écran, appuyez sur Ctrl-o.
2. Pour passer au lien hypertexte suivant, appuyez sur la touche de tabulation ou sur les touches Ctrl-f.
3. Pour revenir au lien précédent, appuyez sur Ctrl-b.
4. Pour sélectionner le lien, appuyez sur Entrée. Le texte cible s'affiche.

Exercez-vous à déplacer le curseur d'un lien à l'autre avec les touches Ctrl-f et Ctrl-b. Avant de passer à la section suivante, essayez de sélectionner l'un des liens dans l'écran d'accueil. N'oubliez pas qu'il suffit d'appuyer sur Ctrl-w pour revenir au premier écran de navigation.

**Remarque :** Pour éviter de perdre cet écran, ne sélectionnez pas de liens dans cet article.

Vous ne pouvez pas afficher de graphique sur un terminal ASCII. En cas de sélection d'un lien vers un graphique, un message signale que le graphique ne peut pas s'afficher. Il vous invite à appuyer sur une touche pour continuer. Par exemple, les articles relatifs aux commandes contiennent des diagrammes de syntaxe ne pouvant être visualisés que sur un affichage graphique. Vous pouvez toutefois visualiser la version avec crochets et accolades de la syntaxe sur un terminal ASCII.

## Défilement des informations (InfoExplorer ASCII)

Vous pouvez vous déplacer dans le texte affiché, dans l'écran de navigation comme dans l'écran de lecture, à l'aide des touches et des combinaisons de touches qui ont été définies à cette fin.

- Pour passer à l'écran précédent, appuyez sur la touche PgHaut ou sur les touches Ctrl-P.
- Pour passer à l'écran suivant, appuyez sur la touche PgBas ou sur les touches Ctrl-N.
- Pour remonter le curseur d'une ligne, appuyez sur la flèche vers le haut.
- Pour descendre le curseur d'une ligne, appuyez sur la flèche vers le bas.

Avant de continuer, faites des essais de déplacement dans le texte sur l'écran du terminal.

## Sélection d'une option de menu (InfoExplorer ASCII)

Chaque écran InfoExplorer contient une barre de menu qui se trouve en haut de l'écran. Cette barre de menu contient la plupart des menus utiles pour les fonctions d'InfoExplorer. Par exemple, le menu Display permet de choisir le menu de navigation que vous souhaitez afficher dans l'écran de navigation et le menu d'aide vous donne des indications sur l'utilisation du programme InfoExplorer.

Pour sélectionner une option de menu, procédez comme suit :

1. Activez la barre de menu en appuyant sur Ctrl-O.
2. Servez-vous des flèches de déplacement du curseur vers la droite et la gauche pour mettre en évidence le titre du menu à afficher et appuyez sur Entrée, ou entrez le caractère souligné dans le titre du menu. Le menu s'affiche comme un panneau déroulant avec plusieurs options.
3. Servez-vous des flèches de déplacement du curseur vers le haut et le bas pour mettre en évidence une option du menu à afficher et appuyez sur Entrée, ou entrez le caractère souligné dans l'option du menu. L'option sélectionnée est appelée ou un autre menu s'affiche dans une boîte de dialogue. Les options de menu s'affichent dans la barre de ce menu.

Une fois un menu affiché, vous souhaitez peut-être le supprimer sans sélectionner d'option. Pour cela, appuyez sur Ctrl-O. Le curseur est réaffiché.

Pour sélectionner une option dans une boîte de dialogue, procédez comme suit :

1. Activez la barre de menu avec Ctrl-O.
2. Mettez une option en évidence soit en tapant le caractère souligné dans l'option voulue, soit en utilisant les touches de déplacement du curseur vers la gauche ou la droite.
3. Sélectionnez l'option en appuyant sur Entrée. La boîte de dialogue se ferme et l'action appropriée est prise.

Une fois une boîte de dialogue affichée, vous souhaitez peut-être le supprimer sans sélectionner d'option. Pour cela, appuyez sur Ctrl-O et sélectionnez **Quit** dans la barre de menu.

Avant de passer à la section suivante, entraînez-vous à l'utilisation des menus. Essayez d'afficher le menu Path et de sélectionner l'option **Show List**. Une boîte de dialogue affiche la liste Path. Pour la fermer, sélectionnez **Quit** dans la barre de menu.

**Remarque :** Pour éviter de perdre cet article, ne sélectionnez pas un emplacement dans la liste Path.

## Retour à un emplacement précédent (InfoExplorer ASCII)

Pendant la session InfoExplorer, le programme conserve deux enregistrements des emplacements où vous vous êtes rendu : la liste History et la liste Path. Ces listes permettent de revenir à des emplacements précédents dans la base de données.

La liste History enregistre chaque emplacement visité par vous dans la base de données. Pour l'afficher, sélectionnez le menu History, puis l'option **List All**. Pour savoir comment utiliser la liste, allez dans le menu d'aide et sélectionnez **List of Helps** dans le menu déroulant. **La rubrique Finding Information in InfoExplorer** dans la liste d'aide apporte des éclaircissements sur la fonction History.

La liste Path conserve le suivi de votre route de navigation en commençant par un article principal (Topic & Task Index, Books, Commands ou Programming Reference). Chaque fois que vous revenez à un écran de navigation principal, votre liste des chemins précédents est écrasée.

Le menu Path contient trois options : **Show List**, **Previous** et **Next**. Sélectionnez l'option **Show List** pour afficher la liste Path. Servez-vous des options **Previous** et **Next** pour passer à l'emplacement respectivement précédent et suivant sans afficher la liste Path.

Pour revenir à un emplacement précédent dans le programme InfoExplorer, procédez comme suit :

1. Activez la barre de menu en appuyant sur Ctrl–O.
2. Tapez P pour afficher le menu Path.
3. Tapez S pour afficher la liste Path. Une boîte de dialogue affiche la liste des emplacements. Si vous vous trouvez dans un écran de navigation principal, vous recevez un message indiquant que la liste est vide.
4. Mettez en évidence l'emplacement où vous souhaitez revenir grâce aux flèches PgHt et PgBas.
5. Sélectionnez l'option **Go to** dans la barre de menu. Le texte s'affiche dans l'écran de lecture.

Avant de poursuivre, entraînez–vous à utiliser les options **Next** et **Previous** pour passer d'un emplacement à l'autre. N'oubliez pas que si vous remontez le chemin jusqu'à un article de navigation principal (comme l'index des tâches), votre liste de chemins est supprimée et vous devez recommencer à sélectionner des articles pour reconstituer un nouveau chemin. Pour revenir à un emplacement ne faisant pas partie du chemin courant, servez–vous de la fonction History.

**Remarque :** Pour éviter de perdre cet écran, n'utilisez pas la fonction Path à partir de cet écran.

## Recherche d'informations (InfoExplorer ASCII)

La première partie de ce chapitre décrit la façon de trouver des informations à l'aide de routes de navigation principales. Mais que faire lorsqu'on ne connaît ni la tâche, ni la commande, ni dans quel manuel rechercher les informations ? Vous pouvez effectuer une recherche sur pratiquement chaque mot dans la base de données InfoExplorer. A l'aide de la fonction de recherche d'InfoExplorer, vous pouvez rechercher des informations par mot ou par expression.

Deux types de recherche sont possibles : la recherche simple et la recherche composée. La *recherche simple* permet de rechercher un mot ou une chaîne de mots (dans leur ordre exact) dans un texte et/ou des titres d'article. (Dans la version ASCII d'InfoExplorer, la sélection Article Title du menu Search permet de limiter la recherche aux titres d'articles.) La *recherche composée* permet de rechercher un mot, un ensemble de mots ou plusieurs ensembles de mots. Vous pouvez affiner la recherche à l'aide des options suivantes :

- **AND, OR** et **BUT NOT**.
- Options de proximité
- Catégories de recherche

Vous pouvez spécifier les bases de données dans lesquelles effectuer les recherches simple et composée. Deux méthodes de spécification d'une base de données existent :

- Vous pouvez spécifier une sélection de base de données temporaire dans la fenêtre Compound Search. Cette sélection s'applique aux recherches simple et composée et reste effective pour la session courante d'InfoExplorer, ou jusqu'à ce que vous en changiez dans la session.
- Vous pouvez spécifier une sélection de base de données par défaut en choisissant Defaults dans le menu des informations et en sélectionnant l'option **DBselect**. Une sélection de base de données effectuée à partir de la fenêtre Defaults s'applique aux recherches simple et composée et reste effective pour les sessions futures d'Info. (Les sélections de base de données temporaires effectuées à partir de la recherche Compound Search peuvent remplacer la sélection par défaut pendant une session InfoExplorer, mais la sélection par défaut revient lorsque vous redémarrez InfoExplorer.)

Ainsi, les étapes suivantes vous guident dans une recherche simple de la commande **qprt**, puis affinent la recherche avec une sélection de base à partir du menu Compound Search.

**Remarque :** Pour éviter de perdre cet article, n'essayez pas d'effectuer une recherche à partir de cet écran. Si votre système est configuré pour l'impression, vous pouvez imprimer un exemplaire de cet article à l'aide du menu info en sélectionnant l'option **Print Article**. Servez-vous de l'exemplaire imprimé pour suivre la procédure.

