

Manuel de l'Utilisateur Expérimenté Solaris

SunSoft
A Sun Microsystems, Inc. Business
Agence Française
13, avenue Morane Saulnier
B.P. 53
78142 Vélizy Cedex

Référence : 801-7481-10
Révision A, Août 1994



© 1994 Sun Microsystems, Inc.

2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

Distribué en France par SunSoft, A Sun Microsystems Inc. Business

Agence française, 13, avenue Morane Saulnier, B.P.53, 78142 Vélizy Cédex, France

Tous droits réservés. Ce produit ainsi que la documentation associée sont protégés par copyright et distribués sous des licences limitant leur utilisation, copie, distribution et décompilation. Aucune partie de ce produit ou de la documentation associée ne peut être reproduite sous quelque forme que ce soit et par quelque moyen que ce soit sans la permission écrite préalable de Sun et des titulaires de licences, le cas échéant.

Certaines parties de ce produit peuvent être dérivées des systèmes UNIX® et BSD Berkeley 4.3, distribués sous licence respectivement par UNIX System Laboratories, Inc., filiale à part entière de Novell, Inc., et l'Université de Californie. Des logiciels de polices de caractères de sociétés indépendantes faisant partie de ce produit ont un copyright et sont distribués sous licence par des fournisseurs de Sun.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

Le produit décrit dans ce manuel est protégé par un ou plusieurs brevets américains et étrangers ou par d'autres brevets en cours d'homologation.

MARQUES DEPOSEES

Sun, Sun Microsystems, le logo Sun, SMCC, SunSoft, le logo SunSoft, Solaris, SunOS, OpenWindows, DeskSet, ONC, ONC+ et NFS sont des marques de fabrique ou des marques déposées par Sun Microsystems, Inc. aux Etats-Unis et dans certains autres pays. UNIX et OPEN LOOK sont des marques déposées d'UNIX System Laboratories, Inc., filiale à part entière de Novell, Inc. [PostScript et Display PostScript sont des marques déposées d'Adobe Systems, Inc.](#) Tous les autres noms de produits mentionnés dans ce manuel sont des marques déposées de leurs propriétaires respectifs.

Toutes les marques SPARC, y compris le logo «SCD Compliant», sont des marques de fabrique ou des marques déposées de SPARC International, Inc. [SPARCstation](#), [SPARCserver](#), [SPARCengine](#), [SPARCstorage](#), [SPARCware](#), [SPARCcenter](#), [SPARCclassic](#), [SPARCcluster](#), [SPARCdesign](#), [SPARC811](#), [SPARCprinter](#), [UltraSPARC](#), [microSPARC](#), [SPARCworks](#) et [SPARCcompiler](#) sont des marques dont la licence exclusive a été octroyée à Sun Microsystems, Inc. Les produits portant la marque SPARC reposent sur une architecture développée par Sun Microsystems, Inc.

Les interfaces OPEN LOOK® et Sun™ Graphical User Interface (GUI) ont été développées par Sun Microsystems, Inc. pour ses utilisateurs et titulaires de licences. Sun reconnaît les efforts de pointe réalisés par Xerox dans le domaine de la recherche et du développement du concept des interfaces utilisateurs visuelles ou graphiques. Sun détient une licence non exclusive de Xerox pour l'Interface Utilisateur Graphique Xerox qui couvre également les titulaires de licences Sun utilisant OPEN LOOK GUI ou étant en conformité avec les agréments de licences écrits Sun.

X Window System est un produit du Massachusetts Institute of Technology.

SUN MICROSYSTEMS, INC. FOURNIT CETTE PUBLICATION "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, EXPLICITE OU IMPLICITE, Y COMPRIS, MAIS SANS SE LIMITER AUX GARANTIES IMPLICITES DE COMMERCIALISABILITE OU DE CONFORMITE A UN USAGE SPECIFIQUE OU DE NON-TRANSGRESSION.

CETTE PUBLICATION PEUT COMPORTER DES IMPRECISIONS TECHNIQUES ET DES ERREURS TYPOGRAPHIQUES. DES MODIFICATIONS SONT APPORTEES PERIODIQUEMENT AUX INFORMATIONS CI-APRES ; CES MODIFICATIONS SERONT INCORPOREES DANS LES EDITIONS ULTERIEURES DE CETTE PUBLICATION. SUN MICROSYSTEMS, INC. SE RESERVE A TOUT MOMENT LE DROIT DE MODIFIER ET/OU D'AMELIORER LE(S) PRODUIT(S) ET/OU PROGRAMME(S) DECRITS DANS CETTE PUBLICATION.



Papier Localisé en Irlande
recyclable

Table des matières

A propos de ce guide	xiii
Conventions spécifiques aux plates-formes.....	xvi
1. Connexion à SunOS	
et lancement d'OpenWindows	1
Connexion.....	1
Le Shell de connexion	2
Déconnexion.....	3
Equivalents clavier.....	4
Remarques préliminaires sur le lancement d'OpenWindows	5
La variable d'environnement OPENWINHOME.....	6
Utilisation du fichier de démarrage approprié	7
Lancement de l'environnement utilisateur OpenWindows	9
Lancement du logiciel OpenWindows	9
Si OpenWindows ne démarre pas.....	11
Sortie de l'environnement OpenWindows.....	11
Options de lancement spéciales d'OpenWindows	12

Lancement d'OpenWindows avec un niveau de sécurité réseau réduit	13
Lancement d'OpenWindows avec différents types de moniteur	13
SPARC Lancement avec différents types de moniteurs	14
2. Commandes SunOS de base	17
Le prompt de commande	17
Saisie des commandes.	18
Correction des fautes de frappe.	18
Saisie de plusieurs commandes et de commandes longues	19
Répétition des commandes antérieures	20
Ajout d'options de commande	21
Redirection et transmission par pipe du résultat d'une commande	21
Exécution de commandes en arrière-plan	23
Aide en ligne sur les commandes OS	24
Affichage des pages manuel avec <code>man</code>	24
Affichage d'un résumé d'une ligne avec <code>whatis</code>	24
Recherche par mots-clés avec <code>apropos</code>	25
3. Gestion des fichiers et des répertoires.	27
Concepts de gestion des fichiers	27
Utilisation des commandes de fichiers	28
Avant de commencer.	28
Création d'un fichier test	29
Liste de fichiers (<code>ls</code>)	29

Copie de fichiers (<code>cp</code>)	30
Déplacement de fichiers et changement de nom (<code>mv</code>)..	30
Suppression de fichiers (<code>rm</code>)	31
Affichage du contenu d'un fichier (<code>more, cat</code>).....	32
Affichage du format d'un fichier (<code>file</code>)	32
Répertoires et arborescence des répertoires.....	33
Arborescence d'un répertoire.....	33
Impression du répertoire de travail (<code>pwd</code>)	34
Votre répertoire utilisateur.....	35
Changement de répertoire de travail (<code>cd</code>).....	35
Création d'un répertoire (<code>mkdir</code>).....	37
Chemins d'accès relatifs	37
Déplacement et changement de nom des répertoires..	38
Copie de répertoires	38
Suppression de répertoires (<code>rmdir</code>)	39
Recherche de différences entre fichiers (<code>diff</code>)	39
Comparaison de trois fichiers différents (<code>diff3</code>).....	42
Utilisation de <code>bdiff</code> sur des fichiers volumineux	42
Recherche de fichiers (<code>find</code>)	42
Droits d'accès aux fichiers et aux répertoires.....	45
Affichage des droits d'accès et des statuts (<code>ls -l</code>)	46
Etablissement d'une liste de fichiers "cachés" (<code>ls -a</code>)	47
Modification des droits d'accès (<code>chmod</code>)	48
Définition de droits d'accès absolus	51

4. Recherche dans des fichiers.....	55
Recherche d'un modèle à l'aide de la commande <code>grep</code> ...	55
Utilisation de la commande <code>grep</code> en tant que filtre.....	56
Utilisation de la commande <code>grep</code> pour des chaînes de plusieurs mots	58
Recherche de lignes ne comportant pas de chaîne donnée	58
Autres possibilités de la commande <code>grep</code>	59
Recherche de métacaractères	60
Guillemets simples ou doubles dans les lignes de commande	62
5. Mots de passe, processus et mémoire disque	63
Utilisation d'un mot de passe	63
Modification du mot de passe	64
Expiration du mot de passe	65
Processus et PID.....	66
Commandes en cours d'exécution (<code>ps</code>).....	66
Interruption d'un processus (<code>kill</code>)	67
Gestion de la mémoire disque	68
Affichage de l'utilisation du disque (<code>df -k</code>)	68
Affichage de l'utilisation d'un répertoire (<code>du</code>).....	68
6. Utilisation de l'éditeur <code>vi</code>	71
Lancement de <code>vi</code>	72
Création d'un fichier	72

La ligne d'état	73
Les deux modes de l'éditeur vi	73
Mode insertion	74
Mode commande	74
Fermeture d'une session	75
Sauvegarde des modifications et sortie de vi	76
Impression d'un fichier	77
Les commandes de base vi	77
Déplacement dans un fichier	78
Insertion de texte	81
Remplacement de texte	82
Annulation de modifications	83
Suppression de texte	84
Copie et déplacement de texte : commandes Yank, Delete et Put	85
Utilisation d'un comptage pour la répétition des commandes	87
Utilisation des commandes ex.	88
Affichage et suppression des numéros de lignes	88
Copie de lignes	89
Déplacement de lignes	90
Suppression de lignes	90
Recherche et remplacement avec vi	90
Recherche d'une chaîne de caractères	91
Affinement de la recherche	91

Remplacement d'une chaîne de caractères.	93
Passage à une ligne donnée	94
Insertion d'un fichier dans un autre	94
Edition de plusieurs fichiers.	95
Edition d'une série de fichiers	95
Copie de lignes d'un fichier à l'autre	96
Sélection des paramètres vi.....	97
Reprise après une panne.....	98
Récapitulatif des principales commandes vi	98
7. Utilisation de la messagerie	103
mailx : Fonctions de base	104
Lancement de mailx	104
Envoi d'une lettre type	105
Lecture de la lettre type	106
Sortie de mailx.....	107
Lecture de lettres	108
Suppression des lettres (et annulation de suppression) ...	110
Impression de lettres.....	111
Envoi de lettres	112
Lettres non transmises	114
Annulation d'une lettre non envoyée	114
Ajout de copies carbone et de copies carbone aveugles	114
Insertion d'une copie de lettre ou de fichier	115
Réponse à une lettre	116

Sauvegarde et réutilisation de lettres	117
Sauvegarde et copie de lettres dans des fichiers	117
Sauvegarde et copie de lettres dans des dossiers	118
Lecture de lettres figurant dans des fichiers et des dossiers	120
Utilisation de <code>vi</code> avec <code>mailx</code>	121
Alias de distribution	123
Définition des alias de distribution dans <code>.mailrc</code> ...	123
Définition des alias de distribution dans <code>/etc/aliases</code>	125
Commandes tilde	129
Aide en ligne : autres commandes <code>mailx</code>	130
8. Utilisation des imprimantes	133
Soumission d'une requête d'impression	133
Soumission de requêtes d'impression à l'imprimante par défaut	134
Soumission de requêtes d'impression à une imprimante donnée	134
Demande de notification de fin d'impression	135
Impression en plusieurs exemplaires	136
Tableau récapitulatif des options <code>lp</code>	136
Vérification de l'état de l'imprimante	137
Vérification de l'état de vos requêtes d'impression ...	138
Recherche des imprimantes disponibles	138
Affichage d'informations sur l'état	139
Affichage de l'état des imprimantes	139

Affichage des caractéristiques d'une imprimante	140
Tableau récapitulatif des options de la commande lpstat	141
Annulation d'une requête d'impression	142
Annulation d'une requête d'impression par numéro d'ID	143
Annulation d'une requête d'impression par nom d'imprimante	143
9. Utilisation du réseau	145
Généralités sur les réseaux	145
Connexion à distance (rlogin)	146
rlogin sans répertoire utilisateur	147
rlogin sous une autre identité	148
rlogin à une machine inconnue	148
Abandon d'une connexion rlogin	149
Suspension d'une connexion rlogin	150
Vérification de votre lieu de connexion (who am i)	150
Copie de fichiers à distance (rcp)	151
Copie de fichiers d'une autre machine vers la vôtre	151
Copie de fichiers de votre machine vers une autre	152
Exécution de commandes à distance (rsh)	152
Affichage d'informations utilisateur (rusers)	153
10. Personnalisation de votre environnement de travail	155
Fichiers d'initialisation	155
Variables d'environnement	156

Le profil utilisateur	157
Définition de la variable PATH	159
Alias (C Shell uniquement)	160
Modification de votre prompt de commande	162
Autres variables d'environnement utiles	163
Définition de droits d'accès par défaut	164
Personnalisation des polices OpenWindows	166
Définition du type de police et de la taille des caractères	166
Affichage de la liste des polices disponibles	169
A. Migration vers Open Windows Version 3.3 ou ultérieure ...	173
SPARC Migration à partir de l'environnement SunView ..	173
SPARC Les fichiers .defaults et .xdefaults	173
Migration à partir de l'environnement OpenWindows pre-Version 3.3	174
La variable d'environnement OPENWINHOME	174
Le fichier .xinitrc	175
Utilisation du fichier de démarrage approprié	176
Propriétés de l'espace de travail	177
Personnalisation du menu Espace de travail	178
B. Modification du clavier	179
Activation/Désactivation de la touche Compose	179
SPARC Reconfiguration du clavier pour les gauchers	180
SPARC Utilisation du script de reconfiguration	180
SPARC Annulation de la reconfiguration du clavier	183

C. Exécution des applications du réseau	187
Utilisation de <code>rlogin</code> pour exécuter une application du réseau	188
Précisions relatives à la sécurité.....	189
A qui s'adresse cette section ?	189
Mécanismes de contrôle d'accès	190
Protocoles d'attribution de droits	191
Gestion de l'accès au serveur.....	192
Exécution de clients sous une autre identité, à distance ou en mode local.....	196
D. SPARC —Interconnexion avec DECnet (DNI).....	199
Configuration de l'interconnexion à un réseau DECnet.....	199
Affichage d'un client distant sur une machine OpenWindows	201
Affichage d'un client distant sur un système VAX	201
E. Configuration des imprimantes et des modems	205
Configuration d'un modem sur votre système	205
Pour obtenir d'autres détails sur les modems	207
Configuration d'une imprimante à utiliser sur votre système.	208
Pour obtenir d'autres détails sur les imprimantes.....	209
Index.....	211

A propos de ce guide

A qui s'adresse ce guide

Ce guide est destiné aux utilisateurs du logiciel système Solaris™. Ce dernier regroupe SunOS™ et OpenWindows™.

Avant de lire ce guide

Votre système doit être installé et prêt à être utilisé. Dans le cas contraire, consultez les manuels *SPARC: Installing Solaris Software* ou *x86: Installing Solaris Software* ou le guide d'installation de votre système avant de continuer.

Bibliographie complémentaire

L'Answerbook du logiciel système Solaris fait référence à un certain nombre de manuels se rapportant au logiciel Solaris. Ceux-ci sont répartis en différents ensembles :

- **Manuels de l'Administrateur Sun**
Ces manuels fournissent des informations détaillées sur l'installation et l'administration du système pour un grand nombre de configurations du système, y compris sur les réseaux de stations de travail Sun.
- **Manuels du Développeur Sun**
Ces manuels fournissent aux développeurs de logiciels les informations qui leur sont nécessaires pour écrire, déboguer et assurer la maintenance des logiciels installés sur le système.

-
- **Manuels de référence Sun**
Ces manuels décrivent l'ensemble des commandes SunOS. Appelées pages *man*, ces informations peuvent être installées en option comme documentation en ligne.
 - **Manuels destinés à l'utilisateur Sun**
Ces manuels fournissent des informations détaillées sur de nombreux aspects du système SunOS, y compris sur l'utilisation des commandes SunOS, le fonctionnement d'OpenWindows, la personnalisation de votre environnement de travail, le traitement des incidents, l'écriture de scripts de commandes, l'utilisation de la messagerie électronique et le travail en réseau.

Pour plus d'informations sur l'Answerbook du logiciel système Solaris, consultez la brochure *Structure de Solaris*.

Conventions typographiques

Ce guide respecte les conventions typographiques suivantes :

- *Les caractères en italiques* sont utilisés pour mettre l'accent sur un élément particulier, pour les arguments de commandes, les variables et les titres de manuels. Par exemple :

Vous devez entrer l'argument *nom_fichier* comme décrit dans *man Pages(1): User Commands*.

- La police **Courier gras** est utilisée dans les exemples d'écran afin d'indiquer les caractères ou les mots que vous devez taper. Par exemple :

```
$ date
```

- La police **Courier** est utilisée pour les listes de programmes, les noms de commandes, les noms de programmes ou les noms de systèmes. Par exemple :

Vous pouvez devenir superutilisateur en tapant `root` au prompt de connexion ou en tapant la commande `su` au prompt de commande.

Cette police est également utilisée pour les parties de texte que le système affiche à l'écran. Par exemple :

```
$ who
bozo      console      Sep 11 15:36
```

- Les boutons de la souris sont identifiés par leur fonction et non par leur emplacement et sont imprimés en majuscules. Par exemple, vous trouverez "Cliquez avec SELECT" et non "Cliquez avec le bouton gauche de la souris".

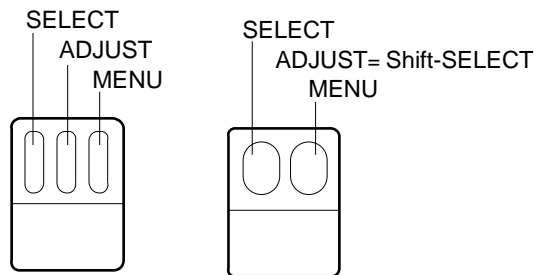


Figure P-1 Fonctions par défaut des boutons de la souris

Sur une souris à trois boutons, les fonctions par défaut sont les suivantes :

- SELECT = bouton gauche de la souris
- ADJUST = bouton central de la souris
- MENU = bouton droit de la souris

Sur une souris à deux boutons, les fonctions par défaut sont les suivantes :

- SELECT = bouton gauche de la souris
- MENU = bouton droit de la souris

Pour obtenir la fonction ADJUST sur une souris à deux boutons, cliquez et maintenez enfoncée la touche Shift de votre clavier tout en appuyant sur le bouton SELECT de votre souris.

- Les noms de touches du clavier se distinguent par leur initiale en majuscules (par exemple, "Appuyez sur la touche Shift"). La touche Return s'appelle touche Enter sur certains claviers.

Remarque : Solaris fonctionne sur machines SPARC et x86, qui n'ont pas le même clavier. Par conséquent, certaines commandes des menus OpenWindows ont des équivalents clavier différents.

Pour la plupart des équivalents clavier, Solaris propose une combinaison de touches méta que vous pouvez utiliser sur ces deux plates-formes. Reportez-vous au Chapitre 1, "Connexion à SunOS et lancement d'OpenWindows," pour une liste des fonctions des commandes avec leurs équivalents clavier.

- Les équivalents clavier (également appelés *raccourcis clavier*) des commandes de Open Windows sont représentés dans ce document de la manière suivante : un tiret haut (-) signifie que vous appuyez simultanément sur deux touches.

Par exemple, Ctrl-Alt indique que vous cliquez en même temps avec les touches Control et Alt ; Ctrl-Alt-c signifie que vous enfoncez simultanément Control, Alt et c.

- Vous utilisez un équivalent clavier en cliquant avec la touche meta puis avec une ou plusieurs autres touches du clavier. Sur un clavier SPARC, la touche meta est représentée par <> ; sur les claviers x86, il faut enfoncer simultanément les touches Ctrl et Alt (Ctrl-Alt). Pour plus d'informations sur ces équivalents clavier, reportez-vous au Chapitre 1, "Connexion à SunOS et lancement d'OpenWindows,".
- Le présent manuel décrit le choix d'une option de menu de la façon suivante : Choisissez Espace de travail ► Menu.
- Lorsqu'il est question du menu tout entier ou de l'option par défaut de ce menu dans le texte, un *cercle* entoure l'élément par défaut du menu. Par défaut, vous affichez le menu entier lorsque vous cliquez sur SELECT.

Conventions spécifiques aux plates-formes

Une procédure ou une section dont le titre est précédé de l'un des signes suivants ne se rapporte qu'à la plate-forme indiquée :

SPARC Représente une section ou une procédure spécifique à la plate-forme SPARC

x86
x86 Représente une section ou une procédure spécifique à la plate-forme x86

Suivant la plate-forme, les remarques sont indiquées de la façon suivante :

SPARC – Remarque appropriée à la plate-forme SPARC.

x86 – Remarque appropriée à la plate-forme x86.

Les contenus des points noirs sont signalés comme suit :

- **SPARC**: Point spécifique à la plate-forme SPARC.
- **x86** : Point spécifique à la plate-forme x86.

Si un code, un tableau ou une figure se rattache à l'une des deux plates-formes, le nom de cette dernière précède le titre. Par exemple :

Table 1-1 SPARC : Titre spécifique à la plate-forme SPARC

Table 1-2 x86 : Titre spécifique à la plate-forme x86

Connexion à SunOS et lancement d'OpenWindows



Le logiciel OpenWindows Version doit déjà être installé sur le disque dur de votre système ou être accessible depuis un serveur sur votre système de fichiers. Si vous n'en êtes pas sûr, contactez votre administrateur système ou consultez les manuels *SPARC: Installing Solaris Software* ou *x86: Installing Solaris Software*.

Ce chapitre explique comment se connecter au système, utiliser l'interpréteur de commande Shell, monter et lancer l'environnement OpenWindows, quitter le système de multi-fenêtrage et se déconnecter. Il décrit également quelques cas particuliers, par exemple comment configurer OpenWindows pour qu'il puisse fonctionner avec deux moniteurs.

Connexion

Une *session de travail standard* se définit comme l'intervalle de temps entre le moment où vous vous connectez au système et celui où vous vous déconnectez. L'environnement multi-utilisateur SunOS vous demande de décliner votre identité chaque fois que vous voulez utiliser le système. Votre *nom de connexion* (également appelé *nom utilisateur* ou *compte*) représente votre identité pour le système et pour les autres utilisateurs. Le *mot de passe* vous permet de limiter l'accès à votre compte aux personnes le connaissant. Si vous n'avez pas de nom de connexion ni de mot de passe, demandez à votre *administrateur système* de créer un compte à votre nom. Vous pourrez ensuite vous connecter.

Pour pouvoir vous connecter au système, vous devez voir apparaître à l'écran :

```
login:
```

Entrez le nom de connexion qui vous a été attribué par l'administrateur système et appuyez sur la touche Return. Par exemple, si votre nom de connexion est `bernard`, tapez :

```
login: bernard
```

et appuyez sur Return. Le système vous demande ensuite votre mot de passe sous la forme suivante :

```
login: bernard  
Password:
```

Tapez votre mot de passe au prompt et appuyez sur Return. (Si aucun mot de passe n'est rattaché à votre compte, le système effectue la connexion sans vous le demander.) Vous pouvez remarquer que le mot de passe n'apparaît pas (*echo*) à l'écran lorsque vous le tapez, ceci afin d'éviter que d'autres personnes en prennent connaissance.

Le Shell de connexion

Dans les sections suivantes, vous allez entrer des commandes SunOS. Lorsque vous transmettez une commande au système, vous envoyez en fait des informations à un interpréteur de commande appelé *Shell*. Ce dernier lit les informations qu'il reçoit et déclenche une action en conséquence au sein du système.

Le Shell par défaut du logiciel système SunOS est le Bourne Shell, mais il existe deux autres interpréteurs de commande dans le système d'exploitation : le C Shell et le Korn Shell. Chacun d'eux comporte ses spécificités.

Remarque : vous pouvez obtenir des informations spécifiques sur n'importe quelle commande SunOS, y compris sur les interpréteurs de commande, en vous reportant à la page manuel (`man`) correspondante. Pour plus d'informations sur les pages manuel, reportez-vous à la section “Affichage des pages manuel avec `man`” dans le Chapitre 2, “Commandes SunOS de base.”

Lorsque vous vous connectez pour la première fois au système (ou que vous ouvrez une nouvelle fenêtre Utilitaire de commande ou Utilitaire Shell) et que vous voyez apparaître le prompt, cela indique qu'un interpréteur de commande a été lancé automatiquement. Ce programme est appelé *Shell de connexion*. Si votre Shell de connexion n'est pas l'interpréteur de commande par défaut de SunOS (Bourne Shell), c'est parce que votre administrateur système vous en a attribué un autre (C Shell ou Korn Shell).

Comme nous l'avons déjà indiqué, chaque interpréteur de commande comporte ses spécificités. Ainsi, l'utilisation de certaines commandes ou procédures peut être limitée à l'un de ces programmes. Toutefois, il convient de préciser qu'à chaque fois qu'une commande ou procédure citée dans ce guide n'est pas utilisable avec l'interpréteur de commande par défaut de SunOS (Bourne Shell), les sections correspondantes en font expressément état.

Déconnexion

Lorsque vous avez terminé votre session de travail et que vous êtes prêt à quitter le système d'exploitation, tapez la commande suivante pour vous déconnecter :

```
$ exit
```

Au bout de quelques instants, le prompt de connexion réapparaît :

```
$ exit
login:
```

Il indique que la procédure de déconnexion est terminée. Le système est maintenant prêt pour une nouvelle connexion, sous votre nom ou celui d'un autre utilisateur.

Remarque : le système d'exploitation SunOS vous permet d'éteindre votre station de travail ou votre terminal *sans* nécessairement vous déconnecter. Vous restez en effet connecté au système, à moins que vous ne choisissiez de vous déconnecter explicitement.

Equivalents clavier

Dans certains cas, vous pouvez accélérer l'exécution d'une commande grâce à des équivalents clavier, appelés *raccourcis clavier*. Ils reproduisent le fonctionnement de la souris et des menus, ainsi que des touches du clavier préconfigurées.

Le tableau suivant présente plusieurs fonctions de commandes avec leurs équivalents clavier pour les machines SPARC et x86.

Remarque : Sur les claviers SPARC, la touche méta correspond à <>, et sur les claviers x86, vous l'obtenez en appuyant sur Ctrl-Alt. .

Pour utiliser un raccourci clavier, appuyez et maintenez enfoncée la première touche (méta ou Control-Alt) puis tapez la deuxième. Par exemple, pour couper un texte sélectionné, maintenez enfoncée la touche méta et appuyez sur *x* si vous disposez d'un système SPARC ; sur un système x86, maintenez enfoncées Control et Alt en même temps et appuyez simultanément sur *x*

*Tableau 1-1*Raccourcis clavier

Operation	Raccourci Clavier	Action
Encore	Meta - a	Répète l'action précédente
Copier	Meta - c	Copie le texte sélectionné vers le presse-papier
Couper	Meta - x	Coupe le texte sélectionné et le garde dans le presse-papier
Chercher	Meta - f	Sélectionne le texte à droite du curseur
Aide	Help or F1	Affiche une fenêtre d'aide concernant l'objet sélectionné par le pointeur
Nouveau	Meta - n	Charge un nouveau fichier

Tableau 1-1 Raccourcis clavier

Operation	Raccourci Clavier	Action
Ouvrir (Fichier)	Meta - o	Ouvre un fichier (par exemple, si vous avez mis en évidence l'icône d'un fichier dans le gestionnaire des fichiers)
Ouvrir (Fenêtre)	Meta - w	Ouvre une icône ou ferme une fenêtre en icône
Coller	Meta - v	Colle la sélection du presse-papier au niveau du point d'insertion
Imprimer	Meta - p	Envoie le fichier vers l'imprimante (par exemple, si vous avez mis en évidence l'icône d'un fichier dans le gestionnaire des fichiers)
Props	Meta - i	Affiche la fenêtre correspondant à l'application indiquée par le pointeur
Refaire	Shift-Meta - p	Annule une annulation
Sauvegarder	Meta - s	Sauvegarde le fichier courant
Stop	Stop or Esc	Arrête l'opération en cours
Défaire	Meta - u	Annule l'opération précédente

Remarques préliminaires sur le lancement d'OpenWindows

Avant de lancer le logiciel OpenWindows, veuillez noter les points suivants. Si aucune de ces situations ne vous concerne, passez directement à la section "Lancement du logiciel OpenWindows" du présent chapitre.

- S'il est de votre responsabilité d'installer OpenWindows sur un serveur, consultez *SPARC: Installing Solaris Software* ou *x86: Installing Solaris Software*.
- Si vous utilisiez jusqu'à présent OpenWindows Version 2, consultez l'Annexe A, "Migration vers Open Windows Version 3.3 ou ultérieure".
- Si vous utilisiez jusqu'à présent l'environnement SunView, consultez l'Annexe A, "Migration vers Open Windows Version 3.3 ou ultérieure".
- Si vous utilisiez jusqu'à présent une version d'OpenWindows antérieure à 3.3, consultez les sections "La variable d'environnement OPENWINHOME" et "Utilisation du fichier de démarrage approprié", dans le présent chapitre.

x86 – Si vous avez modifié le type de souris, le clavier ou la carte adaptateur vidéo de votre système après installation de Solaris, vous devrez lancer le programme `devconfig` pour mettre à jour la configuration système. Reportez-vous à la page `man devconfig(1M)`.

La variable d'environnement OPENWINHOME

Si vous utilisiez jusqu'à maintenant une version d'OpenWindows antérieure à 3.3, vous avez peut-être configuré votre système de façon à utiliser la variable d'environnement `OPENWINHOME`. Il est désormais déconseillé de définir cette variable manuellement ou à partir d'un fichier de démarrage.

En effet, la commande `openwin` définit automatiquement la variable d'environnement `OPENWINHOME` dans `/usr/openwin`. Vous n'avez donc pas à le faire.

Si vous avez défini la variable d'environnement `OPENWINHOME` dans le fichier `.profile` ou `.cshrc` de votre répertoire utilisateur, placez la ligne correspondante en commentaire ou supprimez-la complètement *avant* d'utiliser OpenWindows Version 3.3 ou une version ultérieure.

Pour effacer (ou placer en commentaire) la variable d'environnement `OPENWINHOME` définie dans le fichier `.profile` ou `.cshrc` :

1. Ouvrez le fichier `.profile` ou `.cshrc` au moyen d'un éditeur de texte tel que `vi`.
2. Tapez un dièse (#) avant la variable, comme indiqué ci-après, ou supprimez entièrement la ligne.
Si vous travaillez dans le fichier `.profile`, suivez l'étape a ; si vous travaillez dans le fichier `.cshrc`, suivez l'étape b.
 - a. Dans le fichier `.profile` :

```
#OPENWINHOME=/usr/openwin
```

- b. Dans le fichier `.cshrc` :

```
#setenv OPENWINHOME /usr/openwin
```


3. Sauvegardez et quittez le fichier.

4. Réinitialisez la variable d'environnement `OPENWINHOME` (ou déconnectez-vous, puis reconnectez-vous).

Si vous utilisez l'interpréteur de commande Bourne Shell ou Korn Shell, suivez l'étape a. Si vous utilisez l'interpréteur C shell, suivez l'étape b.

a. Dans le Bourne Shell ou le Korn Shell, tapez :

```
$ unset OPENWINHOME
```

b. Dans un C Shell, tapez :

```
example% unsetenv OPENWINHOME
```

Après avoir réinitialisé la variable d'environnement, vous êtes prêt à utiliser le logiciel OpenWindows, comme indiqué dans la section “Lancement du logiciel OpenWindows” du présent chapitre.

Utilisation du fichier de démarrage approprié

Si vous utilisiez jusqu'à maintenant une version d'OpenWindows antérieure à 3.3, vous devez déterminer le statut de votre fichier `.xinitrc`. Le fichier `.xinitrc` est un fichier de démarrage OpenWindows de votre répertoire utilisateur qui peut contenir des options définies par l'utilisateur.

Pour déterminer le statut du fichier `.xinitrc`, tapez les commandes :

```
$ cd
$ ls -a .xinitrc
```

Selon le résultat de cette commande, suivez l'une des étapes suivantes :

- Si vous n'avez pas de fichier `.xinitrc` (la commande `ls -a` précédente n'ayant pas généré de sortie pour ce fichier), aucune intervention n'est requise. Si votre répertoire utilisateur ne contient pas de fichier `.xinitrc`, OpenWindows utilisera à la place le fichier de démarrage par défaut.

- Si vous avez un fichier `.xinitrc` (dont la commande `ls -a` précédente a généré une sortie) mais que vous ne l'avez encore jamais modifié ou que vous ne souhaitez pas conserver les modifications effectuées, suivez l'étape 1 des "Procédures relatives aux fichiers de démarrage".
- Si vous avez un fichier `.xinitrc` (dont la commande `ls -a` précédente a généré une sortie) et que vous y avez effectué des modifications que vous souhaitez conserver, suivez l'étape 2 des "Procédures relatives aux fichiers de démarrage".

Procédures relatives aux fichiers de démarrage

1. Pour supprimer le fichier `.xinitrc` de votre répertoire utilisateur, tapez la commande suivante :

```
$ rm .xinitrc
```

2. Pour conserver les modifications effectuées dans votre fichier `.xinitrc`, suivez les étapes ci-après :

- a. Transférez `.xinitrc` dans `.xinitrc.save`:

```
$ mv .xinitrc .xinitrc.save
```

- b. Copiez `$OPENWINHOME/lib/Xinitrc` dans le fichier `.xinitrc` de votre répertoire utilisateur :

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

- c. Insérez dans le fichier `.xinitrc` les lignes du fichier `.xinitrc.save` que vous souhaitez conserver.



Attention : lorsque vous éditez le fichier `.xinitrc`, n'ajoutez pas d'autre version de `olwm`, n'ajoutez pas `svenv` et n'effacez pas la ligne contenant `$OPENWINHOME/lib/openwin-sys`.

Lancement de l'environnement utilisateur OpenWindows

Pour lancer l'environnement utilisateur OpenWindows, suivez les étapes ci-après :

1. Au moyen de NFS, montez le logiciel OpenWindows à partir du serveur sur lequel il est installé.

Pour plus d'informations sur le montage du logiciel OpenWindows à partir d'un serveur, consultez *OpenWindows Reference Manual* ou contactez votre administrateur système.

2. Lancez le logiciel OpenWindows à l'aide de la commande `openwin`, en ajoutant les options de lancement supplémentaires, si nécessaire.

Lancement du logiciel OpenWindows

Après avoir monté OpenWindows à partir d'un serveur et en avoir exécuté le script d'installation, vous êtes prêt à lancer le logiciel.

Pour lancer le logiciel OpenWindows, tapez `/usr/openwin/bin/openwin` au prompt du Shell et appuyez sur Return.

```
$ /usr/openwin/bin/openwin
```

Cette commande affiche l'écran OpenWindows Version 3.3 et configure l'environnement de travail OpenWindows.

Lancement rapide d'OpenWindows

Après avoir lancé OpenWindows, vous pouvez configurer le système de façon à utiliser un raccourci qui vous évitera de devoir taper systématiquement le chemin d'accès complet d'OpenWindows.

Si vous utilisez l'interpréteur de commande Bourne Shell ou Korn Shell, vous devez placer une fonction Shell dans votre fichier `.profile`. Si vous utilisez un C Shell, vous devez placer un *alias* dans votre fichier `.cshrc`. Les fichiers `.profile` et `.cshrc` se trouvent dans votre répertoire utilisateur.

Après avoir inséré le raccourci dans le fichier approprié pour votre interpréteur de commande, il vous suffit, pour lancer OpenWindows, de taper :

```
$ openwin
```

Pour insérer le raccourci OpenWindows dans vos fichiers de démarrage, consultez les sections suivantes, “Dans le fichier `.profile`” et “Dans le fichier `.cshrc`”.

Dans le fichier `.profile`

Pour insérer le raccourci OpenWindows dans votre fichier `.profile` :

1. Ouvrez le fichier `.profile` au moyen d'un éditeur de texte tel que `vi`.
2. Entrez la fonction Shell suivante, exactement comme indiqué, dans le fichier :

```
openwin () {  
    /usr/openwin/bin/openwin  
}
```

3. Sauvegardez et quittez le fichier.
4. Déconnectez-vous, puis reconnectez-vous pour activer le raccourci, ou tapez :

```
$ . .profile
```

Chaque fois que vous voudrez lancer OpenWindows, il vous suffira désormais de taper `openwin`.

Dans le fichier `.cshrc`

Pour insérer le raccourci OpenWindows dans votre fichier `.cshrc` :

1. Ouvrez le fichier `.cshrc` au moyen d'un éditeur de texte tel que `vi`.

2. Entrez la commande d'alias suivante, exactement comme indiqué, dans le fichier :

```
alias openwin /usr/openwin/bin/openwin
```

3. Sauvegardez et quittez le fichier.
4. Déconnectez-vous, puis reconnectez-vous pour activer le raccourci, ou tapez :

```
example% source .cshrc
```

Chaque fois que vous voudrez lancer OpenWindows, il vous suffira désormais de taper openwin.

Si OpenWindows ne démarre pas

L'accès au logiciel OpenWindows s'effectue via le répertoire /usr/pioneering. C'est le répertoire par défaut dans lequel est installé OpenWindows. En effet, de nombreuses applications, dont l'Agenda, ne peuvent pas être chargées si OpenWindows n'est pas correctement installé dans /usr/openwin.

Si vous ne parvenez pas à lancer le logiciel OpenWindows à l'aide de la commande /usr/openwin/bin/openwin, cela signifie qu'il n'est pas installé ou qu'il est installé dans un répertoire autre que /usr/openwin. Dans ce cas, consultez le guide *OpenWindows Reference Manual* ou contactez votre administrateur système.

Sortie de l'environnement OpenWindows

Après avoir lancé le logiciel OpenWindows et travaillé dans cet environnement en multi-fenêtrage, vous ne pouvez vous déconnecter comme vous le feriez dans le cas d'une session standard de ligne de commande SunOS. Vous devez en effet quitter l'environnement en multi-fenêtrage avant de vous déconnecter.

Si vous tapez `logout` au prompt du Shell, le message suivant apparaîtra :

```
Il ne s'agit pas d'un shell de connexion.
```

Pour quitter l'environnement OpenWindows, suivez les étapes ci-après :

1. **Déplacez la souris de façon à positionner la flèche (*pointeur*) sur le fond de l'écran (*espace de travail*).**
2. **Appuyez sur le bouton `MENU` de la souris.**
Le menu Espace de travail apparaît.
3. **Faites glisser le pointeur sur ce menu de façon à mettre en évidence la dernière option, `Sortie`.**
4. **Relâchez le bouton de la souris.**
Vous voyez apparaître une fenêtre pop-up vous demandant de confirmer que vous voulez quitter le système de multi-fenêtrage.
5. **Positionnez le pointeur sur `Sortie` et cliquez sur le bouton `SELECT` de la souris.**
Au bout de quelques instants, toutes les fenêtres sont défaites et le prompt du système apparaît dans l'angle inférieur gauche de l'écran.

Options de lancement spéciales d'OpenWindows

Dans la plupart des cas, il suffit pour lancer le logiciel OpenWindows de suivre les étapes décrites dans la section "Lancement du logiciel OpenWindows". Toutefois, dans certaines situations, vous pouvez être appelé à utiliser des options supplémentaires afin de modifier la procédure de lancement d'OpenWindows.

Cette section décrit les situations particulières suivantes :

- Lancement du logiciel OpenWindows avec un niveau de sécurité réseau réduit.
- Lancement du logiciel OpenWindows avec différents types de moniteur et de mémoires tampon d'écran.
- Lancement du logiciel OpenWindows sur plusieurs écrans.

Pour lancer le logiciel OpenWindows avec des options spéciales, vous devez utiliser la commande `openwin` :

```
$ openwin [ options ]
```

Dans l'exemple ci-dessus, *options* désigne les options de ligne de commande qui vous permettent de personnaliser la configuration par défaut du serveur. Les sections suivantes décrivent certaines des options les plus courantes.