Pour exécuter une recherche simple sur la commande **qprt**, procédez comme suit :

1. Affichez le menu Search dans la barre de menu.
2. Sélectionnez l'option **Simple** dans le menu Search. La boîte de dialogue Simple Search s'affiche.
3. Dans le champ Find, tapez `qprt command` et appuyez sur Entrée. La boîte de dialogue Search Match List affiche les informations suivantes :
  - Les bases de données dans lesquelles ont été trouvées des occurrences.
  - Le nombre d'occurrences trouvées.
  - Le nombre d'articles comportant des occurrences.
4. A l'aide des flèches PgHt et PgBas, mettez en évidence **AIX User Guides, System Management Guides, and Commands Reference** dans la liste des bases de données et appuyez sur Entrée. La boîte de dialogue Search : List of Titles affiche la liste des titres d'articles contenant la chaîne de recherche (qprt).

**Remarque :** Si un seul article est concerné, il s'affiche dans un écran de lecture (aucune boîte de dialogue ne s'affiche).

5. Sélectionnez le deuxième titre de la liste **qprt Command** et appuyez sur Entrée. L'article s'affiche dans l'écran de lecture. Vous remarquez que l'expression "qprt Command" est mise en évidence pour montrer l'emplacement du terme dans l'article.
6. Pour afficher l'occurrence suivante, passez dans la barre de menu et sélectionnez le menu Search.
7. Sélectionnez l'option **Next Match** et appuyez sur Entrée. L'occurrence suivante de "qprt command" s'affiche. Cette fonction permet le déplacement dans la base de données, vers d'autres emplacements dans l'article ou vers d'autres articles comportant le terme "qprt command".
8. Pour afficher l'occurrence précédente, sélectionnez l'option **Previous Match** dans le menu Search et appuyez sur Entrée. L'occurrence précédente de "qprt command" s'affiche.
9. Pour afficher à nouveau la liste des titres, sélectionnez l'option **Show Hit List** dans le menu Search et appuyez sur Entrée. La liste des titres s'affiche à nouveau.

S'il existe de nombreuses occurrences, il peut être difficile de localiser l'information exacte. Il vous faudra peut-être lire plusieurs articles pour trouver l'information voulue. Vous avez l'alternative d'utiliser une recherche composée pour rétrécir la recherche. Pour en savoir plus sur une recherche composée, servez-vous de la **liste d'aides** InfoExplorer, qui vous permet d'afficher étape par étape les instructions concernant les procédures les plus utilisées. Dans la liste des aides, sélectionnez **Performing a Compound Search**.

Une autre façon de réduire encore la portée de la recherche est d'aller dans le menu de recherche composée et de sélectionner la base de données la plus susceptible de contenir les informations souhaitées. Suivant votre système, il peut y avoir plusieurs bases de données à explorer. Par exemple, pour limiter la recherche aux informations concernant l'utilisation et la gestion, sélectionnez la base de données **Using, Managing, and Commands**.

Pour spécifier une base de données dans la fenêtre de recherche composée, procédez comme suit :

1. Appuyez sur Ctrl-O pour aller dans la barre de menu.

2. Sélectionnez l'option **DBselect**. La liste Databases List s'affiche.
3. Le symbole > (supérieur à) affiché à gauche de la base de données dans la liste indique que cette base fait partie de la recherche composée courante.
4. Marquez les bases à exclure de la recherche en procédant comme suit :
  - a. Servez-vous des touches PgHt et PgBas pour sélectionner une base de données couramment incluse dans la recherche, que vous voulez supprimer de la recherche. La base de données est mise en évidence.
  - b. Appuyez sur Ctrl-O pour aller dans la barre de menu de la boîte de dialogue Databases List.
  - c. Sélectionnez l'option **Bypass**. Le symbole > disparaît de la marge. Répétez cette procédure pour chaque base à supprimer de la recherche.
5. Marquez les bases à inclure dans la recherche en procédant comme suit :
  - a. Servez-vous des touches PgHt et PgBas pour sélectionner une base de données couramment exclue de la recherche. La base de données est mise en évidence.
  - b. Appuyez sur Ctrl-O pour aller dans la barre de menu de la boîte de dialogue Databases List.
  - c. Sélectionnez l'option **Select**. Le symbole > s'affiche à gauche de la base sélectionnée. Répétez cette procédure pour chaque base à explorer.
6. Lorsque la sélection des bases voulues est terminée, appuyez sur Ctrl-O pour aller dans la barre de menu et sélectionnez l'option **Quit**. La boîte de dialogue Compound Search s'affiche à nouveau (avec toutes les bases de données affichées dans le champ Search) : vous pouvez poursuivre votre recherche.

Lorsque vous spécifiez une sélection de base de données dans le menu Compound Search, celle-ci est conservée jusqu'à modification ultérieure ou sortie de la session InfoExplorer. Cette sélection de base de données s'applique à des recherches simple et composée. Vous pouvez sauvegarder une sélection de base de données en vue de sessions futures d'InfoExplorer à partir de la fenêtre Defaults du menu info.

## Impression d'informations (InfoExplorer ASCII)

Vous pouvez imprimer les informations suivantes à partir de l'interface InfoExplorer ASCII :

- un article provenant d'un écran de lecture,
- un article provenant d'un écran de navigation,
- les références provenant d'un écran de navigation,
- les articles d'une liste de recherche d'occurrences,
- des notes publiques et privées.

**Remarque :** Une ou plusieurs imprimantes doivent être configurées pour votre système.

### Pour imprimer un article d'un écran de lecture

1. Appuyez sur Ctrl-O pour aller dans la barre de menu. Par défaut, l'option **info** est mise en évidence.
2. Appuyez sur Entrée pour afficher les options du menu **info**.
3. Appuyez sur P pour sélectionner l'option **Print Article**. L'article couramment affiché dans l'écran de lecture s'imprime.

### Pour imprimer un article d'un écran de navigation

1. Appuyez sur Ctrl-O pour aller dans la barre de menu. Par défaut, l'option **info** est mise en évidence.
2. Appuyez sur Entrée pour afficher les options du menu **info**. Par défaut, l'option **Print Article** est mise en évidence.
3. Appuyez sur Entrée. L'article de navigation courant s'imprime.

### Pour imprimer des références à partir d'un écran de navigation

1. Appuyez sur Ctrl-O pour aller dans la barre de menu. Par défaut, l'option **info** est mise en évidence.
2. Appuyez sur Entrée pour afficher les options du menu **info**.
3. Appuyez sur la touche PgBas jusqu'à ce que l'option **Print References** soit mise en évidence.
4. Appuyez sur Entrée pour imprimer tous les articles du type références dans l'écran de navigation.

### Pour imprimer les articles d'une liste de recherche d'occurrences

1. Appuyez sur Ctrl-O pour aller dans la barre de menu. Par défaut, l'option **info** est mise en évidence.
2. Appuyez sur S pour sélectionner l'option **Search**.
3. Appuyez sur la touche PgBas jusqu'à ce que l'option **Show Hit List** soit mise en évidence et appuyez sur Entrée. La boîte de dialogue Search: List of Titles s'affiche.
4. Appuyez sur Ctrl-O pour activer la barre de menu de la boîte de dialogue Search : List of Titles.
5. Appuyez sur P pour sélectionner l'option **Print**. Tous les articles auxquels il est fait référence dans la boîte de dialogue Search: List of Titles s'impriment.

### Pour imprimer des notes publiques et privées

1. Appuyez sur Ctrl-O pour aller dans la barre de menu. Par défaut, l'option **info** est mise en évidence.
2. Appuyez sur N pour sélectionner l'option **Notes**. Par défaut, l'option **List All** est mise en évidence.

3. Appuyez sur Entrée pour afficher la boîte de dialogue Notes List.
4. Appuyez sur Ctrl–O pour aller dans la barre de menu de la boîte de dialogue Notes List.
5. Appuyez sur P pour sélectionner l'option **Print**. Toutes les notes de la boîte de dialogue Notes List s'impriment.

## Accès à l'aide (InfoExplorer ASCII)

A partir d'un écran InfoExplorer, vous pouvez visualiser les informations d'aide relatives aux fonctions et aux options disponibles sur l'écran affiché ou vous pouvez rechercher dans une liste de sujets sur lesquels des informations d'aide sont fournies.

### Pour accéder à l'aide relative à un écran spécifique

InfoExplorer utilise beaucoup d'écrans différents en plus des écrans principaux de navigation et de lecture, tels que :

- écrans de glossaires,
- liste des écrans de liens,
- écrans de fichiers,
- écrans de recherche simple et composée.

Chaque type d'écran a un rôle et une fonction particuliers. Le menu d'aide de chaque écran contient une option **On Screen**, qui affiche un article décrivant la fonction disponible sur l'écran courant.

Pour afficher de l'aide sur un écran spécifique, procédez comme suit :

1. Activez la barre de menu en appuyant sur Ctrl–O.
2. Mettez en évidence **Help** à l'aide des touches fléchées et appuyez sur Entrée pour afficher le menu d'aide.
3. Servez-vous des touches fléchées pour sélectionner **On Screen** et appuyez sur Entrée. Un article d'aide remplace l'article de lecture courant.

Avant de passer à la section suivante, essayez de consulter les articles d'aide disponibles pour les écrans ouverts. Servez-vous de la fonction Path pour revenir à cet écran.

### Pour accéder à la liste des aides

Le menu d'aide propose deux options d'aide : **List of Helps** et **On Screen**. **On Screen** permet d'obtenir de l'aide sur l'écran courant. Pour plus d'informations sur cette option, reportez-vous à "Accessing Help on a Specific Screen". L'option **List of Helps** affiche une liste de sujets d'aide relatifs à InfoExplorer. Pour utiliser la liste des aides, procédez comme suit :

1. Activez la barre de menu en appuyant sur Ctrl–O.
2. Mettez en évidence le menu d'aide à l'aide des touches fléchées et appuyez sur Entrée. Le menu d'aide s'affiche.
3. Sélectionnez l'option **List of Helps** pour afficher la liste d'aides.
4. Sélectionnez un sujet d'aide et appuyez sur Entrée.

La liste des aides contient les procédures étape par étape correspondant à toute une gamme de tâches. Avant de passer à la section suivante, sélectionnez une ou deux options de la liste d'aides. Etudiez les aides disponibles, puis entraînez-vous à les utiliser pour apprendre et pratiquer de nouvelles pratiques.

## Touches et combinaisons de touches spéciales (InfoExplorer ASCII)

Dans l'interface ASCII d'InfoExplorer, vous utilisez des touches et des combinaisons de touches spéciales pour vous déplacer au sein d'un écran, sélectionner des éléments d'un menu, accéder à d'autres écrans et démarrer des process. Suivant que vous travaillez à partir d'une zone de texte ou d'une barre de menu, différentes touches et combinaisons de touches sont activées.

### Touches et combinaisons de touches utilisées dans les zones de texte

Les touches et combinaisons de touches suivantes sont disponibles dans les zones de texte :

Touches	Action
<b>Ctrl-W</b>	Permet le déplacement entre l'écran de navigation et l'écran de lecture. Si l'écran de navigation s'affiche, vous pouvez appuyer sur Ctrl-W pour afficher l'écran de lecture. Si l'écran de lecture s'affiche, vous pouvez appuyer sur Ctrl-W pour afficher l'écran de navigation.
<b>Ctrl-F</b> ou <b>Tab</b>	Permet le déplacement sur le lien hypertexte suivant.
<b>Ctrl-B</b>	Permet le déplacement sur le lien hypertexte précédent.
<b>Entrée</b> ou <b>Retour</b>	Active l'opération. Par exemple, si vous avez déplacé le curseur sur un lien hypertexte et avez appuyé sur Entrée, le système suit le lien et affiche les informations cible.
<b>Ctrl-L</b>	Rafraîchit l'écran.
<b>Ctrl-N</b> ou <b>Page Bas</b>	Passe à l'écran suivant (sens vertical).
<b>Ctrl-P</b> ou <b>Page Haut</b>	Passe à l'écran précédent (sens vertical).
<b>Flèche gauche</b>	Déplace le curseur d'un caractère vers la gauche. Si le texte peut défiler horizontalement et que le curseur se trouve à l'extrémité gauche de l'écran, le texte défile d'un caractère.
<b>Flèche droite</b>	Déplace le curseur d'un caractère vers la droite. Si le texte peut défiler horizontalement et que le curseur se trouve à l'extrémité droite de l'écran, le texte défile d'un caractère.
	<b>Remarque :</b> Les touches fléchées ne fonctionnent pas pour le programme InfoExplorer sur les terminaux WYSE.
<b>Flèche vers le haut</b>	Déplace le curseur d'une ligne vers le haut. Si le curseur se trouve sur la ligne supérieure de l'écran, le texte défile verticalement, ligne par ligne.
<b>Flèche vers le bas</b>	Déplace le curseur d'une ligne vers le bas. Si le curseur se trouve sur la dernière ligne de l'écran, le texte défile verticalement, ligne par ligne.
<b>&lt;</b>	Déplace le curseur de 20 caractères vers la gauche. Si le texte peut défiler horizontalement, il défile de 20 caractères vers la gauche.
<b>&gt;</b>	Déplace le curseur de 20 caractères vers la droite. Si le texte peut défiler horizontalement, il défile de 20 caractères vers la droite.

## Touches et combinaisons de touches utilisées dans les barres de menu

Les touches et combinaisons de touches suivantes vous aident à utiliser la barre de menu :

Touches	Action
<b>Ctrl–O</b>	Active ou désactive la barre de menu. Si le curseur se trouve dans la zone texte de l'écran, vous pouvez appuyer sur Ctrl–O pour activer la barre de menu. Si la barre est déjà activée, vous pouvez appuyer sur Ctrl–O pour la désactiver et faire passer le curseur dans la zone de texte.
<b>Echap–Echap ou Ctrl–O</b>	Ferme un menu déroulant et place le curseur dans la zone de texte.
<b>Tab</b>	Passé à l'option suivante de la barre de menu. Si un menu déroulant n'est pas affiché et que vous appuyez sur la touche Flèche droite, l'option suivante s'affiche en vidéo inverse.  Dans les barres de menu des écrans affichés, déplace le curseur de la barre de menu vers la zone de texte si aucun menu déroulant n'est sélectionné.
<b>Flèche gauche</b>	Déplace le curseur sur l'option précédente de la barre de menus. Si un menu déroulant n'est pas affiché et que vous appuyez sur la touche Flèche gauche, l'option précédente s'affiche en vidéo inverse.  Si un menu déroulant s'affiche et que vous appuyez sur la touche Flèche gauche, l'option précédente de la barre de menu est sélectionné et son menu se déroule.
<b>Flèche droite</b>	Déplace le curseur sur l'option suivante de la barre de menus. Si un menu déroulant n'est pas affiché et que vous appuyez sur la touche Flèche droite, l'option suivante s'affiche en vidéo inverse.  Si un menu déroulant s'affiche et que vous appuyez sur la touche Flèche droite, l'option suivante de la barre de menu est sélectionné et son menu se déroule.
<b>Flèche vers le haut</b>	Affiche le menu déroulant de l'option sélectionnée ou déplace la sélection sur l'option précédente du menu déroulant.
<b>Flèche vers le bas</b>	Affiche le menu déroulant de l'option sélectionnée ou déplace la sélection sur l'option suivante du menu déroulant.
<b>Entrée ou Retour</b>	Active l'opération. Par exemple, si une option de la barre de menu s'affiche en vidéo inverse et que vous appuyez sur Entrée, le menu déroulant s'affiche. Si un menu déroulant est déjà affiché et que vous appuyez sur la touche Entrée, l'action appropriée est prise sur l'élément présenté en vidéo inverse.

Dans certains cas, la barre de menu d'une boîte de dialogue contient des options de démarrage ou d'arrêt d'un processus. Les touches suivantes permettent aussi de se déplacer ou de faire défiler des options à l'écran :

Touches	Action
<b>Tab</b>	Permet le déplacement sur le champ suivant.
<b>Flèche gauche</b>	Permet le déplacement sur l'option précédente d'une chaîne d'options.
<b>Flèche droite</b>	Permet le déplacement sur l'option suivante d'une chaîne d'options.
<b>Barre d'espace</b>	Permet le déplacement dans les options d'une chaîne d'options.
<b>Flèche vers le haut</b>	Sélectionne l'élément précédent dans une liste.
<b>Flèche vers le bas</b>	Sélectionne l'élément suivant dans une liste.
<b>Entrée ou Retour</b>	Active l'opération sélectionnée ou met fin à l'entrée de texte et passe au champ suivant.

## Lancement d'une bibliothèque InfoExplorer secondaire (InfoExplorer ASCII)

Vous pouvez demander au programme InfoExplorer de charger une autre bibliothèque de base de données en utilisant l'indicateur **-I** avec la commande **info**. Par exemple, pour démarrer la bibliothèque InfoExplorer contenant la documentation sur C++, FORTRAN et Ada, émettez la commande suivante :

```
info -I compiler
```

## Arrêt d'InfoExplorer (ASCII)

Pour quitter InfoExplorer à partir de l'écran de navigation ou de lecture, procédez comme suit :

1. Activez la barre de menu en appuyant sur Ctrl-O.
2. Mettez en évidence le menu Exit à l'aide des touches fléchées et appuyez sur Entrée. Le menu Exit s'affiche.
3. Sélectionnez l'option **Confirm** en appuyant sur Entrée. L'invite de commande du système d'exploitation s'affiche et le programme InfoExplorer s'arrête.

---

## Personnalisation d'InfoExplorer (ASCII)

Nombre de caractéristiques d'InfoExplorer peuvent être adaptées, pour mieux répondre à vos impératifs et à vos souhaits. Pour cela, vous définissez les valeurs par défaut et les préférences d'InfoExplorer. Par exemple, vous pouvez déterminer quelles bases de données utiliser et quelle imprimante utiliser comme valeur par défaut. Vous avez ainsi la possibilité de choisir la base d'informations à utiliser, l'imprimante par défaut, le nombre d'événements consignés dans l'historique, le type d'impression, etc.

La principale différence entre valeur par défaut et préférence est que la plupart des valeurs par défaut sont initialisées par le programme et que des valeurs par défaut distinctes peuvent être définies pour des bibliothèques distinctes, tandis que les préférences s'appliquent à toutes les bibliothèques. Dans certains cas, si vous modifiez une valeur par défaut (l'article de navigation, par exemple), il vous faut quitter le programme et le relancer. Les commandes d'impression entrent en vigueur une fois sauvegardées. Les préférences, en revanche, sont immédiatement prises en compte.

Les sections suivantes décrivent la personnalisation de l'interface InfoExplorer ASCII.

- Modification des paramètres par défaut InfoExplorer
- Choix des préférences

### Modification des paramètres par défaut d'InfoExplorer (ASCII)

Avec InfoExplorer, vous pouvez définir de nombreuses valeurs par défaut. Il s'agit des paramètres que le programme InfoExplorer utilise au lancement. Sélectionnez **Defaults** dans le sous-menu **Options** du menu déroulant InfoExplorer pour définir les paramètres :

- article de navigation,
- fichiers bookmark, history et note,
- fichier modèle de note,
- bases de données à explorer,
- imprimantes.