Lancement d'OpenWindows avec un niveau de sécurité réseau réduit

Si vous travaillez dans un environnement de réseau ouvert *et que vous n'êtes pas confronté à des problèmes de sécurité réseau*, vous pouvez utiliser l'option `-noauth` pour permettre à d'autres utilisateurs d'ouvrir des applications sur votre système.

La commande suivante remplace la fonction de sécurité par défaut et vous permet d'indiquer le nom des autres utilisateurs ayant accès à votre serveur de multi-fenêtrage :

```
$ openwin -noauth
```

Lancement d'OpenWindows avec différents types de moniteur

Si vous avez un moniteur à échelle de gris (un moniteur monochrome doté d'une mémoire tampon d'écran de 8 bits ou plus), vous pouvez utiliser l'option `grayvis` lorsque vous lancez OpenWindows. Bien qu'elle ne soit pas obligatoire, cette option peut en effet améliorer certains aspects de l'affichage.

Pour utiliser cette option, tapez la commande suivante au prompt du système :

```
$ cd  
$ openwin -dev /dev/fb grayvis
```

SPARC *Lancement avec différents types de moniteurs*

SPARC – Notez que les sections suivantes, relatives au lancement avec plusieurs moniteurs, ne s'appliquent qu'aux machines basées sur SPARC. Les procédures suivantes requièrent une certaine expérience en matière d'administration système. Si vous configurez un système pour la première fois, demandez conseil à votre administrateur système.

Pour lancer l'environnement OpenWindows afin qu'il puisse fonctionner avec plusieurs écrans, vous devez indiquer au système la présence de périphériques supplémentaires, ainsi que les types d'écran que vous voulez utiliser. Vous pouvez pour cela sélectionner les options de type périphérique (device) ou utiliser les valeurs par défaut fournies avec le script `openwin` qui permet de lancer le logiciel OpenWindows.

Les deux options suivantes doivent être utilisées avec la commande `openwin` pour lancer le logiciel de façon qu'il puisse fonctionner avec deux moniteurs :

```
$ openwin [ [ -dev device ] [ deviceoptions ] ]
```

Les doubles crochets indiquent que l'association [*-dev périphérique*] [*options_périphérique*] peut être entrée plusieurs fois sur la ligne de commande (c'est-à-dire une fois par périphérique).

[*-dev périphérique*]

L'option *périphérique* de la ligne de commande indique le type de périphérique à mémoire tampon d'écran que le serveur doit utiliser pour l'affichage.

Si la ligne de commande ne contient pas cette option, le serveur utilise le périphérique par défaut `/dev/fb`. Lorsque l'option `-dev` apparaît plusieurs fois sur la ligne de commande, cela indique que plusieurs écrans sont connectés au même serveur.

[*options_périphérique*]

L'option *options_périphérique* de la ligne de commande est une liste de modificateurs de périphérique qui transforment le comportement du périphérique indiqué dans l'option `-dev`.

Exemples d'options de type périphérique

Cette section donne des exemples de configuration d'écrans superposés ou juxtaposés.

Remarque : dans tous ces exemples, l'ordre des périphériques est important. Le premier périphérique indiqué doit être placé physiquement à gauche ou au-dessus du second périphérique et, en conséquence, le second périphérique doit se trouver à droite ou en-dessous du premier.

[left] [right]

La ligne de commande suivante demande au système de configurer deux écrans. L'écran de gauche (left) est le périphérique à mémoire tampon d'écran par défaut et l'écran de droite (right) est de type monochrome. Cette configuration vous permet de passer d'un écran à l'autre en utilisant la souris.

```
$ openwin -dev /dev/fb left -dev /dev/fbs/bwtwo0 right
```

L'exemple suivant est analogue au précédent. Par défaut, le premier périphérique indiqué sur la ligne de commande se trouve à la gauche du second.

```
$ openwin -dev /dev/fb -dev /dev/fbs/bwtwo0
```

La ligne de commande suivante demande au système de configurer deux écrans. L'écran de droite est le périphérique à mémoire tampon d'écran par défaut et l'écran de gauche est de type monochrome. Cette configuration vous permet de passer d'un écran à l'autre en utilisant la souris.

```
$ openwin -dev /dev/fb right -dev /dev/fbs/bwtwo0 left
```

[top] [bottom]

La ligne de commande suivante demande au système de confirmer deux écrans. L'écran supérieur (top) est de type CG6 et l'écran inférieur (bottom), de type monochrome. Cette configuration vous permet de passer d'un écran à l'autre en utilisant la souris.

```
$ openwin -dev /dev/fbs/cgsix0 top -dev /dev/fbs/bwtwo0 bottom
```

L'exemple suivant est *différent* du précédent. Par défaut, le premier périphérique indiqué sur la ligne de commande est en effet placé à la *gauche* du second.

```
$ openwin -dev /dev/fbs/cgsix0 -dev /dev/fbs/bwtwo0
```

La ligne de commande suivante demande au serveur de configurer deux écrans. L'écran inférieur est de type CG6 et l'écran supérieur, de type monochrome. Cette configuration vous permet de passer d'un écran à l'autre en utilisant la souris.

```
$ openwin -dev /dev/fbs/cgsix0 bottom -dev /dev/fbs/bwtwo0 top
```

Remarques diverses

Voici quelques remarques importantes concernant l'utilisation de plusieurs écrans :

- Par défaut, `olwm` gère tous les écrans.
- Vous ne pouvez pas transférer les fenêtres d'un écran à l'autre.

Commandes SunOS de base



Ce chapitre présente une introduction aux commandes utilisateur du système d'exploitation SunOS. Il explique comment saisir les commandes, corriger les fautes de frappe, saisir plusieurs commandes ou des commandes longues et utiliser les options de commande. Il fournit en outre d'autres indications utiles.

Pour entrer des commandes, vous devez utiliser la fenêtre Utilitaire de commande ou Utilitaire Shell. Pour afficher l'une ou l'autre de ces fenêtres, sélectionnez l'option Menu à partir de l'Espace de travail.

Le prompt de commande

Une fois que vous êtes connecté, l'écran ou la fenêtre qui apparaît contient uniquement le prompt initial. La nature de ce prompt varie selon l'interpréteur de commande Shell utilisé et la façon dont l'administrateur système l'a initialement configuré. Le prompt de commande par défaut du système d'exploitation SunOS étant le signe dollar (\$), c'est le prompt qui est utilisé dans la plupart des exemples de ce manuel.

Si vous décidez par la suite de modifier le prompt de commande, reportez-vous à la section "Modification de votre prompt de commande", dans le Chapitre 10, "Personnalisation de votre environnement de travail", pour connaître la procédure à suivre.

Saisie des commandes

La présence du prompt de commande indique que le système attend une action de votre part. Commencez par entrer la commande `date` au prompt, comme indiqué dans cet exemple (en tapant `date` et en appuyant sur la touche Return) :

```
$ date
Mon Feb 3 10:12:51 PST 1992
$
```

Comme vous le voyez, cette commande affiche la date et l'heure en cours. Entrez à présent la même commande, mais commençant par une majuscule :

```
$ Date
Date: Commande introuvable.
$
```

Comme vous le voyez, le système n'interprète pas le `D` majuscule de la même façon que le `d` minuscule. Presque toutes les commandes du système d'exploitation SunOS sont en minuscules.

Correction des fautes de frappe

Supposons que vous commenciez à taper la commande `Date` et que vous vous aperceviez que vous avez fait une erreur avant d'appuyer sur la touche Return. Le texte que vous avez tapé n'est envoyé au système que lorsque vous avez appuyé sur Return. Vous pouvez donc corriger votre erreur. Pour cela, vous avez deux possibilités :

- Appuyer sur la touche Del ou Back Space pour placer le curseur à l'endroit de l'erreur ; ou bien
- Taper Ctrl-U pour effacer toute la ligne et recommencer. (Maintenez la touche Control enfoncée et appuyez sur "u".)

Essayez ces deux méthodes pour voir comment elles fonctionnent. (Si les touches Del/Back Space peuvent être différentes selon les systèmes, la combinaison Ctrl-U fonctionne, en revanche, sur la plupart des systèmes.)

Saisie de plusieurs commandes et de commandes longues

Vous pouvez entrer plusieurs commandes sur une même ligne. Il suffit de les séparer par un point-virgule (;), comme illustré ci-dessous pour les commandes `date` et `logname` :

```
$ date; logname
Mon Feb 3 10:19:25 PST 1992
bernard
$
```

Comme vous le voyez, l'exemple ci-dessus permet d'afficher la date et l'heure en cours (via la commande `date`), ainsi que le nom de connexion de l'utilisateur actuellement connecté au système (via la commande `logname`).

Si la commande que vous tapez est très longue, vous pouvez utiliser la barre oblique inversée (\) pour continuer la saisie sur une autre ligne. Exemple :

```
$ date; \
logname
Mon Feb 3 10:23:25 PST 1992
bernard
$
```

Bien qu'il ne s'agisse pas de commandes longues, `date` et `logname` sont utilisées dans cet exemple pour illustrer de la façon la plus simple le principe de continuité des commandes sur la ligne suivante. Vous verrez par la suite, lorsque les commandes que vous voudrez utiliser seront trop longues pour tenir sur la largeur de l'écran, que l'emploi de la barre oblique inversée peut être extrêmement utile.

Remarque : si vous utilisez les fenêtres Utilitaire Shell ou Utilitaire de commande de l'environnement OpenWindows, vous n'avez plus besoin de la barre oblique inversée pour continuer la saisie des commandes sur la ligne suivante. En effet, lorsque vous arrivez en fin de ligne, les commandes que vous tapez sont automatiquement renvoyées à la ligne suivante et le système exécute toutes les commandes une fois que vous avez appuyé sur Return.

Répétition des commandes antérieures

Remarque : les fonctions de répétition de commande décrites dans cette section ne sont disponibles qu'avec le C Shell.

Un moyen rapide de répéter la dernière commande consiste à taper `!!` et à appuyer sur Return. Le système conserve l'*historique* des commandes que vous avez tapées et peut répéter ces commandes. Par exemple, si la dernière commande que vous avez entrée est `date` :

```
example% !!
date
Mon Feb 3 10:26:20 PST 1992
example%
```

Vous pouvez également répéter n'importe quelle commande antérieure en tapant `!x`, où `x` est le numéro correspondant à la commande voulue sur la *liste de l'historique*. Pour afficher cette liste, tapez la commande `history` et appuyez sur Return. Voici un exemple de ce que vous voyez apparaître :

```
example% history
1  pwd
2  clear
3  ls -l
4  cd /usr/home/worker
5  logname
6  date
7  history
```

Une autre méthode pour répéter certains éléments de la liste de l'historique consiste à faire suivre le caractère `!` d'un nombre négatif. Par exemple, pour répéter la troisième commande en partant de la fin dans la liste de l'historique, tapez :

```
example% !-2
logname
bernard
example%
```

Lorsque vous utilisez la liste exemple ci-dessus, la commande `logname` est répétée.

Une troisième méthode consiste à faire suivre le caractère `!` des premiers caractères de l'une des commandes antérieures. Par exemple, si vous avez entré précédemment la commande `clear` pour effacer le contenu de votre écran, vous pouvez taper `!cl` pour répéter cette commande. Toutefois, avec cette méthode de répétition des commandes, veillez à indiquer suffisamment de caractères pour différencier la commande dans la liste de l'historique. Si vous n'entrez qu'une lettre après le `!`, le système répètera la dernière commande commençant par cette lettre.

Ajout d'options de commande

De nombreuses commandes sont dotées d'*options* qui permettent d'appeler des fonctions spéciales. Par exemple, la commande `date` comprend l'option `-u`, qui exprime la date au format GMT au lieu de l'heure locale :

```
$ date -u
Mon Feb 3 11:06:51 GMT 1993
$
```

La plupart des options sont représentées par un caractère unique précédé d'un tiret (`-`). Certaines commandes ne comportent pas d'options tandis que d'autres en possèdent plusieurs. Si vous utilisez plusieurs options pour une même commande, vous pouvez taper ces options séparément (`-a -b`) ou ensemble (`-ab`).

Redirection et transmission par pipe du résultat d'une commande

Par défaut, le résultat des commandes est affiché à l'écran. Toutefois, il existe des symboles spéciaux permettant de *rediriger* ce résultat. Vous pouvez, par exemple, le sauvegarder dans un fichier au lieu de l'afficher à l'écran. L'exemple suivant illustre l'utilisation du symbole de redirection (`>`) :

```
$ date > sample.file
$
```

Dans cet exemple, le résultat de la commande `date` est redirigé vers un nouveau fichier, appelé `sample.file`. Le contenu de `sample.file` est ensuite affiché à l'aide de la commande `more` :

```
$ more sample.file
Mon Feb 3 12:56:26 PST 1993
$
```

Comme vous le voyez, `sample.file` contient à présent le résultat de la commande `date`. (Pour plus d'informations sur la commande `more`, reportez-vous au Chapitre 3, "Gestion des fichiers et des répertoires".)

Vous pouvez parfois être appelé à rediriger le résultat d'une commande vers une autre commande. Un jeu de commandes chaînées les unes aux autres de cette façon est appelé *pipe-line*. Le symbole utilisé pour ce type de redirection est une barre verticale (`|`), appelée *pipe*.

Au lieu de sauvegarder le résultat d'une commande dans un fichier, il peut arriver notamment que vous vouliez le rediriger vers une commande d'impression (`lp`) à l'aide du symbole (`|`). Pour envoyer le résultat de la commande `date` directement à l'imprimante, entrez les commandes suivantes :

```
$ date | lp
$
```

Cette combinaison imprime le résultat de la commande `date`. (Pour plus d'informations sur l'utilisation de la commande `lp` pour imprimer des fichiers, reportez-vous à la section "Soumission de requêtes d'impression à l'imprimante par défaut", dans le Chapitre 8, "Utilisation des imprimantes".)

Les exemples de redirection de commande présentés ici sont très simples, mais lorsque vous utiliserez des commandes avancées, vous découvrirez que les fonctions de redirection et de transmission par pipe ont une multitude d'applications possibles.

Exécution de commandes en arrière-plan

Il peut être pratique de lancer une commande à partir du prompt de commande, puis de la placer en *arrière-plan*. Lorsque la commande ne se trouve pas en arrière-plan, le prompt suivant n'apparaît qu'une fois la tâche correspondante exécutée. Toutefois, certaines tâches étant relativement longues, il est appréciable de pouvoir entrer d'autres commandes au cours de leur exécution.

Pour exécuter une commande en arrière-plan, il suffit de taper une perluète (&) à la suite de cette commande. Le nombre qui suit est l'identificateur de processus :

```
$ bigjob &  
[1] 21414  
$
```

La commande `bigjob` est désormais exécutée en arrière-plan et vous pouvez continuer à entrer des commandes pendant ce temps. Une fois le travail terminé, vous verrez apparaître un message du type suivant lorsque vous entrerez à nouveau une commande, comme la commande `date` dans cet exemple :

```
$ date  
Mon Feb 3 10:23:25 PST 1992  
[1] + Done    bigjob  
$
```

Si vous risquez de vous déconnecter pendant l'exécution d'un travail en arrière-plan, utilisez la commande `nohup` (pas d'interruption) pour demander au système d'achever le travail en cours, comme indiqué dans cet exemple. Sinon, celui-ci sera interrompu lorsque vous vous déconnecterez :

```
$ nohup bigjob &  
[1] 21414  
$
```

Aide en ligne sur les commandes OS

Cette section décrit différentes fonctions d'aide en ligne qui vous permettent de consulter des informations de référence depuis votre station de travail ou votre terminal.

Remarque : les fonctions décrites ici forment le *complément* des utilitaires d'aide OpenWindows.

Affichage des pages manuel avec `man`

Si vous connaissez le nom d'une commande mais que vous n'êtes pas sûr de sa fonction, la commande `man` peut vous être utile. Pour plus d'informations sur cette commande, tapez :

```
$ man man
```

Cette commande affiche la première partie des pages manuel SunOS dans la fenêtre. Appuyez sur la barre d'espacement pour visualiser l'écran suivant ou appuyez sur la touche Q pour quitter et retourner au prompt de commande. Utilisez la commande `man` pour visualiser toutes les commandes disponibles et afficher leur syntaxe. Les pages manuel contiennent de nombreux exemples illustrant les différentes utilisations d'une commande.

Affichage d'un résumé d'une ligne avec `whatis`

Pour afficher simplement un résumé d'une ligne sur la fonction d'une commande, utilisez la commande `whatis`, comme indiqué ici :

```
$ whatis date
date (1)          -display or set the date
$
```

Dans l'exemple ci-dessus, notez le numéro entre parenthèses figurant après le nom de la commande. Il indique la section à laquelle appartient la commande. Les commandes sont regroupées en diverses catégories selon leur fonction. La plupart des commandes utilisateur font partie de la section 1. Par convention,

le numéro de section est affiché entre parenthèses après le nom de la commande. Si vous cherchez la page manuel imprimée correspondant à une commande donnée, vous la trouverez classée par ordre alphabétique à l'intérieur de ce groupe.

Recherche par mots-clés avec apropos

Supposons que vous connaissiez la fonction que vous voulez exécuter mais que vous ne soyez pas sûr de la commande à utiliser. La commande `apropos` peut vous être utile dans ce cas. Elle localise les commandes en effectuant une recherche par mots-clés. `apropos` affiche en effet la liste de toutes les commandes dont le résumé d'une ligne contient l'un des mots-clés que vous avez entrés. Cette opération peut être longue car certains mots-clés apparaissent souvent.

Afin d'obtenir quelques exemples de résultats de la commande `apropos`, entrez une ou plusieurs commandes parmi les suivantes :

- `apropos who`
- `apropos execute`
- `apropos apropos`

Si vous entrez un mot-clé qui génère un affichage anormalement long, vous pouvez appuyer sur Ctrl-C pour interrompre la procédure et retourner au prompt de commande. (Maintenez la touche Control enfoncée et appuyez sur "c".)

Gestion des fichiers et des répertoires



La ligne de commande SUNOS est utilisée pour manipuler les fichiers et les répertoires. Vous utilisez les noms de fichiers et de répertoires en tant que paramètres de commande SunOS afin de mener à bien des opérations spécifiques. Avec le Gestionnaire de fichiers OpenWindows, la procédure est différente : les fichiers sont affichés sous forme d'icônes qui peuvent être déplacées ou activées par cliquage, et les commandes sont sélectionnées à partir de menus.

Ce chapitre vous présente les concepts et les procédures de gestion des fichiers et des répertoires vous permettant de travailler à partir de la ligne de commande SunOS. Ces procédures s'appliquent à toute ligne de commande SunOS, que vous utilisiez un Utilitaire Shell ou un Utilitaire de commande sous OpenWindows, ou que vous soyez connecté à un terminal distant. Afin de garantir une utilisation optimale du système d'exploitation SunOS, il est essentiel que vous compreniez les concepts présentés dans ce chapitre.

Concepts de gestion des fichiers

Le *fichier* constitue l'élément central du système d'exploitation SunOS. Les composants de ce système sont pour la plupart traités comme des fichiers, y compris :

- **Les documents** : ils comprennent les fichiers texte, tels que les lettres ou les compte-rendus, les codes source machine, ou tout document que vous avez créé et que vous souhaitez sauvegarder.

- **Les commandes** : la plupart des commandes sont des fichiers *exécutables* ; en d'autres termes, ce sont des fichiers que vous pouvez utiliser afin d'exécuter un programme particulier. A titre d'exemple, la commande `date` du chapitre précédent, qui permet d'obtenir la date du jour, est un fichier exécutable.
- **Les périphériques** : terminal, imprimante et unité(s) de disques sont traités comme des fichiers.
- **Les répertoires** : un répertoire est tout simplement un fichier qui contient d'autres fichiers.

La section suivante décrit les commandes utilisées pour créer, lister, copier, déplacer et supprimer des fichiers. Il vous indique également les procédures à suivre pour répertorier le contenu d'un fichier et en déterminer le type.

Utilisation des commandes de fichiers

Chaque commande présentée dans ce chapitre s'accompagne d'un exemple d'utilisation. Mettez ces exemples en pratique au fur et à mesure de votre lecture. Vous pourrez ainsi acquérir une maîtrise plus rapide de ces différentes commandes et des concepts associés.

Avant de commencer

Avant de commencer à gérer les fichiers, vérifiez que vous êtes dans votre répertoire *utilisateur* (home). Ce répertoire a été défini par votre administrateur système lors de la création de votre compte. L'exécution des tâches indiquées dans les exemples suivants vous évitera de créer, copier, déplacer, ou, pire encore, de supprimer des fichiers dans certaines parties du système que d'autres utilisateurs souhaiteraient conserver en l'état.

Afin d'être sûr que vous vous trouvez dans votre répertoire utilisateur, tapez la commande `cd` (changer de répertoire). Vous accédez alors à votre répertoire utilisateur (répertoire par défaut). Tapez ensuite la commande `pwd` (imprimer

le répertoire de travail) pour visualiser l'emplacement de votre répertoire courant dans le système de fichiers. Le répertoire affiché est votre répertoire utilisateur :

```
$ cd
$ pwd
/export/home/nom_utilisateur
```

Dans cet exemple, le répertoire utilisateur est `/export/home/nom_utilisateur`, *nom_utilisateur* désignant l'utilisateur de ce répertoire.

Création d'un fichier test

Pour créer un fichier vide, utilisez la commande `touch`. S'il n'existe pas déjà un fichier du même nom, la commande `touch` crée un fichier vide (si un fichier du même nom existe déjà, la commande `touch` met à jour la dernière heure d'accès au fichier).

```
$ touch tempfile
$
```

Liste de fichiers (ls)

A l'aide de la commande `ls`, établissez une liste des fichiers du répertoire pour vérifier que vous avez créé votre fichier :

```
$ ls tempfile
tempfile
```

Lorsque vous entrez uniquement la commande `ls`, celle-ci affiche la liste de tous les fichiers de votre répertoire. Si vous entrez la commande `ls` en précisant un nom de fichier, seul ce fichier sera affiché, s'il existe.

Pour plus d'informations sur la commande `ls`, consultez les *man Pages(1): User Commands*.

Copie de fichiers (cp)

Utilisez la commande `cp` pour dupliquer le fichier `tempfile` en un fichier `copyfile` :

```
$ cp tempfile copyfile
$
```

Essayez maintenant d'établir une liste des deux fichiers du répertoire. Notez que les noms des deux fichiers se terminent par "file". Le caractère *générique* (*) peut être utilisé pour remplacer un caractère ou une chaîne de caractères communs. Par conséquent, la commande `ls *file` affiche les fichiers `tempfile` et `copyfile` (et tous les autres fichiers du répertoire dont le nom se termine par `file`) :

```
$ ls *file
copyfile  tempfile
```

Remarquez que le fichier `copyfile` est le premier de la liste, car les fichiers sont répertoriés par ordre alphabétique. (Les lettres en majuscules et les nombres précèdent les lettres en minuscules).

Pour plus d'informations sur la commande `cp`, consultez les *man Pages(1): User Commands* .

Déplacement de fichiers et changement de nom (mv)

Vous pouvez déplacer et renommer les fichiers en utilisant la commande `mv` (déplacer). Dans l'exemple ci-dessous, utilisez la commande `mv` pour renommer le fichier `tempfile` en `emptyfile`.

```
$ mv tempfile emptyfile
$
```


Listez à nouveau les deux fichiers afin de vérifier que le changement de nom a bien été effectué :

```
$ ls *file
copyfile  emptyfile
```

Comme vous pouvez le voir, le fichier `tempfile` a été remplacé par `emptyfile`.

Pour plus d'informations sur la commande `mv`, consultez les *man Pages(1): User Commands*.

Suppression de fichiers (`rm`)

Utilisez la commande `rm` (supprimer) pour supprimer le fichier `copyfile`, et vérifiez le résultat à l'aide de la commande `ls` :

```
$ rm copyfile
$ ls *file
emptyfile
```



Attention : la suppression d'un fichier est une action irréversible. Le fichier supprimé ne pourra pas être restauré si aucune copie de sauvegarde n'a été effectuée. Utilisez avec précaution la commande `rm` et, en particulier, le caractère générique (*). Les fichiers qui ont été supprimés à l'aide de la commande `rm` ne pourront pas être restaurés.

Pour plus d'informations sur la commande `rm`, consultez les *man Pages(1): User Commands*.

Affichage du contenu d'un fichier (`more`, `cat`)

Utilisez la commande `more` pour afficher le contenu d'un fichier. Tapez `more` puis le nom du fichier que vous souhaitez afficher. Le contenu du fichier défile à l'écran. Si le contenu du fichier ne peut pas être affiché en totalité à l'écran, le message suivant apparaît :

```
--More--(nn%) [Appuyez sur la barre d'espace pour continuer,
et sur 'q' pour quitter.]
```

où *nn* désigne le pourcentage déjà affiché.

Vous pouvez également utiliser la commande `cat` pour afficher le contenu d'un fichier. Dans ce cas, la totalité du fichier défile rapidement à l'écran sans marquer de pause. La commande `cat` (concaténer) est généralement utilisée pour créer un gros fichier à partir de deux fichiers ou plus, comme dans l'exemple ci-dessous :

```
$ cat file1 file2 file3 > bigfile
$ ls *file
bigfile
file1
file2
file3
$
```

Pour plus d'informations sur les commandes `more` ou `cat`, consultez les *man Pages(1): User Commands*.

Affichage du format d'un fichier (`file`)

Certains fichiers, tels que les fichiers binaires ou exécutables, ne sont pas imprimables et ne peuvent pas être affichés à l'écran. Vous pouvez utiliser la commande `file` pour connaître le format d'un fichier.

Utilisez la commande `file` pour afficher le format du fichier :

```
$ file copyfile
copyfile:      texte ascii
```

Répertoires et arborescence des répertoires

Vous savez maintenant comment établir une liste, copier, renommer et supprimer des fichiers. Toutefois, vous souhaitez peut-être obtenir des réponses à d'autres questions plus générales, et savoir par exemple où ces fichiers sont situés. La présente section traite de l'arborescence d'un répertoire. Lisez-la attentivement, puis mettez en pratique les exemples présentés dans les sections qui font suite.

Arborescence d'un répertoire

Les fichiers sont rassemblés dans des répertoires eux-mêmes organisés hiérarchiquement. Le répertoire "racine" (root), identifié par le signe "/", figure au sommet de l'arborescence.

Comme le montre la Figure 3-1 ci-après, chaque répertoire du système de fichiers peut posséder un grand nombre de sous-répertoires. Les niveaux de répertoires sont communément différenciés à l'aide du caractère /. Sachant cela, remarquez que le répertoire / (racine) contient des sous-répertoires intitulés, entre autres, /usr, /bin, /home et /lib. Le sous-répertoire /home contient d'autres sous-répertoires, tels que user1, user2 et user3.

Pour accéder aux répertoires (et aux fichiers qu'ils contiennent), précisez les noms des répertoires dans lesquels ils sont situés. Cela s'appelle un *chemin d'accès*. A titre d'exemple, le chemin d'accès du répertoire user3 est /home/user3.

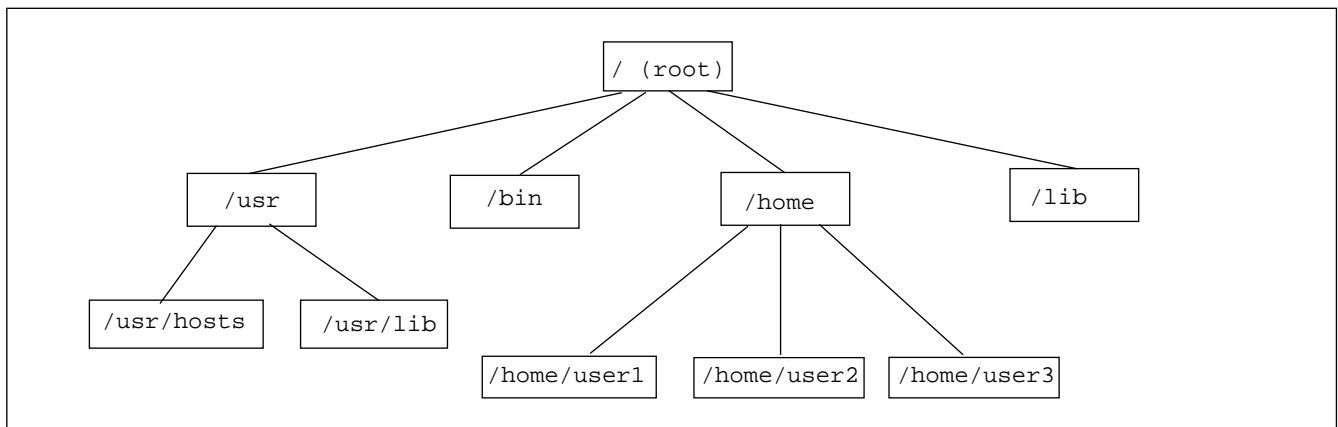


Figure 3-1 Arborescence du système de fichiers

Les noms des sous-répertoires et des fichiers figurant dans un même répertoire doivent être différents. Ils ne peuvent être identiques que s'ils figurent dans des répertoires différents. Par exemple, le répertoire `/usr` contient le sous-répertoire `/usr/lib`. Il n'y a pas de conflit entre `/usr/lib` et `/lib` étant donné que leurs chemins d'accès sont différents.

Les chemins d'accès des fichiers se présentent sous la même forme que ceux des répertoires. Le chemin d'accès d'un fichier indique l'emplacement du fichier en question dans l'arborescence du système de fichiers. Par exemple, si le répertoire `/home/user2` contient un fichier intitulé `report5`, le chemin d'accès de ce fichier est `/home/user2/report5`. Le fichier `report5` se trouve donc dans le répertoire `user2` du répertoire `home`, qui figure lui-même dans le répertoire racine `(/)`.

Les répertoires peuvent contenir des sous-répertoires, des fichiers, ou les deux.

Impression du répertoire de travail (pwd)

La commande `pwd` (imprimer le répertoire de travail) indique votre emplacement dans l'arborescence du système de fichiers :

```
$ pwd
/home/user1
```

Votre résultat sera quelque peu différent de l'exemple ci-dessus, étant donné que votre système de fichiers sera organisé différemment. Souvenez-vous que votre répertoire de travail représente votre emplacement courant dans l'arborescence du système de fichiers.

Votre répertoire utilisateur

Chaque opérateur possède un répertoire *utilisateur* (home). Lorsque vous ouvrez tout d'abord la fenêtre Utilitaire de commande ou Utilitaire Shell dans l'environnement OpenWindows, votre emplacement initial (répertoire de travail) est votre répertoire utilisateur. L'administrateur système a établi ce répertoire lors de la création de votre compte utilisateur.

Changement de répertoire de travail (cd)

La commande `cd` (changer de répertoire) vous permet de vous déplacer dans l'arborescence du système de fichiers :

```
$ cd /usr/lib
$ pwd
/usr/lib
```

Lorsque vous tapez uniquement la commande `cd`, vous revenez à votre répertoire utilisateur. Prenons l'exemple de votre répertoire utilisateur, dont le chemin d'accès est `/home/user1`:

```
$ cd
$ pwd
/home/user1
```

Dans l'interpréteur C Shell, le tilde (`~`) est utilisé comme raccourci pour indiquer votre répertoire utilisateur. Par exemple, pour accéder au sous-répertoire `music` de votre répertoire utilisateur, tapez :

```
exemple% cd ~/music
```

Vous pouvez également utiliser ce raccourci pour indiquer le nom du répertoire utilisateur d'une autre personne. Par exemple :

```
exemple% cd ~nom_utilisateur
```

où *nom_utilisateur* désigne le nom de connexion d'un autre utilisateur et correspond à son répertoire utilisateur.

Remarque : le Bourne shell ne permet pas l'utilisation du raccourci `~`.

Si vous utilisez le Bourne shell, il est possible que votre administrateur système ait configuré le système de telle sorte que vous puissiez taper `$home` pour indiquer votre répertoire utilisateur. Dans ce cas, en tapant :

```
$ cd $home/music
```

vous accédez au sous-répertoire `music` de votre répertoire utilisateur. De même, en tapant :

```
$ cd $homenom_utilisateur
```

vous accédez au répertoire utilisateur de la personne indiquée, où *nom_utilisateur* désigne le nom de connexion d'un autre utilisateur.

Le répertoire situé immédiatement "au-dessus" d'un sous-répertoire s'appelle le *répertoire père*. Dans l'exemple précédent, `/home` est le répertoire père de `/home/user1`. Le symbole `..` ("point-point") représente le répertoire père. Par conséquent, la commande `cd ..` permet d'accéder au répertoire père à partir du répertoire de travail, comme le montre l'exemple ci-dessous :

```
$ pwd
/home/user1
$ cd ..
$ pwd
/home
```

Supposons que votre répertoire de travail en cours soit `/home/user1` et que vous souhaitiez travailler sur des fichiers du répertoire `/home/user2`. Voici un raccourci utile :

```
$ pwd
/home/user1
$ cd ../user2
$ pwd
/home/user2
```

`../user2` demande au système de chercher le répertoire `user2` dans le répertoire père. Cette procédure évite ainsi de taper le chemin d'accès complet `/home/user2`.

Création d'un répertoire (mkdir)

Il est facile de créer un nouveau répertoire. Il vous suffit de taper la commande `mkdir`, puis le nom du répertoire que vous souhaitez créer :

```
$ mkdir veggies
$ cd veggies
$ mkdir broccoli
$ cd broccoli
$ pwd
/home/user2/veggies/broccoli
```

Chemins d'accès relatifs

Le chemin d'accès complet d'un répertoire ou d'un fichier commence par une barre oblique (/) et décrit la structure complète des répertoires situés entre ce fichier (ou ce répertoire) et le répertoire racine. Toutefois, un chemin d'accès beaucoup plus bref peut être utilisé dans la plupart des cas pour définir le fichier ou le répertoire *relativement* au répertoire de travail courant.

Lorsque vous êtes dans un répertoire père, vous pouvez accéder à un sous-répertoire en utilisant uniquement le nom du répertoire, et non pas le chemin d'accès complet. Dans l'exemple précédent, la commande `cd veggies` utilise le chemin d'accès relatif du répertoire `veggies`. Si le répertoire de travail courant est `/home/user2`, le chemin d'accès complet de ce répertoire est `/home/user2/veggies`.

Essayez de créer plusieurs sous-répertoires dans un même répertoire puis déplacez-vous dans ce répertoire. Utilisez les chemins d'accès complets et les chemins d'accès relatifs, puis confirmez votre emplacement à l'aide de la commande `pwd`.

Déplacement et changement de nom des répertoires

Vous renommez un répertoire en le déplaçant vers un répertoire de nom différent. Pour renommer un répertoire, utilisez la commande `mv` :

```
$ pwd
/home/user2/veggies
$ ls
broccoli
$ mv broccoli carrots
$ ls
carrots
```

Vous pouvez également utiliser la commande `mv` pour déplacer un répertoire vers un autre :

```
$ pwd
/home/user2/veggies
$ ls
carrots
$ mv carrots ../veggies2
$ ls ../veggies2
carrots
```

Dans l'exemple ci-dessus, le répertoire `carrots` a été déplacé du répertoire `veggies` vers le répertoire `veggies2` à l'aide de la commande `mv`.

Copie de répertoires

Utilisez la commande `cp -r` pour copier des répertoires et les fichiers qu'ils contiennent :

```
$ cp -r veggies veggies3
$
```


Cette commande copie l'ensemble des fichiers et des sous-répertoires du répertoire `veggies` dans un nouveau répertoire intitulé `veggies3`. Cette copie est une copie *récurrente*, comme l'indique l'option `-r`. Si vous essayez de copier un répertoire sans utiliser cette option, un message d'erreur apparaît.

Suppression de répertoires (`rmdir`)

Pour supprimer un répertoire vide, utilisez la commande `rmdir` de la façon suivante :

```
$ rmdir veggies3
$
```

Si le répertoire que vous souhaitez supprimer contient encore des fichiers ou des sous-répertoires, la commande `rmdir` ne supprimera pas ce répertoire.

Utilisez la commande `rm -r` (en ajoutant l'option *récurrente* `-r` à la commande `rm`) pour supprimer tout le contenu d'un répertoire, y compris les sous-répertoires et leurs fichiers, en procédant de la façon suivante :

```
$ rm -r veggies3
$
```



Attention : les répertoires qui ont été supprimés avec la commande `rmdir` *ne pourront pas* être restaurés, *ni* ceux qui ont été supprimés, avec leur contenu, par la commande `rm -r`.

Recherche de différences entre fichiers (`diff`)

Souvent, plusieurs utilisateurs ayant accès à un fichier effectuent des copies de ce fichier en y apportant des modifications. La commande `diff` vous indique les différences rencontrées entre les versions d'un fichier ASCII. La commande suivante :

```
$ diff leftfile rightfile
```

balaie chaque ligne des fichiers `leftfile` et `rightfile` en recherchant les différences éventuelles. Lorsqu'elle rencontre une ligne (ou des lignes) différente(s), elle détermine si la différence provient de l'ajout, de la suppression ou de la modification d'une ligne, et indique le nombre de lignes concernées, ainsi que leur numéro respectif dans chacun des deux fichiers, suivi du texte correspondant.

Si la différence est due à un ajout, la commande `diff` affiche une ligne sous la forme suivante :

$$l[,l] \text{ a } r[,r]$$

où l désigne un numéro de ligne du fichier `leftfile` et r un numéro de ligne du fichier `rightfile`.

Si la différence est due à une suppression, la commande `diff` affiche la lettre `d` (delete) au lieu de la lettre `a` ; s'il s'agit d'une modification de ligne, la commande `diff` affiche la lettre `c` (change).

Le texte figurant sur les lignes respectives des deux fichiers suit immédiatement le numéro de ligne. Le texte du fichier `leftfile` est précédé d'un chevron gauche (`<`). Le texte du fichier `rightfile` est précédé d'un chevron droit (`>`).

L'exemple ci-dessous fait apparaître deux fichiers contenant les différences trouvées par la commande `diff` :

```
$ cat progr.15.7.
Semaine du 15/7

Jour:  Heure:      Action :           Détails:

M      10:00       Gest.matériel    une semaine sur deux
M      1:30        Gest.logiciels
J      3:00        Gest.doc
V      1:00        Entretien

$ cat progr.22.7
Semaine du 22/7

Jour:  Heure:      Action :           Détails:

L      8:30        Gest.personnel   tous les jours
M      10:00       Gest.matériel    une semaine sur deux
M      1:30        Gest.logiciels
J      3:00        Gest.doc

$ diff progr.15.7 progr.22.7
1c1
< Semaine du 15/7
---
> Semaine du 22/7
4a5
> L      8:30        Gest. personnel   tous les jours
8d8
< V      1:00        Entretien
```

Si les deux fichiers à comparer sont identiques, la commande `diff` n'affiche rien.

Cette section ne présente que certaines des options de la commande `diff`. Pour plus d'informations, consultez les *man Pages(1): User Commands*.

Comparaison de trois fichiers différents (`diff3`)

Si vous souhaitez comparer simultanément trois versions différentes d'un même fichier, utilisez la commande `diff3` de la façon suivante :

```
$ diff3 fichier1 fichier2 fichier3
```

`diff3` compare les trois versions d'un même fichier et affiche les parties de texte différentes, en utilisant les codes suivants :

==== les trois fichiers sont différents

====1 *file1* est différent

====2 *file2* est différent

====3 *file3* est différent

Utilisation de `bdiff` sur des fichiers volumineux

Si vous souhaitez comparer des fichiers très volumineux, utilisez la commande `bdiff` au lieu de `diff`. Son fonctionnement est similaire à celui de `diff` :

```
$ bdiff leftfile rightfile
```

Utilisez `bdiff` au lieu de `diff` pour les fichiers dépassant 3500 lignes.