### Désignation de l'article de navigation par défaut

Par défaut, InfoExplorer affiche l'article Topic & Task Index comme article de navigation principal. Vous pouvez choisir un autre article : au lancement d'InfoExplorer, le nouvel article sera automatiquement affiché comme article de navigation.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

### Désignation de fichiers par défaut

Si vous travaillez avec un stagiaire, vous souhaitez sans doute qu'il suive un chemin déterminé lorsqu'il accède à InfoExplorer. Pour gérer ce type d'action, vous pouvez désigner un fichier comme fichier d'historique par défaut : lorsque vous spécifiez ensuite l'option **.History**, ce fichier vous conduit (vous ou votre stagiaire) via le chemin désigné.

Vous pouvez de la même façon désigner des fichiers bookmarks ou un fichier notes par défaut, et suivre les chemins correspondants dans les sessions suivantes.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Désignation d'un fichier modèle de notes par défaut

Lorsque vous créez une note dans la documentation hypertexte, l'écran de l'éditeur apparaît vide : vous pouvez taper votre note. Si vous souhaitez vous servir d'un format particulier, vous pouvez spécifier un modèle de notes. Si vous en avez créé plusieurs, vous pouvez en désigner un comme modèle par défaut. Si vous utilisez ensuite l'option **Note**, InfoExplorer affiche le modèle par défaut.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Modification de la base de données à explorer

Par défaut, InfoExplorer explore les bases chargées avec le programme. Mais il est parfois inutile de les explorer toutes. Pour mieux cibler les recherches, vous pouvez indiquer les bases qu'il convient d'explorer, pour la session en cours et les suivantes. Et, si vous changez d'avis, il est très simple de revenir à la sélection par défaut.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Désignation des imprimantes par défaut

Le choix d'une imprimante dépend, entre autres, du type d'impression souhaité, simple ou sophistiqué. Sous InfoExplorer, vous pouvez désigner l'imprimante par défaut pour chaque type d'impression. simple ou sophistiquée. L'option **Pretty Print** fait appel au programme **troff**, qui convertit les documents en sorties exploitables par une photocomposeuse. Lorsque vous émettez une commande d'impression pour un écran, la sortie est automatiquement dirigée vers l'imprimante ad hoc. Si vous ne spécifiez pas une file d'attente, la valeur par défaut consiste à utiliser la première file d'attente spécifiée dans votre fichier **/etc/qconfig**.

Par exemple, si vous avez désigné l'imprimante `daves3812` comme imprimante par défaut et choisi l'impression Pretty (sophistiquée), le travail d'impression est automatiquement envoyé sur cette imprimante lorsque vous sélectionnez l'option **Print**.

**Remarque :** L'option Pretty Print n'est active que si le programme de formatage de document **troff** est installé sur le système.

Pour plus de détails sur la procédure, reportez-vous à Setting Defaults (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Choix des préférences dans InfoExplorer (ASCII)

Avec InfoExplorer, vous pouvez définir de nombreuses préférences. Pour cela, sélectionnez l'option **Preferences** du menu déroulant info :

- redéfinir la taille de l'historique,
- choisir l'éditeur de notes,
- définir le type de sortie imprimée.

## Modification de la taille de l'historique

Un historique peut comporter entre 0 (zéro) et 1000 événements: La valeur système par défaut est de 100 événements. Vous pouvez adapter cette taille par défaut à votre situation particulière. Vous pouvez ensuite appliquer les modifications à la session courante uniquement ou les sauvegarder pour les sessions courante et futures.

Pour plus de détails sur la procédure, reportez-vous à Setting Preferences (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Choix de l'éditeur de notes

Plusieurs éditeurs de notes sont disponibles dans l'interface ASCII d'InfoExplorer, notamment INed et vi. L'éditeur de notes par défaut du système est vi. Si vous préférez utiliser Ined ou un autre éditeur pour écrire vos notes, vous pouvez modifier la valeur par défaut. Vous pouvez ensuite appliquer les modifications à la session courante uniquement ou les sauvegarder pour les sessions courante et futures.

Pour plus de détails sur la procédure, reportez-vous à Setting Preferences (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

## Choix du type d'impression

InfoExplorer (ASCII) propose deux types d'impression : simple et sophistiquée. Par défaut, l'impression est de type simple, c'est-à-dire sans gras ni différenciation de polices. Vous pouvez modifier la valeur par défaut.

**Remarque :** L'option Pretty Print n'est active que si le programme de formatage de document **troff** est installé sur le système. Le programme **troff** convertit un document en sortie pour une photocomposeuse.

Vous pouvez choisir une autre valeur, applicable à la seule session en cours ou également aux sessions suivantes.

Pour plus de détails sur la procédure, reportez-vous à Setting Preferences (InfoExplorer ASCII) dans l'aide en ligne InfoExplorer.

---

## Ressources X disponibles sous InfoExplorer

Vous pouvez personnaliser la couleur, la position et la taille des fenêtres sous InfoExplorer en définissant les ressources X dans le fichier **.xdefaults**. Cette méthode s'applique aux types de fenêtres suivants :

Type de fenêtre	Nom de la ressource
Bookmark List	blisTopLevelShell
Citations	citTopLevelShell
Compound Search	compoundTopLevelShell
Database Selection	dbSelectionTopLevelShell
Defaults Editor	defTopLevelShell
Document (Reading)	docTopLevelShell
File (file link)	fileTopLevelShell
File Selection	fileSelTopLevelShell
Footnote	footTopLevelShell
Glossary	glossaryTopLevelShell
Graphic	graphicTopLevelShell
History List	hlistTopLevelShell
Introduction	introTopLevelShell
Library Definition	libinstTopLevelShell
Library Selection	libSelTopLevelShell
List of Titles	tlistTopLevelShell
Message	messageTopLevelShell
Match List	mlSelectionTopLevelShell
Navigation	navTopLevelShell
Note Editor	neditTopLevelShell
Note List	nlistTopLevelShell
Outline	outlineTopLevelShell
Path List	plistTopLevelShell
Preferences	prefTopLevelShell
References	refTopLevelShell
Simple Search	searchTopLevelShell

Le nom de ressource de plus haut niveau sous InfoExplorer est Info-gr. Par exemple, pour modifier la largeur de la fenêtre de navigation, insérez dans le fichier **.xdefaults** le nom de ressource :

```
Info_gr.navTopLevelShell.width: 800
```

Pour afficher les polices ISO, insérez dans le fichier **.xdefaults** le nom de ressource :

```
Info_gr.fontfilestr: isofonts
```

---

# Index

## Symboles

- . répertoires (point), 5-7
- .. répertoires (point point), 5-7

## Nombres

410ap148416, 11-87

## A

### affichage

- calendrier, 12-2
- contenu de fichier, 6-9
- écran disponible, 2-4
- fichier
  - dernières lignes, 6-12
  - premières lignes, 6-12
- ID utilisateur, 1-7
- informations d'aide, 12-3
- informations de contrôle des accès, 9-12
- informations sur un groupe d'utilisateurs, 9-6
- logiciels, 2-5
- messages de l'agenda, 12-2
- nom de connexion, 1-6
- nom de console, 2-3
- nom du système, 1-7
- nom du terminal, 2-3
- polices disponibles, 2-4
- répertoire de fichiers
  - contenu, 5-10
  - courant, 5-9
  - texte affiché en gros caractères, 4-8

affichage à trois chiffres, 1-3

alias, commandes, shell Korn ou POSIX, 11-15

alias de commandes, shell Korn ou POSIX, 11-15

- substitution de tilde, 11-16

arguments, 3-3

### arithmétique

- conversion des unités, 12-5
- facteur premier, 12-4

### ASCII – PostScript

- automatisation de la conversion, 7-16
- conversion de fichiers, 7-16

## B

### backup

- commande, 8-12
- compression préalable de fichiers, 8-9
- utilisation, 8-12
- utilisation de la commande smit, 8-13

### bande

- copie, 8-9
- vérification d'intégrité, 8-9

bibliothèque, préférée, interface Window, 10-15

## C

calendar, commande, 12-2

calendrier, affichage, 12-2

caractère joker, 6-3

chaînage, définition, 3-2, 4-5

chaînes, recherche dans des fichiers texte, 6-10

changement de nom

- fichier, 6-6

- répertoires, 5-8

chemin, répertoire, 5-6

chemin d'accès

- absolu, 5-7, 6-3

- définition, 6-3

- répertoire, 5-6

classes, utilisateur, 9-4

commande acledit, 9-13

commande aclget, 9-12

commande aclput, 9-12

commande aixterm, 2-8

commande alias, 3-10

commande apropos, 12-1

commande at, 3-19, 3-20

commande atq, 3-20

commande banner, 4-8

commande bsh, 11-4, 11-70

commande cal, 12-2

commande capture, 4-7

commande cat, 4-3, 6-10

commande cd, 5-6, 5-9

commande chfont, 10-11

commande chmod, 9-8

commande chown, 9-4

commande chpq, 7-16

commande clear, 4-6

commande colrm, 6-15

commande compress, 8-9

commande cp, 5-9, 6-7

commande cpio -o, 8-7

commande csh, 11-4, 11-92

commande cut, 6-13

commande de formatage de texte, 3-11

commande del, 6-5, 6-18

commande df, 5-4

commande diff, 6-11

commande dircmp, 5-13

commande dosdel, 6-20

commande dosdir, 6-20

commande dosread, 6-19

commande doswrite, 6-19

commande echo, 4-6, 4-7

commande env, 2-7

commande exit, 1-5

commande export, 10-11

commande factor, 12-4

commande fc, 11-11

commande fdformat, 8-5

commande find, 6-8

commande flcopy, 8-6

commande format, 8-5

commande fsck, 8-6

commande grep, 4-6, 6-10

commande groups, 9-4

commande head, 6-12

commande id, 1-7  
 commande kill, 3-21  
 commande ksh, 11-4, 11-10  
 commande learn, 12-3  
 commande leave, 12-4  
 commande li, 5-10  
 commande ln, 6-17  
 commande lock, 9-14  
 commande logname, 1-6  
 commande ls, 5-11  
 commande lscfg, 2-2  
 commande lscons, 2-3  
 commande lsdisk, 2-4  
 commande lsfont, 2-4  
 commande lsgroup, 9-6  
 commande lskbd, 2-4  
 commande lspp, 2-5  
 commande man, 3-7  
 commande mkdir, 5-8  
 commande more, 6-10  
 commande mv, 6-6  
 commande mkdir, 5-8  
 commande mwm, 10-6  
 commande nice, 3-16  
 commande nl, 6-14  
 commande pack, 8-9  
 commande page, 6-10  
 commande passwd, 1-9  
 commande paste, 6-13  
 commande pg, 6-9  
 commande pr, 7-13  
 commande printenv, 2-8  
 commande ps, 3-14  
 commande psh, 11-4, 11-10  
 commande pwd, 5-9  
 commande qcan, 7-7  
 commande qchk, 7-8  
 commande qhld, 7-11  
 commande qmov, 7-12  
 commande qpri, 7-10  
 commande qprt, 7-4  
 commande r (répétition), 3-9  
 commande renice, 3-17  
 commande restore, 8-14  
 commande rm, 6-6, 6-18  
 commande rmdir, 5-12  
 commande Rsh, 11-4, 11-72  
 commande rsh, 11-4  
 commande script, 4-7  
 commande sh, 11-4  
 commande shutdown, 1-5  
 commande smit, 3-7, 7-6, 8-13  
 commande sort, 6-10  
 commande stty, 2-6, 10-12  
 commande su, 1-4  
 commande tail, 6-12  
 commande tapechk, 8-9  
 commande tar, 8-16  
 commande tcopy, 8-9  
 commande tee, 4-6  
 commande touch, 1-4  
 commande tsh, 11-4  
 commande tty, 2-3  
 commande uname, 1-7  
 commande uncompress, 8-11  
 commande units, 12-5  
 commande unpack, 8-11  
   objet, 8-9  
 commande virscan, 9-15  
   ajout de signatures de virus, 9-16  
   format des fichiers signature, 9-15  
 commande wc, 6-12  
 commande whatis, 3-8  
 commande whereis, 3-7  
 commande who, 1-7  
 commande who am i, 1-6  
 commande whoami, 1-6  
 commande write, 12-7  
 commande xlock, 9-14  
 commandes  
   alias, création, 3-10  
   commandes longues sur plusieurs lignes,  
     entrée, 3-3  
   diagramme de syntaxe, 3-4  
   distinction majuscules/minuscules, 3-2  
   entrée, 3-2  
   espaces, 3-2  
   fonction, description, 3-8  
   formatage de texte, 3-11  
   généralités, 3-2  
   indicateurs, utilisation, 3-3  
   informations, affichage, 3-7  
   instructions d'usage, 3-6  
   intégrées, 11-37  
     shell Bourne, 11-76  
     shell C, 11-94  
   modification de l'historique, 3-10  
   nom abrégé, création, 3-10  
   nom d'une commande, définition, 3-3  
   paramètre, 3-3  
   plusieurs commandes sur une seule ligne,  
     entrée, 3-2  
   répétition, 3-9  
   sauvegarde de commande entrée, 3-8  
   shell Bourne, 11-73  
   shell C, 11-94  
   shell Korn ou POSIX, 11-32  
   substitution de chaînes, 3-9  
   syntaxe, 3-2  
 commandes du gestionnaire système basé sur le  
   web, 7-11  
 commandes intégrées, 11-37  
   shell Bourne, 11-76  
   shell C, 11-94  
 commandes spéciales, shell Bourne, 11-76  
 comparaison de fichiers, 6-11  
 compression de fichiers, 8-9  
 concaténation de fichiers texte, 4-3  
 connexion  
   distante, 1-1  
   messages, suppression, 1-4  
   multiples, 1-4  
   nom, affichage, 1-6  
   procédure, 1-3  
 console, affichage du nom, 2-3  
 contrôle des accès  
   affichage d'informations, 9-12  
   définition des informations, 9-12

- droits d'accès étendus, 9-10
- édition des informations, 9-13
- liste, 9-9, 9-10
- contrôle des travaux
  - shell C, 11-118
  - shell Korn ou POSIX, 11-51
- conventions d'appellation
  - fichier, 6-3
  - répertoires, 5-6
- conversion des unités de mesure, commande
  - units, 12-5
- copie
  - de/sur bande, 8-9
  - fichier sur bande ou disque, 8-7
  - sur/à partir de disquettes, 8-6
- copie de l'écran dans un fichier, 4-7
- correspondance de trame
  - shell Bourne, 11-89
  - shell Korn ou POSIX, 11-26
- coupe de sections, 6-13

## D

- déclaration de caractères
  - shell Bourne, 11-74
  - shell Korn ou POSIX, 11-12
- démarrage, contrôle des fenêtres et des applications, 10-6
- diagramme de syntaxe, pour une commande, 3-4
- didacticiel, 12-3
- disquette
  - copie, 8-6
  - formatage, 8-5
  - manipulation, 8-3
- distante
  - connexion, 1-1
  - imprimante, 7-3
- document here, 4-5, 11-28
- documents entrée en ligne, 4-5
- droit d'accès
  - de base, 9-9
  - fichier, 9-8
  - répertoire, 9-8
- droit d'accès de base, 9-9
- droits d'accès, étendus, 9-10
- droits d'accès étendus, 9-10

## E

- écran
  - affichage de texte écran par écran, 6-10
  - affichage en gros caractères, 4-8
  - copie dans un fichier, 4-7
  - copie de l'écran dans un fichier, 4-6
  - désignation de l'article de navigation par défaut, A-15
  - effacement, 4-6
- écrans, liste des écrans disponibles, 2-4
- éditeur, 6-5, 11-53
- éditeur ed, 6-5
- éditeur emacs, 11-53
- éditeur InEd, 6-5
- éditeur vi, 11-57
- édition, en ligne, shell Korn ou POSIX, 11-53
- édition en ligne, shell Korn ou POSIX, 11-53
  - Mode d'édition vi, 11-57

- mode emacs, 11-53
- édition en ligne (shell Korn)
  - mode d'édition vi, 11-57
  - mode emacs, 11-53
- effacement de l'écran, 4-6
- élimination d'une sortie, 4-4
- enseignement, didacticiel, 12-3
- entrée standard
  - copie dans un fichier, 4-6
  - définition, 4-2
  - réacheminement, 4-3
- environnement
  - affichage de l'environnement courant, 2-7
  - définition, utilisateur, 10-4
  - système, 2-1
- espace, affichage, 5-4
- états de sortie, shell Korn ou POSIX, 11-31
- évaluation arithmétique, shell Korn ou POSIX, 11-24
- expressions, recherche de fichiers, 6-8
- expressions régulières, 6-4

## F

- facteur premier, commande factor, 12-4
- fenêtre Window Manager, lancement, 10-6
- fichier
  - affichage
    - contenu, 6-9
    - dernières lignes, 6-12
    - premières lignes, 6-12
  - ajout d'une seule ligne de texte, 4-7
  - arborescence, 5-2
  - archivage, 8-16
  - ASCII, 6-2
  - binaire, 6-2
  - caractère joker, 6-3
  - changement de nom, 6-6
  - chemin d'accès, 5-6, 6-3
  - collage de texte, 6-13
  - colonne, suppression, 6-15
  - comparaison, 5-13, 6-11
  - compression, 8-9
  - concaténation, 4-3
  - conventions d'appellation, 6-3
  - copie, 6-7
    - de bande ou disque, 8-7
    - de DOS, 6-19
    - de l'écran, 4-7
    - vers DOS, 6-19
  - création avec un réacheminement à partir du clavier, 4-3
  - décompression, 8-11
  - décompte
    - lignes, 6-12
    - mots, 6-12
    - octets, 6-12
  - découpe de champs sélectionnés, 6-13
  - définition, 5-1
  - déplacement, 6-6
  - descripteur, 4-4
  - droit d'accès, 6-2, 9-4
  - environnement, 10-4
  - exécutable, 6-2
  - expressions correspondantes, recherche, 6-8