Recherche de fichiers (`find`)

La commande `find` recherche les fichiers répondant aux critères précisés, en partant du répertoire indiqué. Par exemple, vous pouvez rechercher les noms des fichiers dont le nom correspond à un modèle particulier, ou ceux qui ont été modifiés au cours d'une période donnée.

Contrairement à la plupart des commandes, les options de la commande `find` comportent plusieurs caractères ; le nom du répertoire de départ doit précéder l'intitulé de ces options, comme indiqué sur la ligne de commande suivante :

```
$ find repertoire options
```

où *répertoire* désigne le nom du répertoire de départ, et *options* désigne les options relatives à la commande `find`.

Chaque option décrit un critère de sélection de fichier. Pour être sélectionné, un fichier doit répondre à l'ensemble des critères indiqués. Ainsi, plus le nombre d'options utilisées est grand, plus la recherche est affinée. L'option `-print` permet d'afficher les résultats de la recherche. (Comme expliqué ultérieurement, vous pouvez utiliser la commande `find` pour exécuter des commandes. Dans ce cas, vous pouvez lui demander de ne pas afficher les fichiers sélectionnés, pour éviter la répétition.)

L'option `-name nom_fichier` demande à la commande `find` de sélectionner les fichiers qui correspondent à *nom_fichier*. Ici, *nom_fichier* est le composant le plus à droite du chemin d'accès complet d'un fichier. A titre d'exemple, le composant le plus à droite du chemin d'accès du fichier `/usr/lib/calendar` est `calendar`. Cette partie du nom d'un fichier est souvent appelée *nom principal*.

Pour afficher la liste des fichiers se terminant par `s` dans le répertoire courant et dans ses sous-répertoires, tapez :

```
$ find . -name '*s' -print
./programs
./programs/graphics
./programs/graphics/gks
./src/gks
$
```

Il existe d'autres options, telles que :

`-name nom_fichier`

Sélectionne les fichiers dont le composant le plus à droite correspond à *nom_fichier*. Placez *nom_fichier* entre apostrophes s'il comporte des paramètres de substitution de nom de fichier.

`-user ID_utilisateur`

Sélectionne les fichiers appartenant à *ID_utilisateur*. *ID_utilisateur* pouvant être un nom de connexion ou un numéro d'ID utilisateur.

`-group groupe`

Sélectionne les fichiers appartenant à *groupe*.

`-mtime n`

Sélectionne les fichiers qui ont été modifiés dans une période de *n* jours.

`-newer fichier_contrôle`

Sélectionne les fichiers dont la modification est plus récente que celle de *fichier_contrôle*.

Vous pouvez indiquer un ordre de priorité en groupant les options entre parenthèses (despécialisées) (par exemple, `\(options\)`). A l'intérieur de parenthèses despécialisées, vous pouvez utiliser l'indicateur `-o` entre les options pour que la commande `find` sélectionne les fichiers répondant à l'une des catégories uniquement, et non aux deux :

```
$ find . \( -name AAA -o -name BBB \) -print
./AAA
./BBB
```

Vous pouvez inverser le sens de l'option en la faisant précéder d'un point d'exclamation despécialisé. Dans ce cas, la commande `find` sélectionne les fichiers pour lesquels l'option *ne* s'applique *pas* :

```
$ find . \!-name BBB -print
./AAA
```

Vous pouvez également utiliser la commande `find` pour appliquer des commandes au fichier que celle-ci a sélectionné, en utilisant l'option :

`-exec commande '{ }' \;`

Cette option se termine par un point virgule despécialisé (`\;`). Les accolades entre apostrophes sont remplacées par les noms des fichiers sélectionnés par la commande `find`.

A titre d'exemple, vous pouvez utiliser la commande `find` pour supprimer automatiquement les fichiers de travail temporaires. Si vous nommez vos répertoires de travail de manière cohérente, vous pouvez utiliser la commande

`find` pour les rechercher et les détruire, où qu'ils se trouvent. Par exemple, si vous nommez vos répertoires de travail `junk` ou `dummy`, cette commande les trouvera, puis les supprimera :

```
$ find . \( -name junk -o -name dummy \) -exec rm '{}' \;
```

Pour plus d'informations, reportez-vous aux *man Pages(1): User Commands*.

Droits d'accès aux fichiers et aux répertoires

Remarque : lisez attentivement cette section. Une bonne compréhension des droits d'accès aux fichiers est souvent importante dans votre travail quotidien.

Les droits d'accès aux fichiers permettent de protéger les fichiers et les répertoires contre toute opération de lecture ou d'écriture non autorisée. Dans la plupart des cas, d'autres utilisateurs pourront lire vos fichiers, mais ne pourront pas les modifier. Dans d'autres cas, vous pourrez être amené à partager des fichiers exécutables (des programmes) avec d'autres utilisateurs. Les droits d'accès aux fichiers vous permettent de contrôler les conditions d'accès à vos fichiers.

Les principaux types de droits d'accès aux fichiers et aux répertoires sont les suivants :

- `r` – droit de *lecture* (read) . La lecture du fichier doit être possible pour que vous puissiez le consulter ou le copier. La lecture d'un répertoire doit être possible pour que vous puissiez établir une liste de son contenu.
- `w` – droit d'*écriture* (write). L'écriture du fichier doit être possible pour que vous puissiez le modifier, le supprimer ou le renommer. La lecture d'un répertoire doit être possible pour que vous puissiez y ajouter ou y supprimer des fichiers.
- `x` – droit d'*exécution* (execute). Un fichier possédant un droit d'exécution est un fichier que vous pouvez exécuter, tel qu'un programme. Un répertoire doit être exécutable pour que vous puissiez accéder à l'ensemble de ses sous-répertoires.

Il existe trois catégories d'utilisateurs pour lesquels des droits d'accès peuvent être définis :

- Propriétaire : l'utilisateur
- Groupe : les autres personnes appartenant au même groupe que l'utilisateur (par exemple, tous les utilisateurs de la comptabilité). Les groupes sont établis et mis à jour par l'administrateur système.
- Autres : toute autre personne

Affichage des droits d'accès et des statuts (ls -l)

Vous avez déjà utilisé la commande `ls` pour établir une liste des fichiers. La commande `ls` possède de nombreuses options. Utilisez l'option `-l` pour afficher une liste *détaillée*. Les fichiers et les répertoires sont répertoriés par ordre alphabétique. La figure 3-2 illustre la méthode d'affichage des fichiers par ordre alphabétique.

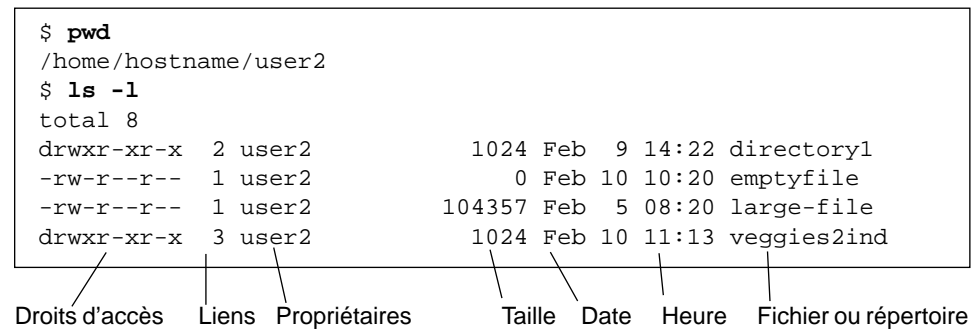


Figure 3-2 Affichage des attributs des droits d'accès et des statuts

Le premier caractère de la ligne indique le type du fichier. Le tiret (-) représente un fichier ordinaire, la lettre `d` un répertoire (directory) et les autres caractères, des types de fichiers spéciaux.

Les neuf caractères suivants indiquent les attributs des droits d'accès au fichier ou au répertoire correspondant. Ces neuf caractères sont formés de trois groupes de trois caractères, indiquant respectivement les attributs des droits d'accès du propriétaire, du groupe du propriétaire et des autres (ni le propriétaire, ni les membres de son groupe). Les attributs des droits d'accès au fichier `emptyfile` sont `rw-r--r--`. Ils indiquent que le propriétaire peut lire et écrire dans ce fichier, que tous les utilisateurs peuvent le lire, mais qu'aucun

utilisateur ne peut l'exécuter. Les attributs des droits d'accès au répertoire `veggies2` sont `rw-r-xr-x`. Ils indiquent que tous les utilisateurs ont un droit d'exécution et de lecture, mais que seul le propriétaire peut apporter des modifications.

Outre les droits d'accès au fichier, la figure ci-avant indique les informations suivantes :

- Nombre de liens vers ce fichier ou de ce répertoire
- Nom du propriétaire (en l'occurrence, `user2`)
- Nombre d'octets (caractères) du fichier
- Date et heure de la dernière mise à jour du fichier ou du répertoire
- Nom du fichier ou du répertoire

Utilisez la commande `cd` pour accéder à votre répertoire utilisateur, puis mettez en pratique la commande `ls -l`. Vos résultats seront bien entendu différents de ceux de l'exemple ci-dessus.

Essayez maintenant de taper la commande suivante :

```
$ ls -l nom_répertoire
```

où *nom_répertoire* désigne le nom d'un répertoire réel dans votre système de fichiers. Lorsque vous indiquez le nom d'un répertoire, la commande `ls -l` affiche des informations sur l'ensemble des fichiers ou des sous-répertoires (éventuels) qui se trouvent dans ce répertoire.

Etablissement d'une liste de fichiers "cachés" (`ls -a`)

Certains fichiers ne peuvent pas être répertoriés par la commande ordinaire `ls`. Le nom de ces fichiers commence par le caractère `.` (point), comme par exemple `.cshrc`, `.login` et `.profile`. Utilisez la commande `ls -a` pour établir une liste de ces fichiers point :

```
$ ls -a
.
..
.cshrc
.login
.profile
emptyfile
```

Remarquez que les fichiers commençant par un point (.) sont affichés en premier. Il existe deux fichiers spéciaux dans cette liste : le fichier . qui représente le répertoire courant et le fichier . . qui représente le répertoire père.

En règle générale, les fichiers commençant par un point (.) sont utilisés par les utilitaires du système et n'ont pas à être modifiés par l'utilisateur. Il existe cependant des exceptions.

Modification des droits d'accès (chmod)

Utilisez la commande `chmod` pour modifier les droits d'accès à un fichier ou à un répertoire. Pour pouvoir effectuer cette opération, vous devez être le propriétaire d'un fichier ou d'un répertoire, ou être superutilisateur. La commande `chmod` se présente sous la forme suivante :

```
chmod droits_d'accès nom
```

où *droits_d'accès* désigne les droits d'accès à modifier et *nom*, le nom du fichier ou du répertoire sur lequel portera la modification.

Les droits d'accès peuvent être indiqués sous différentes formes. La forme suivante est l'une des plus simples :

1. Utilisez une ou plusieurs lettres pour indiquer le type d'utilisateur auquel les droits d'accès s'appliquent :
 - u (pour *utilisateur*)
 - g (pour *groupe*)
 - o (pour *autres utilisateurs*)
 - a (pour *toutes les catégories ci-dessus*)
2. Indiquez si les droits d'accès doivent être ajoutés (+) ou supprimés (-).
3. Utilisez une ou plusieurs lettres pour indiquer les droits d'accès :
 - r (pour *lire*)
 - w (pour *écrire*)
 - x (pour *exécuter*)

Dans l'exemple suivant, le droit d'écriture est ajouté au répertoire `carrots` pour les utilisateurs appartenant au même groupe (ainsi, les attributs du *droit d'accès* sont `g+w` et *name* est le nom du répertoire `carrots`) :

```
$ ls -l carrots
drwxr-xr-x  3 user2          1024 Feb 10  11:15 carrots
$ chmod g+w carrots
$ ls -l carrots
drwxrwxr-x  3 user2          1024 Feb 10  11:15 carrots
$
```

Comme vous pouvez le voir, le tiret (-) du jeu de caractères correspondant au groupe d'utilisateurs a été remplacé par la lettre `w` suite à l'exécution de cette commande.

Pour éviter que les utilisateurs appartenant à un groupe différent du vôtre puissent lire et exécuter ce répertoire (les attributs du *droit d'accès* sont `o-rx`), vous devez entrer les commandes suivantes :

```
$ ls -l carrots
drwxrwxr-x  3 user2          1024 Feb 10  11:15 carrots
$ chmod o-rx carrots
$ ls -l carrots
drwxrwx---  3 user2          1024 Feb 10  11:15 carrots
$
```

Les lettres `r` (pour lire) et `x` (pour exécuter) du jeu de caractères correspondant aux autres utilisateurs ont été remplacées par un tiret (-).

Lorsque vous créez un nouveau fichier ou un nouveau répertoire, les droits d'accès sont affectés automatiquement par le système.

En règle générale, les paramètres d'affectation par défaut des nouveaux fichiers sont :

```
-rw-r--r--
```

ceux des nouveaux répertoires sont :

```
drwxr-xr-x
```

Par conséquent, pour que le nouveau fichier `turnip` soit exécutable par son propriétaire (`user2`), vous devez entrer les commandes suivantes :

```
$ ls -l turnip
-rw-r--r--  3 user2          1024 Feb 10  12:27 turnip
$ chmod u+x turnip
$ ls -l turnip
-rwxr--r--  3 user2          1024 Feb 10  12:27 turnip
$
```

Si vous souhaitez affecter simultanément les trois catégories d'utilisateurs, utilisez l'option `a`. Pour que le nouveau fichier `garlic` soit exécutable par l'ensemble des utilisateurs, vous devez entrer les commandes suivantes :

```
$ ls -l garlic
-rw-r--r--  3 user2          1024 Feb 10  11:31 garlic
$ chmod a+x garlic
$ ls -l garlic
-rwxr-xr-x  3 user2          1024 Feb 10  11:31 garlic
$
```

L'indicateur `x` apparaît alors dans les trois catégories.

Vous pouvez également modifier les attributs des droits d'accès à des groupes de fichiers ou à des répertoires, en utilisant le caractère générique *. Par exemple, pour changer les attributs des droits d'accès à l'ensemble des fichiers du répertoire courant `veggies`, afin que vous seul puissiez écrire dans ces fichiers, vous devez entrer les commandes suivantes :

```
$ pwd
/home/user2/veggies
$ ls -l
-rwxrwxrwx  3 user2          21032 Feb 12  10:31 beats
-rwxrwxrwx  2 user2           68 Feb 10  11:09 corn
-rwxrwxrwx  3 user2       12675 Feb 08  09:31 garlic
-rwxrwxrwx  1 user2         1024 Feb 14  16:38 onions
$ chmod go-w *
$ ls -l
-rwxr-xr-x  3 user2          21032 Feb 12  10:31 beats
-rwxr-xr-x  2 user2           68 Feb 10  11:09 corn
-rwxr-xr-x  3 user2       12675 Feb 08  09:31 garlic
-rwxr-xr-x  1 user2         1024 Feb 14  16:38 onions
$
```

Dans l'exemple ci-dessus, la commande `pwd` indique que le répertoire auquel vous appliquez la commande `chmod` est le répertoire courant.

Définition de droits d'accès absolus

Jusqu'ici, les explications relatives au droit d'accès se sont limitées à l'utilisation de la commande `chmod` en vue de modifier les attributs de droits d'accès *relativement* à leurs valeurs par défaut. L'utilisation de la commande `chmod` sous une forme différente, par application de codes numériques indiquant les droits d'accès, vous permet de définir de manière *absolue* un droit d'accès à un fichier ou à un répertoire.

La syntaxe correspondante de la commande `chmod` est la suivante :

```
chmod code_numérique nom
```

où *code_numérique* désigne le code numérique et *nom* le nom du fichier ou du répertoire pour lequel le droit d'accès est modifié.

Le code numérique complet est composé de trois chiffres, utilisés pour chacune des trois catégories prédéfinies : utilisateur, groupe et autres. Par exemple, la commande suivante définit un droit de lecture, d'écriture et d'exécution absolu pour les catégories utilisateur et groupe, et uniquement un droit d'exécution pour la catégorie autres :

```
$ chmod 771 garlic
```

Le Tableau 3-1 montre comment le code 771 est utilisé pour représenter les droit d'accès au fichier `garlic`.

Tableau 3-1 Droits d'accès au fichier `garlic`

Droits d'accès	Utilisateur	Groupe	Autres
Lire	4	4	0
Ecrire	2	2	0
Exécuter	1	1	1
Total	7	7	1

Chaque colonne du Tableau 3-1 représente l'une des catégories prédéfinies : utilisateur, groupe et autres. Pour définir un droit de lecture, insérez le chiffre 4 dans la colonne prévue à cet effet. Pour définir un droit d'écriture, insérez 2. Pour définir un droit d'exécution, insérez 1. Le total des trois colonnes figurant dans la dernière rangée du tableau représente le code numérique complet.

L'écran suivant, dans lequel la commande `ls -l` indique les résultats obtenus, constitue un autre exemple de définition d'un droit d'accès absolu :

```
$ ls -l onion
-rw-r--r-- 3 user2          1024 Feb 10  11:46 onion
$ chmod 755 onion
$ ls -l onion
-rwxr-xr-x 3 user2          1024 Feb 10  11:48 onion
$
```

Pour le fichier `onion`, les droits d'accès sont définis de telle manière que l'utilisateur puisse lire, écrire, et exécuter ce fichier ; les membres de la catégorie groupe peuvent le lire et l'exécuter, de même que les membres de la catégorie autres. Le code numérique utilisé pour définir les droits d'accès au fichier `onion` est défini dans le Tableau 3-2.

Tableau 3-2 Droits d'accès au fichier `onion`

Droits d'accès	Utilisateur	Groupe	Autres
Lire	4	4	4
Ecrire	2	0	0
Exécuter	1	1	1
Total	7	5	5

Pour attribuer un droit de lecture, d'écriture et d'exécution du fichier `cabbage` aux trois catégories du tableau ci-dessus, vous devez bien entendu entrer les commandes suivantes :

```
$ ls -l cabbage
-rw-r--r-- 3 user2          1024 Feb 10 11:51 cabbage
$ chmod 777 cabbage
$ ls -l cabbage
-rwxrwxrwx 3 user2          1024 Feb 10 11:53 cabbage
$
```

Cet exemple est décomposé dans le Tableau 3-3.

Tableau 3-3 Droits d'accès au fichier `cabbage`

Droits d'accès	Utilisateur	Groupe	Autres
Lire	4	4	4
Ecrire	2	2	2
Exécuter	1	1	1
Total	7	7	7

Le code numérique `777` représente le niveau maximal de droits d'accès que vous pouvez définir.

Comme pour changer un droit d'accès relatif, vous pouvez utiliser le caractère générique `*` afin de définir un droit d'accès absolu à l'ensemble des fichiers du répertoire courant. Imaginons que vous souhaitiez définir un droit d'accès absolu à l'ensemble des fichiers du répertoire courant `veggies` : si vous entrez les commandes suivantes, vous aurez un droit de lecture, d'écriture, et d'exécution, votre groupe aura un droit de lecture et d'exécution, et tous les autres utilisateurs auront uniquement un droit d'exécution.

```
$ pwd
/home/user2/veggies
$ ls -l
-rwxrwxrwx 3 user2      21032 Feb 12 10:31 beats
-rwxrwxrwx 2 user2        68 Feb 10 11:09 corn
-rwxrwxrwx 3 user2    12675 Feb 08 09:31 garlic
-rwxrwxrwx 1 user2     1024 Feb 14 16:38 onions
$ chmod 751 *
$ ls -l
-rwxr-x--x 3 user2      21032 Feb 12 10:31 beats
-rwxr-x--x 2 user2        68 Feb 10 11:09 corn
-rwxr-x--x 3 user2    12675 Feb 08 09:31 garlic
-rwxr-x--x 1 user2     1024 Feb 14 16:38 onions
$
```

Dans l'exemple ci-dessus, la commande `pwd` indique que le répertoire sur lequel vous effectuez cette procédure est le répertoire courant. La commande `ls -l` est affichée uniquement pour illustrer le changement des droits d'accès. Lors de la définition des droits d'accès absolus, il n'est pas nécessaire de connaître les types des droits d'accès en vigueur dans le système.

Pour plus d'informations sur la commande `chmod`, consultez les *man Pages(1): User Commands*.

Recherche dans des fichiers



Ce chapitre explique comment rechercher un mot clé ou une chaîne de caractères dans des répertoires ou des fichiers à l'aide de la commande `grep` sous SunOS.

Recherche d'un modèle à l'aide de la commande `grep`

La commande `grep` permet de rechercher une chaîne de caractères donnée dans un fichier. La syntaxe de base de la commande `grep` est la suivante :

```
$ grep chaîne fichier
```

où *chaîne* désigne le mot ou l'expression à rechercher et *fichier* le fichier dans lequel la recherche doit être effectuée.

Remarque : une *chaîne* est composée d'un caractère ou d'un ensemble de caractères ; il peut s'agir d'une lettre unique, d'un mot, ou d'une phrase. Les chaînes peuvent contenir des "caractères blancs", des signes de ponctuation ou des caractères (de contrôle) invisibles.

Par exemple, pour connaître le poste téléphonique d'Edgar Allan Poe, tapez `grep`, tout ou partie de son nom, puis le fichier contenant l'information :

```
$ grep Poe extensions
Edgar Allan Poe      x72836
$
```

Le modèle entré peut correspondre à plusieurs lignes :

```
$ grep Allan extensions
David Allan          x76438
Edgar Allan Poe      x72836
$ grep Al extensions
Louisa May Alcott    x74236
David Allan          x76438
Edgar Allan Poe      x72836
$
```

Etant donné que la commande `grep` distingue les majuscules des minuscules, il est recommandé de respecter la combinaison du modèle :

```
$ grep allan extensions
$ grep Allan extensions
David Allan          x76438
Edgar Allan Poe      x72836
$
```

La commande `grep` n'a pas abouti dans le premier cas car aucune entrée commençant par un "a" minuscule n'a été trouvée.

Utilisation de la commande `grep` en tant que filtre

La commande `grep` est souvent utilisée en tant que "filtre" conjointement à d'autres commandes. Elle permet d'éliminer par filtrage les informations inutiles fournies par une commande. Utiliser la commande `grep` en tant que filtre consiste à limiter le résultat de la commande à l'aide de `grep`. Pipe est symbolisé par "`|`".

Dans l'exemple ci-dessous, seuls les fichiers dont l'extension est ".ps" et qui ont été créés au mois de mai, sont affichés :

```
$ ls -l *.ps | grep May
```

La première partie de la commande,

```
ls -l *.ps
```

permet d'obtenir la liste de fichiers suivante :

```
$ ls -l *.ps
-rw-r--r-- 1 elvis      7228 Avr 22 15:07 change.ps
-rw-r--r-- 1 elvis     2356 May 22 12:56 clock.ps
-rw-r--r-- 1 elvis     1567 Jun 22 12:56 cmdtool.ps
-rw-r--r-- 1 elvis    10198 Jun 22 15:07 command.ps
-rw-r--r-- 1 elvis     5644 May 22 15:07 buttons.ps
$
```

La seconde partie

```
| grep May
```

limite la recherche au modèle May, à l'aide de la commande grep :

```
$ ls -l *.ps | grep May
-rw-r--r-- 1 elvis     2356 May 22 12:56 clock.ps
-rw-r--r-- 1 elvis     5644 May 22 15:07 buttons.ps
$
```

Utilisation de la commande `grep` pour des chaînes de plusieurs mots

Pour rechercher un modèle comportant plusieurs mots, vous devez limiter la chaîne par des guillemets simples ou doubles :

```
$ grep "Louisa May" postes
Louisa May Alcott      x74236
$
```

La commande `grep` peut rechercher une chaîne dans un ensemble de fichiers. Lorsqu'elle trouve un modèle correspondant dans plusieurs fichiers, elle indique le nom du fichier suivi de deux points (:), puis de la ligne correspondant au modèle :

```
$ grep est *
auteurs:Ernest Hemmingway
alaska:L'Alaska est le plus vaste Etat des Etats-Unis.
wilde:livre. Un livre est plus ou moins bien écrit.
$
```

Recherche de lignes ne comportant pas de chaîne donnée

Il est possible de rechercher dans un fichier toutes les lignes ne comportant *pas* de chaîne donnée à l'aide de l'option `-v` de la commande `grep`. L'exemple ci-dessous décrit la recherche, dans le répertoire utilisateur `medici`, de toutes les lignes ne comportant pas la lettre `e` :

```
$ ls
auteurs      alaska      hinterland  tuteurs     wilde
$ grep -v e *
auteurs:Mon Mar 14 10:00 PST 1936
wilde:That is all.
$
```

Autres possibilités de la commande `grep`

La commande `grep` permet également de rechercher des cibles prédéfinies en tant que modèles à l'aide d'*expressions régulières*. Ces dernières se composent de chiffres, de lettres et de caractères ayant une signification particulière pour la commande `grep`. Ces caractères spéciaux, appelés *métacaractères*, ont également une signification particulière pour le système et doivent être mis entre guillemets ou despécialisés. Lorsque vous utilisez une expression régulière `grep` sur une ligne de commande, vous devez l'entourer de guillemets ou despécialiser les métacaractères (tels que `&` `!` `.` `*` `$` `?` ou `\`) à l'aide d'une barre oblique inverse (`\`).

- Un caret (^) désigne le début d'une ligne. Ainsi, la commande :

```
$ grep '^b' list
```

permet d'afficher toutes les lignes du fichier `list` commençant par la lettre “b.”

- Un symbole dollar (\$) désigne la fin d'une ligne. La commande :

```
$ grep 'b$' list
```

permet d'afficher les lignes se terminant par la lettre “b”. Et la commande :

```
$ grep '^b$' list
```

permet d'afficher les lignes du fichier `list` dont le *seul* caractère est “b”.

- Un point (.) situé à l'intérieur d'une expression régulière permet de rechercher tout caractère unique. Ainsi, la commande :

```
$ grep 'an.' list
```

permet de trouver tous les ensembles de trois caractères commençant par “an”, y compris “rang”, “année”, “rangement” ou “plan” (car les espaces sont également pris en compte).

- Lorsqu'un astérisque (*) est placé à la suite d'un caractère, il est interprété par la commande `grep` comme "zéro, une, ou plusieurs occurrences de ce caractère." Lorsque l'astérisque est placé à la suite d'une expression régulière, il est interprété comme "zéro, une, ou plusieurs occurrences des caractères correspondant au modèle."

Etant donné que l'astérisque comprend zéro occurrence, son utilisation n'est pas très intuitive. En effet, pour obtenir la liste des mots contenant les lettres "qu", vous pouvez utiliser la commande :

```
$ grep 'qu*' list
```

Mais, si vous recherchez tous les mots contenant la lettre "n," vous devez taper :

```
$ grep 'nn*' list
```

Alors que pour trouver les mots contenant le modèle "nn," il vous faudrait taper :

```
$ grep 'nnn*' list
```

Vous pouvez lancer ces commandes pour juger du résultat obtenu.

- Pour trouver zéro, une, ou plusieurs occurrences correspondant à *n'importe quel* caractère du fichier `list`, tapez :

```
$ grep .* list
```

Recherche de métacaractères

Imaginons que vous recherchiez les lignes du texte contenant un symbole dollar (\$). Si vous faites précéder le symbole dollar d'une barre oblique inverse (\) dans l'expression régulière, la commande `grep` ne tiendra pas compte de la signification particulière de (despécialisera) ce symbole. Cela est valable également pour les autres métacaractères (& ! . * ? et même \).

Par exemple, la commande

```
$ grep ^\.
```

permet de rechercher les lignes commençant par un point. Elle est particulièrement utile pour la recherche des demandes de formatage des fichiers `nroff` ou `troff` (qui commencent par un point).

Le Tableau 4-1 présente une liste des modèles de recherche les plus fréquemment utilisés avec la commande `grep`.

Tableau 4-1 Modèles de recherche de la commande `grep`

Caractère	Recherche
<code>^</code>	Début d'une ligne de texte
<code>\$</code>	Fin d'une ligne de texte
<code>.</code>	Tout caractère unique
<code>[. . .]</code>	Tout caractère unique dans la liste ou l'intervalle entre crochets
<code>[^ . . .]</code>	Tout caractère non compris dans la liste ou l'intervalle
<code>*</code>	Zéro, une ou plusieurs occurrences du caractère ou de l'expression régulière précédent(e)
<code>.*</code>	Zéro, une ou plusieurs occurrences de tout caractère unique
<code>\</code>	Supprime la signification particulière du caractère suivant

Ces caractères de recherche peuvent également être utilisés pour les recherches dans l'éditeur de texte `vi`.

Guillemets simples ou doubles dans les lignes de commande

Comme indiqué précédemment, un texte doit être placé entre guillemets pour être interprété en tant que mot unique. Par exemple, pour rechercher la phrase “On s’en fiche, les gars” dans tous les fichiers à l’aide de la commande `grep`, tapez :

```
$ grep "On s'en fiche, les gars" *
```

Une phrase de plusieurs mots peut aussi être unifiée à l’aide de guillemets simples ou apostrophes ('). Les apostrophes permettent également d’indiquer que certains caractères, tels que \$, doivent être interprétés littéralement. (Le métacaractère de la commande `history` ! est toujours interprété en tant que tel, même placé entre guillemets, sauf si vous le despécialisez à l’aide d’une barre oblique inverse.). Quoi qu’il en soit, il est préférable de despécialiser les caractères tels que & ! \$? . ; et \ chaque fois que vous les utilisez en tant que caractères typographiques ordinaires.

Par exemple, si vous tapez :

```
$ grep $ list
```

vous verrez apparaître *toutes* les lignes du fichier `list`, alors que si vous tapez :

```
$ grep '\$' list
```

seules les lignes contenant le caractère “\$” seront affichées.

Pour plus d’informations sur la commande `grep`, consultez les *man Pages(1): User Commands*.

Mots de passe, processus et mémoire disque

5 

SunOS offre une quantité de commandes permettant d'effectuer des tâches système à partir de la ligne de commande. Ce chapitre explique comment choisir un mot de passe, établir une liste des processus en cours sur votre machine, arrêter des processus indésirables et connaître la quantité d'espace disque utilisée.

Utilisation d'un mot de passe

Afin de garantir la sécurité de votre système, SunOS préconise l'utilisation d'un mot de passe. Pour être certain que vous seul pouvez bénéficier de l'accès à votre compte, il est recommandé de changer de mot de passe plusieurs fois au cours de l'année. Si vous soupçonnez quelqu'un de s'être servi de votre compte sans votre accord, changez immédiatement de mot de passe.

Quelques principes sont à considérer lors du choix d'un mot de passe, à savoir:

- Choisissez un mot de passe que vous pourrez retenir sans avoir besoin de le noter. Mieux vaut qu'il soit facile à deviner que facile à oublier.
- Choisissez un mot de passe comportant un minimum de six caractères, dont un au moins devra être un chiffre.
- Ne choisissez pas votre nom (ou vos initiales), ni celui de votre conjoint.
- Ne choisissez pas non plus des noms d'animaux ou d'objets auxquels vous vous intéressez.
- N'utilisez pas uniquement des majuscules.

- Si vous possédez plusieurs comptes, utilisez un mot de passe différent pour chacun d'eux.
- Tous les caractères sont autorisés. Toutefois, certains caractères, tels que Ctrl-C, Ctrl-Z, Ctrl-U, Ctrl-S, Esc, Tab et dans certains cas # et @ peuvent être significatifs pour le terminal. Evitez donc de les utiliser car le terminal pourrait les interpréter non pas comme des caractères de texte, mais comme des signaux, et cela vous empêcherait d'entrer correctement votre mot de passe.

Modification du mot de passe

Pour modifier votre mot de passe, entrez la commande `passwd` :

```
$ passwd
Modification du mot de passe pour hankw sur worker en cours...
Old password:
New password:
Entrez votre nouveau mot de passe une nouvelle fois :
$
```

1. **Lorsque le système vous demande d'indiquer votre ancien mot de passe en regard du prompt `Old Password:`, entrez le mot de passe que vous utilisez actuellement.**
(Si aucun mot de passe n'est associé à votre compte, le système sautera l'invite `Old Password:`.) Le système ne renvoie pas (n'affiche pas) votre mot de passe à l'écran, afin que personne ne puisse le découvrir.
2. **Lorsque le système vous demande d'indiquer votre nouveau mot de passe en regard du prompt `New Password:`, entrez le mot de passe que vous avez choisi d'utiliser.**
De même, le mot de passe que vous entrez n'est pas renvoyé à l'écran.
3. **Au dernier prompt, `Retype new password:`, entrez votre nouveau mot de passe une deuxième fois.**
Cela permet de vérifier si vous avez bien entré ce que vous vouliez.

Si vous n'entrez pas exactement le même mot de passe que celui du prompt précédent, le système rejette la demande de modification du mot de passe et vous répond `Désolé`. Si cela se produit fréquemment, demandez à votre administrateur système de vous attribuer un nouveau mot de passe.

Remarque : les mots de passe contenant moins de six caractères ne sont pas autorisés. En outre, le nouveau mot de passe doit différer d'au moins trois caractères de l'ancien mot de passe.

Expiration du mot de passe

Si votre système dispose de l'option d'expiration du mot de passe (fournie avec les options de la commande `passwd`), votre mot de passe aura une durée de validité maximale ou bien maximale *et* minimale. Cette durée de validité est fixée par votre administrateur système.

Lorsque la date de péremption (ou la durée de validité maximale) de votre mot de passe est atteinte, vous êtes invité à changer de mot de passe. Vous en êtes informé au moment de l'ouverture de session, lorsque le message suivant apparaît :

```
Your password has expired. Choose a new one.
```

Le système lance alors automatiquement le programme `passwd` et vous invite à entrer un nouveau mot de passe.

Si, par exemple, la durée de validité *minimale* de votre mot de passe a été fixée à deux semaines et que vous tentez de modifier votre mot de passe avant que cette durée ne soit écoulée, le message suivant apparaît :

```
Sorry, less than 2 weeks since the last change.
```

L'option `-d` de la commande `passwd` vous permet de consulter les informations relatives à l'expiration de votre mot de passe :

```
$ passwd -d
username 2-14-92 14 60
```

L'écran affiche, dans l'ordre, la date de création du mot de passe en cours, sa durée de validité minimale et sa durée de validité maximale. (Ces informations n'apparaissent que dans l'hypothèse où l'option d'expiration du mot de passe est utilisée.)

Pour plus d'informations sur l'utilisation des mots de passe et de leur expiration, consultez les *man Pages(1): User Commands*.

Processus et PID

Lorsque le système a interprété une commande, un *processus* indépendant associé à un identificateur de processus (PID) unique, est créé pour l'exécution de cette commande. Le PID permet au système de suivre l'état en cours de chaque processus.

Commandes en cours d'exécution (*ps*)

La commande *ps* permet d'afficher une liste des processus en cours. Outre l'*identificateur de processus* (répertorié sous *PID*) pour chaque processus que vous possédez (que vous avez créé en entrant une commande), la commande *ps* indique également le *terminal* à partir duquel ce processus a été lancé (*TTY*), le *temps d'unité centrale* que ce dernier a utilisé jusqu'à présent (*TIME*) et la *commande* qu'il exécute (*COMMAND*).

Si vous précisez l'option *-l*, la commande *ps* affiche d'autres informations sur les processus en cours, telles que le *statut* de chaque processus (répertorié sous *S*). Les codes utilisés pour l'affichage de ces informations sont les suivants :

- O - Le processus est en cours sur un processeur.
- S - Sommeil : le processus est en attente de fin d'exécution d'un événement.
- R - Exécutable : le processus est dans la file d'attente d'exécution.
- I - Inactif : le processus est en cours de création.
- Z - Zombie : le processus est terminé et le père n'est pas en attente.
- T - Analyse : le processus a été arrêté par un signal car il est en cours d'analyse par le père.
- X - SXBRK : le processus attend davantage de mémoire principale.

Des modifications peuvent se produire au cours de l'exécution de la commande *ps*. Cette commande ne fournissant qu'un cliché de l'état en cours, les informations ne sont valables qu'un court instant après son lancement. Il se peut qu'elles soient devenues radicalement différentes au moment où vous en prenez connaissance.

La commande `ps` offre d'autres options que celles énumérées ci-dessus. Pour les connaître, consultez les *man Pages(1): User Commands*.

Interruption d'un processus (kill)

La commande `kill` permet d'interrompre immédiatement les processus de commande dont vous souhaitez vous débarrasser. Elle est particulièrement utile lorsque vous avez lancé par erreur une commande dont l'exécution prend du temps.

Pour interrompre un processus, procédez de la façon suivante :

1. Tapez `ps` pour obtenir la liste des PID du ou des processus.
2. Tapez `kill`, suivi du ou des PID.

Cette procédure est illustrée dans l'exemple suivant :

```
$ ps
PID    TTY    TIME    COMMAND
1291    co     0:12    -bin/csh (csh)
3250    p0     0:00    ps
1286    p1     0:05    -bin/csh (csh)
3248    p1     0:05    vi commands
$ kill 1291
[1]  Terminated      -bin/csh/ (csh)
$
```

Pour connaître encore plus rapidement le PID, limitez le résultat de la commande `ps` en exécutant la commande `grep` de la manière suivante :

```
$ ps | grep nom_commande
```

où *nom_commande* désigne le nom du processus de commande que vous souhaitez interrompre. (Pour l'utilisation de la commande `grep`, reportez-vous au Chapitre 4, "Recherche dans des fichiers.")

Si vous devez absolument interrompre un processus, vous pouvez préciser l'option `-9` dans la commande `ps`, comme illustré ci-dessous :

```
$ kill -9 n.PID
```

où *n.PID* désigne l'identificateur du processus que vous souhaitez interrompre.

Gestion de la mémoire disque

L'espace disque étant limité, il est recommandé de vérifier régulièrement l'espace utilisé.

Affichage de l'utilisation du disque (`df -k`)

La commande `df -k` permet d'afficher la quantité d'espace utilisé sur chaque disque installé (directement accessible) sur votre système. Il vous suffit de taper :

```
$ df -k
```

pour voir apparaître la capacité de chaque disque installé sur votre système, la quantité disponible et le pourcentage d'espace utilisé.

Les systèmes de fichiers dont la capacité utilisée est supérieure ou égale à 90% doivent être débarrassés des fichiers inutiles. Pour effectuer cette opération, vous pouvez transférer ces fichiers sur un disque ou une bande moins saturée en utilisant la commande `cp` pour les copier, puis la commande `rm` pour les supprimer. Vous pouvez également les supprimer directement. Il va de soi que vous ne devez pratiquer ces séances de “ménage” que pour des fichiers qui vous appartiennent.

Affichage de l'utilisation d'un répertoire (`du`)

La commande `du` vous permet d'afficher l'utilisation d'un répertoire et de ses sous-répertoires en blocs de 512 octets ; c'est à dire en unités de 512 octets ou caractères.

La commande `du` affiche l'espace disque utilisé pour chaque sous-répertoire. Pour obtenir une liste des sous-répertoires d'un système de fichiers, lancez la commande `cd` pour le chemin associé à ce système de fichiers, puis exécutez le pipeline suivant :

```
$ du | sort -r -n
```

Ce pipeline, qui utilise les options *reverse* et *numeric* de la commande `sort`, permet la localisation des répertoires volumineux. Vous obtiendrez la taille (en octets) et la date de dernière modification des fichiers de chaque répertoire en utilisant la commande `ls -l`. Les fichiers anciens ou les fichiers de texte de plus de 100 Koctets aboutissent souvent à un dépassement de capacité de la mémoire.

`vi` (prononcé “vi-ā”, abréviation d’éditeur pleine page) est l’éditeur de texte standard de SunOS. Comme `vi` ne fonctionne pas en multi-fenêtrage, il peut être utilisé sur un terminal quelconque pour l’édition d’un grand nombre de formats de fichiers.