- expressions régulières, 6-4
- extraction, 8-16
- formatage
  - pour affichage, 6-9
  - pour impression, 7-13
- fusion de lignes, 6-13
- généralités, 6-1
- gestion, 6-5
- identification du type, 6-9
- liaison, 6-16, 6-17
- lié (suppression), 6-18
- localisation de section, 3-7
- métacaractère, 6-4
- mode d'accès, 9-4
  - définition, 6-16
- modification
  - à partir d'un fichier lié, 6-16
  - droit d'accès, 9-8
  - propriété, 9-4
- numérotation de lignes, 6-14
- propriété, 6-16, 9-4
- recherche d'une chaîne, 6-10
- regroupement, 4-3
- restauration, 8-14
  - utilisation de la commande smit, 8-15
- sauvegarde, 8-12
- sortie, à partir d'un point spécifié, 6-12
- suppression, 6-5
- suppression du DOS, 6-20
- tri du texte, 6-10
- types
  - affichage, 6-9
  - répertoire, 6-2
  - spécial, 6-2
  - standard, 6-2
- fichier .env, 10-5
- fichier .mwmrc, 10-8
- fichier .profile, 10-4, 10-5
- fichier .Xdefaults, 10-7
- fichier .xinitrc, 10-6
- fichier /etc/environment, 10-4
- fichier /etc/profile, 10-3
- fichier d'environnement, 10-4
- fichier de connexion
  - fichier .env, 10-5
  - fichier .profile, 10-4, 10-5
  - fichier /etc/environment, 10-4
  - fichier /etc/profile, 10-3
- fichier de ressources, modification, 10-7, 10-8
- fichier lié, suppression, 6-18
- fichier PostScript, conversion à partir d'ASCII, 7-16
- fichier profile, utilisation, 10-3
- fichier racine, 5-3
- fichiers ASCII, impression sur une imprimante PostScript, 7-14
- fichiers de lancement
  - shell C, 11-92
  - système, 10-3
- fichiers DOS
  - conversion, 6-19
  - copie, 6-19
  - liste du contenu, 6-20
  - suppression, 6-20
- fichiers texte
  - colonnes, suppression, 6-15
  - concaténation, 4-3
  - création à partir d'une entrée du clavier, 4-3
  - lignes, numérotation, 6-14
  - recherche de chaînes, 6-10
  - sections
    - collage, 6-13
    - découpe, 6-13
  - tri, 6-10
- file, commande, 6-9
- file d'attente
  - impression, 7-2
  - unité, 7-2
- filtres, définition, 4-5
- fonction auto-hold, interface Window, 10-15
- fonction coprocess, shell Korn ou POSIX, 11-29
- formatage de disquettes, 8-5
- formatage de texte
  - caractères étendus mono-octets, 3-11
  - prise en charge des caractères internationaux, 3-11
  - prise en charge des caractères multi-octets, 3-12

## G

- gestion du temps
  - création de rappels, 12-4
  - rédaction de messages de l'agenda, 12-2
- graphique, préférence d'affichage, 10-15

## H

- help
  - affichage d'informations, 12-3
  - commande, 12-3
- historique
  - événement, modification de taille, 10-15
  - liste, définition de la taille, A-16
  - modification, 3-10
  - substitution, shell C, 11-105
- historique des commandes, shell Korn ou POSIX, 11-11
- history, command, 3-8

## I

- ID, utilisateur, 9-4
- ID de connexion, 9-3
- impression, 7-1
  - blocage de travaux d'impression, 7-11
  - déplacement de travaux d'impression., commande du gestionnaire système basé sur le web, 7-12
  - formatage de fichiers, 7-13
  - impression de fichiers ASCII sur une imprimante PostScript, 7-14
  - libération de travaux d'impression, 7-11
- impression à partir d'InfoExplorer, A-10
- imprimante, 7-1
  - affichage d'état d'un travail, 7-10
  - annulation d'un travail, 7-7
  - distante, 7-3
  - état, 7-9
  - file d'attente, 7-2
  - lancement d'un travail, 7-4

- locale, 7-3
- programme expéditeur, 7-3
- qdaemon, 7-2
- réelle, 7-3
- spouleur, 7-2
- unité de file d'attente, 7-2
- valeur par défaut
  - interface ASCII, A-16
  - interface Window, 10-14
- virtuelle, 7-3
- imprimante locale, 7-3
- imprimante PostScript, impression de fichiers
  - ASCII, 7-14
- Imprimante réelle, 7-3
- imprimante virtuelle, 7-3
- indicateur de commande —, 3-3
- indicateurs
  - , 3-3
  - dans une commande, 3-3
- instructions d'usage, pour une commande, 3-6
- interface ASCII
  - généralités, A-1
  - personnalisation, généralités, A-15
  - valeur par défaut, généralités, A-15
- interface ASCII d'InfoExplorer
  - aide, A-2, A-11
  - arrêt, A-14
  - bibliothèque secondaire, lancement, A-14
  - déplacement dans le texte, A-5
  - écran
    - déplacement, A-4
    - types, A-3
  - impression, A-10
  - liens hypertexte, sélection, A-5
  - options de menu, sélection, A-6
  - recherche, A-7
  - sortie, A-14
- interface Window
  - personnalisation, 10-13
  - ressources X disponibles, A-18
- invite, changement, 10-12

**L**

- lancement
  - fenêtre Window Manager, 10-6
  - shell Bourne, 11-70
  - shell C, 11-92
  - shell Korn ou POSIX, 11-10
  - X, 10-6
- langages, particulières, 2-8
- langues particulières, 2-8
- lecture de l'affichage à trois chiffres, 1-3
- liaisons
  - création, 6-17
  - fixes, 6-16
  - suppression, 6-18
  - symbolique, 6-16
  - types, 6-16
- liens, généralités, 6-16
- ligne de texte, ajout à un fichier, 4-7
- lignes, décompte, 6-12
- liste de commandes
  - &gt;, 4-2

- &gt;&gt;, 4-3
- &lt;&lt;&lt;&lt;, 4-3
- |, 4-5
- acldedit, 9-13
- aclget, 9-12
- aclput, 9-12
- aixterm, 2-8
- alias, 3-10
- apropos, 12-1
- at, 3-19, 3-20
- atq, 3-20
- backup, 8-12
- banner, 4-8
- bsh, 11-4, 11-70
- cal, 12-2
- calendar, 12-2
- capture, 4-7
- cat, 4-3, 6-10
- cd, 5-6, 5-9
- chfont, 10-11
- chmod, 9-8
- chown, 9-4
- chpq, 7-16
- clear, 4-6
- colrm, 6-15
- commande cpio -o, 8-7
- compress, 8-9
- connexion, 1-4
- cp, 5-9, 6-7
- cpio -i, 8-7
- csh, 11-4, 11-92
- cut, 6-13
- del, 6-5, 6-18
- df, 5-4
- diff, 6-11
- dircmp, 5-13
- dosdel, 6-20
- dosdir, 6-20
- dosread, 6-19
- doswrite, 6-19
- echo, 4-6, 4-7
- env, 2-7
- exit, 1-5
- export, 10-11
- factor, 12-4
- fc, 11-11
- fdformat, 8-5
- file, 6-9
- find, 6-8
- flcopy, 8-6
- format, 8-5
- fsck, 8-6
- grep, 4-6, 6-10
- groupe, 9-4
- head, 6-12
- help, 12-3
- history, 3-8
- id, 1-7
- kill, 3-21
- ksh, 11-4, 11-10
  - commandes intégrées spéciales, 11-37, 11-38, 11-39, 11-41, 11-42, 11-43

- commandes intégrées standard, 11-44, 11-45, 11-46, 11-47, 11-48
- learn, 12-3
- leave, 12-4
- li, 5-10
- ln, 6-17
- lock, 9-14
- logname, 1-6
- logout, 1-5
- ls, 5-11
- lscfg, 2-2
- lscons, 2-3
- lsdisp, 2-4
- lsfont, 2-4
- lsgroup, 9-6
- lskbd, 2-4
- lslpp, 2-5
- man, 3-7
- mkdir, 5-8
- more, 6-10
- mv, 6-6
- mkdir, 5-8
- mwm, 10-6
- nice, 3-16
- nl, 6-14
- pack, 8-9
- page, 6-10
- passwd, 1-9
- paste, 6-13
- pg, 6-9
- pr, 7-13
- printenv, 2-8
- ps, 3-14
- psh, 11-4, 11-10
- pwd, 5-9
- qcan, commande du gestionnaire système basé sur le web, 7-7
- qchk, commande du gestionnaire système basé sur le web, 7-8
- qhld, 7-11
- qmov, 7-12
- qpri, 7-10
- qprt, 7-4
- r, 3-9
- renice, 3-17
- restore, 8-14
- rm, 6-6, 6-18
- rmdir, 5-12
- Rsh, 11-4, 11-72
- rsh, 11-4
- script, 4-7
- sh, 11-4
- shutdown, 1-5
- smit, 3-7, 7-6, 8-13
- sort, 6-10
- stty, 2-6, 10-12
- su, 1-4
- tail, 6-12
- tapechk, 8-9
- tar, 8-16
- tcopy, 8-9
- tee, 4-6
- touch, 1-4
- tsh, 11-4

- tty, 2-3
- uname, 1-7
- uncompress, 8-11
- units, 12-5
- unpack, 8-11
- virscan, 9-15
- wc, 6-12
- whatis, 3-8
- whereis, 3-7
- who, 1-7
- who am i, 1-6
- whoami, 1-6
- write, 12-7
- xlock, 9-14
- logiciels, affichage d'informations, 2-5
- login
  - commande, 1-4
  - ID utilisateur (autre), 1-4
- logout
  - commande, 1-5
  - procédure, 1-5

## M

- mappes, clavier, 2-4
- mappes clavier, liste des mappes clavier disponibles, 2-4
- message envoyé en temps réel, envoi à d'autres utilisateurs, 12-7
- messages
  - affichage à l'écran, 4-6
  - envoi à d'autres utilisateurs, 12-7
  - envoi vers la sortie standard, 4-6
- messages de l'agenda, commande calendar, 12-2
- métacaractère, 6-4
- mode d'accès
  - classe d'utilisateur, 9-4
  - contrôle, 9-4
  - droit d'accès de base, 9-9
  - fichier, 9-4
  - informations sur le groupe, affichage, 9-6
  - répertoires, 9-4
  - représentation
    - numérique, 9-6
    - symbolique, 9-5
  - valeur par défaut
    - représentation numérique, 9-6
    - représentation symbolique, 9-5
- mots, décompte, 6-12
- mots de passe
  - consignes, 1-8
  - modification ou définition, 1-9
  - Null, 1-10
- mots réservés, shell Korn ou POSIX, 11-14