Sans être un traitement de texte à proprement parler car il n’est pas conçu pour traiter des textes formatés, cet éditeur permet de saisir et d’éditer un texte. Pour générer des impressions formatées, l’éditeur `vi` a recours à un programme d’émulation de composeur, tel que `nroff`, `troff` ou `ditroff`. Ce programme permet de formater un texte créé sous `vi` en insérant des codes qui sont ensuite interprétés par l’émulateur.

`vi` contient un grand nombre de commandes, dont la plupart ont des fonctions qui se recoupent. Ceci peut être assez déroutant pour les nouveaux utilisateurs. Toutefois, ce chapitre a pour objectif de vous offrir un aperçu des principales commandes `vi`. Lorsque vous commencerez à utiliser `vi`, vous vous apercevrez qu’il s’agit d’un éditeur de texte extrêmement performant, et qu’il vous faudra un certain temps avant d’être opérationnel.

Il existe une version de `vi` en lecture seule, appelée `view`. Lorsque vous ouvrez un fichier sous `view`, vous pouvez utiliser les commandes `vi`, mais vous ne pouvez pas écrire (ou sauvegarder) vos modifications. Cette version permet de consulter un fichier `vi` sans risquer de le modifier par erreur.

Lancement de vi

Les sous-sections suivantes vous expliquent comment démarrer vi, saisir du texte dans un fichier, sauvegarder (écrire) ce fichier et quitter vi. Vous allez également pouvoir créer un fichier d'entraînement qui vous servira pour la suite de ce chapitre.

Création d'un fichier

Démarrez vi, puis éditez le fichier `paint` comme indiqué dans cet exemple :

```
$ vi paint
```

Si le fichier `paint` existe déjà, vi ouvre le fichier existant ; s'il s'agit d'un nouveau fichier, vi le crée. Dans le but de l'exercice, `paint` doit être un nouveau fichier.

L'écran d'édition de vi suivant apparaît :

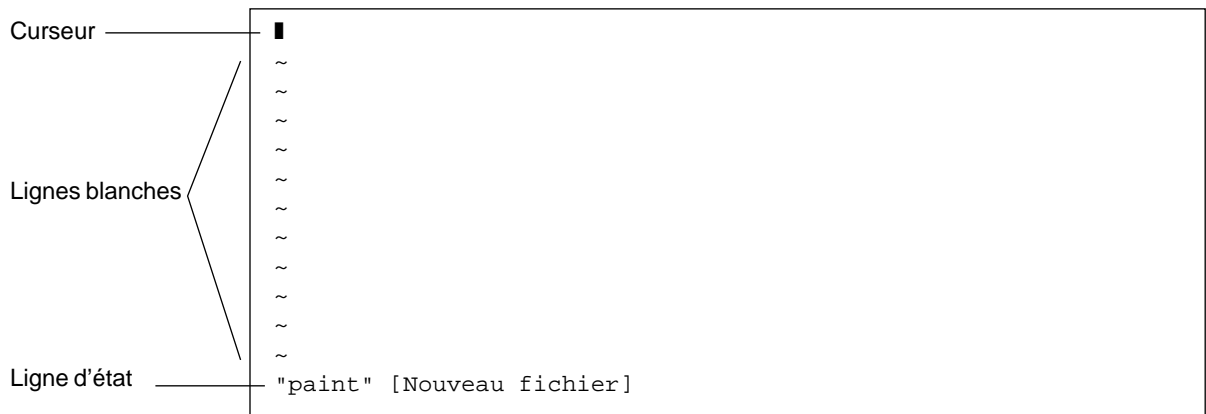


Figure 6-1 L'écran d'édition de vi

Le curseur apparaît dans le coin supérieur gauche de l'écran. Les lignes blanches sont désignées par une série de tildes (~) placés les uns en-dessous des autres.

Vous pouvez également démarrer `vi` sans préciser de nom de fichier, en entrant simplement `vi`. Vous pourrez alors nommer le fichier au moment où vous quitterez l'éditeur.

La ligne d'état

La dernière ligne de l'écran, appelée *ligne d'état*, contient le nom du fichier et le nombre de lignes et de caractères qu'il comporte. Lorsque vous créez un fichier, comme dans notre exemple, la ligne d'état signale qu'il s'agit d'un nouveau fichier.

Les deux modes de l'éditeur vi

L'éditeur `vi` offre deux modes de fonctionnement : le mode insertion et le mode commande. Le *mode insertion* permet de saisir du texte dans un fichier, alors que le *mode commande* permet d'entrer des commandes exécutant des fonctions `vi` particulières. Le mode commande est utilisé par défaut.

Comme `vi` n'indique pas le mode en cours, les nouveaux utilisateurs ont souvent du mal à opérer la distinction entre les deux modes. Mais, il vous suffit de retenir quelques concepts fondamentaux dès le départ pour éviter de connaître l'habituel "stress `vi`."

Lorsque vous ouvrez pour la première fois un fichier sous `vi`, vous êtes toujours en mode commande. Avant de pouvoir saisir dans ce fichier, vous devez taper l'une des deux commandes d'entrée `vi` suivantes : `i` ("insérer") pour insérer du texte à l'emplacement du curseur, ou `a` ("ajouter") pour insérer du texte *après* le curseur. (L'utilisation de ces commandes et d'autres commandes d'entrée `vi` est décrite plus en détail dans la suite de ce chapitre.)

Pour repasser en mode commande, appuyez sur `Esc`. Si vous ne savez plus dans quel mode vous vous trouvez, appuyez sur `Esc` pour vous assurer que vous êtes bien en mode commande, puis continuez. Si vous appuyez sur `Esc` alors que `vi` est déjà en mode commande, le système émet un signal sonore et l'écran clignote, mais cela n'a pas d'importance.

Mode insertion

Pour insérer du texte dans le fichier d'entraînement `paint`, entrez la commande d'insertion `i` de `vi`. `vi` quitte le mode commande et passe en mode insertion.

Entrez ensuite quelques courtes lignes de texte, en appuyant sur Return à la fin de chacune d'elles. Les caractères entrés apparaissent à gauche du curseur et repoussent les caractères existants vers la droite. Pour l'instant, vous pouvez corriger vos erreurs en effectuant un retour en arrière à l'aide de la touche Back Space et en retapant la ligne avant d'appuyer sur Return. Vous apprendrez plus tard à éditer le texte que vous avez entré.

Lorsque vous avez fini d'insérer votre texte dans le fichier `paint`, appuyez sur Esc pour repasser en mode commande. Le curseur se positionne sur le dernier caractère saisi. Vous pouvez alors entrer d'autres commandes `vi`.

Si, selon vous, `vi` a un comportement inattendu, vérifiez que vous n'êtes pas en mode "Caps Lock". Dans ce cas, le texte saisi apparaît en majuscules. Sur certains systèmes, la touche F1 (généralement située à côté de la touche Esc) a la même fonction que la touche Caps Lock. Il peut donc vous arriver d'appuyer dessus par erreur au lieu d'appuyer sur Esc.

Remarque : vous pouvez être appelé à effacer ou à rafraîchir l'écran afin de supprimer, par exemple, les messages intempestifs du système. Pour rafraîchir l'écran, passez en mode commande et appuyez sur Ctrl-L. Cette opération correspond à la commande Rafraîchir d'OpenWindows.

Mode commande

Lorsque vous ouvrez un fichier sous `vi`, vous vous trouvez en mode commande. Ce mode vous permet d'entrer des commandes qui mettent en oeuvre un grand nombre de fonctions. La plupart des commandes `vi` se composent d'une ou deux lettres et d'un chiffre en option. Il existe généralement une version des commandes en majuscules et une version en minuscules. Ces deux versions assurent des fonctions apparentées mais néanmoins distinctes. Par exemple, la commande `a` permet d'ajouter les caractères tapés à la droite du curseur, alors que la commande `A` permet de les ajouter à la *fin* de la ligne.

La plupart des commandes `vi` s'exécutent sans que vous ayez besoin d'appuyer sur Return. Il est toutefois nécessaire de le faire pour les commandes commençant par deux points (:). Il arrive que l'on classe ces commandes dans un troisième mode `vi`, le *mode dernière ligne*. Car lorsque vous entrez deux points en mode commande, les deux points et ce qui suit s'affichent sur la dernière ligne de l'écran. Mais il ne faut pas oublier que toutes les commandes `vi` sont lancées à partir du mode commande.

Les commandes précédées de deux points sont en fait des commandes `ex`. `vi` et `ex` sont deux interfaces distinctes du même logiciel d'édition de texte. Alors que `vi` est une interface orientée écran, `ex` est une interface orientée ligne. Toutes les commandes `ex` sont disponibles à partir de l'éditeur `vi`. Lorsque vous appuyez sur la touche ":", vous passez en fait à l'interface orientée ligne, `ex`. Cela vous permet d'exécuter de nombreuses commandes de manipulation de fichiers sans quitter `vi`. Pour plus d'informations, reportez-vous à la section "Utilisation des commandes `ex`", dans le présent chapitre.

Fermeture d'une session

Lorsque vous éditez un fichier sous `vi`, les modifications ne sont pas effectuées directement dans ce fichier, mais dans une copie de ce dernier créée par `vi` dans un espace de mémoire temporaire appelé *mémoire tampon*. La version permanente du fichier sur le disque n'est modifiée que lorsque vous écrivez (sauvegardez) le contenu de la mémoire tampon.

Cette méthode a des avantages et des inconvénients. D'un côté, vous pouvez quitter un fichier en annulant toutes les modifications apportées au cours d'une session d'édition et en laissant la version du disque intacte. Mais d'un autre côté, vous pouvez perdre le contenu (non sauvegardé) de la mémoire tampon de travail en cas de blocage du système. (Les utilisateurs de terminaux distants, connectés via des lignes téléphoniques, sont particulièrement vulnérables à ces interruptions inattendues.)

Il vaut donc mieux sauvegarder votre travail régulièrement, et, à plus forte raison, si vous y apportez des modifications substantielles.



Attention : bien qu'il soit possible d'ouvrir plusieurs sessions `vi` simultanément sur un même fichier, cela n'est pas recommandé. Vous pourriez avoir des difficultés à distinguer les modifications entrées dans le fichier de celles qui s'y sont superposées au cours d'une session simultanée.

Sauvegarde des modifications et sortie de vi

vi offre bon nombre de commandes plus ou moins synonymes permettant de sauvegarder le contenu de la mémoire tampon dans un fichier et de quitter vi. Ces commandes vous offrent la possibilité de sauvegarder, de sauvegarder et quitter ou de quitter sans sauvegarder.

Sauvegarde

Pour sauvegarder le contenu de la mémoire tampon (c'est-à-dire l'écrire dans le fichier du disque), entrez :

`:w`

et appuyez sur Return.

Sauvegarde et sortie

Pour sauvegarder et quitter, entrez :

`:wq`

et appuyez sur Return. Vous pouvez également taper ZZ.

La commande ZZ ne doit pas être précédée de deux-points ni suivie de Return.

Sortie sans sauvegarde

Si vous n'avez apporté aucune modification à un fichier et que vous souhaitez simplement le quitter, entrez :

`:q`

et appuyez sur Return. Si vous avez apporté des modifications, vi ne vous permettra pas de quitter le fichier à l'aide de la commande :q. Il affichera le message : `No write since last change (:quit! overrides)`.

Si vous ne voulez pas sauvegarder les modifications que vous avez apportées, entrez :

```
:q!
```

et appuyez sur Return.

Impression d'un fichier

Une fois que vous avez quitté un fichier `vi`, vous pouvez l'imprimer à l'aide de la commande suivante :

```
$ lp nom_fichier
```

où *nom_fichier* désigne le nom du fichier `vi` à imprimer. L'impression est envoyée à l'imprimante par défaut. Le fichier est alors imprimé non formaté, ligne par ligne, tel qu'il apparaît à l'écran. Pour plus d'informations sur les commandes d'impression, reportez-vous au Chapitre 8, "Utilisation des imprimantes".

Les commandes de base vi

La présente section décrit plusieurs catégories de commandes `vi` permettant :

- Le déplacement dans un fichier
- L'insertion de texte
- La modification et le remplacement de texte
- L'annulation des modifications apportées au texte
- La suppression de texte
- La copie et le déplacement de texte
- La répétition de commandes

Déplacement dans un fichier

Dans les sections précédentes, vous avez appris à créer, sauvegarder, imprimer et quitter un fichier `vi`. Maintenant que vous avez créé un fichier, vous devez connaître les concepts qui vous permettront de vous déplacer dans ce fichier. Ouvrez votre fichier d'entraînement et essayez, l'une après l'autre, les commandes décrites dans cette section.

Déplacement du curseur

Lorsque vous démarrez `vi`, le curseur se situe dans le coin supérieur gauche de l'écran `vi`. En mode commande, plusieurs commandes clavier vous permettent de déplacer le curseur : certaines touches alphabétiques, les touches de déplacement (flèches), les touches Return, Back Space (ou Del) et la barre d'espacement.

Remarque : la plupart des commandes `vi` font la distinction entre majuscules et minuscules. Ainsi, la “même” commande, selon qu'elle est entrée en caractères minuscules ou majuscules peut avoir des effets radicalement différents.

Déplacement à l'aide des flèches

Si votre clavier dispose de touches de déplacement (flèches), essayez-les. Vous devez pouvoir déplacer le curseur librement à l'écran à l'aide des flèches de déplacement vers le haut, vers le bas, vers la droite et vers la gauche. Mais ces flèches vous permettent uniquement de vous déplacer dans un texte ou des champs de saisie déjà existants.

Si vous utilisez `vi` à partir d'un terminal distant, les touches de déplacement du curseur peuvent ne pas fonctionner correctement. Cela dépend de l'émulateur de votre terminal. Si ces touches ne fonctionnent pas, vous pouvez utiliser à la place :

- La touche `h` pour vous déplacer vers la gauche.
- La touche `l` pour vous déplacer vers la droite.
- La touche `j` pour vous déplacer vers le bas.
- La touche `k` pour vous déplacer vers le haut.

Déplacement mot par mot

Appuyez sur **w** (“word”) pour déplacer le curseur mot par mot vers la droite.

Appuyez sur **b** (“back”) pour déplacer le curseur mot par mot vers la gauche.

Appuyez sur **W** ou sur **B** pour déplacer le curseur du signe de ponctuation le plus proche au blanc suivant ou précédent.

Appuyez sur **e** (“end”) pour positionner le curseur sur le dernier caractère du mot en cours.

Déplacement vers le début ou la fin d’une ligne

Appuyez sur **^** pour amener le curseur au début de la ligne en cours.

Appuyez sur **\$** pour amener le curseur à la fin de la ligne en cours.

Déplacement ligne par ligne

Appuyez sur la touche **Return** pour positionner le curseur au début de la ligne suivante.

Déplacement vers la gauche

Appuyez sur la touche **Back Space** pour déplacer le curseur d’un caractère vers la gauche.

Déplacement vers la droite

Appuyez sur la barre d’espace pour déplacer le curseur d’un caractère vers la droite.

Déplacement vers le haut

Appuyez sur **H** (“high”) pour amener le curseur en haut de l’écran.

Déplacement vers le centre

Appuyez sur **M** (“middle”) pour amener le curseur au centre de l’écran.

Déplacement vers le bas

Appuyez sur **L** (“low”) pour amener le curseur en bas de l’écran.

Pagination et défilement

Si vous essayez de vous déplacer vers le bas alors que le curseur se trouve en bas de l'écran ou vers le haut alors que ce dernier est en haut de l'écran, le texte défile vers le haut ou vers le bas. Dans un petit fichier, cela permet d'afficher une plus grande partie de texte, mais il peut être fastidieux de se déplacer de cette manière dans un fichier plus long.

Vous avez pu constater que si vous positionnez le curseur au-delà du bas ou du haut de l'écran, le texte défilait vers le haut ou vers le bas. Si elle est intéressante pour les petits fichiers, cette méthode est fastidieuse pour les fichiers plus longs.

Vous pouvez feuilleter ou faire défiler un fichier, un écran ou un demi-écran vers l'arrière ou vers l'avant. (Vous pouvez entrer un peu plus de texte dans le fichier `paint`, de façon à obtenir un fichier plus long, pour pouvoir tester ces commandes.)

Il existe une différence fondamentale entre la pagination et le défilement. Le défilement consiste à déplacer le curseur *ligne par ligne* vers le haut ou vers le bas du texte, à la manière d'un rouleau de papier, alors que la pagination consiste à déplacer le curseur *page-écran par page-écran* vers le haut ou vers le bas du texte. Sur un système rapide, vous ne remarquerez peut-être pas la différence. Mais si vous travaillez sur un terminal distant ou dans toute autre condition susceptible de ralentir le fonctionnement de votre système, cette différence peut être très sensible.

Défilement d'une page-écran

Pour faire défiler une page-écran vers le bas, appuyez sur Ctrl-F ("Forward"). (Maintenez la touche Control enfoncée et appuyez sur la touche F.) Le curseur se positionne dans le coin supérieur gauche du nouvel écran.

Défilement d'un demi-écran vers l'avant

Pour faire défiler un demi-écran vers le bas, appuyez sur Ctrl-D ("Down").

Défilement d'un écran vers l'arrière

Pour faire défiler en arrière (vers le haut) une page-écran, appuyez sur Ctrl-B.

Défilement d'un demi-écran vers l'arrière

Pour faire défiler en arrière un demi-écran, appuyez sur Ctrl-U.

Insertion de texte

`vi` dispose de plusieurs commandes pour l'insertion de texte. Cette section décrit les plus utiles d'entre elles. Pour utiliser ces différentes commandes, qui font passer `vi` en mode insertion, vous devez vous trouver en mode commande. Pour vous en assurer, appuyez sur la touche Esc.

Ajout

La commande `a` (ajout) permet d'insérer du texte à la *droite* du curseur. Positionnez le curseur sur une ligne, tapez `a`, puis le texte à ajouter. Appuyez sur Esc lorsque vous avez terminé.

La commande `A` permet d'insérer du texte à la *fin* d'une ligne. Positionnez le curseur sur une ligne de texte, puis tapez `A`. Le curseur est renvoyé à la fin de la ligne, où vous pouvez insérer votre texte. Appuyez sur Esc lorsque vous avez terminé.

Insertion

Vous pouvez insérer du texte à la gauche du curseur en tapant `i` en mode commande.

La commande `I` permet d'insérer du texte en début de ligne. (Elle ramène le curseur en début de ligne, quelle que soit sa position sur la ligne.) Comme pour toutes les commandes décrites dans cette section, vous devez appuyer sur Esc pour repasser en mode commande après avoir entré le texte à insérer.

Ouverture de lignes

Les commandes suivantes permettent d'ouvrir de nouvelles lignes, que ce soit au-dessus ou en-dessous de la position courante du curseur.

Tapez `o` pour ouvrir une ligne *au-dessous* de la position courante du curseur, puis insérez, au besoin, quelques lignes de texte. Appuyez sur Esc lorsque vous avez terminé.

Tapez `O` pour ouvrir une ligne *au-dessus* de la position actuelle du curseur.

Remplacement de texte

Le remplacement de texte implique de substituer une partie de texte à une autre. L'utilisation des différentes méthodes de remplacement de texte de `vi` dépend des circonstances.

Remplacement d'un mot

Pour remplacer un mot, positionnez le curseur au début du mot à remplacer. Tapez `cw`, suivi du nouveau mot, puis appuyez sur Esc.

Pour remplacer une *partie* de mot seulement, positionnez le curseur sur le mot, à *droite* de la partie à conserver. Tapez `cw`, entrez la correction, puis appuyez sur Esc.

Remplacement d'une ligne

Pour remplacer une ligne, positionnez le curseur sur cette ligne et tapez `cc`. La ligne est remplacée par une ligne blanche sur laquelle vous pouvez saisir une nouvelle chaîne de texte, quelle que soit sa longueur. Appuyez ensuite sur Esc.

Remplacement d'une partie de ligne

Pour remplacer une partie de ligne seulement, positionnez le curseur à *droite* de la partie à conserver. Tapez `C`, entrez la correction, puis appuyez sur Esc. Cela modifie la partie comprise entre le curseur et la fin de la ligne.

Substitution de caractère(s)

Pour substituer un ou plusieurs caractères au caractère situé sous le curseur, tapez `s`, suivi du (ou des) nouveau(x) caractère(s). Appuyez sur Esc pour repasser en mode commande.

Remplacement d'un caractère

Cette commande permet de remplacer le caractère mis en évidence par le curseur par un autre caractère. Positionnez le curseur sur le caractère et tapez `r`, suivi d'un caractère de remplacement unique. Une fois le remplacement effectué, `vi` repasse automatiquement en mode commande (vous n'avez pas besoin d'appuyer sur Esc).

Inversion de caractères

Avec `vi`, il suffit d'appuyer sur deux touches pour corriger les caractères inversés. Par exemple, si vous vous apercevez que vous avez tapé "lse" à la place de "les", vous pouvez corriger cette erreur en positionnant le curseur sur la première lettre à déplacer (dans le cas présent, la lettre `s`), puis en tapant `xp`. Le `s` et le `e` seront intervertis – et `vi` repassera automatiquement en mode commande.

Coupure et assemblage de lignes

Pour couper une ligne sans modifier le texte, amenez le curseur à l'endroit où vous souhaitez couper la ligne, tapez `r` (pour "remplacer"), et appuyez sur la touche Return. Si vous tapez `r` alors que le curseur est positionné sur un caractère et que vous appuyez sur Return, ce caractère sera remplacé par Return.

Pour réunir deux lignes, positionnez le curseur sur la ligne supérieure et tapez un `J` majuscule. (Vous n'avez pas besoin d'appuyer sur Esc après avoir tapé `J`.)

Annulation de modifications

Lorsque vous éditez un texte et que vous modifiez le contenu d'un fichier `vi`, vous pouvez être appelé à revenir sur vos modifications. Les commandes d'annulation de `vi` vous permettent d'annuler une opération et de poursuivre votre travail.

Annulation de la dernière commande

Si vous commettez une erreur sous `vi` ou si vous avez changé d'avis une fois l'opération terminée, vous pouvez annuler votre dernière commande en appuyant sur `u` immédiatement après l'exécution de cette dernière. (Vous n'avez pas besoin d'appuyer sur `Esc` après avoir tapé `u`.) Si vous appuyez sur `u` une *seconde* fois, la commande Défaire est annulée.

Annulation des modifications apportées à une ligne

Tapez `U` pour défaire toutes les modifications que vous avez apportées à une ligne. Cette commande fonctionne uniquement si vous n'avez pas déplacé le curseur en dehors de la ligne. (Vous n'avez pas besoin d'appuyer sur `Esc` après avoir tapé `U`.)

Suppression de texte

Ces commandes `vi` permettent de supprimer les caractères, mots ou lignes que vous avez indiqués. Etant donné que `vi` reste en mode commande, toute nouvelle insertion de texte devra être précédée d'une commande permettant de passer en mode insertion.

Suppression d'un caractère

Pour supprimer un caractère, positionnez le curseur sur le caractère à supprimer, puis tapez `x`.

La commande `x` supprime également l'espace qui était occupé par le caractère ; lorsqu'une lettre est supprimée au milieu d'un mot, les lettres restantes sont accolées. La commande `x` vous permet également de supprimer des espaces sur une ligne.

Pour supprimer un caractère précédant le (à gauche du) curseur, tapez `X` (en majuscule).

Suppression d'un mot ou d'une partie de mot

Pour supprimer un mot, positionnez le curseur au début du mot, puis tapez `dw`. Le mot et l'espace qu'il occupait sont supprimés.

Pour supprimer une partie de mot, positionnez le curseur sur le mot, à *droite* de la partie à conserver, puis tapez `dw` pour supprimer le reste du mot.

Suppression d'une ligne

Pour supprimer une ligne, positionnez le curseur sur la ligne, puis tapez `dd`. La ligne et l'espace qu'elle occupait sont supprimés.

Suppression d'une partie de ligne

Vous pouvez également supprimer une partie de ligne.

Pour supprimer tout le texte figurant à la *droite* du curseur, positionnez le curseur à droite de la partie à conserver, puis tapez `D`.

Pour supprimer tout le texte figurant à la *gauche* du curseur, positionnez le curseur à droite de la partie à supprimer, puis tapez `d0` (d-zéro).

Suppression de la fin du fichier

Pour supprimer tout le texte figurant entre la ligne en cours et la fin du fichier, tapez `dG`. Cette commande supprime également la ligne sur laquelle figure le curseur.

Suppression du début du fichier

Pour supprimer tout le texte figurant entre le début du fichier et la ligne en cours, tapez `d1G`. Cette commande supprime également la ligne sur laquelle figure le curseur.

Copie et déplacement de texte : commandes Yank, Delete et Put

La plupart des traitements de texte permettent de “copier et coller” et de “couper et coller” des lignes de texte. L'éditeur `vi` comporte également ces fonctions. Les commandes `vi` correspondant à la fonction “copier-coller” s'appellent *yank* et *put* ; et celles correspondant à la fonction “couper-coller” s'appellent *delete* et *put*.

Pour copier ou déplacer de petits blocs de texte sous `vi`, vous devez utiliser la combinaison de commandes `yank`, `delete` et `put`.

Copie de lignes

Deux commandes sont nécessaires pour copier une ligne : `yy` ou `Y` (“yank”) et `p` (“put below”), pour insérer la nouvelle ligne au-dessous de la ligne à copier, ou `P` (“put above”) pour l’insérer au-dessus. Les commandes `Y` et `yy` ont la même fonction.

Pour copier une ligne, positionnez le curseur sur la ligne choisie et tapez `yy`. Ensuite, amenez le curseur sur la ligne située au-dessus de l’endroit où vous voulez insérer (copier) la ligne extraite, puis tapez `p`. La copie de la ligne extraite apparaît sur une nouvelle ligne située *en-dessous* du curseur.

Pour placer la ligne extraite sur une nouvelle ligne située *en-dessus* du curseur, tapez `P`.

La commande `yy` fonctionne également par comptage. Par exemple, si vous voulez extraire 11 lignes, il vous suffit de taper `11yy`. Onze lignes sont extraites à partir de l’emplacement du curseur et `vi` vous le signale en affichant le message suivant en bas de l’écran : `11 lignes extraites` .

Vous pouvez également lancer les commandes `P` ou `p` juste après l’une des commandes de suppression décrites plus haut. Le texte que vous avez supprimé sera inséré respectivement au-dessus et en-dessous du curseur.



Attention : seules les commandes de déplacement du curseur sont autorisées entre la commande d’extraction ou de suppression et celle d’insertion. Si vous effectuez une autre suppression ou extraction avant d’avoir inséré le texte à son nouvel emplacement, les lignes extraites seront perdues.

Déplacement de lignes

Le déplacement de lignes fait également appel à deux commandes : `dd` (“delete”), puis `p` ou `P`.

Pour déplacer une ligne, positionnez le curseur sur la ligne choisie, puis tapez `dd`. Et si par exemple vous voulez déplacer 5 lignes, tapez `5dd`.

Ensuite, amenez le curseur sur la ligne située en-dessus de l’endroit où vous voulez réinsérer la ligne supprimée, puis tapez `p`. Le texte est inséré sur une nouvelle ligne en-dessous du curseur.

Vous pouvez également insérer la ligne supprimée en-dessus du curseur en tapant `P`.

Utilisation de mémoires tampon nommées

Pour insérer un même ensemble de lignes à différents endroits dans un document, vous pouvez extraire (ou supprimer) ces lignes et les placer dans une mémoire tampon nommée. Pour attribuer un nom à une mémoire tampon, faites précéder la commande d'un guillemet (") et du nom de la mémoire tampon. Par exemple, pour extraire quatre lignes et les placer dans la mémoire tampon *a*, tapez "`a4yy`". Vous pouvez utiliser plusieurs mémoires tampon. Ainsi, vous pouvez très bien supprimer un texte à un endroit et l'insérer à plusieurs endroits différents. Pour supprimer 12 lignes et les placer dans la mémoire tampon *b*, tapez "`b12dd`".

Pour insérer le texte, faites précéder la commande `p` ou `P` de "*n*", *n* étant le nom de la mémoire tampon. Pour insérer les lignes sauvegardées dans la mémoire tampon *b*, tapez "`bP`".

Vous pouvez écraser le contenu des mémoires tampon nommées en insérant de nouvelles lignes de texte. Les mémoires tampon sont sauvegardées jusqu'à ce que vous quittiez `vi`.

Lorsque vous utilisez les mémoires tampon nommées, vous pouvez supprimer et extraire d'autres parties de texte sans que cela n'affecte les lignes déjà sauvegardées dans des mémoires tampon nommées (sauf si vous écrasez volontairement une mémoire tampon nommée).

Utilisation d'un comptage pour la répétition des commandes

De nombreuses commandes `vi` peuvent être précédées d'un facteur de répétition (appelé *comptage*). Un nombre précède la commande, indiquant combien de fois l'opération doit être répétée.

La plupart des commandes décrites dans les sections précédentes peuvent recevoir un comptage. Par exemple, `3dd` active trois fois consécutives la commande de suppression de ligne, et supprime donc trois lignes, `2dw` supprime deux mots et `4x` quatre caractères ou espaces. Vous pouvez également utiliser le comptage pour les commandes de déplacement du curseur telles que `3w` et `2Ctrl-F`. Vous vous familiariserez avec ces fonctions en apprenant à vous servir des commandes `vi`. A la section "Récapitulatif des

principales commandes `vi`” proposée à la fin du présent chapitre, les commandes pour lesquelles le comptage est possible portent la mention “[count]”.

Si vous tapez un point (`.`), la dernière commande de modification de texte est répétée. Par exemple, si vous venez de supprimer une ligne à l’aide de la commande `dd`, vous pouvez positionner le curseur sur une autre ligne et supprimer cette dernière simplement en tapant un point.

Utilisation des commandes `ex`

Les commandes `ex` sont plus précises et plus pratiques que `yank`, `delete` et `put` pour le traitement de blocs de texte importants. Au lieu de compter les lignes à l’écran, puis de rechercher le point d’insertion, il vous suffit d’indiquer à `vi` une série de lignes à déplacer ou à copier, puis la ligne précédant le point d’insertion. (Il n’y a pas de point d’insertion pour les commandes de suppression.)

Affichage et suppression des numéros de lignes

Pour *afficher* les numéros de lignes, tapez `:set nu` et appuyez sur Return.

Les numéros de lignes s’affichent dans la marge de gauche. Uniquement visibles à l’écran, ils n’apparaissent pas à l’impression du fichier.

```
1 A te voir si bleu, si profond,
2 Et si brusquement, ô désert !
3 Comme les Grecs de Xénophon
4 Je me suis écrié : la mer !
~
~
~
~
~
~
~
:set nu
```

Pour supprimer les numéros de lignes, tapez `:set nonu` et appuyez sur Return.

Copie de lignes

La commande de copie `ex` présente le format général suivant :

```
:n`ligne,n`ligne co n`ligne
```

Les deux premiers nombres (séparés par une virgule) indique la série de lignes à copier. Le troisième désigne la ligne située *avant* le point d'insertion.

Par exemple, pour copier les lignes 1 à 5 du fichier `paint` et les insérer après la ligne 12, tapez :

```
:1,5 co 12
```

et appuyez sur Return.

Vous pouvez utiliser les abréviations suivantes pour indiquer des séries de lignes :

- Le point (`.`) signifie “à partir de la ligne en cours”.
- Le symbole dollar (`$`) signifie “jusqu’à la fin du fichier”.

Ainsi, pour copier la série de lignes allant “de la ligne en cours à la ligne 5” et insérer ce bloc après la ligne 12, tapez :

```
:. ,5 co 12
```

Pour copier la série de lignes allant “de la ligne 6 à la fin du fichier” et insérer ce bloc après la ligne 2, tapez :

```
:6,$ co 2
```

Déplacement de lignes

La commande de déplacement `ex` est analogue à la commande de copie décrite précédemment :

```
: n°line , n°line m n°line
```

Les séries de lignes et les points d'insertion sont désignés de la même façon et les abréviations (`.` et `$`) sont les mêmes. La seule différence est que la fonction “déplacement” permet d'extraire un bloc pour l'insérer à un autre emplacement.

Par exemple, pour déplacer les lignes 1 à 5 en dessous de la ligne 12, tapez :

```
: 1,5 m 12
```

et appuyez sur Return.

Suppression de lignes

Pour supprimer une série de lignes, utilisez la syntaxe de commande suivante :

```
: n°ligne , n°ligne d
```

Par exemple, pour supprimer les lignes 1 à 5, tapez :

```
: 1,5 d
```

Recherche et remplacement avec `vi`

Il existe plusieurs façons de localiser une chaîne de caractères donnée dans un fichier avec `vi`, qui offre notamment une fonction de remplacement systématique très performante.

Recherche d'une chaîne de caractères

Une *chaîne de caractères* est composée d'un ou de plusieurs caractères consécutifs. Elle peut contenir des lettres, des chiffres, des signes de ponctuation, des caractères spéciaux, des espaces, des tabulations ou des retours chariot. Il peut s'agir d'un mot grammatical ou d'une partie de mot.

Pour rechercher une chaîne de caractères, tapez / suivi de la chaîne recherchée et appuyez sur Return. `vi` positionne le curseur sur la première occurrence de la chaîne trouvée. Par exemple, pour rechercher la chaîne "meta", tapez /meta et appuyez sur Return.

Tapez `n` pour passer à l'occurrence *suivante* de la chaîne ou `N` pour passer à l'occurrence *précédente*.

Pour effectuer une recherche en arrière dans un fichier, vous pouvez utiliser ? à la place de /. Dans ce cas, les directions `n` et `N` sont inversées.

Normalement, les commandes de recherche font la distinction entre majuscules et minuscules : une recherche du mot "chine" ne trouvera pas le mot "Chine." Pour éviter cette distinction, tapez `:set ic`. Pour revenir au mode par défaut, qui distingue les majuscules des minuscules, tapez `:set noic`.

Si `vi` trouve la chaîne recherchée, le curseur s'arrête à la première occurrence de cette dernière. Dans le cas contraire, `vi` affiche le message : `Motif introuvable` sur la dernière ligne de l'écran.

Certains caractères spéciaux (/ & ! . ^ * \$ \ ?) ont une signification particulière pour le processus de recherche et doivent être «despécialisés» lorsqu'ils sont utilisés pour une recherche. Pour despécialiser un caractère spécial, faites-le précéder d'une barre oblique inversée (\). Par exemple, pour rechercher la chaîne "d'accord ?", tapez `/d'accord\?` et appuyez sur Return.

Etant donné que ces caractères spéciaux peuvent être interprétés comme des commandes par la fonction de recherche, vous devez indiquer qu'ils sont à inclure dans la chaîne recherchée en les faisant précéder d'une barre oblique inversée. Pour despécialiser une barre oblique inversée, tapez `\\`.

Affinement de la recherche

Vous pouvez affiner la recherche en associant à la chaîne des indicateurs signalant les états suivants :

- Début de ligne
- Fin de ligne
- Début de mot
- Fin de mot
- Caractères génériques

Pour que la chaîne soit recherchée en début de ligne, faites-la précéder d'un caret (^). Par exemple, pour rechercher la prochaine ligne commençant par "Rechercher", tapez :

```
/^Rechercher
```

Pour que la chaîne soit recherchée en fin de ligne, faites-la suivre d'un symbole dollar (\$). Par exemple, pour rechercher la prochaine ligne se terminant par "rechercher.", tapez :

```
/rechercher\.$
```

(Le point est déspecialisé par la barre oblique inversée.)

Pour que la chaîne recherchée corresponde au début d'un mot, faites-la précéder de \`<`, et pour qu'elle corresponde à la fin d'un mot, faites-la suivre de \`>`. Ainsi, pour que la chaîne recherchée corresponde à un mot plutôt qu'à une chaîne, combinez les deux indicateurs de début et de fin de mot dans les attributs de la recherche. Par exemple, pour trouver la prochaine occurrence du mot (par opposition à la chaîne) "rechercher", tapez :

```
/\<rechercher\>
```

Pour remplacer une partie de la chaîne recherchée par n'importe quel caractère, tapez un point (.) dans la chaîne, à l'emplacement à partir duquel la concordance de caractères doit être trouvée. Par exemple, pour trouver la prochaine occurrence de "pain" ou "main", tapez :

```
/ain
```

Comme il s'agit d'une recherche de chaîne et non de mot, ce modèle de recherche peut également trouver les occurrences de "pains" ou "mains".

Pour rechercher des variantes de caractères dans une chaîne, placez les variantes entre crochets. Le schéma de recherche `/[pm] chaîne` permettra de trouver les chaînes commençant soit par p, soit par m. D'autre part, `/[m-p] chaîne` permettra de trouver les chaînes commençant par une lettre de l'alphabet comprise entre m et p.

Pour que la chaîne recherchée corresponde à zéro, une ou plusieurs occurrences du dernier caractère, tapez un astérisque (*) dans la chaîne. Vous pouvez combiner les crochets et l'astérisque pour trouver des variantes déterminées. Par exemple, pour trouver toutes les chaînes commençant par une lettre de l'alphabet et se terminant par "information" *et* pour trouver toutes les occurrences de la chaîne "information", tapez :

```
/[a-z]*information
```

Remplacement d'une chaîne de caractères

La procédure de remplacement d'une chaîne de caractères repose sur le même principe que les procédures de recherche précédemment décrites. Toutes les concordances de caractères utilisées dans la fonction de recherche peuvent être également utilisées dans la fonction de recherche-remplacement.

Pour remplacer une chaîne de caractères, tapez :

```
:g/chaîne_recherchée/s//chaîne_replacement/g
```

et appuyez sur Return.

Pour remplacer toutes les occurrences de la chaîne "pain" par la chaîne "main", tapez :

```
:g/pain/s//main/g
```

et appuyez sur Return.

Vous pouvez modifier cette commande de telle sorte que `vi` interrompe la recherche et vous demande de confirmer le remplacement de chacune des occurrences. La commande suivante utilise `gc` (où `c` correspond à “confirmer”). Dans cette commande, `vi` s’arrête à chaque occurrence de “pain” et vous demande de confirmer le remplacement. Répondez par `y` pour oui ou par `n` pour non.

```
:g/pain/s//main/gc
```

Remarque : vous pouvez annuler toute opération de recherche-remplacement avec demande de confirmation en appuyant sur `Ctrl-C`.

Passage à une ligne donnée

Pour passer à la dernière ligne d’un fichier ouvert, tapez `G`. Pour revenir à la première ligne, tapez `1G`.

Vous pouvez accéder à n’importe quelle autre ligne en tapant le numéro correspondant, suivi de `G`.

Par exemple, si vous quittez le fichier `paint` lorsque la ligne 51 est en cours d’édition, vous pouvez réaccéder à cette ligne lors de l’ouverture du fichier en tapant `51G`.

Insertion d’un fichier dans un autre

`vi` facilite la “lecture” (l’insertion) d’un fichier dans le fichier en cours d’édition grâce à une commande du type :

```
:n'ligne r nom_fichier
```

Si vous ne précisez pas de numéro de ligne, `vi` insère le fichier à l’emplacement du curseur.

Par exemple, pour insérer le fichier `orwell` à la ligne 84 du fichier `paint`, tapez :

```
:84 r orwell
```

ou positionnez le curseur sur la ligne 84 et tapez :

```
:r orwell
```

Edition de plusieurs fichiers

`vi` vous permet d'éditer plusieurs fichiers à la fois. Si vous voulez éditer le fichier `orwell` alors que le fichier `paint` est en cours d'édition, procédez comme suit :

1. **Commencez par sauvegarder votre travail en cours dans le fichier `paint`. Tapez `:w` et appuyez sur Return.**
2. **Pour éditer le fichier `orwell`, tapez `:n orwell` et appuyez sur Return.**
3. **Apportez des modifications d'édition dans le fichier `orwell`, puis sauvegardez votre travail.**
4. **Une fois votre travail dans `orwell` terminé et sauvegardé, trois possibilités vous sont offertes :**
 - Quitter `vi` en tapant `:q` et en appuyant sur Return.
 - Retourner au fichier `paint` en tapant `:n n°` et en appuyant sur Return.
 - Basculer d'un fichier à l'autre à l'aide de la commande `:n n°`.

Edition d'une série de fichiers

Pour éditer une série de fichiers lors du démarrage de l'éditeur, entrez le nom de ces fichiers à la suite de `vi` au niveau du prompt de commande :

```
$ vi paint orwell
```

Les fichiers apparaissent dans l'ordre dans lequel ils sont entrés. Le fichier `paint` apparaît donc en premier. Une fois l'édition de `paint` terminée, tapez `:n` pour passer au fichier suivant, `orwell`. Pour passer au fichier suivant sans sauvegarder les modifications dans le fichier en cours, tapez `:n!` à la place de `:n`.

Si les noms des fichiers se ressemblent (par exemple, `test1`, `test2`, `test3`), vous pouvez utiliser des caractères génériques pour désigner un groupe de fichiers :

```
$ vi test*
```

Les noms des fichiers apparaîtront par ordre alphabétique pour l'édition.

Copie de lignes d'un fichier à l'autre

Pour copier des lignes d'un fichier à l'autre, procédez comme suit :

1. Editez le premier fichier.
2. Sauvegardez les lignes à copier dans des mémoires tampon nommées à l'aide de la commande `yank`. Par exemple, pour sauvegarder 10 lignes dans la mémoire tampon `a`, tapez `"a10Y`.
3. Sans sortir de `vi`, éditez le fichier suivant (`orwell` dans le présent exemple) :

```
:n orwell
```

4. Ajoutez les lignes du premier fichier à l'aide de la commande `put`. Par exemple, pour insérer le contenu de la mémoire tampon `a` en dessous du curseur, tapez `ap`.

Le contenu des mémoires tampon est perdu lorsque vous quittez `vi`. N'utilisez pas la commande `quit` (`:q`) tant que vous n'avez pas terminé toutes les opérations liées aux mémoires tampon nommées.

Sélection des paramètres vi

Le fonctionnement et l'aspect de `vi` dépendent d'un certain nombre de variables, dont vous pouvez consulter la liste (et la sélection en cours) lorsque `vi` est en cours, en tapant :

```
:set all
```

et en appuyant sur Return.

Reprise après une panne

En cas de panne du système, le contenu de votre mémoire tampon risque d'être perdu. Mais il est souvent possible de récupérer votre travail en redémarrant `vi` à l'aide du format de commande suivant :

```
vi -r nom_fichier
```

nom_fichier représente le fichier qui était en cours d'édition au moment de la panne. Une fois le système relancé, vous recevez généralement un courrier électronique vous informant de la présence d'un fichier de récupération.

Récapitulatif des principales commandes `vi`

Le tableau suivant offre un rappel pratique des principales commandes `vi`.

Tableau 6-1 Principales commandes `vi`

Commande	Explication
<i>Lancement de <code>vi</code></i>	
<code>vi nom_fichier</code>	Ouverture ou création d'un fichier
<code>vi</code>	Ouverture d'un nouveau fichier qui sera nommé ultérieurement
<code>vi -r nom_fichier</code>	Récupération d'un fichier après une panne
<code>view nom_fichier</code>	Ouverture d'un fichier en lecture seule
<i>Commandes du curseur</i>	
<code>h</code>	Déplacement d'un caractère vers la gauche
<code>j</code>	Déplacement d'une ligne vers le bas
<code>k</code>	Déplacement d'une ligne vers le haut
<code>l</code>	Déplacement d'un caractère vers la droite
<code>w</code>	Déplacement d'un mot vers la droite
<code>W</code>	Déplacement d'un mot vers la droite (depuis un signe de ponctuation)
<code>b</code>	Déplacement d'un mot vers la gauche

Tableau 6-1 Principales commandes vi (Continued)

Commande	Explication
B	Déplacement d'un mot vers la gauche (depuis un signe de ponctuation)
e	Déplacement vers la fin du mot en cours
Return	Déplacement d'une ligne vers le bas
Back Space	Déplacement d'un caractère vers la gauche
Barre d'espace	Déplacement d'un caractère vers la droite
H	Déplacement vers le haut de l'écran
M	Déplacement vers le centre de l'écran
L	Déplacement vers le bas de l'écran
Ctrl-F	Défilement d'une page-écran vers l'avant
Ctrl-D	Défilement d'un demi-écran vers l'avant
Ctrl-B	Défilement d'une page-écran vers l'arrière
Ctrl-U	Défilement d'un demi-écran vers l'arrière
<i>Insertion de caractères et de lignes</i>	
a	Insertion de caractères à droite du curseur
A	Insertion de caractères en fin de ligne
i	Insertion de caractères à gauche du curseur
I	Insertion de caractères en début de ligne
o	Insertion d'une ligne en dessous du curseur
O	Insertion d'une ligne en dessus du curseur
<i>Remplacement de texte</i>	
cw	Remplacement d'un mot (ou d'une partie de mot) à droite du curseur
c	Remplacement d'une ligne
C	Remplacement de la partie de ligne allant de l'emplacement du curseur à la fin de la ligne
s	Substitution d'une chaîne au(x) caractère(s) situés à droite du curseur

Tableau 6-1 Principales commandes vi (Continued)

Commande	Explication
r	Remplacement du caractère situé à droite du curseur par un autre caractère
␣ Return	Coupure de ligne
J	Jointure de la ligne en cours à la ligne suivante
xp	Inversion du caractère désigné par le curseur et du caractère situé à droite de ce dernier
~	Changement du type de caractères (majuscules ou minuscules)
u	Annulation de la dernière commande
U	Annulation des modifications apportées sur la ligne en cours
:u	Annulation de la dernière commande de la dernière ligne
<i>Suppression de texte</i>	
x	Suppression du caractère désigné par le curseur
X	Suppression du caractère situé à gauche du curseur
dw	Suppression du mot (ou d'une partie de mot) situé à droite du curseur
dd	Suppression de la ligne contenant le curseur
D	Suppression de la partie de ligne située à droite du curseur
dG	Suppression de la fin du fichier
d1G	Suppression du début du fichier jusqu'à l'emplacement du curseur
:5,10 d	Suppression des lignes 5 à 10
<i>Copie et déplacement de texte</i>	
yy	Extraction ou copie de ligne
Y	Extraction ou copie de ligne
p	Insertion de la ligne extraite ou supprimée sur la ligne située en dessous de la ligne en cours

Tableau 6-1 Principales commandes vi (Continued)

Commande	Explication
P	Insertion de la ligne extraite ou supprimée sur la ligne située en dessus de la ligne en cours
:1,2 co 3	Copie des lignes 1 à 2 à la ligne située en dessous de la ligne 3
:4,5 m 6	Déplacement des lignes 4 à 5 à la ligne située en dessous de la ligne 6
<i>Affichage des numéros de lignes</i>	
:set nu	Affichage des numéros de lignes
:set nonu	Suppression des numéros de lignes
<i>Distinction des majuscules et des minuscules</i>	
:set ic	Recherche sans distinction majuscules-minuscules
:set noic	Recherche avec distinction majuscules-minuscules
<i>Recherche d'une ligne</i>	
G	Passage à la dernière ligne du fichier
1G	Passage à la première ligne du fichier
21G	Passage à la ligne 21
<i>Recherche et remplacement</i>	
/chaîne	Recherche de chaîne
?chaîne	Recherche vers l'arrière de chaîne
n	Recherche de l'occurrence suivante de chaîne dans le sens de la recherche
N	Recherche de l'occurrence précédente de chaîne dans le sens de la recherche
:g/recherche/s//remplace/g	Recherche et remplacement
<i>Effacement de l'écran</i>	
Ctrl-L	Effacement (rafraîchissement) d'un écran brouillé

Tableau 6-1 Principales commandes vi (Continued)

Commande	Explication
<i>Insertion d'un fichier dans un autre fichier</i>	
:r <i>nom_fichier</i>	Insertion (lecture) du fichier après le curseur
:34 r <i>nom_fichier</i>	Insertion du fichier après la ligne 34
<i>Sauvegarde et sortie</i>	
:w	Sauvegarde des modifications (écriture dans la mémoire tampon)
:w <i>nom_fichier</i>	Ecriture dans la mémoire tampon du fichier nommé
:wq	Sauvegarde de modifications et sortie de vi
ZZ	Sauvegarde des modifications et sortie de vi
:q!	Sortie sans sauvegarde des modifications

Utilisation de la messagerie



SunOS contient un programme, appelé `mailx`, qui permet l'envoi et la réception de courrier électronique (*email*). `mailx` inclut des fonctions de lecture, d'écriture, d'envoi, de réception, de sauvegarde et de suppression des messages. Comme il n'est pas structuré en fenêtres, le programme `mailx` peut être exécuté sur n'importe quel terminal. Bien que vous préféreriez peut-être utiliser la messagerie structurée en fenêtres, le programme `mailx` peut se révéler utile pour la rédaction de courtes notes. Vous trouverez en outre de précieuses informations sur la configuration de vos propres alias de distribution.

Remarque : si vous êtes dans l'environnement OpenWindows et que l'icône de la Messagerie apparaît à l'écran, quittez la Messagerie avant d'exécuter les exemples de ce chapitre. La présence de deux processus de messagerie actifs risque de générer des erreurs et des messages d'alerte. Vous pouvez très bien envoyer des messages dans une fenêtre Utilitaire de commande ou l'Utilitaire Shell, mais si vous consultez votre courrier et que vous sauvegardez ou supprimez des messages, cela risque d'affecter le contenu de votre corbeille d'arrivée et de gêner le fonctionnement de l'utilitaire.

`mailx` : *Fonctions de base*

Cette section présente les fonctions de base qui vous permettront de vous familiariser avec `mailx`. Dans les sections ultérieures, vous découvrirez des caractéristiques et des fonctions qui vous permettront de vous perfectionner dans la manipulation de ce programme.

Le programme `mailx` requiert comme adresse unique le nom de connexion du destinataire voulu et celui de la machine. Si le destinataire se trouve sur la même machine que l'émetteur, seul le nom de connexion est nécessaire. Chaque utilisateur possède une *boîte à lettres* pour la réception de son courrier. Cette boîte à lettres se trouve généralement dans le répertoire `/var/mail/nom_utilisateur`, où *nom_utilisateur* désigne votre nom de connexion.

Le programme `mailx` vous informe de la réception de messages et place ces derniers dans votre boîte à lettres. Une fois que vous avez consulté vos messages, `mailx` les stocke automatiquement dans un fichier, appelé `mbox`, qui se trouve également dans votre répertoire utilisateur.

Lancement de `mailx`

Pour lancer `mailx`, tapez la commande suivante en regard du prompt, puis appuyez sur la touche Return :

```
$ mailx
```

Si vous n'avez pas de courrier en attente, votre terminal affiche le message :

```
Pas de courrier pour nom_utilisateur
$
```

où *nom_utilisateur* désigne votre nom de connexion.

Envoi d'une lettre type

Pour vous familiariser rapidement avec `mailx`, vous pouvez vous envoyer une lettre type. En regard du prompt, entrez à nouveau la commande `mailx`, mais en précisant cette fois votre adresse (c'est-à-dire votre nom de connexion plus le nom de votre machine). Par exemple, si votre nom de connexion est `rose` et que le nom de votre machine est `texas`, vous aurez pour adresse `rose@texas`. (Le symbole `@` se lit "arobas".) Si vous êtes connecté à un réseau local, vous n'avez à indiquer que votre nom de connexion ; en cas de doute, contactez votre administrateur système.

```
$ mailx rose@texas
```

Le programme affiche en réponse une ligne `Subject`:

```
$ mailx rose@texas
Subject:
```

Vous pouvez, si vous le souhaitez, taper un ou deux mots pour identifier le contenu de la lettre que vous vous envoyez et appuyer sur `Return`. Tapez ensuite le corps en utilisant des lignes courtes et en appuyant sur `Return` à la fin de chaque ligne. (Notez que vous pouvez effectuer des corrections uniquement en vous déplaçant au moyen de la touche `Back Space` et en retapant le contenu de la ligne *avant* d'avoir appuyé sur `Return`.)

Votre lettre type doit se présenter comme l'exemple suivant (pour générer un espace entre deux lignes, il suffit d'appuyer deux fois de suite sur Return) :

```
$ mailx rose@texas
Subject: à moi-même,message test

Rose,

Du bout de tes doigts fins
Au plus profond de tes yeux de jade
Tu es une douce brise de printemps
Ma chère et tendre Rose.

A bientôt,

Richard
```

Pour envoyer votre lettre type, appuyez sur Return pour terminer la dernière ligne du texte, puis sur Ctrl-D. Une fois la lettre envoyée, le système affiche un prompt de commande.

Lecture de la lettre type

Pour lire votre lettre type, entrez à nouveau la commande mailx. Vous verrez apparaître un écran plus ou moins similaire au suivant :

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Entrez un point
d'interrogation (?) pour obtenir de l'aide.
"/var/mail/rose": 2 messages 1 nouveau(x)
  U  2 hal@uncertain   Fri Feb 14 12:01   14/318 état financier
>N  1 rose@texas      Mon Feb 17 08:12   21/453 à moi-même
&
```

La première ligne identifie la version de mail utilisée ; la deuxième désigne la boîte à lettres, généralement placée dans `/var/mail/nom_utilisateur`, où sont déposés les messages reçus. La troisième ligne de cet exemple est l'en-tête de la lettre que vous vous êtes envoyée. Le "N" placé au début de la ligne indique qu'il s'agit d'une "nouvelle" lettre. La lettre "U" (unread) indique qu'il s'agit

également d'une nouvelle lettre, mais qu'elle n'a pas été lue lors de la précédente session du programme `mailx`. (Pour plus d'informations sur le contenu de cet écran, consultez la section "Lecture de lettres", dans le présent chapitre.)

A réception, chaque lettre se voit attribuer un numéro. Par exemple, la lettre que Rose s'est envoyée à elle-même porte le numéro 1.

Pour lire une lettre, il suffit de taper au prompt `mailx` le numéro correspondant, précédé d'une perluète (&), comme indiqué ci-dessous :

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Entrez un point
d'interrogation (?) pour obtenir de l'aide.
"/var/mail/rose": 1 message 1 new
>N 1 rose@texas   Fri Jul 14 12:01 21/453 à moi-même
& 1

To: rose@texas
From: rose@texas
Subject: à moi-même,message test

Chère Rose,

Du bout de tes doigts fins
Au plus profond de tes yeux de jade
Tu es une douce brise de printemps
Ma chère et tendre Rose.

A bientôt,

Richard

&
```

Sortie de mailx

Lorsque vous avez terminé votre travail sous `mailx`, vous pouvez quitter le programme en utilisant l'une de ces deux commandes : `q` (quit) ou `x` (exit).

Si vous tapez `q` à la suite du prompt `mailx` et que vous appuyez sur Return,

```
& q
```

vous verrez apparaître un message du type suivant :

Un message sauvegardé dans *répertoire_utilisateur*/mbox.

où *répertoire_utilisateur* désigne le chemin de votre répertoire utilisateur.

Lorsque vous utilisez `q` pour quitter `mailx` après avoir lu vos messages, `mailx` sauvegarde le contenu de votre boîte à lettres dans le fichier `mbox` de votre répertoire utilisateur, ainsi que les éventuelles modifications ou suppressions que vous avez effectuées.

En revanche, si vous tapez `x` au prompt `mailx` et que vous appuyez ensuite sur Return,

```
& x
```

le programme `mailx` ne sauvegarde pas les modifications ni les suppressions, et ne transfère pas les lettres déjà consultées dans le fichier `mbox`.

Lecture de lettres

Si vous avez du courrier, `mailx` vous en informe chaque fois que vous vous connectez par le message :

You have mail

ou

You have new mail

Pour lire vos lettres, tapez `mailx` en regard du prompt de commande et appuyez sur Return. Si vous n'avez pas de courrier en attente, vous verrez apparaître le message :

Pas de courrier pour *nom_utilisateur*

Sinon, vous obtiendrez la liste suivante :

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Entrez un point
d'interrogation (?) pour obtenir de l'aide.
"/var/mail/rose": 4 messages 1 new 2 unread
  1 rose@texas      Fri Feb 14 12:01 21/453 à moi-même
U  2 hank@fretful   Fri Feb 14 18:31 19/353 so lonely I
U  3 farmer@freeway Sat Feb 15 10:22 24/557 looks like my
>N 4 hoover@woofer  Sun Feb 16 23:59 14/280 big old furry

&
```

Le programme `mailx` affiche des informations relatives à lui-même (numéro de version et date) et des instructions concernant l'aide en ligne (Entrez un point d'interrogation (?) pour obtenir de l'aide.).

Sur la ligne suivante, `mailx` indique l'adresse de votre boîte à lettres, le nombre de lettres reçues, ainsi que leur statut.

`mailx` affiche ensuite la liste numérotée des lettres contenues dans votre boîte à lettres. Les colonnes qui apparaissent sur chaque ligne désignent, de gauche à droite :

- **Statut** : indique si la lettre est nouvelle (N), non lue (U) ou lue (pas de marque). La présence d'un ">" en début de ligne désigne la lettre en cours. Les lettres supprimées sont identifiées par un astérisque (*).
- **Numéro** : indique l'ordre de réception de chacune des lettres.
- **Expéditeur** : indique le nom de l'utilisateur (et généralement de la machine) qui a envoyé la lettre.
- **Heure** : indique la date et l'heure d'envoi de la lettre.
- **Taille** : indique le nombre de lignes/caractères de la lettre.
- **Objet** : indique l'objet de la lettre, tel qu'il a été défini par l'expéditeur.

Lorsque votre boîte à lettres est très remplie, toutes les lettres n'apparaissent pas sur la liste. Dans ce cas, tapez :

- `z` – pour afficher l'écran d'en-têtes de lettres suivant.
- `h-` – pour afficher l'écran d'en-têtes de lettres précédent.

- `h` – pour réafficher à tout moment la liste d'en-têtes de lettres.

Pour consulter la lettre en cours (identifiée par `>`), appuyez sur Return. Si vous appuyez une deuxième fois sur Return, c'est la lettre suivante qui s'affiche. Pour consulter une lettre quelconque de la liste, tapez son numéro et appuyez sur Return.

Suppression des lettres (et annulation de suppression)

Après avoir consulté une lettre, vous pouvez la supprimer au lieu de la sauvegarder dans le fichier `mbx` (option par défaut lorsque vous quittez le programme `mailx`).

Pour supprimer la dernière lettre que vous avez consultée, il vous suffit de taper `d` au prompt `mailx`. Pour supprimer une autre lettre de votre boîte à lettres, utilisez la syntaxe :

`d` *numéro*

Par exemple, pour supprimer la deuxième lettre, entrez la commande suivante depuis `mailx` :

```
& d 2
```

Vous pouvez également supprimer plusieurs lettres à la fois. Par exemple, pour supprimer les lettres 1 *et* 3, entrez la commande :

```
& d 1 3
```

Pour supprimer un ensemble de lettres consécutif (les lettres 1 à 3, par exemple), entrez la commande :

```
& d 1-3
```

Avant de quitter `mail`, vous pouvez *annuler la suppression* des lettres de votre boîte à lettres qui étaient destinées à être supprimées, en utilisant la syntaxe :

`u` *numéro*

puis Return. Par exemple, pour annuler la suppression de la deuxième lettre, entrez la commande :

```
& u 2
```

Pour annuler la dernière suppression effectuée, il suffit de taper `u` en regard du prompt `mailx` immédiatement après la commande. Par exemple, si votre dernière commande était `d 2-5` et que vous tapez `u`, l'annulation de suppression portera sur les messages 2, 3, 4 et 5.

Notez que lorsque vous quittez `mailx` avec la commande `q`, les suppressions sont définitives, c'est-à-dire qu'elles ne peuvent plus être annulées. En revanche, si vous voulez quitter `mailx` en conservant votre boîte à lettres intacte, vous pouvez utiliser la commande `x`. Comme nous l'avons déjà expliqué, la commande `x` permet en effet de quitter le programme en conservant la marque `U` devant les lettres qui ont été lues, en gardant intactes les lettres destinées à être supprimées, et ainsi de suite.

Impression de lettres

Vous pouvez imprimer une lettre en la transmettant par pipe à une commande d'impression. Vous devez pour cela utiliser la syntaxe :

```
| numéro lp
```

en regard du prompt `mailx`. (Le symbole `|` est appelé *pipe*.) Par exemple, pour imprimer la lettre 2, tapez :

```
& | 2 lp
```

et appuyez sur Return. Si vous n'indiquez pas de numéro de lettre, `mailx` transmet à l'imprimante la lettre en cours. Pour plus d'informations sur la transmission par pipe, consultez la section "Redirection et transmission par pipe du résultat d'une commande", dans le Chapitre 2, "Commandes SunOS de base".

Envoi de lettres

Pour envoyer une lettre à l'aide du programme `mailx`, vous devez connaître le nom de connexion du (ou des) destinataire(s). Si le destinataire utilise une autre machine que vous, vous devez également préciser le nom de cette dernière. Pour accéder à cette information, vous pouvez utiliser les commandes `who`, `finger` ou `rusers`.

La commande `who` affiche la liste de tous les utilisateurs actuellement connectés à votre serveur de fichiers. Cette liste contient le nom de connexion des utilisateurs, le type de terminaux sur lesquels ils travaillent, ainsi que leur date et heure de connexion. Exemple :

```
$ who
  elmer      tty15      Feb 20 10:22
  susan      tty04      Feb 20 10:37
  stormy     tty07      Feb 20 11:49
  hankw      tty06      Feb 20 12:02
```

La commande `finger` affiche le même type d'informations que la commande `who`, mais de façon plus détaillée. Les informations qui s'affichent dépendent de la façon dont votre administrateur système a configuré cette commande. A titre d'exemple, vous pouvez voir apparaître un écran du type :

```
$ finger
  Login      Name           TTY      Idle    When
  elmer      Elmer Brown    tty15     43      Thu 10:22
  susan      Susan Lake     tty04     12      Thu 10:37
  stormy     Stormy Ball    tty07     12      Thu 11:49
  hankw      Hank Wilson    tty06     22      Thu 12:02
```

La commande `rusers` affiche des informations sur les utilisateurs actuellement connectés à votre réseau local. Reportez-vous au Chapitre 9, "Utilisation du réseau" pour connaître la procédure d'utilisation de la commande `rusers`.

Après avoir déterminé les informations nécessaires sur l'utilisateur, suivez les étapes ci-après pour envoyer votre lettre.

1. Tapez la commande `mailx`, suivie de l'adresse d'un utilisateur :

```
$ mailx utilisateur@machine
```

où *utilisateur* désigne le nom de connexion du destinataire et *machine* le nom de sa machine.

- Si vous avez déjà lancé `mailx`, vous pouvez taper simplement `m` au prompt `mailx`, suivi du nom de connexion du destinataire et du nom de sa machine :

```
& m utilisateur@machine
```

- Pour envoyer la même lettre à plusieurs destinataires, vous devez séparer les différentes adresses par un espace ou une virgule. Exemple :

```
$ mailx hank@fretful sally@dakota tex@twister
```

ou

```
$ mailx hank@fretful,sally@dakota,tex@twister
```

2. Lorsque vous appuyez sur Return, le programme `mailx` vous invite à entrer un objet. Tapez l'objet de votre lettre et appuyez à nouveau sur Return.

3. Tapez le corps de votre lettre. Lorsque vous voulez créer une nouvelle ligne, appuyez sur Return.

Une phrase renvoyée à la ligne sur votre écran n'est pas considérée comme une nouvelle ligne tant que vous n'avez pas appuyé sur Return.

Remarque : la longueur maximale de chaque ligne de texte à l'intérieur de la lettre est de 256 caractères. Lorsque vous dépassez cette limite, l'écran est gelé. Vous devez alors appuyer sur Ctrl-C pour interrompre la lettre.

4. Après avoir terminé votre lettre, appuyez sur Return pour positionner le curseur sur une nouvelle ligne. Appuyez ensuite sur Ctrl-D pour envoyer la lettre.

Lettres non transmises

Si vous indiquez une adresse utilisateur incorrecte lorsque vous envoyez votre lettre, le système affiche en réponse le message :

```
utilisateur@machine...User unknown
```

et renvoie la lettre dans votre boîte à lettres. La prochaine fois que vous taperez la commande `mailx`, l'en-tête vous informera qu'un message vous a été renvoyé, comme dans l'exemple suivant :

```
N 1 Mailer-Daemon Fri Jan 3 11:13 8/49 Returned mail: User unknown
```

Par ailleurs, lorsqu'une lettre ne peut être transmise à son destinataire, elle est copiée dans un fichier de votre répertoire utilisateur nommé `dead.letter`.

Annulation d'une lettre non envoyée

Vous pouvez à tout moment annuler une lettre *avant* de l'envoyer, en appuyant à deux reprises sur Ctrl-C.

Ajout de copies carbone et de copies carbone aveugles

Avant d'envoyer une lettre, vous pouvez demander l'envoi de "copies carbone" à des personnes autres que les principaux destinataires. Vous pouvez également envoyer des "copies carbone aveugles". (Les destinataires de votre lettre peuvent lire les adresses de distribution des copies carbonées, mais pas celles des copies carbonées aveugles.)

De nombreux utilisateurs s'envoient à eux-mêmes une copie carbone ou une copie carbone aveugle afin de garder un exemplaire de leur lettre pour archivage.

Il existe trois méthodes pour joindre des copies carbonées à une lettre :

- Vous pouvez modifier le contenu du fichier `.mailrc` (de votre répertoire utilisateur) au moyen d'un éditeur de texte, en y insérant la ligne suivante :

```
set askcc
```

Le programme `mailx` affiche alors le prompt `Cc:` (copie carbone) à la suite du prompt de l'objet. Entrez les adresses des utilisateurs auxquels sont destinées les copies carbone, en les séparant par des espaces.

- Une fois que vous avez tapé le corps de votre lettre, mais avant d'appuyer sur Ctrl-D, appuyez sur Return pour passer à la ligne et utilisez :

```
~c adresse(s)
```

Si vous envoyez des copies carbone à plusieurs destinataires, séparez les différentes adresses par des espaces. Exemple :

```
~c hank@fretful george@lonesome stormy@snoozer
```

- Une ligne `Cc:` est en outre créée par la commande `~h`, qui affiche l'en-tête complet de la lettre. La commande `~h` vous invite à renseigner ligne par ligne les rubriques `To:`, `Subject:`, `Cc:` et `Bcc:` (copie carbone aveugle). Vous pouvez insérer des informations sur les lignes vierges et saisir en recouvrement sur les lignes déjà renseignées. Comme toutes les autres commandes tilde, la commande `~h` doit toujours être entrée sur une ligne vierge.

Remarque : `~c`, `~h` et les autres commandes tilde sont décrites dans la section "Commandes tilde" du présent chapitre.

Insertion d'une copie de lettre ou de fichier

Vous pouvez insérer la copie d'une lettre de votre boîte à lettres dans une autre lettre que vous êtes en train de rédiger. Selon le même principe, vous pouvez également insérer la copie de n'importe quel fichier texte.

Insertion d'une lettre

La syntaxe à utiliser pour l'insertion d'une lettre est :

`~m numéro`

où *numéro* désigne le numéro de la lettre à insérer. Par exemple, pour envoyer un courrier à un autre utilisateur en y insérant la copie de la lettre numéro 3 de votre boîte à lettres, procédez comme suit :

1. **Entrez la commande `~m 3` sur une ligne vierge, puis appuyez sur Return.**
2. **mailx affiche le message**, Insertion en cours : 3 (suite)
3. **Vous n'avez pas accès au contenu du message 3, mais l'utilisateur, lui, pourra le visualiser. Vous pouvez alors continuer à travailler sur votre lettre (suite) ou l'envoyer telle quelle.**
4. **Pour visualiser la lettre complète, y compris le message joint, tapez la commande `~p`.**

Insertion d'un fichier

Vous pouvez également insérer dans une lettre la copie de n'importe quel fichier texte, en utilisant la syntaxe :

`~r nom_fichier`

tandis que vous rédigez votre lettre. Par exemple, pour insérer le fichier `outline` dans la lettre en cours, tapez :

`~r outline`

Réponse à une lettre

Pour répondre à un courrier, il suffit d'entrer la commande :

`r numéro`

à la suite du prompt `mailx`. (Si vous omettez d'indiquer le numéro de lettre, `mailx` répond à la lettre en cours.) Par exemple, pour répondre à l'expéditeur de la lettre 2, entrez la commande :

```
& r 2
```

`mailx` attribue automatiquement une adresse à votre lettre et affiche une ligne `Re: Subject:`, identique à la ligne `Subject: originale`. Vous pouvez alors envoyer votre réponse comme une lettre normale.

`R` est une variante de la commande réponse qui envoie votre réponse, non seulement à l'expéditeur, mais également à tous les destinataires de la lettre originale. Utilisez cette commande uniquement lorsque c'est absolument nécessaire, afin d'éviter de générer des courriers inutiles.

Remarque : vous pouvez insérer une lettre dans votre réponse comme indiqué dans la section précédente. Pour insérer la copie de la lettre à laquelle vous êtes en train de répondre, il suffit d'entrer la commande `~m`, sans indiquer le numéro de la lettre.

Sauvegarde et réutilisation de lettres

Outre les procédures d'envoi et de réception de courrier, vous pouvez également être appelé à sauvegarder des lettres et à les réutiliser par la suite. `mailx` vous permet de sauvegarder vos lettres en les plaçant dans des fichiers texte normaux ou dans des fichiers spéciaux appelés dossiers. Ces deux méthodes sont décrites ci-dessous.

`mailx` distingue les opérations de *sauvegarde* et de *copie* de lettres. La sauvegarde consiste en effet à insérer une lettre dans un fichier ou dans un dossier en la supprimant de la boîte à lettres, tandis que la copie place cette lettre dans un fichier ou dans un dossier, mais en la conservant dans la boîte à lettres.

Sauvegarde et copie de lettres dans des fichiers

La syntaxe à utiliser à la suite du prompt `mailx` pour sauvegarder une lettre dans un fichier est la suivante :

`s numéro nom_fichier`

où *numéro* désigne le numéro de la lettre à sauvegarder et *nom_fichier* le fichier dans lequel vous voulez la sauvegarder. Par exemple, pour sauvegarder la lettre 3 dans un fichier appelé `~/notes/finance`, vous devez taper :

```
& s 3 ~/notes/finance
```

(N'oubliez pas que dans un chemin, le `~` représente votre répertoire utilisateur.)

Vous pouvez également sauvegarder plusieurs lettres à la fois dans le même fichier. Par exemple, pour sauvegarder les lettres 3, 5, 6, 7 et 8 dans `~/notes/finance`, vous pouvez taper :

```
& s 3 5-8 ~/notes/finance
```

Si le fichier indiqué n'existe pas, `mailx` le crée. S'il existe, `mailx` place la lettre sauvegardée à la fin de ce fichier.

Lorsque vous sauvegardez un fichier, il est supprimé de votre boîte à lettres. `mailx` affiche un astérisque (*) en regard de l'en-tête des lettres qui ont été sauvegardées.

Pour conserver la lettre dans votre boîte à lettres lorsque vous l'insérez dans un fichier, utilisez la commande `copy`, comme indiqué ci-après :

```
& c 3 ~/notes/finance
```

Sauvegarde et copie de lettres dans des dossiers

Pour éviter d'avoir à taper le chemin d'accès complet de vos fichiers, vous pouvez sauvegarder ou copier vos lettres dans des dossiers de messagerie. Les dossiers sont des fichiers spéciaux, stockés dans un répertoire de dossiers.

La sauvegarde ou la copie de lettres dans des dossiers présente l'avantage de rassembler automatiquement vos lettres dans le même répertoire, ce qui les rend facilement accessibles et évite à l'utilisateur d'avoir à taper des chemins d'accès complexes.

Configuration du répertoire de dossiers

Pour utiliser des dossiers, vous devez préalablement définir un répertoire de dossiers. Ce processus s'effectue en deux étapes :

1. Création du répertoire à l'aide de la commande `mkdir`

Par exemple, si vous voulez nommer votre répertoire de dossiers `Lettres`, vous devez commencer par le créer :

```
$ mkdir Lettres
```

2. Edition du fichier `.mailrc` de votre répertoire utilisateur (contenant les options `mailx`) au moyen d'un éditeur de texte afin d'y définir le chemin du répertoire de dossiers.

Vous devez ici éditer la variable `set folder` pour y inclure le chemin d'accès complet du répertoire de dossiers nouvellement créé, comme dans l'exemple :

```
set folder=/home/austin/rose/Lettres
```

ou indiquer votre répertoire utilisateur au moyen du raccourci C Shell.

```
set folder=~ /Lettres
```

Votre répertoire de dossiers est maintenant prêt à recevoir les lettres sauvegardées dans des dossiers. (La modification du fichier `.mailrc` prendra effet la prochaine fois que vous lancerez `mailx`.)

Désignation des dossiers

Que vous sauvegardiez ou que vous copiez des lettres dans des dossiers ou dans des fichiers, vous devez utiliser les mêmes commandes, à ceci près que le nom du dossier doit être précédé du signe (+) à la place du chemin d'accès. Le + indique en effet à `mailx` que le dossier doit être classé dans le répertoire de dossiers (`Lettres`).

Par exemple, pour sauvegarder la lettre 3 dans un dossier `projets`, tapez :

```
& s 3 +projets
```

`mailx` interprète cette commande comme “sauvegarder la lettre 3 dans `~/Lettres/projets`”. (Si ce dossier n’existe pas encore, `mailx` le crée.)

Copiez la lettre dans un dossier en tapant :

```
& c 3 +projets
```

Envoi d’une lettre directement dans un fichier ou un dossier

Vous pouvez envoyer des copies de vos lettres directement dans l’un de vos fichiers ou dossiers. Pour envoyer une copie dans un dossier, il suffit d’entrer le nom du dossier dans la zone `Cc:` ou `Bcc:`. La procédure à suivre est analogue pour l’envoi d’une copie dans un fichier, mais vous devez indiquer le chemin d’accès complet.

Lecture de lettres figurant dans des fichiers et des dossiers

Pour lire les lettres sauvegardées dans un fichier, utilisez la syntaxe :

```
mailx -f nom_fichier
```

D’après l’exemple ci-dessus, vous pourrez lire le fichier `~/memos/finance` en tapant :

```
$ mailx -f ~/memos/finance
```

Vous pouvez lire les lettres sauvegardées dans un dossier à l’aide d’une commande similaire, à ceci près que vous devrez utiliser le signe `+` à la place du chemin d’accès. Par exemple, pour lire les lettres contenues dans le dossier `projets`, vous devrez taper :

```
$ mailx -f +projets
```

Cette commande lance `mailx` en ouvrant le fichier ou le dossier désigné. Seuls les en-têtes des lettres contenus dans ce fichier ou ce dossier sont affichés. Pour sélectionner la lettre à consulter, vous devez taper son numéro à la suite du prompt `mailx` et appuyer sur Return.

Vous pouvez également travailler sur des dossiers de messagerie sous le programme `mailx`. Pour afficher la liste de vos dossiers, tapez la commande suivante au prompt `mailx` :

```
& folders
```

Pour passer de votre boîte à lettres à l'un des dossiers, utilisez la syntaxe :

```
& folder +nom_dossier
```

Pour revenir ensuite à votre boîte à lettres, tapez la commande ci-après à la suite du prompt de la Messagerie :

```
& %
```

Pour retourner au dossier précédent, tapez :

```
& #
```

Utilisation de `vi` avec `mailx`

Vous pouvez utiliser l'éditeur de texte `vi` pour rédiger vos courriers depuis `mailx`. Cet outil vous permet en outre de corriger vos erreurs et d'ajouter ou de supprimer des informations avant d'envoyer vos lettres. Si c'est la première fois que vous utilisez `vi`, reportez-vous au Chapitre 6, "Utilisation de l'éditeur `vi`", pour connaître la procédure à suivre.

Vous pouvez utiliser les commandes `vi` dans la messagerie pour insérer, supprimer et modifier votre texte.

Pour rédiger une lettre avec `vi` :

1. Entrez la commande `mailx`, suivie d'une adresse, soit à la suite du prompt `mailx (&)`, soit à la suite du prompt de commande `($)`.
2. Tapez l'objet sur la ligne `Subject:` et appuyez sur Return.
3. Lancez `vi` en entrant la commande `~v` sur une nouvelle ligne. L'écran `vi` apparaît, représentant un fichier vide de votre répertoire `/tmp`.
4. Utilisez les commandes `vi` pour saisir et éditer le corps de votre lettre.
5. Lorsque vous avez terminé, quittez `vi` avec la commande `:wq` ou `ZZ`.

Une fois que vous avez quitté `vi`, `mailx` affiche le message `(continue)` : vous pouvez alors ajouter la lettre à la liste (en dehors de `vi`) ou l'envoyer en appuyant sur Ctrl-D.

Alias de distribution

Un *alias de distribution* est un nom unique représentant un groupe d'utilisateurs.

Vous pouvez utiliser des alias de distribution lorsque vous envoyez fréquemment des courriers à un même groupe de personnes. Par exemple, si vous écrivez régulièrement à `hank@fretful`, `george@lonesome` et `sally@dakota`, vous pouvez créer un alias de distribution `amigos`. Ainsi, chaque fois que vous déciderez d'envoyer un courrier à `amigos`, il sera transmis à ces trois personnes.

Vous pouvez définir vos alias de distribution à deux endroits différents :

- Dans votre fichier `.mailrc`
- Dans le fichier `/etc/aliases`

Les alias de distribution définis dans `.mailrc` fonctionnent différemment de ceux qui sont définis dans `/etc/aliases`. Ces différences sont résumées dans le Tableau 7-1, situé à la fin de cette section.

Définition des alias de distribution dans `.mailrc`

Pour définir des alias dans le fichier `.mailrc`, tenez compte des points suivants :

- Les alias de distribution du fichier `.mailrc` sont *privés*, c'est-à-dire que vous seul pouvez les utiliser. Par exemple, si vous définissez un alias de distribution appelé `amigos` dans `.mailrc` et qu'un autre utilisateur tente d'envoyer un courrier à `amigos`, il recevra le message d'erreur `unknown user`.
- Lorsque le courrier est envoyé, les alias de `.mailrc` sont automatiquement développés de façon à faire apparaître tous les destinataires de l'alias de distribution. Par exemple, si vous envoyez un courrier à `amigos`, il est transmis comme si vous aviez tapé le nom de chacun des destinataires. Le fait que vous avez utilisé un alias de distribution pour envoyer votre courrier est transparent pour les destinataires.

`.mailrc` se trouve dans votre répertoire utilisateur. Ce fichier contient un certain nombre de paramètres qui déterminent le fonctionnement de `mailx` et de la Messagerie.

Pour ajouter un alias de distribution dans `.mailrc`, tapez :

```
$ vi ~/.mailrc
```

Remarque : vous pouvez éditer le fichier `.mailrc` avec n'importe quel éditeur de texte. Dans l'exemple ci-dessus, l'éditeur utilisé est `vi`. Si c'est la première fois que vous utilisez `vi`, reportez-vous au Chapitre 6, "Utilisation de l'éditeur `vi`", pour connaître la procédure à suivre.

Dans le fichier, chaque alias de distribution doit figurer sur une seule ligne. S'il est trop long, il peut s'afficher sur deux lignes à l'écran, mais ne doit pas comporter de retour chariot. Chaque alias de distribution doit contenir les informations suivantes, séparées par des espaces :

- Le mot "alias"
- Le nom de l'alias de distribution (un mot maximum)
- Les destinataires (noms de connexion et noms de machine) de l'alias de distribution, séparés par des espaces

L'exemple ci-dessous fait apparaître deux alias de distribution. Le premier (`amigos`) contient trois personnes et le second (`softball`), huit personnes. Vous remarquerez dans `softball` que les noms sont automatiquement renvoyés à la ligne à l'écran. Cela n'a pas d'importance, pourvu qu'aucun retour chariot n'ait été inséré.