## N

- network, affichage du nom, avec la commande uname, 1-7
- NFS, 5-2
- nom abrégé pour des commandes, création, 3-10
- nom des ressources, interface Window, A-18
- nom, affichage
  - connexion, 1-6
  - système d'exploitation, 1-7

- nombre entier, 11-24
- notes
  - définition de l'éditeur préféré, interface ASCII, A-17
  - impression, InfoExplorer, A-10
  - modèle par défaut, interface ASCII, A-16
- numéro d'i-node, 5-5, 6-16
- numéro d'identification, 3-13
- numéro d'i-node, 6-2
- numéro de référence index node, 5-5

## O

- octets, décompte, 6-12
- opérateur de réacheminement avec ajout (&gt;&gt;), 4-3
- opérateur de réacheminement des entrées (&lt;&lt;&lt;&lt;), 4-3
- opérateur de réacheminement des sorties (&gt;), 4-2
- option, dans une commande, 3-3
- options, dans une commande, 3-3

## P

- pages man, recherche par mot-clé, 12-1
- paramètre
  - dans une commande, 3-3
  - shell Korn ou POSIX, 11-17
- paramètres positionnels, shell Bourne, 11-89
- personnalisation
  - environnement système, 10-11
  - InfoExplorer
    - interface ASCII, A-15
    - interface Window, 10-13
- PID, description, 3-13
- pipes, définition, 4-5
- police, modification, 10-11
  - interface Window, 10-14
- polices, liste des polices disponibles, 2-4
- préférences, généralités, interface Window, 10-14
- prise en charge des caractères internationaux, formatage de texte, 3-11
- prise en charge des caractères multi-octets, formatage de texte, 3-12
- process
  - affichage de l'état, 3-15
  - affichage des process actifs, 3-14
  - annulation, 3-17
    - process d'avant-plan, 3-17
  - arrêt, 3-18
    - process d'arrière-plan, 3-21
  - arrière-plan, 3-13
  - avant-plan, 3-13
  - définition de la priorité initiale, 3-16
  - démon, 3-13
  - déprogrammation, 3-20
  - description, 3-13
  - lancement, 3-14
  - liste des process planifiés, 3-20
  - modification de priorité, 3-17
  - planification pour un fonctionnement en différé, 3-18
  - relance d'un process arrêté, 3-18
  - zombie, 3-14
- process d'arrière-plan, définition, 3-13

- process d'avant-plan, définition, 3-13
- process démon, description, 3-13
- process zombie, 3-14
- programme, copie de la sortie dans un fichier, 4-6
- programme expéditeur, imprimante, 7-3
- programme InfoExplorer
  - arrêt, A-14
  - généralités, interface ASCII, A-1
  - lancement, A-3
  - sortie, A-14
- programme InfoExplorer (ASCII)
  - article de navigation,, désignation d'une valeur par défaut, A-15
  - bases de données à explorer, modification de valeur par défaut, A-16
  - éditeur de notes, choix, A-17
  - imprimante, désignation par défaut, A-16
  - personnalisation, A-15
  - préférences, modification des paramètres, A-16
  - type de sortie imprimée, modification, A-17
  - valeur par défaut, modification, A-15
- programme InfoExplorer (Windows), personnalisation, 10-13

## Q

- qdaemon, 7-2

## R

- rappels avec leave, commande leave, 12-4
- réacheminement
  - entrée standard, 4-3
  - sortie (erreur) standard, 4-4
  - sortie standard, 4-2
  - sortie vers un fichier, 4-2
- réacheminement des entrées, 4-2
- réacheminement des entrées/sorties, 11-91
  - à partir du coprocess, 11-30
  - shell Bourne, 11-91
  - shell C, 11-116
  - shell Korn ou POSIX, 11-28
- recherche
  - base de données par défaut, interface ASCII, A-16
  - mot-clé, 12-1
- recherche par mot-clé, commande apropos, 12-1
- références, impression à partir d'écrans InfoExplorer, A-10
- Répertoire personnel, 5-7
- répertoires, 5-6
  - abréviations, 5-7
  - affichage
    - contenu, 5-10
    - courant, 5-9
  - changement de nom, 5-8
  - changement de propriété, 9-4
  - chemin d'accès, 5-6
  - comparaison du contenu, 5-13
  - conventions d'appellation, 5-6
  - copie, 5-9
  - création, 5-8
  - de travail, 5-6
  - définition, 5-1
  - déplacement, 5-8

- généralités, 5-5
- liaison, 6-16
- liste de fichiers, 5-10
- liste des fichiers DOS, 6-20
- mode d'accès, 9-4
- modification, 5-9
- modification des droits d'accès, 9-8
- organisation, 5-6
- parent, 5-6
- personnel, 5-6
- racine, définition, 5-1
- sous-répertoires, 5-6
- spécification par abréviation, 5-7
- structure, 5-6
- suppression, 5-12
- types, 5-5
- ressource, description, 10-7
- ressources X, interface Window, A-18

## S

- sauvegarde
  - bande, avantages, 8-4
  - consignes, 8-2
  - objet, 8-1
- sécurité
  - fichier, 9-1
  - menaces, 9-2
  - système, 9-1
- shell
  - programme, 11-7
  - script
    - création, 11-7
    - spécification (shell), 11-8
  - variables, exportation, 10-11
- Shell Bourne, lancement, 11-70
- shell Bourne
  - commandes
    - intégrées, 11-76
    - liste, 11-75
    - utilisation, 11-73
  - commandes spéciales, 11-76
  - correspondance de trame, 11-89
  - déclaration de caractères, 11-74
  - environnement, 11-70
  - mots réservés, 11-75
  - paramètres positionnels, 11-89
  - réacheminement des entrées/sorties, 11-91
  - substitution conditionnelle, 11-88
  - substitution de commande, 11-82
  - substitution de noms de fichiers, 11-89
  - traitement des signaux, 11-74
  - variables, 11-84
    - définies par l'utilisateur, 11-83
    - prédéfinies, 11-87
    - substitution, 11-83
- shell C
  - commandes
    - intégrées, 11-94
    - utilisation, 11-94
  - contrôle des travaux, 11-118
  - expressions, 11-102
  - lancement, 11-92
  - opérateur, 11-102

- réacheminement des entrées/sorties, 11-116
- règles d'utilisation, 11-93
- substitution d'alias, 11-108
- substitution d'historique, 11-105
- substitution de commande, 11-103
- substitution de noms de fichiers, 11-111
- substitution de variables, 11-109
- traitement des signaux, 11-93
- variables prédéfinies et d'environnement, 11-113
- shell Korn ou POSIX
  - alias de commandes, 11-15
    - substitution de tilde, 11-16
  - commandes
    - composées, 11-33
    - fonctions, 11-35
    - intégrées, 11-37
    - utilisation, 11-32
  - commandes intégrées, 11-37
  - contrôle des travaux, 11-51
  - coprocess, réacheminement des entrées/sorties, 11-30
  - correspondance de trame, 11-26
  - déclaration de caractères, 11-12
  - édition, 11-53
  - environnement, 11-10
  - états de sortie, 11-31
  - évaluation arithmétique, 11-24
  - expressions conditionnelles, 11-49
  - fonction coprocess, 11-29
  - historique des commandes, 11-11
  - lancement, 11-10
  - mots réservés, 11-14
  - réacheminement des entrées/sorties, 11-28
  - séparation de zones, 11-25
  - substitution de commandes, 11-23
  - substitution de noms de fichiers, 11-26
  - substitution de paramètres, 11-17
  - suppression des caractères de déclaration, 11-27
  - traitement des signaux, 11-51
  - variables
    - définies par l'utilisateur, 11-20
    - prédéfinies, 11-20
- shell POSIX, 11-9
- shell restreint, lancement, 11-72
- shell standard, expressions conditionnelles, 11-49
- shells
  - Bourne
    - commandes intégrées, 11-76
    - environnement, 11-70
    - lancement, 11-70
    - paramètres positionnels, 11-89
    - réacheminement des entrées/sorties, 11-91
    - substitution conditionnelle, 11-88
    - substitution de commande, 11-82
    - substitution de noms de fichiers, 11-89
    - substitution de variables, 11-83
    - variables, 11-84
    - variables définies par l'utilisateur, 11-83
    - variables prédéfinies, 11-87