```
alias amigos hank@fretful george@lonesome sally@dakota
alias softball earl@woofer tex@twister elmer@farmhouse
jane@freeway hank@fretful jj@walker sally@dakota steve@hardway
```

Pour envoyer un courrier aux destinataires d'un alias de `.mailrc`, il vous suffit d'indiquer comme adresse le nom de l'alias de distribution, *sans* préciser le nom de votre machine. Par exemple, si vous envoyez le courrier :

```
$ mail amigos
Subject: Déjeuner

Bonjour à tous. Cela pourrait être sympa de déjeuner ensemble
vendredi. Avis aux amateurs.
```

les destinataires verront apparaître l'écran suivant (notez que la ligne `To:` a été développée) :

```
To: hank@fretful george@lonesome sally@dakota
Subject: Déjeuner

Bonjour à tous. Cela pourrait être sympa de déjeuner ensemble
vendredi. Avis aux amateurs.
```

Définition des alias de distribution dans `/etc/aliases`

Pour définir des alias de distribution dans le fichier `/etc/aliases`, tenez compte des points suivants :

- Les alias de distribution de `/etc/aliases` sont *publics*. Cela signifie que si vous définissez un alias de distribution appelé `softball`, n'importe qui peut envoyer un courrier à `softball@votre-machine` et utiliser cet alias.
- Lorsque le courrier est envoyé, les alias de distribution de `/etc/aliases` ne sont *pas* développés. Par exemple, si vous avez envoyé un courrier à `softball@machine`, l'alias apparaîtra tel quel à réception. Les destinataires connaîtront l'alias de distribution, mais pas nécessairement le nom des autres destinataires.

Les alias de distribution créés dans `/etc/aliases` ont un format légèrement différent de ceux qui sont créés dans `.mailrc`. Chaque alias de `/etc/aliases` doit en effet contenir :

- Le nom de l'alias de distribution, suivi de deux-points (`:`)

- Les destinataires (noms d'accès et noms de machine), séparés par des virgules. Notez que l'alias de distribution ne doit *pas* nécessairement figurer sur une seule ligne.

Pour modifier votre fichier `/etc/aliases`, vous devez préalablement devenir superutilisateur. Si le titre de superutilisateur est protégé par un mot de passe, vous devez connaître ce mot de passe.

Pour devenir superutilisateur du système, tapez :

```
$ su
Password:
#
```

Notez que le prompt de commande change lorsque vous devenez superutilisateur.

L'exemple suivant montre comment ajouter l'alias `softball@texas` dans le fichier `/etc/aliases` par défaut.

```
# vi /etc/aliases
##
#Aliases can have any mix of upper and lower case on the left-
#hand side,
#but the right-hand side should be proper case (usually lower)
#
#    >>>>>>>>The program "newaliases" will need to be run after
#    >> NOTE >>this file is updated for any changes to
#    >>>>>>>>show through to sendmail.
#
#@( # )aliases 1.10 89/01/20 SMI
##
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's
# mail problems.
Postmaster: root

# Alias for mailer daemon; returned messages from our MAILER-
# DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster
```



```
# Aliases to handle mail to programs or files, eg news or vacation
# decode: "|/usr/bin/uudecode"
nobody: /dev/null

# Sample aliases:
# Alias for distribution list, members specified here:
#staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie

# Alias for distribution list, members specified elsewhere:
#keyboards: :include:/usr/jfarrell/keyboards.list

# Alias for a person, so they can receive mail by several names:
#epa:eric

#####
# Local aliases below #
#####
softball@texas: earl@woofer tex@twister elmer@farmhouse
jane@freeway hank@fretful jj@walker sally@dakota steve@hardway
:wq          (to quit vi and save the /etc/aliases file)
# exit      (to exit root)
$
```

Vous pouvez éditer le fichier `/etc/aliases` au moyen de n'importe quel éditeur de texte. Dans l'exemple ci-dessus, l'éditeur utilisé est `vi`. Si c'est la première fois que vous utilisez `vi`, reportez-vous au Chapitre 6, "Utilisation de l'éditeur `vi`", pour connaître la procédure à suivre.

Vous remarquerez que les signes `#` qui figurent dans le fichier `/etc/aliases` ont été insérés à titre de *commentaires* sur le texte et les exemples d'alias. La présence de ces caractères évite que le système ne traite ces informations comme des alias.

N'insérez pas de signes `#` devant les alias que vous ajoutez dans ce fichier, à moins que vous ne vouliez désactiver un de ces alias.

Pour envoyer un courrier aux destinataires d'un alias contenu dans /etc/aliases, vous devez indiquer comme adresse le nom de cet alias, ainsi que le nom de votre machine. Par exemple, si vous envoyez le courrier :

```
$ mail softball@texas
Subject: Entraînement au club ce soir
```

```
Je te propose de te retrouver au club ce soir. A mon avis, il faut
nous entraîner si nous voulons être au point pour samedi. Essaie
de ne pas venir trop tard.
```

les destinataires verront apparaître l'écran suivant :

```
To: softball@texas
Subject: Entraînement au club ce soir
```

```
Je te propose de te retrouver au club ce soir. A mon avis, il faut
nous entraîner si nous voulons être au point pour samedi. Essaie
de ne pas venir trop tard.
```

Notez que la ligne To : n'a *pas* été développée.

A chaque fois que vous envoyez un courrier en utilisant un alias de distribution de ce type, veillez à bien préciser le nom de la machine sur laquelle il réside. Par exemple, si vous avez défini un alias de distribution appelé `riders` sur la machine `freeway`, vous devez envoyer votre courrier à `riders@freeway`.

Le Tableau 7-1 établit une comparaison succincte entre les alias de distribution créés dans `.mailrc` et ceux qui ont été créés dans `/etc/aliases`.

Tableau 7-1 Comparaison des alias de distribution des fichiers `.mailrc` et `/etc/aliases`

	.mailrc	/etc/aliases
Faut-il être superutilisateur (<code>root</code>) pour faire des modifications ?	non	oui
Envoi du message à :	alias	<i>alias@machine</i>
Liste des destinataires visible par ces derniers ?	oui	non
Noms séparés par des virgules ?	non	oui
Tous les noms sur une même ligne ?	oui	non
Utilisation de l'alias de distribution par d'autres utilisateurs ?	non	oui

Pour plus d'informations sur les alias de distribution, tapez `man aliases` ou `man addresses` en regard du prompt du système.

Commandes tilde

Lorsque vous rédigez une lettre, vous pouvez utiliser les commandes tilde pour exécuter certaines fonctions. Ces commandes se présentent sous la forme d'un tilde (~) suivi d'un caractère unique. Le tableau suivant décrit les commandes tilde les plus utiles. Certaines d'entre elles vous ont déjà été présentées dans les sections précédentes de ce chapitre.

Remarque : si vous devez insérer un tilde dans l'une de vos lettres, tapez deux tildes à la suite. Seul l'un des deux sera affiché.

Tableau 7-2 Commandes tilde (`mailx`)

Commande	Fonction
<code>~!commande</code>	Convertit la commande en une commande Shell.
<code>~.</code>	Simule l'activation Ctrl-D pour marquer la fin d'un fichier.
<code>~?</code>	Affiche la liste résumée des commandes tilde.
<code>~b nom_utilisateur</code>	Ajoute un ou plusieurs noms utilisateur à la liste de copies carbone aveugles (Bcc:).
<code>~c nom_utilisateur</code>	Ajoute un ou plusieurs noms utilisateur à la liste de copies carbone (Cc:).
<code>~d</code>	Insère le contenu du fichier <code>dead.letter</code> dans la lettre en cours.
<code>~f numéro</code>	Envoie la lettre indiquée. Valable uniquement pour l'envoi d'un message lors de la lecture d'un courrier.
<code>~h</code>	Affiche les lignes d'en-tête : Subject, To, Cc, and Bcc.
<code>~m numéro</code>	Insère le contenu de la lettre indiquée dans la lettre en cours. Valable uniquement pour l'envoi d'un message lors de la lecture d'un courrier.
<code>~p</code>	Imprime le message saisi à l'écran.
<code>~q</code>	Simule l'activation à deux reprises de Ctrl-C. S'il n'est pas vide, le corps du message en cours est sauvegardé dans <code>dead.letter</code> .
<code>~r nom_fichier</code>	Insère le contenu du fichier indiqué.
<code>~s chaîne</code>	Remplace la ligne Subject par la chaîne indiquée.
<code>~t nom</code>	Ajoute le(s) nom(s) indiqué(s) à la liste des destinataires.
<code>~w nom_fichier</code>	Insère la lettre en cours sans l'en-tête dans le fichier indiqué.
<code>~x</code>	Quitte <code>mailx</code> . Identique à <code>~q</code> , à ceci près que le message n'est pas sauvegardé dans le fichier <code>dead.letter</code> .

Aide en ligne : autres commandes `mailx`

`mailx` contient deux commandes d'aide qui affichent respectivement les listes de commandes et de fonctions. Lorsque vous êtes en mode commande, vous pouvez taper `?` à la suite du prompt `mailx (&)` pour afficher la liste des commandes utilisées dans ce mode. De même, en mode saisie (lorsque vous

écrivez une lettre, par exemple), vous pouvez entrer la commande équivalente `~?` pour afficher la liste des commandes tilde (également appelées "caractères des spécialisés tilde").

Les pages d'aide (man) contiennent elles aussi des informations plus étendues sur `mailx`, mais sous une forme plus technique. Pour afficher ces informations, entrez la commande :

```
$ man mailx
```

ou consultez les *man Pages(1): User Commands*.

Utilisation des imprimantes



Le service d'impression LP (sous-système signifiant *imprimante en ligne*) constitue l'outil d'impression de SunOS. Il offre un large éventail de fonctions, qui ne sont pas toutes présentées dans ce manuel. Le présent chapitre décrit uniquement les procédures nécessaires à l'exécution des quelques tâches élémentaires offertes par ce service d'impression :

- Soumission d'une requête d'impression (envoi d'un fichier à l'imprimante)
- Vérification de l'état d'une imprimante
- Annulation d'une requête d'impression

Une description complète du service d'impression LP vous est proposée dans le manuel *User Accounts, Printers, and Mail Administration*.

Soumission d'une requête d'impression

Pour imprimer un fichier à partir du prompt de commande, vous devez demander son impression à l'imprimante à l'aide de la commande `lp`. Dès sa réception par le service d'impression, la requête est placée dans la file d'attente de l'imprimante, son ID d'impression est affiché et le prompt de l'Utilitaire Shell réapparaît.

Soumission de requêtes d'impression à l'imprimante par défaut

Si vous avez défini une imprimante par défaut pour le service d'impression LP, vous pouvez soumettre des requêtes d'impression de la manière suivante, sans préciser le nom de l'imprimante :

```
$ lp nom_fichier
```

où *nom_fichier* désigne le fichier à imprimer.

Le fichier indiqué est placé dans la file d'attente d'impression de l'imprimante par défaut et l'*ID d'impression* est affiché.

Par exemple, pour imprimer le fichier `/etc/passwd`, tapez :

```
$ lp /etc/passwd
l'ID de la requête est pinecone-8 (1 fichier)
$
```

Pour plus d'informations sur la définition d'une imprimante par défaut, consultez le manuel *User Accounts, Printers, and Mail Administration*.

Soumission de requêtes d'impression à une imprimante donnée

Que vous ayez ou non défini une imprimante par défaut, vous avez la possibilité de soumettre des requêtes d'impression à n'importe quelle imprimante configurée pour votre système. Pour ce faire, tapez :

```
$ lp -d nom_imprimante nom_fichier
```

où *nom_imprimante* désigne le nom de l'imprimante à laquelle vous souhaitez soumettre votre requête et *nom_fichier* le fichier à imprimer.

Le fichier indiqué est placé dans la file d'attente de l'imprimante de destination, et l'*ID d'impression* est affiché.

Par exemple, pour imprimer le fichier `/etc/passwd` sur l'imprimante `acorn`, tapez :

```
$ lp -d acorn /etc/passwd
l'ID de la requête est acorn-9 (1 fichier)
$
```

Lorsque vous soumettez une requête à une imprimante qui n'est pas configurée sur votre système, un message d'information apparaît, comme indiqué dans l'exemple suivant :

```
$ lp -d thorn /etc/passwd
UX:lp: ERROR: Le serveur d'impression LP ne reconnaît pas
              le périphérique de destination "thorn".
$
```

Pour plus d'informations sur la configuration d'une imprimante, consultez le manuel *User Accounts, Printers, and Mail Administration*. Pour connaître les imprimantes disponibles sur votre système, reportez-vous à la section "Vérification de l'état de l'imprimante," dans le présent chapitre.

Demande de notification de fin d'impression

Lorsque vous soumettez l'impression d'un long fichier, vous pouvez demander au service d'impression LP de vous avertir de la fin de l'impression. Cette notification peut vous être adressée de deux manières :

- Par l'intermédiaire de la messagerie électronique
- Par l'affichage d'un message dans la Console

Pour être informé de la fin de l'impression par l'intermédiaire de la messagerie électronique, utilisez l'option `-m` lors de la soumission de la requête d'impression :

```
$ lp -m nom_fichier
```

Pour demander l’affichage d’un message dans la Console, utilisez l’option `-w` lors de la soumission de la requête d’impression :

```
$ lp -w nom_fichier
```

où *nom_fichier* désigne le nom du fichier à imprimer.

Impression en plusieurs exemplaires

Vous pouvez demander l’impression d’un fichier en plusieurs exemplaires. Dans ce cas, le fichier est imprimé autant de fois qu’indiqué dans l’option `-n` de la commande `lp`. La requête d’impression étant considérée comme un travail d’impression unique, une seule page de garde sera imprimée.

Pour demander une impression en plusieurs exemplaires, entrez la commande suivante :

```
$ lp -nnombre nom_fichier
```

où *nombre* désigne le nombre d’exemplaires désirés et *nom_fichier* le fichier à imprimer.

Par exemple, pour imprimer le fichier `/etc/passwd` en quatre exemplaires, tapez :

```
$ lp -n4 /etc/passwd
l’ID de la requête est pinecone-9 (1 fichier)
$
```

Tableau récapitulatif des options lp

Vous pouvez personnaliser votre requête d’impression en utilisant les options de la commande `lp` pour préciser un imprimé, un jeu de caractères, un filtre, un titre, une bannière, etc. Les options les plus fréquemment utilisées de la commande `lp` sont résumées dans le Tableau 8-1. Vous pouvez les utiliser séparément ou bien les combiner dans votre ligne de commande, dans l’ordre souhaité. Si vous indiquez plusieurs options, séparez-les par un espace et répétez le tiret (`-`).

Par exemple, si vous voulez préciser une imprimante de destination, demander la notification par l'intermédiaire de la messagerie électronique et imprimer le fichier en six exemplaires, vous devez taper :

```
$ lp -d nom_imprimante -m -n6 nom_fichier
```

où *nom_imprimante* désigne l'imprimante souhaitée et *nom_fichier* le fichier à imprimer.

Tableau 8-1 Récapitulatif des principales options de `lp`

Option	Description
-d	Destination. Précise le nom de l'imprimante de destination.
-m	Messagerie. Envoie un message électronique au demandeur une fois l'impression terminée.
-n	Nombre. Précise le nombre d'exemplaires à imprimer.
-t	Titre. Précise le titre (imprimé uniquement sur la bannière) d'une requête d'impression.
-o nobanner	Option. Supprime l'impression de la bannière pour une requête donnée.
-h	En-tête. Place un en-tête sur toutes les pages imprimées.
-c	Copie. Copie le fichier avant l'impression.
-w	Ecriture. Affiche un message sur votre terminal une fois le fichier imprimé.

Une liste complète des options vous est fournie dans les *man Pages(1): User Commands*.

Vérification de l'état de l'imprimante

La commande `lpstat` vous permet de connaître l'état du service d'impression LP. Vous pouvez vérifier l'état de vos travaux dans la file d'attente de l'imprimante, déterminer les imprimantes disponibles ou les ID d'impression des travaux que vous souhaitez annuler.

Vérification de l'état de vos requêtes d'impression

Pour connaître l'état de vos requêtes d'impression dans la file d'attente, tapez:

```
$ lpstat
```

pour obtenir une liste des fichiers envoyés en impression.

Dans l'exemple suivant, sur le système `pine`, un seul fichier figure dans la liste d'attente de l'imprimante `pinecone` :

```
$ lpstat
pinecone-10          fred          1261   Mar 12 17:34 on pine
$
```

La commande `lpstat` affiche une ligne pour chaque travail d'impression, contenant l'ID d'impression, puis le nom de l'utilisateur qui l'a soumis, la taille en octets, ainsi que la date et l'heure de soumission de la requête.

Recherche des imprimantes disponibles

Pour connaître les imprimantes configurées pour votre système, tapez :

```
$ lpstat -s
```

L'état de l'ordonnanceur s'affiche, suivi de la destination par défaut et d'une liste des systèmes et imprimantes auxquels vous avez accès.

Dans l'exemple suivant, sur le système `elm`, l'ordonnanceur est activé, l'imprimante par défaut est `pinecone` et les deux imprimantes `pinecone` et `acorn` sont disponibles en réseau :

```
$ lpstat -s
l'ordonnanceur fonctionne
imprimante par défaut : pinecone
serveur d'impression de pinecone : pine
serveur d'impression d'acorn : oak
$
```

Affichage d'informations sur l'état

L'option `-t` de `lpstat` affiche un bref compte-rendu de l'état du service d'impression LP.

Pour l'afficher, tapez :

```
$ lpstat -t
```

Toutes les informations disponibles s'affichent.

Dans l'exemple suivant, la file d'attente de l'imprimante ne contient aucun travail. Lorsque les fichiers sont placés dans la file, l'état de leur requête d'impression s'affiche également :

```
$ lpstat -t
l'ordonnanceur fonctionne
imprimante par défaut : pinecone
serveur d'impression d'acorn : oak
pinecone accepte les requêtes depuis Wed Jan  2 18:20:10 PST 1991
acorn accepte les requêtes depuis Mon Mar  4 15:53:47 PST 1991
l'imprimante pinecone est inutilisée. est activée depuis Wed Jan
2 18:20:22 PST 1991. disponible.

l'imprimante acorn est inutilisée. activée depuis Mon Mar  4
15:53:44 PST 1991. disponible.
$
```

Affichage de l'état des imprimantes

Vous pouvez demander des informations sur l'état de chacune des imprimantes à l'aide de l'option `-p` de `lpstat`. Cette option indique si l'imprimante est active ou non, à quel moment elle a été mise en service ou hors service et si elle est disponible pour recevoir des requêtes d'impression.

Pour demander l'état de toutes les imprimantes d'un système, tapez :

```
$ lpstat -p
```

Dans l'exemple suivant, deux imprimantes sont inactives, en service et disponibles. Si l'une d'elles a des travaux dans sa file d'attente, ces derniers sont également affichés.

```
$ lpstat -p
l'imprimante pinecone est inutilisée. activée depuis Wed Jan  2
18:20:22 PST 1991. disponible.
l'imprimante acorn est inutilisée. activée depuis Mon Mar  4
15:53:44 PST 1991. disponible.
$
```

Pour connaître l'état d'une imprimante donnée en indiquant son nom, tapez :

```
$ lpstat -p nom_imprimante
```

où *nom_imprimante* désigne l'imprimante dont vous souhaitez connaître l'état.

Affichage des caractéristiques d'une imprimante

Si vous voulez connaître toutes les caractéristiques d'une imprimante, utilisez l'option `-p` en même temps que l'option `-l` (long) de `lpstat`. Cette commande est particulièrement utile pour rechercher le type d'une imprimante et son type de contenu.

Pour afficher les caractéristiques de toutes les imprimantes, tapez :

```
$ lpstat -p -l
```

Toutes les informations relatives à la configuration du service d'impression LP pour chaque imprimante s'affichent sous forme de tableau.

Dans l'exemple suivant, toutes les zones sont à blanc, sauf celles indiquant le type de l'imprimante pinecone et son type de contenu.

```
$ lpstat -p pinecone -l
l'imprimante pinecone est inutilisée. activée depuis Wed Jan  2
18:20:22 PST 1991. disponible.
    Types de contenu : PS
    Types d'imprimante : PS
    Description :
    Utilisateurs autorisés :
        (tous)
    Imprimés autorisés :
        (aucun)
    Pas de bannière
    Jeux de caractères :
        (aucun)
    Interligne par défaut :
    Taille par défaut de la page :
```

```
$
```

Tableau récapitulatif des options de la commande lpstat

La commande `lpstat` vous permet de demander plusieurs types d'informations liées à l'état de l'impression. Le Tableau 8-2 offre un récapitulatif des options les plus utilisées de la commande `lpstat`. Vous pouvez utiliser ces options séparément ou les combiner dans votre ligne de commande, dans l'ordre souhaité. Si vous combinez des options, séparez-les par un espace et répétez le tiret (-).

Par exemple, pour afficher un long compte-rendu d'état pour une imprimante donnée, tapez :

```
$ lpstat -p nom_imprimante -l
```

où *nom_imprimante* désigne l'imprimante dont vous souhaitez connaître l'état.

Tableau 8-2 Récapitulatif des principales options de la commande `lpstat`

Option	Description
-a	Acceptation. Indique si les destinataires de l'impression acceptent les requêtes.
-c	Classe. Indique les classes et leurs membres.
-d	Destination. Indique la destination par défaut.
-f	Imprimés. Indique les imprimés.
-o	Sortie. Indique l'état de la sortie.
-p [<i>compte-rendu</i>] [D] [-1]	Imprimante/Description/Long compte-rendu. Indique l'état des imprimantes.
-r	Demande. Demande l'état de l'ordonnanceur.
-R	Indique la position du travail dans la file d'attente.
-s	Etat. Indique un résumé des états.
-S	Jeux. Indique les jeux de caractères.
-u [<i>nom_utilisateur</i>]	Utilisateur. Indique les requêtes par utilisateur
-v	Indique les périphériques.

Pour la liste complète des options, consultez les *man Pages(1): User Commands*.

Annulation d'une requête d'impression

La commande `cancel` permet d'annuler une requête d'impression placée en file d'attente ou en cours d'impression. Pour annuler une requête d'impression, vous devez connaître son ID d'impression. Celui-ci contient toujours le nom de l'imprimante, un tiret et un numéro. Il s'affiche lorsque vous soumettez une requête d'impression. Si vous ne vous souvenez pas de votre ID d'impression, tapez `lpstat` et appuyez sur Return. Seul l'utilisateur ayant soumis la requête ou une personne connectée sous l'ID connexion `root` (superutilisateur) ou `lp` est autorisé à annuler une requête d'impression.

Annulation d'une requête d'impression par numéro d'ID

Pour annuler une requête d'impression, tapez :

```
$ cancel ID_requête
```

où *ID_requête* désigne le numéro de l'ID d'impression que vous souhaitez annuler.

Un message s'affiche pour vous indiquer que la requête d'impression est annulée. L'impression du travail suivant de la file d'attente démarre.

Dans l'exemple suivant, deux requêtes d'impression sont annulées :

```
$ cancel pinecone-3 pinecone4  
la requête "pinecone-3" est annulée  
la requête "pinecone-4" est annulée  
$
```

Annulation d'une requête d'impression par nom d'imprimante

Vous pouvez également annuler uniquement le travail en cours d'impression (si vous l'avez soumis) en entrant le nom de l'imprimante à la place de l'ID d'impression :

```
$ cancel nom_imprimante
```

où *nom_imprimante* désigne le nom de l'imprimante à laquelle vous envoyez la requête d'impression.

Un message s'affiche pour vous indiquer que la requête d'impression est annulée. L'impression du travail suivant de la file d'attente démarre.

Dans l'exemple suivant, la requête en cours d'impression est annulée :

```
$ cancel pinecone  
la requête "pinecone-3" est annulée  
$
```

Votre administrateur système peut se connecter sous l'ID connexion `root` (superutilisateur) ou `lp` et annuler la requête en cours d'impression en utilisant le nom de l'imprimante comme argument de la commande `cancel`.

Utilisation du réseau



Un *réseau* est un groupe d'ordinateurs configurés pour communiquer les uns avec les autres. Lorsque votre machine fait partie d'un réseau, vous pouvez utiliser les ressources des autres machines du réseau tout en restant connecté à la vôtre. Vous pouvez vous connecter à d'autres machines ou exécuter depuis votre propre poste de travail des commandes à distance qui affectent les autres machines.

Ce chapitre traite des notions suivantes :

- Généralités sur les réseaux
- Connexion à des machines distantes
- Copie de fichiers à distance
- Exécution de commandes sur des machines distantes
- Demande d'informations d'état sur des machines distantes

Si votre machine n'est pas connectée à un réseau, les informations qui sont présentées ici ne vous concernent pas. Toutefois, il peut être intéressant pour vous de parcourir brièvement ce chapitre pour avoir un aperçu des avantages que peut procurer la connexion en réseau.

Généralités sur les réseaux

Une connexion en réseau permet l'échange d'informations entre les différentes machines du réseau. Les réseaux sont souvent répartis par catégorie en : *réseaux locaux* (LANs) qui couvrent de courtes distances ne dépassant pas

généralement quelques centaines de mètres ; *réseaux longue distance* (WAN) pouvant atteindre plusieurs milliers de kilomètres ; et *réseaux campus* (CAN), de taille intermédiaire.

Un réseau composé de plusieurs réseaux interconnectés est appelé *interréseau*. Par exemple, votre machine peut faire partie à la fois d'un réseau local interne à votre bâtiment et d'un interréseau reliant ce réseau local à d'autres du même type, répartis sur l'ensemble du pays. Comme la différence entre un réseau et un interréseau est généralement imperceptible pour l'utilisateur, ce manuel désigne communément sous le terme "réseau" l'une et l'autre de ces deux notions.

Les machines connectées à un réseau utilisent pour communiquer un *protocole de réseau* ou un langage de réseau commun afin que les informations soient bien transmises aux adresses appropriées. Les *protocoles d'interréseau*, souvent appelés *relais*, relient, pour leur part, des réseaux entre eux.

Connexion à distance (`rlogin`)

La commande `rlogin` vous permet de vous connecter à d'autres machines UNIX de votre réseau.

Pour vous connecter à distance à une autre machine, tapez :

```
$ rlogin nom_machine
```

où *nom_machine* désigne le nom de la machine distante.

Si un prompt de mot de passe apparaît, tapez le mot de passe de la machine distante et appuyez sur Return. Si le nom de votre machine se trouve dans le fichier `/etc/hosts.equiv`, l'autre machine le "reconnaît" et vous n'avez pas à taper le mot de passe.

```
$ rlogin lonesome
Password: (type password)
Last login: Mon Jan 6 09:37:55 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
( Les commandes suivantes ont été exécutées sur lonesome.)
$ pwd
/home/keithp
$ logout
Connection closed.
$
```

`rlogin` sans répertoire utilisateur

Dans l'exemple ci-dessus, l'utilisateur `keithp` est connecté à `lonesome` dans le répertoire `/home/keithp`, comme indiqué par la commande `pwd`. Lorsque vous vous connectez à une machine sur laquelle vous ne disposez pas de répertoire utilisateur, `rlogin` vous le signale par un message et vous connecte dans le répertoire racine (`/`) de cette machine :

```
$ rlogin fretful
Password:
No directory! Loggin in with home=/
Last login: Fri Jan 3 10:21:59 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
( Les commandes suivantes ont été exécutées sur fretful.)
$ pwd
/
$ logout
Connection closed.
$
```

rlogin *sous une autre identité*

Il peut arriver que vous vouliez vous connecter à une machine distante sous une autre identité que la vôtre. Par exemple, si vous travaillez sur la machine d'une autre personne (sous son nom utilisateur) et que vous voulez vous connecter à votre machine sous votre propre identité. L'option `-l` de `rlogin` vous le permet. La syntaxe de cette commande est la suivante :

```
rlogin nom_machine -l nom_utilisateur
```

L'exemple suivant montre comment l'utilisateur `keithp` de la machine `blue` se connecte à la machine `lonesome` sous le nom `earl` :

```
$ rlogin lonesome -l earl
Password:
Last login: Wed Jan 8 07:12:25 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
( Les commandes suivantes ont été exécutées sur lonesome.)
$ pwd
/home/earl
$ logout
Connection closed.
$
```

Notez que lorsque vous vous connectez à une machine distante sous l'identité de quelqu'un d'autre, vous accédez au répertoire de connexion de cet utilisateur.

rlogin *à une machine inconnue*

Si vous tentez de vous connecter à une machine distante dont le nom est inconnu pour la vôtre, `rlogin` consulte en vain la base de données des ordinateurs centraux et affiche le message suivant :

```
$ rlogin stranger
stranger: unknown host
$
```

Abandon d'une connexion `rlogin`

Pour mettre fin à une connexion à distance, vous devez normalement taper `logout` à la fin de votre session de travail. Si toutefois pour une raison quelconque vous ne pouvez clore votre session de cette manière, vous pouvez abandonner la connexion en tapant le caractère tilde, suivi d'un point (`~.`), au début d'une ligne. La connexion à la machine distante prend alors fin et vous revenez à votre machine de départ.

Si vous vous connectez en chaîne à plusieurs machines, c'est-à-dire en accédant à chacune d'entre elles via une machine donnée, et que vous utilisez la commande `~.` pour abandonner la connexion à l'une de ces machines, vous revenez à votre machine de départ :

```
$ rlogin dakota
Password:
Last login: Fri Jan 10 09:14:43 from blue
Sun Microsystems, Inc.      SunOS 5.1      October 1992
( Les commandes suivantes ont été exécutées sur dakota. )
$ ~. ( Le caractère ~ peut ne pas apparaître à l'écran. )
Connection closed.
$
```

Si vous voulez revenir à l'une des connexions intermédiaires, tapez deux tilde suivis d'un point (`~~.`), comme indiqué ci-après :

```
$ rlogin lonesome
Password:
Last login: Tue Jan 7 08:12:49 from blue
Sun Microsystems, Inc.      SunOS 5.1      October 1992
( La commande suivante a été exécutée sur lonesome. )
$ rlogin dakota
Password:
Last login: Tue Jan 7 10:17:40 from lonesome
Sun Microsystems, Inc.      SunOS 5.1      October 1992
( La commande suivante a été exécutée sur dakota. )
$ ~~. ( Les caractères ~~ peuvent ne pas apparaître à l'écran. )
Connection closed.
$
```

Suspension d'une connexion `rlogin`

Lorsque vous voulez suspendre provisoirement une connexion à distance pour y revenir ultérieurement, tapez le caractère tilde (~), suivi de Ctrl-Z. Le processus de connexion à distance est alors stoppé et vous revenez à la machine depuis laquelle vous vous êtes connecté.

Pour réactiver la connexion, tapez `fg`. Vous pouvez également taper le signe pourcentage (%), suivi du numéro du processus qui a été stoppé (en l'absence de numéro, le processus réactivé par défaut selon cette méthode est le dernier qui a été suspendu).

```
$ rlogin lonesome
Password:
Last login: Tue Jan 7 08:12:49 from blue
Sun Microsystems, Inc. SunOS 5.1 October 1992
( La commande suivante a été exécutée sur lonesome.)
~^Z ( Les caractères ^Z peuvent ne pas apparaître à l'écran..)
Stopped
( La commande suivante a été exécutée sur blue.)
$ pwd
/home/keithp
$ %
rlogin lonesome

( La commande suivante a été exécutée sur lonesome.)
$ logout
Connection closed.
$
```

Analogue à la commande d'abandon `~~.`, la commande double tilde Ctrl-Z suspend la connexion `rlogin` en cours et vous ramène à une connexion intermédiaire.

Vérification de votre lieu de connexion (`who am i`)

Après vous être connecté à plusieurs machines distantes, sous des noms de connexion différents, vous pouvez être appelé à vérifier l'endroit exact où vous vous trouvez. La commande `who am i` affiche le nom de la machine à laquelle vous êtes actuellement connecté, ainsi que votre identité.

Tapez `man rlogin` au prompt de commande, ou consultez les *man Pages(1): User Commands*.

Copie de fichiers à distance (rcp)

La commande `rcp` vous permet de copier des fichiers d'une machine à une autre. Elle utilise les fichiers `/etc/hosts.equiv` et `/etc/passwd` de la machine distante afin de déterminer la validité de vos droits d'accès. La syntaxe de `rcp` est analogue à celle de `cp`.

Remarque : pour copier des sous-répertoires ainsi que leur contenu d'une machine à une autre, utilisez `rcp -r`.

Copie de fichiers d'une autre machine vers la vôtre

Pour effectuer une copie d'une machine distante vers votre machine, vous devez utiliser la syntaxe suivante :

```
rcp nom_machine:source destination
```

où *nom_machine* désigne le nom de la machine distante, *source* le nom du (ou des) fichier(s) à copier et *destination* le chemin d'accès au répertoire de votre machine dans lequel vous voulez copier ce(s) fichier(s).

L'exemple suivant montre comment copier le fichier `/home/dakota/doc/letter` de la machine distante `dakota` dans le répertoire `/tmp` de la machine locale `blue`:

```
$ rcp dakota:/home/dakota/doc/letter /tmp
$
```

Lorsque vous utilisez `rcp`, vous pouvez également combiner différentes abréviations et syntaxes. Par exemple, pour copier tous les fichiers portant l'extension `.doc` du répertoire utilisateur `hank` de la machine distante `fretful` dans le répertoire courant de la machine locale `blue`, vous devez taper :

```
$ rcp fretful:~hank/*.doc .
$
```

Copie de fichiers de votre machine vers une autre

Pour effectuer une copie de votre machine locale vers une machine distante, vous devez inverser la syntaxe, comme indiqué ci-après :

```
rcp source nom_machine:destination
```

où *source* désigne le(s) fichier(s) à copier, *nom_machine* le nom de la machine distante et *destination* le chemin d'accès au répertoire de la machine distante dans lequel vous voulez copier ce(s) fichier(s).

L'exemple suivant montre comment copier le fichier `austin` de votre répertoire `~/usa/texas` dans le répertoire `~hank/cities` de la machine éloignée `fretful` (n'oubliez pas que `~` est votre répertoire utilisateur et `~hank` celui de l'utilisateur `hank`) :

```
$ rcp ~/usa/texas/austin fretful:~hank/cities
$
```

Pour plus d'informations sur la commande `rcp` et ses options, consultez les *man Pages(1): User Commands*.

Exécution de commandes à distance (rsh)

La commande `rsh` (*remote shell*) vous permet d'exécuter une commande unique sur une machine éloignée sans avoir à vous connecter de façon formelle. Elle vous permet d'économiser un temps précieux lorsque vous n'avez à exécuter qu'une opération sur cette machine.

Pour exécuter une commande sur une machine éloignée, tapez :

rsh *nom_machine commande*

L'exemple suivant illustre le contenu du répertoire
/home/lonesome/guitar tel qu'il apparaît sur la machine lonesome :

```
$ rsh lonesome ls /home/lonesome/guitar
collings      gibson      santacruz
fender        martin      taylor
$
```

Comme les commandes rlogin et rcp, rsh utilise les fichiers
/etc/hosts.equiv et /etc/passwd de la machine éloignée pour
déterminer si vous disposez de droits d'accès appropriés.

Pour plus d'informations sur la commande rsh et ses options, consultez les
man Pages(1): User Commands.

Affichage d'informations utilisateur (rusers)

La commande rusers (*remote users*) affiche la liste des utilisateurs connectés
aux autres machines du réseau. Utilisée seule, la commande rusers affiche la
liste de toutes les machines du réseau et des utilisateurs qui leur sont
connectés, comme illustré ci-après :

```
$ rusers
aspen      susan
blue       keithp
dakota     sally
farmhouse  elmer
freeway    lindab    johnj     karenm
fretful    hank
lonesome   george
twister    tex
$
```

Notez que trois utilisateurs sont connectés à la machine freeway.

Pour afficher des informations sur une machine spécifique, tapez la commande `rusers`, suivie du nom de la machine, comme illustré ci-après :

```
$ rusers freeway
freeway      lindab      johnj      karenm
$
```

L'option `-l` de la commande `rusers` permet d'afficher des informations plus détaillées, comme les noms des utilisateurs, des machines et des terminaux, l'heure de connexion de chaque utilisateur, son temps d'inactivité (s'il est supérieur à une minute) et, le cas échéant, le nom des machines depuis lesquelles se sont connectés les utilisateurs :

```
$ rusers -l freeway
lindab      freeway:ttyd8      Feb 10 08:12      5:29
johnj      freeway:console    Feb 10 09:16
karenm      freeway:ttyp0      Feb 10 11:56      36
$
```

Vous pouvez également utiliser l'option `-l` sans indiquer de nom de machine.

Pour plus d'informations sur la commande `rusers` et ses options, consultez les *man Pages(1): User Commands*.

Personnalisation de votre environnement de travail

10 

Le système d'exploitation SunOS vous permet de gérer et de définir la plupart des caractéristiques de votre environnement de travail. Ces opérations s'effectuent en modifiant les *variables d'environnement* des *fichiers d'initialisation* de votre système. Lorsque vous vous connectez, le système lit les fichiers d'initialisation et utilise les variables d'environnement pour configurer le système. En définissant les variables d'environnement, vous pouvez "personnaliser" votre système pour travailler plus rapidement et plus efficacement.

Ce chapitre explique comment personnaliser votre système en modifiant les fichiers d'initialisation et en définissant les variables d'environnement les plus courantes. Il fournit également les procédures à suivre pour personnaliser les commandes SunOS, modifier le prompt de votre système, définir des droits d'accès par défaut aux fichiers et personnaliser les polices OpenWindows.

Fichiers d'initialisation

Les fichiers d'initialisation utilisés pour la configuration de votre système dépendent du Shell par défaut défini par l'administrateur système lors de l'installation de votre système. Le Bourne Shell est le Shell par défaut du système d'exploitation SunOS, mais vous pouvez également utiliser le C Shell ou le Korn Shell. Chacun de ces Shells possède son ou ses propre(s) fichier(s) d'initialisation.

Si vous ne savez pas exactement quel est votre Shell par défaut (en d'autres termes votre *Shell de connexion*) :

1. Tapez `echo $SHELL`:

```
$ echo $SHELL
/bin/sh
```

2. Si le résultat est le suivant :

- `/bin/sh` : votre Shell de connexion est le Bourne Shell
- `/bin/csh` : votre Shell de connexion est le C Shell
- `/bin/ksh` : votre Shell de connexion est le Korn Shell

Lorsque vous vous connectez à votre système, celui-ci exécute généralement le fichier profil système, `/etc/profile`, indépendamment du Shell utilisé. En règle générale, ce fichier appartient à l'administrateur système. Il est accessible à tous les utilisateurs, en lecture mais non en écriture.

Après avoir exécuté le profil système, votre système exécute le *profil utilisateur*, qui est constitué d'un ou de plusieurs fichiers d'initialisation définissant votre environnement de travail. Par exemple, si vous êtes dans l'environnement OpenWindows, votre système vérifie ce fichier (ou ce groupe de fichiers) chaque fois que vous ouvrez une fenêtre Utilitaire Shell ou Utilitaire de commande.

Selon le Shell qui a été défini par défaut, votre profil utilisateur peut être l'un des suivants :

- `.profile` (pour le Bourne Shell et le Korn Shell)
- `.login` et `.cshrc` (pour le C Shell)

Le(s) fichier(s) profil utilisateur se trouvent dans votre répertoire utilisateur et vous permettent de configurer votre environnement de travail à votre convenance.

Variables d'environnement

Votre environnement système est défini d'après les spécifications figurant dans les fichiers d'initialisation. Pour modifier temporairement votre environnement (pour la session de travail en cours), vous pouvez émettre des commandes

directement au prompt de commande. Toutefois, si vous souhaitez modifier votre environnement de travail de manière plus permanente, vous pouvez enregistrer les variables d'environnement "permanentes" dans les fichiers `.profile`, `.login`, ou `.cshrc`.

Pour afficher les variables d'environnement en cours pour votre système :

1. Tapez la commande `env` et appuyez sur Retour :

```
$ env
HISTORY=100
HOME=/home/texas/keith
HZ=100
LOGNAME=keith
MAIL=/var/mail/keith
MANSECTS=\1:1m:1c:1f:1s:1b:2:\3:3c:3i:3n:3m:3k:3g:3e:3x11:3xt:3
w:3b:9:4:5:7:8
PATH=/usr/bin
SHELL=/bin/sh
TERM=sun
TZ=EST5EDT
```

Remarque : vous pouvez également utiliser la commande `env` pour définir votre Shell de connexion. Celui-ci est précisé par la variable d'environnement `SHELL`. Le Shell défini dans l'exemple ci-dessus est `/bin/sh` (Bourne Shell).

Le profil utilisateur

Cette section décrit certaines variables d'environnement, parmi les plus fréquemment utilisées. Il est possible que bon nombre de ces variables soient déjà dans votre profil utilisateur. Comme indiqué précédemment, votre fichier profil utilisateur (`.profile` pour le Bourne Shell et le Korn Shell, et `.cshrc` pour le C Shell) se trouve dans votre répertoire utilisateur.

Remarque : pour obtenir la liste des fichiers cachés ("fichiers point"), utilisez la commande `ls -la`.

La liste suivante présente une partie des variables d'environnement qui peuvent être incluses dans votre profil utilisateur. La syntaxe de définition des variables d'environnement dépend du Shell que vous utilisez :

- **CDPATH** : indique les répertoires dans lesquels le système doit rechercher lorsque vous indiquez un nom de répertoire sans chemin d'accès complet.
- **HISTORY** : définit le nombre de commandes disponibles pour la commande `history` (pour le C Shell uniquement).
- **HOME** : définit le chemin d'accès absolu de votre répertoire utilisateur. Le système utilise ces informations pour déterminer le répertoire dans lequel vous souhaitez passer lorsque vous tapez la commande `cd` sans aucun argument.
- **LANG** : définit la langue locale. Les valeurs à utiliser sont les suivantes : Japanese, German, French, Swedish et Italian.
- **LOGNAME** : définit votre nom de connexion. Par défaut, cette variable utilise automatiquement le nom de connexion indiqué dans la base de données `passwd`. Ce nom fait partie intégrante du processus de connexion. Pour plus d'informations sur la base de données `passwd`, consultez le manuel *User Accounts, Printers, and Mail Administration*.
- **LPDEST** : définit votre imprimante par défaut.
- **MAIL** : définit le chemin d'accès de votre boîte aux lettres, qui se trouve généralement dans le répertoire `/var/mail/nom_utilisateur`, où *nom_utilisateur* désigne votre nom de connexion. Pour plus d'informations sur ce fichier, reportez-vous au Chapitre 7, "Utilisation de la messagerie".
- **MANSECTS** : définit les sections disponibles de la documentation d'aide en ligne.
- **PATH** : établit la liste ordonnée des répertoires dans lesquels le système recherche le programme à exécuter lorsque vous entrez une commande. Si le nom du répertoire adéquat ne figure pas dans cette liste, vous devez l'indiquer ou taper le chemin d'accès complet chaque fois que vous entrez une commande.

Par défaut, cette variable utilise automatiquement la valeur indiquée dans votre fichier `.profile` (Bourne Shell ou Korn Shell), ou `.cshrc` (C Shell). Cette valeur fait partie intégrante du processus de connexion.

- `PS1` : définit votre prompt de commande. Le prompt par défaut du Bourne Shell et du Korn Shell est le signe dollar (\$), et celui du C Shell, le signe pourcent (%). Le prompt par défaut de la racine dans ces deux Shells est le signe dièse (#).
- `SHELL` : définit le Shell utilisé par `vi` et par les autres utilitaires.
- `TERMINFO` : définit le chemin d'accès d'un terminal non supporté et intégré à la base de données `terminfo`. Il n'est pas nécessaire de définir cette variable pour les terminaux par défaut dans cette base de données. Pour plus d'informations sur la base de données `terminfo`, consultez le manuel *User Accounts, Printers, and Mail Administration*.
- `TERM` : définit les caractéristiques du terminal que vous utilisez. Lorsque vous exécutez un éditeur, le système recherche un fichier ayant le même nom que celui qui a été indiqué cette variable. Il recherche tout d'abord dans le chemin (éventuel) référencé par la variable `TERMINFO`, puis dans le répertoire par défaut, `/usr/share/lib/terminfo`, afin de déterminer les caractéristiques du terminal. Si la recherche est infructueuse, le terminal est identifié comme "non intelligent."
- `TZ` : définit le fuseau horaire pour votre horloge système.

Définition de la variable `PATH`

La variable d'environnement `PATH` est utilisée pour localiser les commandes dans l'arborescence des répertoires SunOS. En définissant la variable `PATH`, vous créez un ensemble fixe de répertoires dans lesquels le système recherche systématiquement les fichiers à utiliser lorsque vous entrez le nom d'une commande.

Par exemple, si vous n'avez pas défini de variable `PATH` et que vous souhaitez copier un fichier, vous devez entrer le chemin d'accès complet de la commande `cp`, à savoir `/usr/bin/cp`. En revanche, si la variable `PATH` que vous avez définie comporte le répertoire `/usr/bin`, il vous suffit de taper `cp` pour que votre système exécute la commande. En effet, le système recherche la commande `cp` dans chaque répertoire mentionné dans la variable `PATH`, et l'exécute lorsqu'il l'a trouvée. Lorsque la variable `PATH` comporte les répertoires contenant les commandes système SunOS les plus courantes, vous pouvez travailler plus rapidement.

Pour le Bourne Shell et le Korn Shell, la variable `PATH` figure dans le fichier `.profile` de votre répertoire utilisateur et utilise la syntaxe suivante :

```
PATH=.: /usr/bin: /répertoire_utilisateur/bin
```

où *répertoire_utilisateur* désigne le chemin d'accès de votre répertoire utilisateur.

Pour le C Shell, la variable `PATH` figure dans le fichier `.cshrc` de votre répertoire utilisateur et utilise la syntaxe suivante :

```
set path=(. /usr/bin répertoire_utilisateur/bin)
```

où *répertoire_utilisateur* désigne le chemin d'accès de votre répertoire utilisateur.

Remarque : dans le C Shell, vous pouvez utiliser le raccourci `~` pour représenter le chemin d'accès de votre répertoire utilisateur.

Si vous modifiez la variable `PATH` pendant que vous exécutez le C Shell, utilisez la commande `source` pour que les modifications soient prises en compte dans la fenêtre en cours sans que vous ayez à sortir du système :

```
example% source .cshrc
```

Si vous êtes en train d'exécuter le Bourne Shell ou le Korn Shell, tapez la commande suivante pour que les modifications soient prises en compte dans la fenêtre en cours sans que vous ayez à sortir du système :

```
$ . .profile
```

Alias (C Shell uniquement)

Les alias sont des raccourcis utiles lorsque vous tapez fréquemment certaines commandes. Par exemple, la commande de suppression (`rm`) définie par défaut ne vous demande pas de confirmer la suppression des fichiers. Cela est parfois

gênant car une erreur de frappe peut entraîner la suppression d'un autre fichier que le fichier souhaité. Cependant, le C Shell vous permet d'utiliser la variable `alias` pour remédier à cet inconvénient, en ajoutant la ligne suivante à votre fichier `.cshrc` :

```
alias rm 'rm -i'
```

Avec cette ligne dans le fichier `.cshrc`, la commande `rm` donne le même résultat que `rm -i`, qui est la forme interactive de `rm`. Le système vous demande systématiquement de confirmer la commande avant toute suppression de fichiers. (Dans l'exemple ci-dessus, les apostrophes entourant la commande `rm -i` servent à inclure l'espace entre `rm` et `-i`. Sans ces apostrophes, le C Shell ne peut pas interpréter correctement le texte figurant après l'espace.)

Pour que les modifications que vous avez apportées au fichier `.cshrc` soient immédiatement prises en compte dans la fenêtre en cours, utilisez la commande `source`. Cette commande demande au système de lire le fichier `.cshrc` et d'exécuter les commandes qui s'y trouvent :

```
example% source .cshrc
```

Modification de votre prompt de commande

La syntaxe qui vous permet de modifier votre prompt de commande est différente selon que vous utilisez le Bourne Shell, le Korn Shell ou le C Shell.

Bourne Shell et Korn Shell

Pour le Bourne Shell ou le Korn Shell, redéfinissez votre prompt de commande à l'aide de la commande `PS1`. Trois exemples figurent ci-dessous :

```
PS1=": "
PS1="'hostname': "
PS1="'hostname'{'id'}": "
```

- Le premier exemple définit comme prompt le signe deux-points (:) suivi d'un espace.
- Le deuxième exemple définit comme prompt le nom de votre machine suivi de deux-points et d'un espace.
- Le troisième exemple définit comme prompt le nom de votre machine suivi de votre nom de connexion entre accolades {}, de deux-points et d'un espace.

Tapez l'un des exemples ci-dessus pour modifier le prompt de commande en cours. Le prompt restera le même jusqu'à ce que vous le changiez à nouveau ou que vous sortiez du système.

Si vous souhaitez que vos modifications soient permanentes, ajoutez l'un des exemples ci-dessus (ou un prompt que vous aurez créé) à votre fichier `.profile`. Le prompt que vous avez indiqué apparaîtra à chaque demande de connexion ou à chaque lancement d'un nouveau Shell.

C Shell

Pour le C Shell, personnalisez votre prompt de commande à l'aide de la commande `set prompt`. Trois exemples figurent ci-dessous :

```
set prompt="% "
set prompt="'hostname'\!: "
set prompt="'hostname'{'id'}": "
```

- Le premier exemple définit comme prompt le signe pourcent suivi d'un espace.
- Le deuxième exemple définit comme prompt le nom de votre machine suivi du numéro d'historique de la commande (hostname1, hostname2, hostname3, et ainsi de suite).
- Le troisième exemple définit comme prompt le nom de votre machine suivi de votre nom de connexion entre accolades, de deux-points et d'un espace.

Tapez l'un des exemples ci-dessus pour modifier votre prompt de commande. Le prompt restera le même jusqu'à ce que vous le changiez à nouveau ou que vous sortiez du système.

Si vous souhaitez que vos modifications soient permanentes, ajoutez l'un des exemples ci-dessus (ou un prompt que vous aurez créé) à votre fichier `.cshrc`. Le prompt que vous avez indiqué apparaîtra à chaque demande de connexion ou à chaque lancement d'un nouveau Shell.

Autres variables d'environnement utiles

Bon nombre d'autres variables d'environnement peuvent être définies dans votre fichier `.profile` ou `.cshrc`. Pour en obtenir la liste exhaustive, consultez les *man Pages(1): User Commands*. Les exemples suivants sont de brèves descriptions des options les plus couramment utilisées.

La variable `set noclobber` empêche tout écrasement involontaire des fichiers lorsque vous utilisez la commande `cp` pour copier un fichier. Cette variable concerne uniquement le C Shell. Entrez les commandes suivantes dans votre fichier `.cshrc` :

```
set noclobber
```

La variable `set history` permet de définir le nombre de commandes enregistrées dans votre liste d'historique. La commande `history` est utile pour visualiser les commandes que vous avez entrées précédemment. Le

fichier d'historique peut également être utilisé pour répéter des commandes antérieures. Cette variable concerne uniquement le C Shell. Entrez les commandes suivantes dans votre fichier `.cshrc` :

```
set history=100
```

En ce qui concerne le Bourne Shell et le Korn Shell, vous pouvez obtenir le même résultat en entrant la commande suivante dans votre fichier `.profile` :

```
HISTORY=100
```

Définition de droits d'accès par défaut

La commande `umask` permet de définir des droits d'accès par défaut pour l'ensemble des fichiers et des répertoires que vous créez. Si vous souhaitez, par mesure de sécurité, accorder aux membres de votre groupe ou à tous les utilisateurs, un droit de lecture et d'exécution uniquement (`-rwxr-xr-x`) sur vos répertoires et vos fichiers, vous pouvez définir la commande `umask` dans vos fichiers `.cshrc` et `.profile` de façon à protéger chaque nouveau fichier ou répertoire que vous créez par ces droits d'accès.

Comme la commande `chmod`, `umask` utilise un code numérique pour représenter les droits d'accès absolus aux fichiers. Toutefois, la méthode de calcul du code de la commande `umask` est différente de celle de la commande `chmod`.

Tout d'abord, si la valeur définie pour `umask` est `000`, tous les fichiers que vous créez ont les droits d'accès (en lecture, écriture, mais non exécution) suivants :

```
rw-rw-rw- (mode 666)
```

et tous les répertoires créés ont les droits d'accès (en lecture, écriture et d'exécution) suivants :

```
rwxrwxrwx (mode 777)
```

Pour déterminer la valeur à utiliser pour la commande `umask`, vous devez soustraire la valeur des droits d'accès souhaités (au moyen de la valeur que vous définiriez pour la commande `chmod`) des droits d'accès par défaut en cours affectés aux fichiers. Le résultat de l'opération représente la valeur à utiliser pour la commande `umask`.

Par exemple, pour décrémenter la valeur du mode par défaut des fichiers de 666 (`rw-rw-rw-`) à 644 (`rw-r--r--`), soustrayez 644 de 666. Le chiffre obtenu, 022, est la valeur numérique à utiliser pour la commande `umask`, comme indiqué ci-dessous :

```
umask 022
```

Comme pour le code numérique de la commande `chmod`, les trois chiffres à utiliser avec la commande `umask` sont les suivants :

- Le premier chiffre contrôle les droits d'accès de l'utilisateur propriétaire.
- Le deuxième chiffre contrôle les droits d'accès d'un groupe d'utilisateurs.
- Le troisième chiffre contrôle les droits d'accès des autres utilisateurs.

Le Tableau 10-1 indique les droits d'accès aux fichiers correspondant à chacun des chiffres du code numérique de la commande `umask`.

Table 10-1 Droits d'accès associés à `umask`

Code umask	Droits
0	rwX
1	rw-
2	r-X
3	r--
4	-wX
5	-w-
6	--X
7	---(aucun)

Pour plus d'informations sur la commande `umask`, consultez les *man Pages(1): User Commands*.

Personnalisation des polices OpenWindows

Si vous le souhaitez, vous pouvez personnaliser la taille et le style des polices affichées dans vos applications OpenWindows. Les sections suivantes décrivent les procédures à appliquer pour personnaliser les polices.

Définition du type de police et de la taille des caractères

La police par défaut utilisée dans les fenêtres est Lucida Sans taille 12 (moyen), et Lucida Sans Bold pour les titres. Si vous le souhaitez, vous pouvez définir un autre type et une autre taille de police pour le texte et le titre des fenêtres. Vous pouvez modifier une seule fenêtre, ou effectuer une modification permanente pour l'ensemble de vos applications dans la fenêtre Propriétés de l'Espace de travail. Les sous-sections suivantes décrivent chacune de ces options.

Polices non proportionnelles et polices proportionnelles

Les polices sont regroupées en deux grandes catégories : *non proportionnelles* et *proportionnelles*. Tous les caractères d'une police non proportionnelle requièrent le même espace. En revanche, l'espace occupé par les caractères d'une police proportionnelle varie en fonction de la largeur de chacun d'entre eux. Les polices proportionnelles sont plus agréables à l'oeil. Toutefois, les polices non proportionnelles conviennent mieux à certaines applications, telles que l'Utilitaire de commande, l'Utilitaire Shell et l'émulateur de terminal `xterm`.

Choix entre les polices non-proportionnelles et proportionnelles

Remarquez que la police utilisée par défaut dans l'Utilitaire de commande et l'Utilitaire Shell est proportionnelle. Bien qu'elle soit plus agréable à l'oeil, elle génère des problèmes au niveau de l'alignement des caractères (en cas d'espacement ou de tabulation), comme toute police proportionnelle utilisée lors des sessions en fenêtres. Si l'alignement des espaces et des tabulations vous pose des problèmes, il est préférable de choisir une police non-proportionnelle pour ces fenêtres. Dans les exemples qui suivent, seules des polices non-proportionnelles sont utilisées pour les fenêtres de terminal ; les exemples relatifs aux autres fenêtres et à leurs titres utilisent des polices proportionnelles.

Définition de la police pour une seule fenêtre

Cette section explique comment ouvrir une seule application en modifiant le type et la taille des polices. Remarquez que les changements ne peuvent pas être effectués sur les fenêtres existantes ; pour afficher une nouvelle police, vous devez lancer à nouveau l'application en tapant son nom sur la ligne de commande.

La commande de base indiquée ci-dessous précise le nom de l'application, l'option `-fn` (nom de la police), ainsi que le type et la taille de la police. La perluète (`&`) fait repasser le prompt système dans la fenêtre après indication de la commande, pour que vous puissiez continuer à utiliser cette dernière.

```
$ application -fn type_police_taille &
```

Les exemples suivants indiquent comment utiliser la commande pour ouvrir une application en appliquant le type et la taille de police indiqués.

- L'exemple ci-dessous indique comment démarrer une session Utilitaire de commande en utilisant la police proportionnelle Lucida Sans Typewriter Bold.

La taille en points n'étant pas indiquée, la taille par défaut (taille 12) est utilisée.

```
$ cmdtool -fn lucidasans-typewriter-bold &
```

- L'exemple ci-dessous indique comment démarrer une session Utilitaire Shell en utilisant la police Lucida Sans Typewriter Bold et fait passer la taille de la police de 12 à 14.

Remarquez que lorsque vous modifiez la taille d'une police, la taille de la fenêtre est modifiée en conséquence.

```
$ shelltool -fn lucidasans-typewriter-bold-14 &
```

- L'exemple ci-dessous indique comment ouvrir une session en fenêtre `xterm` en utilisant la police `terminal-bold` taille 16 :

```
$ xterm -fn terminal-bold-16 &
```

- L'exemple ci-dessous indique comment ouvrir une fenêtre `Edit` en utilisant la police `Helvetica Bold` taille 14 :

```
$ textedit -fn helvetica-bold-14 &
```

Utilisez l'option `-fn` dans une application quelconque, et choisissez le type et la taille de police souhaités. La section "Liste des polices disponibles" de ce chapitre explique comment répertorier l'ensemble des polices disponibles pour les applications `OpenWindows`. La section "Choix d'une police dans la liste", qui figure également dans ce chapitre, présente plusieurs programmes de démonstration permettant de visualiser les polices avant de les utiliser.

Affectation permanente des polices

Si vous exécutez fréquemment des applications en utilisant des polices personnalisées, vous souhaiterez peut-être personnaliser le menu de votre espace de travail. Pour cela, modifiez la catégorie `Menu des programmes` dans la fenêtre `Propriétés de l'Espace de travail`. Vous éviterez ainsi de taper systématiquement les options de la ligne de commande. A titre d'exemple, si vous souhaitez exécuter souvent l'Editeur de texte en utilisant une police de taille supérieure, ajoutez la ligne de commande suivante dans le menu de programmes :

```
textedit -fn lucidasans-typewriter-14
```

Votre `Menu des programmes` peut comporter plusieurs entrées pour la même application si vous souhaitez avoir accès à des polices de taille différente. Cela est utile si vous exécutez une application en utilisant souvent des tailles de

police différentes. Par exemple, si vous souhaitez exécuter l'Editeur de texte en utilisant des polices de taille 12, 14, ou 18, ajoutez les commandes suivantes à votre Menu des programmes :

```
textedit -fn lucidasans-typewriter-12  
textedit -fn lucidasans-typewriter-14  
textedit -fn lucidasans-typewriter-18
```

Après avoir ainsi personnalisé votre Menu des programmes dans la fenêtre Propriétés de l'Espace de travail, vous pouvez appeler l'Editeur de texte en appliquant la taille de police souhaitée. Il suffit pour cela de choisir l'élément correspondant dans la liste de votre Menu des programmes.

Remarque : les lignes de commande ajoutées au Menu des programmes ne doivent pas être suivies d'une perluète (&).

Affichage de la liste des polices disponibles

Si vous souhaitez utiliser dans les applications OpenWindows des polices qui ne figurent pas dans les exemples précédents, commencez par en afficher la liste puis sélectionnez les polices souhaitées.

Liste des polices disponibles

Vous pouvez visualiser la liste complète des polices disponibles en entrant la commande `xlsfonts` au prompt affiché dans une fenêtre d'émulateur de terminal. Il est conseillé de recourir à l'Utilitaire de commande car il est probable que la liste ne pourra pas figurer en totalité à l'écran, et cet utilitaire dispose d'un ascenseur qui vous permettra de visualiser intégralement la liste.

Remarque : la liste générée par `xlsfonts` est très longue ; il existe plus de 400 polices disponibles. Si la liste affichée à l'écran ne comporte pas le nombre de polices attendu, contactez votre administrateur système. Il est possible qu'un sous-ensemble des polices disponibles ait été installé.

Chaque police a un nom complet et un nom abrégé. Par exemple, le nom complet de la police `lucidasans-typewriter` est :

`-b&h-lucida sans typewriter-medium-r-normal-sans-12-120-72-72-m-0-iso8859-1`

Les polices affichées par la commande `xlsfonts` figurent sous leur forme complète suivie de leur forme abrégée. Pour plus de commodité, utilisez uniquement les formes abrégées.

Après avoir choisi une police, suivez les instructions de la section “Définition du type de police et de la taille des caractères” de ce chapitre, afin de personnaliser les polices dans vos fenêtres d’applications.

Choix d’une police dans la liste

Remarque : vous trouverez ci-après les options à sélectionner dans les sous-menus du Menu des programmes de l’Espace de travail pour effectuer les opérations souhaitées.

Trois programmes de démonstration vous permettent de visualiser à l’écran les types et les tailles de police disponibles :

- `Fontview`

`Fontview` est un programme de démonstration disponible dans le menu Espace de travail ► Menu ► Démon. Lorsque le sous-menu Démon apparaît, positionnez le pointeur dans la fenêtre correspondante et appuyez sur `MENU` pour effectuer vos sélections.

- `Text`

`Text` est un programme de démonstration disponible dans le menu Espace de travail ► Menu ► Démon. Lorsque le sous-menu Démonstrations apparaît, positionnez le pointeur dans la fenêtre correspondante et appuyez sur `MENU` pour effectuer vos sélections.

- `xfontsel`

Pour exécuter le programme `xfontsel`, tapez `xfontsel` & sur la ligne de commande.

```
$ xfontsel &
```


Migration vers Open Windows Version 3.3 ou ultérieure



Il se peut que vous disposiez d'une version de l'environnement utilisateur qui n'est plus compatible avec Solaris et qui utilise OpenWindows comme environnement utilisateur par défaut. Par exemple, si vous utilisiez l'environnement utilisateur SunView ou une version d'OpenWindows antérieure à la version 3, vous pouvez migrer vers OpenWindows 3.3 ou vers des versions ultérieures. Cette annexe vous explique comment procéder.

Remarque : le logiciel SunView n'est plus supporté sous OpenWindows Version 3.3 ou une version ultérieure. Une fois que vous avez migré vers la version 3.3, il ne vous est plus possible d'utiliser SunView, comme c'était le cas dans les versions précédentes d'OpenWindows.

SPARC *Migration à partir de l'environnement SunView*

Si vous migrez vers OpenWindows à partir de l'environnement SunView, les informations qui suivent vous seront d'une grande utilité.

SPARC *Les fichiers .defaults et .Xdefaults*

Pour personnaliser votre environnement OpenWindows de la même manière que SunView, vous pouvez convertir votre fichier `.defaults` (utilisé par le logiciel SunView) en un fichier `.Xdefaults` (utilisé par le logiciel

OpenWindows). Si votre répertoire utilisateur contient un fichier `.defaults`, lancez le programme `convert_to_Xdefaults(1)` dans ce répertoire en procédant de la manière suivante :

```
$ cd
$ $OPENWINHOME/bin/convert_to_Xdefaults .defaults
```

Cette commande crée dans votre répertoire utilisateur un fichier `.Xdefaults`, permettant de personnaliser votre environnement OpenWindows au moment du lancement du logiciel.

Migration à partir de l'environnement OpenWindows pre-Version 3.3

Lisez attentivement cette section si vous passez d'une version d'OpenWindows antérieure à la Version 3.3 à OpenWindows Version 3.3 ou ultérieure. La plupart des informations données ici intéressent plus particulièrement les anciens utilisateurs d'OpenWindows Version 2.

La variable d'environnement OPENWINHOME

Si vous disposez actuellement d'une version du logiciel OpenWindows antérieure à la version 3.3, vous avez peut-être configuré votre système avec la variable d'environnement `OPENWINHOME`. Or, sur la version 3.3, il n'est plus nécessaire de définir cette variable, que ce soit manuellement ou à partir d'un fichier de démarrage.

Lorsque vous lancez la commande `openwin`, la variable d'environnement `OPENWINHOME` est automatiquement définie dans `/usr/openwin`. Vous n'avez donc pas à le faire.

Si vous avez défini la variable d'environnement `OPENWINHOME` dans le fichier `.profile` ou `.cshrc` de votre répertoire utilisateur, mettez la ligne correspondante en commentaire ou supprimez-la avant de lancer la version 3.3 d'OpenWindows ou une version ultérieure.

Pour supprimer ou mettre en commentaire la variable d'environnement `OPENWINHOME` dans le fichier `.profile` ou `.cshrc`, procédez comme suit :

- 1. Ouvrez le fichier `.profile` ou `.cshrc` à l'aide d'un éditeur de texte tel que `vi`.**

2. Tapez un signe dièse (#) en regard de la variable, comme indiqué ci-dessous, ou bien supprimez entièrement la ligne.

Si vous travaillez dans le fichier `.profile`, inspirez-vous de l'exemple a et si vous travaillez dans le fichier `.cshrc`, de l'exemple b

a. Dans le fichier `.profile`, tapez :

```
#OPENWINHOME=/usr/openwin
```

b. Dans le fichier `.cshrc`, tapez :

```
#setenv OPENWINHOME /usr/openwin
```

3. Sauvegardez, puis quittez le fichier.

Le fichier `.xinitrc`

Voici quelques remarques importantes concernant les fichiers `.xinitrc` et `$OPENWINHOME/lib/Xinitrc` :

1. Dans l'environnement OpenWindows Version 2, le script `openwin` générait automatiquement une copie de `$OPENWINHOME/lib/Xinitrc` dans un fichier appelé `.xinitrc` de votre répertoire utilisateur. Or, ce n'est plus le cas dans l'environnement OpenWindows Version 3.3. Il s'agit d'un changement important pour les raisons suivantes :
 - a. Le script de démarrage `openwin` utilise le fichier de démarrage `$OPENWINHOME/lib/Xinitrc`, sauf si votre répertoire utilisateur contient un fichier `.xinitrc`, qui remplace le fichier par défaut.
 - b. Il est important d'utiliser le fichier `$OPENWINHOME/lib/Xinitrc` par défaut livré avec le logiciel OpenWindows Version 3.3. (Toutefois, si vous souhaitez conserver certaines modifications que vous avez apportées au fichier `.xinitrc` sous la version 2, vous pouvez suivre les instructions fournies dans cette section.)
2. Si votre système fonctionne avec plusieurs écrans, vous n'avez plus besoin de plusieurs exemplaires de `olwm`.

Utilisation du fichier de démarrage approprié

Si vous disposez d'une version du logiciel OpenWindows antérieure à 3.3, il est important de définir l'état de votre fichier `.xinitrc`. Il s'agit d'un fichier de démarrage OpenWindows de votre répertoire utilisateur, qui peut contenir des options définies par l'utilisateur.

Pour définir l'état de votre fichier `.xinitrc`, tapez les commandes suivantes :

```
$ cd
$ ls -a .xinitrc
```

Selon le résultat de cette commande, effectuez l'une des opérations suivantes :

- Si vous n'avez pas de fichier `.xinitrc` (la dernière commande `ls -a` n'ayant pas généré de sortie pour ce fichier), n'engagez aucune procédure. Si votre répertoire utilisateur ne contient aucun fichier `.xinitrc`, OpenWindows utilise à la place le fichier de démarrage par défaut du système.
- Si vous avez un fichier `.xinitrc` (dont la dernière commande `ls -a` a généré une sortie) mais que vous n'y avez jamais apporté de modifications ou que vous ne souhaitez pas conserver ces dernières, passez à l'étape 1 des "Procédures relatives au fichier de démarrage."
- Si vous avez un fichier `.xinitrc` (dont la dernière commande `ls -a` a généré une sortie) mais que vous y avez apporté des modifications que vous souhaitez conserver, passez à l'étape 2 des "Procédures relatives au fichier de démarrage."

Procédures relatives au fichier de démarrage

- 1. Pour supprimer le fichier `.xinitrc` de votre répertoire utilisateur, tapez la commande suivante :**

```
$ rm .xinitrc
```

- 2. Pour conserver les modifications apportées à votre fichier `.xinitrc`, suivez les étapes ci-après :**

a. Déplacez le fichier `.xinitrc` dans le fichier `.xinitrc.save` :

```
$ mv .xinitrc .xinitrc.save
```

b. Copiez le fichier `$OPENWINHOME/lib/Xinitrc` dans le fichier `.xinitrc` de votre répertoire utilisateur :

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

c. Ajoutez les lignes du fichier `.xinitrc.save` que vous souhaitez conserver dans le fichier `.xinitrc`.



Attention : lorsque vous éditez le fichier `.xinitrc`, n'ajoutez pas de deuxième version de `olwm`, n'ajoutez pas `svenv` et ne supprimez pas la ligne contenant le fichier `$OPENWINHOME/lib/openwin-sys`.

Propriétés de l'espace de travail

Dans les précédentes versions d'OpenWindows (antérieures à 3.3), les modifications du menu Propriétés de l'Espace de travail étaient sauvegardées dans le fichier `.Xdefaults` de votre répertoire utilisateur, alors que dans la version 3.3 et les versions ultérieures, elles sont sauvegardées dans le fichier `.OWdefaults` figurant également dans votre répertoire utilisateur. Même si le fichier `.Xdefaults` existe toujours, la priorité est donnée à la personnalisation du fichier `.OWdefaults`.

Le fichier `.Xdefaults` doit être utilisé *uniquement* pour effectuer les modifications personnalisées ne pouvant pas être apportées dans les Propriétés de l'Espace de travail. Par exemple, vous pouvez éditer le fichier `.Xdefaults` à l'aide d'un éditeur de texte tel que `vi` pour personnaliser des applications ne fonctionnant pas sous OpenWindows ou pour ajouter des macros au pré-processeur C. L'utilisation du menu Propriétés de l'Espace de travail n'empêche nullement ces personnalisations.

Si votre répertoire utilisateur contient déjà un fichier `.Xdefaults` que vous ne souhaitez pas personnaliser, il n'est pas nécessaire de supprimer ce fichier. Sa présence n'est pas gênante étant donné que le fichier `.OWdefaults` est prioritaire.

Personnalisation du menu Espace de travail

Sous OpenWindows Version 3.3 ou ultérieure, vous pouvez personnaliser l'option Menu de l'Espace de travail à partir des Propriétés de l'Espace de travail. Dans les versions antérieures à OpenWindows 3.3, cette opération était effectuée lors de l'édition du fichier `.openwin-menu` de votre répertoire utilisateur.

Remarque : si votre répertoire utilisateur ne contient pas le fichier `.openwin-menu`, la procédure suivante n'est pas nécessaire. Vous pouvez personnaliser le menu Espace de travail à partir des Propriétés de l'Espace de travail.

Si vous disposez d'un fichier `.openwin-menu`, suivez les étapes ci-après pour utiliser les Propriétés de l'Espace de travail afin de personnaliser votre menu Espace de travail.

Si la ligne suivante apparaît dans votre fichier `.openwin-menu` :

```
"Programs"MENU $OPENWINHOME/lib/openwin-menu-programs
```

supprimez-la et remplacez-la par celle-ci :

```
"Programs"INCLUDE openwin-menu-programs
```

Si votre fichier `.openwin-menu` ne contient pas la ligne à remplacer, insérez tout de même la ligne de remplacement ci-dessus.

L'insertion ou le remplacement de cette ligne permet d'ajouter l'option par défaut Menu dans votre menu Espace de travail, afin de le personnaliser à partir des Propriétés de l'Espace de travail.

Si vous obtenez plusieurs options identiques dans le menu Espace de travail, terminez la procédure d'édition en supprimant simplement les lignes redondantes du fichier `.openwin-menu`.

Modification du clavier



La présente annexe vous explique comment reconfigurer le clavier. Elle décrit les options de reconfiguration des touches spéciales du clavier. Elle explique également comment activer et désactiver la touche Compose du clavier.

Pour plus d'informations sur la reconfiguration des boutons de la souris (pour une utilisation par un gaucher, par exemple), reportez-vous au *Manuel de l'Utilisateur Solaris*.

Activation/Désactivation de la touche Compose

x86 – Sur les systèmes x86, la touche Compose est définie comme suit : Ctrl-Shift-F1.

Si vous n'utilisez pas la touche Compose, vous pouvez la désactiver pour ne pas risquer d'appuyer dessus par inadvertance. Déterminez tout d'abord le code clavier correspondant à Multi_key:

```
$ xmodmap -pk | grep Multi_key
```

Le système affiche une ligne du type :

```
nn 0xff20 (Multi_key)
```

L'information importante est le code de touche à deux chiffres représenté par *nn* au début de la ligne. Ce code vous permet de créer la ligne suivante dans le fichier `.xinitrc` :

```
xmodmap -e 'keycode nn = NoSymbol'
```

Pour réactiver la touche Compose, mettez la ligne ci-dessus en commentaire dans votre fichier `.xinitrc`, puis relancez OpenWindows.

SPARC *Reconfiguration du clavier pour les gauchers*

Le script de reconfiguration des touches décrit dans cette section (pour les claviers de Type-4 et Type-5) permet de reconfigurer la plupart des touches spéciales des pavés droite et gauche du clavier (les blocs de touches situés de part et d'autre de la partie centrale du clavier).

SPARC – Notez que les sections suivantes, relatives à la reconfiguration des touches pour une meilleure utilisation par les gauchers, ne s'appliquent qu'aux machines SPARC.

SPARC *Utilisation du script de reconfiguration*

Pour créer et utiliser le script de reconfiguration, suivez les étapes ci-après :

1. Créez un fichier appelé `lefty.data` à l'aide d'un éditeur de texte.

Ce fichier peut figurer dans n'importe quel répertoire, mais l'étape 4 doit être exécutée dans le répertoire où il a été créé.

2. Tapez le script comme indiqué dans le Tableau B-1 "Le script

`lefty.data`."

Une ligne précédée d'un point d'exclamation est une ligne de commentaire qui n'exécute aucune opération.

3. Sauvegardez les modifications, puis quittez l'éditeur.

4. En regard du prompt, tapez : `xmodmap lefty.data`

Vous devez vous trouver dans le répertoire du fichier script.

5. Cliquez sur l'Espace de travail à l'aide de la souris pour que le script prenne effet.

Une fois ces étapes terminées, vous pouvez utiliser le clavier configuré pour un gaucher.

Tapez le script suivant dans le fichier *lefty.data*, comme décrit à l'étape 1.

Tableau B-1 Le script *lefty.data*

```
!  
! lefty.data  
!  
! Données utilisées par xmodmap pour configurer les touches de fonction droite et  
! gauche des claviers Sun type-4 pour les gauchers. Pour utiliser ce type de données,  
! tapez la commande suivante où <filename> désigne le nom du fichier (par exemple,  
! lefty-data).  
!  
! xmodmap <filename>  
!  
! Les commentaires ci-dessous correspondent aux différentes affectations de touches  
! suivantes.  
!  
! swap L2 (Again) with R1 (Pause)  
! swap L3 (Props) with R6 (KP_Multiply)  
! swap L4 (Undo) with R4 (KP_Equal)  
! swap L5 (Front) with R9 (KP_9)  
! swap L6 (Copy) with R7 (KP_7)  
! swap L7 (Open) with R12 (KP_6)  
! swap L8 (Paste) with R10 (Left)  
! swap L9 (Find) with R15 (KP_3)  
! swap L10 (Cut) with R13 (KP_1)  
!  
! chng R3 (Break) to L1 (Stop)  
! chng R2 (Print) to R10 (Left)
```

Tableau B-1 Le script `lefty.data`

```
! chng R5 (KP_Divide) to R12 (Right)
!
! chng Linefeed to Control-R
!
keycode 10 = R1      R1      Pause
keycode 28 = L2      L2      SunXK_Again
keycode 32 = R6      R6      KP_Multiply
keycode 54 = L3      L3      SunXK_Props
keycode 33 = R4      R4      KP_Equal
keycode 52 = L4      L4      SunXK_Undo
keycode 56 = R9      R9      KP_9      Prior
keycode 77 = L5      L5      SunXK_Front
keycode 58 = R7      R7      KP_7      Home
keycode 75 = L6      L6      SunXK_Copy
keycode 79 = Right   R12     KP_6
keycode 100 = L7     L7      SunXK_Open
keycode 80 = Left    R10     KP_4
keycode 98 = L8      L8      SunXK_Paste
keycode 102 = R15    R15     KP_3      Next
keycode 121 = L9     L9      SunXK_Find
keycode 104 = R13    R13     KP_1      End
keycode 119 = L10    L10     SunXK_Cut
keycode 30 = L1      L1      SunXK_Stop
keycode 29 = Left    R10     KP_4
keycode 53 = Right   R12     KP_6

keycode 118 = Control_R
add control = Control_R
```

SPARC Annulation de la reconfiguration du clavier

Il existe deux méthodes pour ramener le clavier à sa configuration initiale. La première consiste à quitter, puis à relancer OpenWindows. La seconde méthode, qui est préférable si vous voulez reconfigurer les touches régulièrement, est de créer un second script et de l'initialiser à chaque fois que vous voulez effectuer une reconfiguration.

Pour créer un second script, procédez comme suit :

1. Créez un fichier appelé `nolefty.data` à l'aide d'un éditeur.

Ce dernier doit figurer dans le répertoire contenant le script `lefty.data`.

2. Tapez le script comme indiqué dans le Tableau B-2, "Le script `nolefty.data`".

Une ligne précédée d'un point d'exclamation est une ligne de commentaire qui n'exécute donc aucune opération.

3. Sauvegardez les modifications, puis quittez l'éditeur.

4. Au prompt, tapez :

```
$ xmodmap nolefty.data
```

Pour que le fichier `nolefty.data` soit réellement actif, vous devez entrer la commande ci-dessus dans le même répertoire que le fichier script.

Tableau B-2 Le script `nolefty.data`

```
!  
! nolefty.data  
!  
! Data for xmodmap to reset the left and right function keys after being set for  
! left-handed use on the Sun type-4 keyboard. To use this data type the following  
! where <filename> is the name of this file.  
!  
! xmodmap <filename>  
!  
!Reassign standard values to left function keys
```

Tableau B-2 Le script `nolefty.data`

```

!
keycode 10 = L2      L2      SunXK_Again
keycode 32 = L3      L3      SunXK_Props
keycode 33 = L4      L4      SunXK_Undo
keycode 56 = L5      L5      SunXK_Front
keycode 58 = L6      L6      SunXK_Copy
keycode 79 = L7      L7      SunXK_Open
keycode 80 = L8      L8      SunXK_Paste
keycode 102 = L9     L9      SunXK_Find
keycode 104 = L10    L10     Sun XK_Cut
!
! Reassign standard values to right function keys.
!
keycode 28 = R1      R1      Pause
keycode 29 = R2      R2      Print
keycode 30 = R3      R3      Scroll_Lock  Break
keycode 52 = R4      R4      KP_Equal
keycode 53 = R5      R5      KP_Divide
keycode 54 = R6      R6      KP_Multiply
keycode 75 = R7      R7      KP_7         Home
keycode 77 = R9      R9      KP_9         Prior
keycode 98 = Left    R10     KP_4
keycode 100 = Right  R12     KP_6
keycode 119 = R13    R13     KP_1         End
keycode 121 = R15    R15     KP_3         Next
!
! Reassign the Linefeed key as such and remove from control map.

```

Tableau B-2 Le script `nolefty.data`

```
!  
remove control = Control_R  
5keycode 118 = Linefeed
```

Exécution des applications du réseau



Cette annexe décrit certaines fonctions avancées de l'environnement OpenWindows qui vous permettent d'exécuter des applications résidant sur une autre machine de votre réseau.

Remarque : cette annexe ne concerne qu'une petite partie des utilisateurs. Si vous comptez utiliser des applications du réseau, demandez à votre administrateur système quelles sont les applications disponibles sur votre réseau.

Dans l'environnement OpenWindows, toutes les applications qui apparaissent sur votre écran (comme la Messagerie et l'Agenda) résident généralement sur votre machine. Cependant, si vous êtes connecté à un réseau, vous pouvez également lancer des applications résidant sur une autre machine et les afficher sur votre écran. Cela vous permet d'économiser les ressources de traitement de votre propre machine et d'accéder à un vaste réseau d'applications.

Cette annexe décrit la méthode la plus simple pour exécuter une application sur une machine distante et l'afficher sur votre écran. Toutefois, la procédure à suivre peut varier en fonction de votre environnement informatique. Pour plus d'informations sur l'exécution des applications du réseau, consultez la section "Précisions relatives à la sécurité".

Pour exécuter une application distante à l'aide de cette procédure, vous devez répondre aux conditions suivantes :

- Vous devez disposer de droits d'accès sur la machine distante.

- Votre répertoire utilisateur doit pouvoir être installé sur le système de fichiers réseau (NFS) de la machine distante.
- L'application et les bibliothèques appropriées doivent être installées sur la *machine* distante.

Si ces conditions ne sont pas claires, contactez votre administrateur système.

Utilisation de `rlogin` pour exécuter une application du réseau

Pour exécuter une application du réseau sur une machine distante, vous devez impérativement vérifier que vos variables d'environnement sont correctement définies :

- La variable d'environnement `HOME` de votre Shell sur la machine distante doit correspondre à votre répertoire utilisateur.
- La variable d'environnement `DISPLAY` de votre Shell sur la machine distante doit désigner votre écran.
- Si les bibliothèques OpenWindows n'ont pas été installées dans le répertoire standard des bibliothèques partagées `/usr/lib` ou `/usr/local`, vous devez assigner à la variable d'environnement `LD_LIBRARY_PATH` le répertoire approprié (`$OPENWINHOME/lib`).

L'exemple ci-après indique comment exécuter un Utilitaire de commande sur une machine distante à l'aide de `rlogin`. Dans cet exemple, le répertoire utilisateur réside sur la machine distante, sous le chemin `/home/mydirectory`, et le logiciel OpenWindows dans le répertoire `/usr/openwin` de cette même machine. Modifiez les variables *mon_répertoire* et *ma_machine* à votre convenance et remplacez `cmdtool` par le nom de l'application que vous voulez exécuter.