## C

- commandes intégrées, 11-94
- contrôle des travaux, 11-118
- lancement, 11-92
- Réacheminement des entrées/sorties, 11-116
- substitution d'alias, 11-108
- substitution d'historique, 11-105
- substitution de commande, 11-103
- substitution de noms de fichiers, 11-111
- substitution de variables, 11-109
- traitement des signaux, 11-93
- variables prédéfinies et d'environnement, 11-113
- caractéristiques, 11-3
- disponibles, 11-4
- explication, 11-1
- Korn ou POSIX
  - commande, 11-11, 11-15, 11-23
  - commandes composées, 11-33
  - commandes intégrées, 11-37
  - contrôle des travaux, 11-51
  - déclaration de caractères, 11-12
  - édition en ligne, 11-53, 11-57
  - environnement, 11-10
  - états de sortie, 11-31
  - évaluation arithmétique, 11-24
  - expressions conditionnelles, 11-49
  - fonction coprocess, 11-29
  - lancement, 11-10
  - mots réservés, 11-14
  - paramètre, 11-17
  - réacheminement des entrées/sorties, 11-28
  - substitution de noms de fichiers, 11-26
  - traitement des signaux, 11-51
  - utilisation de commandes, 11-32
- restreint, lancement, 11-72
- script, spécification (shell), 11-8
- script shell, création, 11-7
- sécurisé (lancement), 11-4
- terminologie, définition, 11-5
- types, 11-4
- SMIT, impression, contrôle, 7-4
- sortie
  - élimination des sorties via le fichier /dev/null, 4-4
  - réacheminement vers un fichier, 4-2
- sortie (erreur) standard, réacheminement, 4-4
- sortie standard
  - ajout à un fichier, 4-3
  - définition, 4-2
  - réacheminement, 4-2
- spouleur d'impression, 7-2
- substitution conditionnelle, shell Bourne, 11-88
- substitution d'alias, shell C, 11-108
- substitution de commande
  - shell Bourne, 11-82
  - shell C, 11-103
- substitution de commandes, shell Korn ou POSIX, 11-23
- substitution de noms de fichiers
  - shell Bourne, 11-89
  - shell C, 11-111
  - shell Korn ou POSIX, 11-26

- substitution de tilde, alias de commandes, shell Korn ou POSIX, 11-16
- substitution de variables
  - shell Bourne, 11-83
  - shell C, 11-109
  - shell Korn ou POSIX, 11-20
- suppression
  - fichier, 6-5
  - répertoires, 5-12
- suppression des caractères de déclaration, shell Korn ou POSIX, 11-27
- suppression des messages de connexion, 1-4
- système
  - affichage du nom, 1-7
  - environnement, 2-1
  - fichiers de lancement, 10-3
  - gestion, tâches du système de fichiers, 5-2
  - invite, modification, 10-12
  - mise sous tension, 1-3
  - personnalisation de l'environnement, 10-11
  - sécurité, 9-1
  - shutdown, 1-5
  - variables par défaut, 10-3
- système d'exploitation
  - affichage du nom, avec la commande uname, 1-7
  - connexion, 1-3
  - déconnexion, 1-5
  - système de fichiers CD-ROM, 5-2
  - système de fichiers journalisé, 5-2
  - système de fichiers réseau (NFS), 5-2
  - systèmes de fichiers
    - définition, 5-1
    - espace disponible, affichage, 5-4
    - exemple, illustration, 6-3
    - généralités, 5-2
    - racine, 5-3
    - réparation interactive, 8-6
    - structure, 5-2
    - types
      - système de fichiers journalisé, 5-2
      - système de fichiers réseau (NFS), 5-2
    - vérification d'intégrité, 8-6

## T

- terminal
  - affichage de paramètres, 2-8
  - affichage du nom, 2-3
  - verrouillage, 9-14
    - utilisation de la commande lock, 9-14
- texte
  - affichage en gros caractères, 4-8
  - ajout à un fichier, 4-7
- touches de contrôle
  - affichage de paramètres, 2-6
  - modification, 10-12
- traitement des signaux
  - shell Bourne, 11-74
  - shell C, 11-93
  - shell Korn ou POSIX, 11-51
- travail
  - déprogrammation, 3-20
  - liste des process planifiés, 3-20

- planification, 3-18
- travail d'impression
  - affichage de l'état, 7-8
  - annulation, 7-7
  - blocage, 7-11
  - définition, 7-2
  - définition d'une priorité, 7-10
  - déplacement, 7-12
  - formatage de fichiers, 7-13
  - lancement, 7-4
  - libération, 7-11
- type de fichier d'impression, annulation de la détermination automatique, 7-16
- types, système de fichiers CD-ROM, 5-2

## U

- unité /dev/rfd0, 8-3
- unité /dev/rmt0, unité de bande, utilisation, 8-3
- unités, affichage d'informations, 2-2
- unités de mesure, conversion, 12-5
- utilisateur
  - affichage d'un ID système, 1-7
  - affichage du système courant, 1-7
  - classes, 9-4
  - envoi de messages à d'autres utilisateurs, 12-7
  - groupe
    - affichage d'informations, 9-6
    - définition, 9-4
  - ID, passage à un autre, 1-4

## V

- valeur par défaut
  - article de navigation InfoExplorer
    - interface ASCII, A-15
    - interface Window, 10-13

- base de données InfoExplorer
  - interface ASCII, A-16
  - interface Window, 10-14
- fichier bookmark InfoExplorer
  - interface ASCII, A-15
  - interface Window, 10-13
- fichier history InfoExplorer
  - interface ASCII, A-15
  - interface Window, 10-13
- fichier notes InfoExplorer
  - interface ASCII, A-15
  - interface Window, 10-13
- imprimante InfoExplorer
  - interface ASCII, A-16
  - interface Window, 10-14
- modèle de notes InfoExplorer
  - interface ASCII, A-16
  - interface Window, 10-14, A-16
- variables
  - exportation de shell, 10-11
  - shell Bourne, 11-84
    - définies par l'utilisateur, 11-83
    - prédéfinies, 11-87
  - shell C, variables prédéfinies et d'environnement, 11-113
  - shell Korn ou POSIX
    - définies par l'utilisateur, 11-20
    - prédéfinies, 11-20
- variables d'environnement, affichage de valeurs, 2-8
- verrouillage de votre terminal, 9-14
- virus, test de fichiers, 9-15

## X

- X Window System, lancement, 10-6

## Vos remarques sur ce document / Technical publication remark form

**Titre / Title :** Bull AIX 4.3 - Guide de l'utilisateur Système d'exploitation et unités

**N° Référence / Reference N° :** 86 F2 97HX 02

**Daté / Dated :** Novembre 1999

### ERREURS DETECTEES / ERRORS IN PUBLICATION

### AMELIORATIONS SUGGEREES / SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Vos remarques et suggestions seront examinées attentivement.

Si vous désirez une réponse écrite, veuillez indiquer ci-après votre adresse postale complète.

Your comments will be promptly investigated by qualified technical personnel and action will be taken as required.

If you require a written reply, please furnish your complete mailing address below.

NOM / NAME : \_\_\_\_\_ Date : \_\_\_\_\_

SOCIETE / COMPANY : \_\_\_\_\_

ADRESSE / ADDRESS : \_\_\_\_\_

Remettez cet imprimé à un responsable BULL ou envoyez-le directement à :

Please give this technical publication remark form to your BULL representative or mail to:

**BULL ELECTRONICS ANGERS  
CEDOC  
34 Rue du Nid de Pie – BP 428  
49004 ANGERS CEDEX 01  
FRANCE**

# Technical Publications Ordering Form

## Bon de Commande de Documents Techniques

To order additional publications, please fill up a copy of this form and send it via mail to:  
 Pour commander des documents techniques, remplissez une copie de ce formulaire et envoyez-la à :

**BULL ELECTRONICS ANGERS**  
**CEDOC**  
**ATTN / MME DUMOULIN**  
**34 Rue du Nid de Pie – BP 428**  
**49004 ANGERS CEDEX 01**  
**FRANCE**

**Managers / Gestionnaires :**  
**Mrs. / Mme :** C. DUMOULIN +33 (0) 2 41 73 76 65  
**Mr. / M :** L. CHERUBIN +33 (0) 2 41 73 63 96  
**FAX :** +33 (0) 2 41 73 60 19  
**E-Mail / Courrier Electronique :** srv.Cedoc@franp.bull.fr

Or visit our web site at: / Ou visitez notre site web à:  
<http://www-frec.bull.com> (PUBLICATIONS, Technical Literature, Ordering Form)

CEDOC Reference # N° Référence CEDOC	Qty Qté	CEDOC Reference # N° Référence CEDOC	Qty Qté	CEDOC Reference # N° Référence CEDOC	Qty Qté
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	
__ __ __ __ __ [__]		__ __ __ __ __ [__]		__ __ __ __ __ [__]	

[\_\_] : no revision number means latest revision / pas de numéro de révision signifie révision la plus récente

NOM / NAME : \_\_\_\_\_ Date : \_\_\_\_\_

SOCIETE / COMPANY : \_\_\_\_\_

ADRESSE / ADDRESS : \_\_\_\_\_

PHONE / TELEPHONE : \_\_\_\_\_ FAX : \_\_\_\_\_

E-MAIL : \_\_\_\_\_

**For Bull Subsidiaries / Pour les Filiales Bull :**

Identification: \_\_\_\_\_

**For Bull Affiliated Customers / Pour les Clients Affiliés Bull :**

**Customer Code / Code Client :** \_\_\_\_\_

**For Bull Internal Customers / Pour les Clients Internes Bull :**

**Budgetary Section / Section Budgétaire :** \_\_\_\_\_

**For Others / Pour les Autres :**

**Please ask your Bull representative. / Merci de demander à votre contact Bull.**



**BULL ELECTRONICS ANGERS  
CEDOC  
34 Rue du Nid de Pie – BP 428  
49004 ANGERS CEDEX 01  
FRANCE**

**86 F2 97HX 02**

PLACE BAR CODE IN LOWER  
LEFT CORNER



Utiliser les marques de découpe pour obtenir les étiquettes.  
Use the cut marks to get the labels.

AIX  
AIX 4.3 - Guide  
de l'utilisateur  
Système  
d'exploitation  
et unités  
86 F2 97HX 02

AIX  
AIX 4.3 - Guide  
de l'utilisateur  
Système  
d'exploitation  
et unités  
86 F2 97HX 02

AIX  
AIX 4.3 - Guide  
de l'utilisateur  
Système  
d'exploitation  
et unités  
86 F2 97HX 02