```
$ rlogin machine_distante
.
.
( Exécutez les commandes suivantes sur la machine distante. ) ]
.
.
$ HOME=/home/mon_répertoire
$ DISPLAY=ma_machine:0
$ LD_LIBRARY_PATH=/usr/openwin/lib
$ /usr/openwin/bin/cmdtool &
```

Après avoir tapé la dernière ligne, vous voyez apparaître à l'écran la fenêtre Utilitaire de commande. Bien que vous puissiez dialoguer avec cette application comme avec n'importe lequel des programmes apparaissant sur votre écran, l'Utilitaire de commande est en fait exécuté sur la machine distante.

S'il ne présente guère d'intérêt d'exécuter l'Utilitaire de commande de cette façon puisqu'il est déjà disponible sur votre machine et sollicite relativement peu les ressources de traitement, cet exemple est valable pour n'importe quelle application distante.

Précisions relatives à la sécurité

Cette section traite de certains aspects fondamentaux de la sécurité réseau qui peuvent se révéler utiles lorsque vous exécutez des applications en réseau, à savoir :

- Les mécanismes de contrôle d'accès basés respectivement sur l'utilisateur et sur la machine
- Les protocoles d'attribution de droits MIT-MAGIC-COOKIE-1 et SUN-DES-1
- Quand et comment modifier le mécanisme de contrôle d'accès du serveur
- Comment exécuter des applications sous une autre identité, à distance ou en mode local

A qui s'adresse cette section ?

Vous n'avez pas à modifier la configuration de sécurité par défaut du logiciel OpenWindows Version 3.3 ou ultérieure, à moins que vous vous n'utilisiez l'une des configurations suivantes :

- Vous exécutez une application présentant des liens avec les versions de `xlib` ou `libcps` antérieures à OpenWindows Version 2 ou à X11R4.
- Vous exécutez une application présentant des liens statiques avec les bibliothèques d'OpenWindows Version 2 et vous voulez utiliser le protocole d'attribution de droits SUN-DES-1.
- Vous exécutez une application sur un serveur distant.

Mécanismes de contrôle d'accès

Un mécanisme de contrôle d'accès est une procédure visant à déterminer quels *clients* ou applications auront accès au serveur X11. Seuls les clients disposant des droits appropriés pourront se connecter au serveur ; tous les autres se verront refuser l'accès à l'aide d'un message d'erreur.

Il existe deux types de mécanisme de contrôle d'accès, l'un étant basé *sur l'utilisateur* et l'autre *sur la machine*. (Le premier attribue l'accès à un compte utilisateur, le second à une *machine* spécifique.) A moins que vous n'ayez utilisé l'option `-noauth` avec `openwin`, ces mécanismes de contrôle d'accès sont tous les deux actifs. Pour plus d'informations, consultez la section "Gestion de l'accès au serveur", dans le présent chapitre.

Accès basé sur l'utilisateur

Le mécanisme de contrôle d'accès basé sur l'utilisateur, ou sur les droits, vous permet d'attribuer des droits explicites à un utilisateur donné sur une machine donnée. Le client de cet utilisateur transmet au serveur les données d'attribution de droits. Si ces données sont conformes à celles du serveur, l'utilisateur est autorisé à se connecter.

Accès basé sur la machine

Le mécanisme de contrôle d'accès basé sur la machine est un mécanisme général qui vous donne accès à une machine spécifique, depuis laquelle tous les utilisateurs peuvent se connecter au serveur. C'est une forme de contrôle d'accès plus souple : si la machine a accès au serveur, tous ses utilisateurs sont également autorisés à s'y connecter.

Ce mécanisme est principalement utilisé pour la compatibilité ascendante. Les applications générées avec les versions de `Xlib` ou `libcps` antérieures à OpenWindows Version 2 ou à X11R4 ne reconnaissent pas le nouveau mécanisme de contrôle d'accès basé sur l'utilisateur. Pour permettre à ces applications d'accéder au serveur, l'utilisateur doit sélectionner le mécanisme basé sur la machine ou générer l'application à nouveau avec des versions plus récentes de `Xlib` et de `libcps`.

Remarque : les clients de versions antérieures de `xlib` ou de `libcps` doivent si possible être générés à nouveau avec des versions plus récentes de ces bibliothèques pour pouvoir se connecter au serveur via le nouveau mécanisme de contrôle d'accès basé sur l'utilisateur.

Protocoles d'attribution de droits

La présente version du logiciel OpenWindows supporte deux protocoles d'attribution de droits, `MIT-MAGIC-COOKIE-1` et `SUN-DES-1`, qui diffèrent par leurs types de données, mais utilisent, en revanche, le même mécanisme de contrôle d'accès. Le serveur ne peut mettre en oeuvre qu'un protocole à la fois. `MIT-MAGIC-COOKIE-1`, qui fait appel au mécanisme basé sur l'utilisateur, est le protocole par défaut du logiciel OpenWindows.

`MIT-MAGIC-COOKIE-1`

Le protocole d'attribution de droits `MIT-MAGIC-COOKIE-1` a été développé par le Massachusetts Institute of Technology. Au démarrage du serveur, un “*magic cookie*” est attribué à ce dernier et à l'utilisateur qui a initialisé le système. A chaque tentative de connexion, le client de cet utilisateur envoie au serveur le “*magic cookie*” avec le paquet de connexion, pour comparaison avec celui du serveur. Si les deux “*magic cookies*” sont identiques, la connexion est établie ; sinon, elle est refusée.

`SUN-DES-1`

Développé par Sun Microsystems, le protocole d'attribution de droits `SUN-DES-1` est basé sur l'appel de procédure distante sécurisé (Secure RPC) et requiert l'algorithme de chiffrement de données DES (Data Encryption Software). Les droits d'accès sont basés sur l'identité réseau de l'utilisateur, indépendamment de la machine utilisée. Cette information est chiffrée et envoyée au serveur avec le paquet de connexion. Le serveur déchiffre l'information et, s'il reconnaît l'identité réseau, autorise la connexion.

Ce protocole offre un niveau de sécurité plus élevé que le protocole `MIT-MAGIC-COOKIE-1`. En aucun cas, un autre utilisateur ne peut accéder au serveur sous votre identité réseau, alors qu'il peut le faire avec le “*magic cookie*”.

La section “Attribution des droits d’accès à l’aide du protocole SUN-DES-1” du présent chapitre explique comment permettre à d’autres utilisateurs d’accéder à votre serveur en ajoutant leur nom sur la liste d’accès du serveur.

Modification du protocole d’attribution de droits par défaut

Vous pouvez remplacer le protocole d’attribution de droits par défaut, MIT-MAGIC-COOKIE-1, par SUN-DES-1, l’autre protocole supporté, ou encore ne spécifier aucun mécanisme de contrôle d’accès basé sur l’utilisateur. Il suffit pour cela d’indiquer des options avec la commande `openwin`. Par exemple, pour remplacer le protocole par défaut MIT-MAGIC-COOKIE-1 par SUN-DES-1, lancez le logiciel OpenWindows de la façon suivante :

```
$ openwin -auth sun-des
```

Pour exécuter OpenWindows sans le mécanisme de contrôle d’accès basé sur l’utilisateur, utilisez l’option de ligne de commande `-noauth` :

```
$ openwin -noauth
```



Attention : l’option `-noauth` réduit le niveau de sécurité. Elle revient à exécuter OpenWindows avec le mécanisme de contrôle d’accès basé sur la machine seulement puisque le mécanisme basé sur l’utilisateur est désactivé par le serveur. Par conséquent, quiconque peut exécuter des applications sur votre machine a accès également à votre serveur.

Gestion de l’accès au serveur

A moins que vous n’utilisiez l’option `-noauth` avec l’option `openwin` (voir la section “Modification du protocole d’attribution de droits par défaut”), le mécanisme de contrôle d’accès basé sur l’utilisateur et le mécanisme basé sur la machine sont tous les deux actifs. Le serveur commence par contrôler le mécanisme basé sur l’utilisateur, puis le mécanisme basé sur la machine. La configuration de sécurité par défaut utilise MIT-MAGIC-COOKIE-1 comme mécanisme basé sur l’utilisateur et une liste vide comme mécanisme basé sur la machine. Seul le mécanisme basé sur l’utilisateur est donc réellement actif.

L'option `-noauth` commande au serveur de désactiver le mécanisme de contrôle d'accès basé sur l'utilisateur et initialise la liste de contrôle d'accès basé sur la machine en y ajoutant la machine.

Deux programmes permettent de modifier le mécanisme de contrôle d'accès du serveur : `xhost` et `xauth`. Ces programmes accèdent à des fichiers binaires créés par le protocole d'attribution de droits et contenant des données d'attribution de droits spécifiques à la session. L'un de ces fichiers est destiné uniquement au serveur, tandis que l'autre réside dans le répertoire `$HOME` de l'utilisateur :

- `.Xauthority` Fichier de droits client

Vous devez utiliser le programme `xhost` pour modifier la liste de contrôle d'accès basé sur la machine du serveur. Vous pouvez y ajouter des machines, ou en supprimer. Si vous utilisez la configuration par défaut (liste vide) et que vous ajoutez un nom de machine à l'aide de `xhost`, vous réduirez le niveau de sécurité. Le serveur autorisera alors l'accès à la machine que vous avez ajoutée et à tous les utilisateurs précisant le protocole d'attribution de droits par défaut. La section "Accès basé sur la machine" explique pourquoi le mécanisme de contrôle d'accès basé sur la machine offre un niveau de sécurité moindre.

Le programme `xauth` accède aux données du protocole d'attribution de droits via le fichier `.Xauthority`. Vous pouvez extraire ces informations de votre fichier `.Xauthority` pour les faire fusionner avec les données du fichier `.Xauthority` d'un autre utilisateur et permettre à ce dernier d'accéder à votre serveur ou au serveur auquel vous êtes connecté.

Voir la section "Attribution des droits d'accès à l'aide du protocole MIT-MAGIC-COOKIE-1" pour obtenir des exemples d'utilisation de `xhost` et `xauth`.

Fichier de droits client

Le fichier de droits client, `.Xauthority`, contient des entrées du type :

<code>connection-protocol</code>	<code>auth-protocol</code>	<code>auth-data</code>
----------------------------------	----------------------------	------------------------

Par défaut, `.Xauthority` contient MIT-MAGIC-COOKIE-1 comme *protocole d'attribution de droits* et les entrées associées à l'écran local comme *protocole de connexion* et *données d'attribution de droits*. Par exemple, sur la machine *anyhost*, le fichier `.Xauthority` peut comporter les entrées suivantes :

<code>anyhost:0</code>	MIT-MAGIC-COOKIE-1	<code>82744f2c4850b03fce7ae47176e75</code>
<code>localhost:0</code>	MIT-MAGIC-COOKIE-1	<code>82744f2c4850b03fce7ae47176e75</code>
<code>anyhost/unix:0</code>	MIT-MAGIC-COOKIE-1	<code>82744f2c4850b03fce7ae47176e75</code>

Au démarrage du client, une entrée correspondant au *protocole de connexion* est lue dans `.Xauthority` et les *protocole* et *données d'attribution de droits* sont envoyés au serveur avec le paquet de connexion. Dans la configuration par défaut, `xhost` génère des listes vides de contrôle d'accès basé sur la machine et déclarent que les droits sont accordés.

Si vous avez remplacé le protocole d'attribution de droits par défaut par SUN-DES-1, les entrées de `.Xauthority` contiennent SUN-DES-1 comme *protocole d'attribution de droits* et l'identité réseau de l'utilisateur comme *données d'attribution de droits*. L'identité réseau présente le format suivant :

<code>unix.ID_utilisateur@NISnom_domaine</code>

Par exemple, sur la machine *anyhost*, le fichier `.Xauthority` peut comporter les entrées suivantes, où `unix.15339@EBB.Eng.Sun.COM` désigne l'identité réseau de l'utilisateur indépendante de la machine :

<code>anyhost:0</code>	SUN-DES-1	<code>"unix.15339@EBB.Eng.Sun.COM"</code>
<code>localhost:0</code>	SUN-DES-1	<code>"unix.15339@EBB.Eng.Sun.COM"</code>
<code>anyhost/unix:0</code>	SUN-DES-1	<code>"unix.15339@EBB.Eng.Sun.COM"</code>

Remarque : Si vous ne connaissez pas votre identité réseau ou identité réseau indépendante de la machine, contactez votre administrateur système.

Attribution des droits d'accès à l'aide du protocole MIT-MAGIC-COOKIE-1

Si vous utilisez le protocole d'attribution de droits MIT-MAGIC-COOKIE-1, suivez les étapes ci-après pour permettre à un autre utilisateur d'accéder à votre serveur :

1. **Sur la machine qui exploite le serveur, utilisez `xauth` pour extraire une entrée correspondant à `nom_machine:0` dans un fichier.**

Dans cet exemple, `nom_machine` correspond à `anyhost` et le fichier s'appelle `xauth.info` :

```
$ $OPENWINHOME/bin/xauth nextract - anyhost:0 > $HOME/xauth.info
```

2. **Envoyez le fichier contenant l'entrée associée à l'utilisateur qui a émis la demande d'accès (en utilisant la Messagerie, `rcp` ou une autre méthode de transfert de fichiers).**

Remarque : il est plus prudent d'envoyer le fichier contenant les informations sur l'attribution des droits à l'aide de la Messagerie qu'avec la commande `rcp`. Si vous utilisez `rcp`, évitez de placer le fichier dans un répertoire facilement accessible par un autre utilisateur.

3. **L'autre utilisateur doit faire fusionner cette entrée dans son fichier**

`.Xauthority`.

Dans cet exemple, la `machine_utilisateur` fait fusionner `xauth.info` dans le fichier `.Xauthority` :

```
$ $OPENWINHOME/bin/xauth nmerge - < xauth.info
```

Remarque : Les *données d'attribution de droits* sont spécifiques à la session en cours. Elles sont donc valables jusqu'à la réinitialisation du serveur.

Attribution des droits d'accès à l'aide du protocole SUN-DES-1

Si vous utilisez le protocole d'attribution de droits SUN-DES-1, suivez les étapes ci-après pour permettre à un autre utilisateur d'accéder à votre serveur :

1. Sur la machine qui exploite le serveur, utilisez `xhost` pour déclarer le nouvel utilisateur au serveur.

Dans cet exemple, le nouvel utilisateur *quidam* est autorisé à accéder à *ma_machine*:

```
$ xhost + quidam@
```

2. Le nouvel utilisateur doit utiliser `xauth` pour ajouter l'entrée dans son fichier `.Xauthority`.

Dans cet exemple, l'identité réseau indépendante de la machine du nouvel utilisateur est *unix.15339@EBB.Eng.Sun.COM*. Notez que cette commande doit être tapée sur une seule ligne, sans retour chariot. A la suite du symbole pipe, tapez un espace suivi du reste de la commande .

```
$ echo 'add nom_machine:0 SUN-DES-1 "unix.15339@EBB.Eng.Sun.COM"' |
$OPENWINHOME/bin/xauth
```

Exécution de clients sous une autre identité, à distance ou en mode local

Les clients X utilisent la valeur de la variable d'environnement `DISPLAY` pour déterminer le nom du serveur auquel ils doivent se connecter. Pour des raisons de compatibilité ascendante,

Pour exécuter des clients sous une autre identité, que ce soit à distance ou en mode local, suivez les étapes ci-après :

1. Sur la machine exploitant le serveur, attribuez des droits d'accès à un autre utilisateur.

Selon le protocole d'attribution de droits utilisé, suivez la procédure décrite dans la section "Attribution des droits d'accès à l'aide du protocole MIT-MAGIC-COOKIE-1".

2. Attribuez comme valeur à `DISPLAY` le nom de la machine exploitant le serveur.

Dans cet exemple, la machine est désignée par *machine_distante* :

```
$ DISPLAY=machine_distante:0
```

3. Exécutez le programme client comme indiqué.

```
$ client_program&
```

Le client s'affiche sur la machine distante, *machine_distante*.

SPARC — Interconnexion avec DECnet (DNI)



Cette annexe décrit les procédures d'interconnexion de l'environnement OpenWindows et de l'environnement DECwindows™ via le protocole de transport NSP DECnet.

SPARC – Notez que ce chapitre sur l'interconnexion avec DECnet ne s'applique qu'aux machines SPARC. L'interconnexion à un réseau DECnet est disponible uniquement avec 8.x DNI.

Il existe deux scénari DNI :

- Exécution d'un client X11 sur un système VAX (sous le système d'exploitation VMS®) et affichage de la fenêtre client sur une machine OpenWindows
- Exécution d'un client X11 sur une machine OpenWindows et affichage de la fenêtre client sur un système VAX

Ces deux scénarios sont décrits dans les sections qui font suite. Elles sont précédées d'une section initiale qui explique les procédures de configuration du logiciel DNI software pour l'un des deux scénarios.

Configuration de l'interconnexion à un réseau DECnet

Pour configurer l'interconnexion à un réseau DECnet, suivez les étapes ci-après

1. Permettre une connexion via DNI.

Les bibliothèques serveur et client d'OpenWindows utilisent une version dynamiquement chargeable de la bibliothèque de transport `libdni` de DNI. Afin que les bibliothèques client et serveur puissent charger `libdni`, vous devez définir la variable d'environnement `DNI_X_ENABLE` à la valeur du répertoire dans lequel `libdni.so` est installé.

L'exemple ci-dessous suppose que vous avez chargé DNI via `pkgadd` dans l'emplacement par défaut :

```
$ DNI_X_ENABLE=/opt/SUNWconn/dni/lib
```

2. Démarrer le serveur OpenWindows.

Le serveur OpenWindows supporte par défaut le mécanisme de contrôle d'accès "MIT-MAGIC-COOKIE". Ce mécanisme est basé sur l'utilisateur et non sur l'ordinateur : il vous permet de choisir les utilisateurs et non les "machines" qui peuvent se connecter au serveur. Dans le mode par défaut, la commande `xhost` génère une liste vide, et indique uniquement que le mécanisme de sécurité est activé. Vous pouvez désactiver ce mode (pour revenir à celui des versions antérieures de serveur OpenWindows) en utilisant l'option `-noauth` de la commande `openwin`.

```
$ openwin -noauth
```

3. Demander au propriétaire de la machine exécutant le logiciel OpenWindows d'utiliser la commande `xhost` pour que DEC® VAX® puisse accéder au serveur OpenWindows via une connexion X11.

Afin que les clients X11 puissent se connecter au serveur OpenWindows via le logiciel DNI, les adresses du noeud DECnet doivent être mappées vers leurs noms de noeud DECnet. Pour effectuer cette opération, vous devez créer et initialiser la base de données NCP. Cette opération doit également être effectuée sur le système DEC VAX.

```
$ xhost decvax::
```

Les deux-points consécutifs désignent le protocole de transport DECNet.

Affichage d'un client distant sur une machine OpenWindows

Vous pouvez exécuter des clients X11 à partir de VMS en utilisant la commande `dnilogin` de SunLink DNI pour vous connecter au système VAX. Définissez en premier lieu votre variable d'environnement `DISPLAY` sur votre machine en tant que serveur X11 d'une machine distante. Exécutez ensuite un client X11 en entrant le nom du client, représenté ici par `x11_client`. (Pour plus d'informations sur l'utilisation du système d'exploitation VMS, consultez le *VMS DECwindows User's Guide, Running Applications Across the Network*).

Par exemple :

```
$ dnilogin decvax
.
.
.
$ define DECW$DISPLAY OW_machine::0
$ spawn/nowait run x11_client
```

Affichage d'un client distant sur un système VAX

Vous pouvez exécuter des clients X11 sur une machine OpenWindows et les afficher sur un serveur DECwindows en définissant la variable `DISPLAY` pour le système VAX distant.

Avant de pouvoir exécuter l'un quelconque des clients X11, vous devez compiler et installer les polices OpenWindows sur le serveur DECwindows. Ces polices sont disponibles dans la version X11R4 du MIT ou dans le module des polices optionnelles du logiciel OpenWindows Version. Suivez les étapes ci-dessous pour installer les polices adéquates sur le serveur DECwindows :

- 1. Installez les versions sources des polices optionnelles sur OpenWindows (en utilisant les versions sources des polices de la version X11R4 du MIT) sur la machine OpenWindows.**
- 2. Pour consulter les instructions d'installation des polices, consultez le *OpenWindows Server Programmer's Guide*.**

3. Copiez les versions sources des polices dans un répertoire du système VAX.

```
$ cd $OPENWINHOME/share/src/fonts/misc
$ dnicp *.bdf `decvax::[vaxdir]`
```

4. Compilez les polices du curseur dans le système VAX.

Les fichiers suivants s'affichent : `olcursor.decw$font;1`
`olglyph10.decw$font;1...`

```
$ font olcursor.bdf
$ font olglyph10.bdf
...
```

5. Copiez les polices dans le répertoire `sysfont` :

```
$ set def sys$sysroot:[sysfont.decw.user_cursor16]
$ copy [vaxdir]olcursor.decw$font;1 *
```

Remarque : pour copier les polices dans le répertoire `sysfont`, vous devez être connecté en tant que “system” à DEC VAX.

6. Vous devez également suivre les étapes 2 à 4 pour les autres polices du curseur et les polices Lucida qui se trouvent dans

`$OPENWINHOME/share/src/fonts/75dpi` et
`$OPENWINHOME/share/src/fonts/100dpi`.

Remarque : les polices Lucida doivent être installées dans
`sys$sysroot:[sysfont.decw.user_75dpi]` et
`sys$sysroot:[sysfont.decw.user_100dpi]`.

La liste suivante montre le jeu de polices minimum qui doit être installé afin de pouvoir exécuter les utilitaires DeskSet d'OpenWindows. Si vous utilisez des polices par défaut dans vos applications, ce jeu de polices est suffisant. Toutefois, vous pouvez installer des polices supplémentaires si nécessaire.

- `olcursor.bdf`

- olglyph10.bdf
- olglyph12.bdf
- olglyph14.bdf
- olglyph19.bdf
- luBS08.bdf
- luBS10.bdf
- luBS12.bdf
- luBS14.bdf
- luRS08.bdf
- luRS10.bdf
- luFS12.bdf
- lutBs12.bdf
- lutRS10.bdf
- lutRS12.bdf

7. Redémarrez le serveur DECwindows.

8. Vous pouvez vérifier que les polices ont été installées en établissant la liste des polices disponibles du serveur DECwindows :

```
$ DISPLAY=decvax::0
$ xlsfonts | grep Sun      (polices curseur)
$ xlsfonts | grep Lucida
```

9. Vérifiez que vous avez accordé un droit d'affichage au noeud OpenWindows sur le serveur DECwindows, en utilisant le menu Sécurité du Gestionnaire de Sessions DECwindows.

10. Exécutez une application X11 (par exemple, un utilitaire DeskSet de OpenWindows).

```
$ DISPLAY=decvax::0
$ mailtool
```

Remarque : la commande `DNI_X_ENABLE` doit être définie à l'emplacement de la bibliothèque de transport DNI `libdni`. Reportez-vous à l'Etape 1 de la section "Configuration de l'interconnexion à un réseau DECnet", dans le présent chapitre.

Si un message d'erreur du type suivant s'affiche, vous devez installer la police appropriée sur le serveur DECwindows afin de pouvoir exécuter l'application.

```
XView warning: Cannot load font '-b&h-lucida-medium-r-*-*-*-*  
80-*-*-*-*-*' (Font package)
```

Ce message signifie que la police `luRS10.bdf` doit être installée.

Pour plus d'informations sur les polices, consultez la brochure *OpenWindows Server Programmer's Guide*.

Configuration des imprimantes et des modems



Configuration d'un modem sur votre système

Un *modem* est un périphérique permettant à votre système de transmettre et de recevoir des informations via les lignes téléphoniques. L'outil graphique Administration Tool permet de réaliser facilement la configuration d'un modem. Il vous suffit de suivre les étapes suivantes :

1. Connectez le modem à votre système.

Pour ce faire, vous devez généralement procéder aux opérations suivantes :

- Connectez physiquement le modem à votre système, ou installez-le (si vous disposez d'une carte modem).
- Définissez les éventuelles options de commande requises et/ou configurez le débit en bauds, le port et les autres options du modem.
- Branchez le modem ou son adaptateur dans une prise de courant, le cas échéant.

Reportez-vous à la documentation fournie par le fabricant du modem et à celle relative à l'installation de votre système pour savoir comment effectuer ces tâches.

2. Devenez superutilisateur.

A moins que vous ne soyez membre du groupe spécial sysadmin UNIX (GID 14), vous devez devenir *superutilisateur* de votre système pour pouvoir utiliser Administration Tool. En effet, le superutilisateur détient des droits particuliers de modification des fichiers système.

Pour devenir superutilisateur, lancez la commande su :

```
$ su
Password:      (entrez ici le mot de passe superutilisateur)
#
```

Si le compte superutilisateur est protégé par un mot de passe, vous devez le connaître. Dans le cas contraire, appuyez simplement sur Return à l'invite Password. Si vous ne connaissez pas le mot de passe superutilisateur de votre système, demandez l'aide d'un administrateur système de votre site.

3. Lancez Administration Tool.

Tapez :

```
# /usr/bin/admintool &
#
```

L'outil graphique Administration Tool apparaît au bout de quelques secondes.

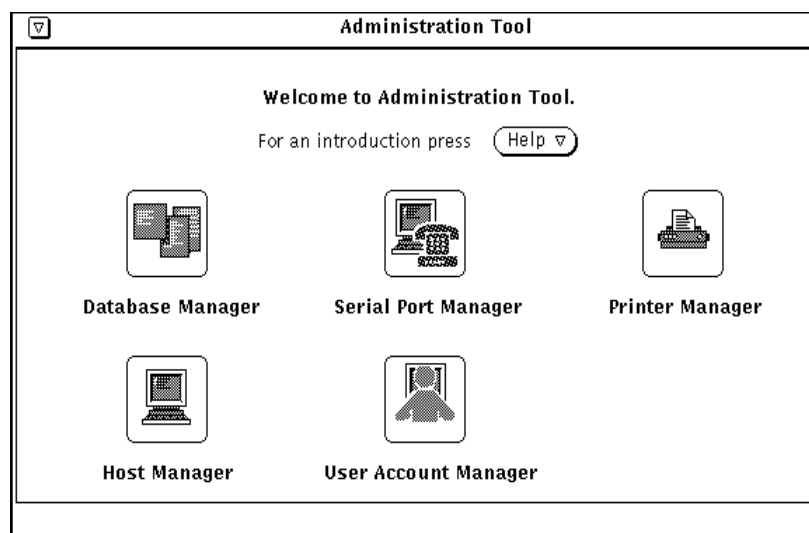


Figure E-1 L'outil Administration Tool

4. Choisissez Serial Port Manager.

Pour lancer Serial Port Manager, cliquez sur l'icône Serial Port Manager de la fenêtre Administration Tool. Une nouvelle fenêtre s'ouvre, qui contient les commandes et les options permettant de gérer les modems et les terminaux.

5. Suivez les instructions de l'aide en ligne.

L'aide en ligne de Serial Port Manager fournit des instructions étape par étape relatives à l'utilisation de Serial port Manager pour l'installation d'un modem. Pour visualiser les instructions d'aide, choisissez Help... dans la fenêtre Serial Port Manager.

Pour obtenir d'autres détails sur les modems

Pour obtenir des informations supplémentaires sur l'installation et la configuration des modems, reportez-vous au manuel *Peripherals Administration*, qui fait partie de *Solaris 2.4 System Administrator AnswerBook*.

Configuration d'une imprimante à utiliser sur votre système

L'outil graphique Administration Tool est tout indiqué pour configurer une imprimante. Les instructions suivantes expliquent comment configurer une imprimante locale afin de pouvoir l'utiliser sur votre propre système. Pour plus de détails et d'instructions sur la configuration d'imprimantes en réseau, reportez-vous au manuel *Peripherals Administration*, qui fait partie de *Solaris 2.4 System Administrator AnswerBook*.

Voici les instructions à suivre pour configurer une imprimante locale :

1. Connectez l'imprimante à votre système.

Pour ce faire, vous devez généralement procéder aux opérations suivantes :

- Connectez physiquement l'imprimante à votre système.
- Définissez les éventuelles options de commande requises et/ou configurez le débit en bauds, le port et les autres options de l'imprimante.
- Branchez l'imprimante dans une prise de courant.

Vous pouvez connecter l'imprimante au système et la mettre sous tension avant ou après avoir utilisé Administration Tool pour l'ajouter. Reportez-vous au manuel fourni par le fabricant de l'imprimante et au manuel d'installation de votre système pour savoir comment définir les options de commande et connaître les instructions de câblage. Les câbles d'imprimantes sont généralement connectés à un port série, mais dans certains cas, selon les exigences de l'imprimante en question, vous pouvez utiliser un port parallèle.

2. Devenez superutilisateur.

A moins que vous ne soyez membre du groupe spécial sysadmin UNIX (GID 14), vous devez devenir *superutilisateur* de votre système pour pouvoir utiliser Administration Tool. En effet, le superutilisateur détient des droits particuliers de modification des fichiers système.

3. Pour devenir superutilisateur, lancez la commande su :

```
$ su
Password:      (entrez ici le mot de passe superutilisateur)
#
```

Si le compte superutilisateur est protégé par un mot de passe, vous devez le connaître. Dans le cas contraire, appuyez simplement sur Return à l'invite Password. Si vous ne connaissez pas le mot de passe superutilisateur de votre système, demandez l'aide d'un administrateur système de votre site.

4. Lancez Administration Tool.

Tapez :

```
# /usr/bin/admintool &
#
```

L'outil graphique Administration Tool apparaît après quelques secondes. (Voir Figure E-1, page 207.)

5. Choisissez Printer Manager.

Pour lancer Printer Manager, cliquez sur l'icône Printer Manager de la fenêtre Administration Tool. Une nouvelle fenêtre s'ouvre, qui contient les commandes et les options permettant de gérer les imprimantes.

6. Suivez les instructions de l'aide en ligne.

L'aide en ligne de Printer Manager fournit des instructions étape par étape relatives à l'utilisation de Printer Manager pour l'installation d'une imprimante. Pour visualiser les instructions d'aide, choisissez Help... dans la fenêtre Printer Manager.

Pour obtenir d'autres détails sur les imprimantes

Pour obtenir des informations supplémentaires sur l'installation et la gestion des imprimantes, reportez-vous au manuel *Peripherals Administration*, qui fait partie de *Solaris 2.4 System Administrator AnswerBook*.

Index

Symbols

..., symbole 36

A

absolus, droits d'accès
 Voir également droits d'accès
administrateur système 1
affichage
 contenu d'un répertoire distant 152
 de votre lieu de connexion à distance 150
 droits d'accès à un fichier 46
 du contenu d'un fichier 31–32
 état de l'imprimante 139–140
 liste de la boîte à lettres 109
 utilisateurs de votre serveur de fichiers 112
 utilisateurs distants 153–154
 utilisation du disque 68
 utilisation du répertoire 68–69
affichages, lancement d'OpenWindows avec plusieurs moniteurs 14
aide en ligne 24
alias 160
alias de distribution 123
 Voir aussi messagerie
application

en réseau
 avec DNI (DECnet) 199
sur une machine distante 188

B

boîte à lettres 104
 définition du chemin d'accès 158
 Voir aussi messagerie
Bourne Shell
 fichiers d'initialisation 155–156

C

C Shell
 fichiers d'initialisation 155–156
 historique des commandes 20–21
C shell
 alias 160
caractères des spécialisés tilde, *Voir* commandes tilde
cercle, sur les menus xvi
chaîne 55
chaîne de caractères 91
changement de nom
 de fichiers 30
 de répertoires 38
changement de répertoire 35–37

chemin d'accès 33
chemin d'accès relatif 37
clavier
 annulation de la reconfiguration
 (pour les gauchers) 183
 reconfiguration pour les gauchers 180
commande `apropos` 25
commande `bdiff` 42
commande `cancel` 142
commande `cat` 32
commande `cd` 28, 35–37
commande `chmod` 48, 51
 et `umask` 165
commande `cp` 30, 39
commande d'ajout 81
commande d'insertion 81
commande d'ouverture de lignes 81
commande `date` 18
commande `df` 68
commande `diff` 40–41
commande `diff3` 42
commande `du` 69
commande `env` 157
commande `file` 32
commande `find` 42–45
commande `finger` 112
commande `grep`
 et chaînes 58
 et expressions régulières 59–60
 et métacaractères 60–61
 syntaxe 55–56
 utilisée en tant que filtre 56
commande `history`
 définition 158
commande `history` 20
commande `kill` 67
commande `logname` 19
commande `lp` 133
 tableau des options 137
commande `lpstat` 137–142
 syntaxe 141
 tableau des options 142
commande `ls` 30
 et liste des fichiers cachés 47
 et option de liste longue 46–47
commande `man` 3
commande `man` 24
commande `mkdir`
 et messagerie 119
commande `mkdir` 37
commande `more` 32
commande `mv` 30
commande `nohup` 23
commande `passwd` 65
commande `ps` 66
commande `put` 86
commande `pwd` 28, 35
commande `rcp` 151–152
commande `rlogin` 146–150
 et abandon d'une connexion 149
 et suspension d'une connexion 150
commande `rm` 31, 39
commande `rmdir` 39
commande `rsh` 152–153
commande `rusers` 112
commande `rusers` 153
commande `sort` 69
commande `touch` 29
commande `umask` 164–165
commande `view` 71
commande `whatis` 24
commande `who` 112
commande `yank` 86
commandes
 définition des fonctions 24
 distinction majuscules/minuscules 18
 exécution en arrière-plan 23
 exécution sur une machine distante
 152–153
 options 21
 recherche par mots-clés 25
 redirection du résultat 21–22
 répétition des commandes
 antérieures 20–21

-
- saisie de commandes longues 19
 - saisie de plusieurs commandes 19
 - syntaxe 24
 - transmission par pipe du résultat 22
 - commandes `ex` 75, 88–90
 - commandes SunOS xiv
 - commandes `vi`
 - répétition 87
 - commentaires sur le texte 127
 - compatibilité avec SunView 173
 - fichier `.defaults` (SunView) 173
 - fichier `.Xdefaults` 173
 - programme
 - `convert_to_Xdefaults` 174
 - compte 1
 - concaténer
 - Voir également* commande `cat` 32
 - connexion
 - au système de multi-fenêtrage 12
 - connexion au système 1, 17
 - à distance 146–151
 - sous une autre identité 148
 - connexions à distance, *Voir* commande `rlogin`
 - copie
 - de fichiers 30
 - de lettres dans des dossiers 118–120
 - de lettres dans des fichiers 118
 - de répertoires 39
 - depuis une machine distante 151
 - vers une machine distante 152
 - copie de
 - lignes d'un fichier `vi` à l'autre 96
 - texte `vi` 86
 - texte `vi` à l'aide des commandes `ex` 89
 - copie récurrente 39
 - copies aveugles 114
 - copies carbone 114
 - courrier électronique, *Voir* messagerie
 - fichier `.cshrc` 156–165
- D**
- de 155
 - déconnexion du système 3–4
 - `.defaults` 173
 - défaut
 - droits d'accès aux fichiers 164–??
 - démarrage
 - compatibilité avec SunView 173
 - touche `Compose`,
 - activation/désactivation 179
 - déplacement
 - de fichiers 30
 - de répertoires 38
 - déplacement de
 - texte `vi` 86
 - texte `vi` à l'aide des commandes `ex` 90
 - distinction majuscules/minuscules
 - commandes 18
 - distinction majuscules-minuscules
 - et éditeur `vi` 78, 91
 - dossiers, *Voir* messagerie 118
 - droit d'écriture 45
 - droit d'exécution 45
 - droit de lecture 45
 - droits
 - définition des valeurs par défaut 165
 - droits d'accès
 - absolus, définition 51–54
 - catégories 46
 - définition par défaut 164–??
 - et le caractère générique `*` 51
 - modification 48–51
 - types 45
 - droits sur les fichiers par défaut 165
- E**
- écho 2
 - écrans
 - lancement d'OpenWindows avec
 - deux écrans juxtaposés 15
 - lancement d'OpenWindows avec

deux écrans superposés 16
éditeur *vi*, *voir vi*
email, *Voir* messagerie
en réseau, applications
avec DECnet 199
expressions régulières 59

F

fichier
 .cshrc 156
 .login 156
 .mailrc 123
 .profile 156
 /etc/aliases 125
 /etc/hosts.equiv 151, 153
 /etc/passwd 151, 153, 155
 /etc/profile 156
 affichage du contenu 31–32
 caché, liste 47
 copie 30
 création 29
 d'initialisation 155–156
 définition 27
 définition de droits d'accès par défaut
 164–??
 définition des droits par défaut 165
 droits d'accès 45–46
 et commande *diff* 40–42
 impression 134
 insertion du contenu d'un fichier dans
 une lettre 116
 modification des droits d'accès 48–51
 recherche 42–45
 sauvegarde de lettres dans un fichier
 117–118
 sécurité 45–54
 vérification du format 32
fichier mbox 104
 Voir aussi messagerie
fichier point
 .xinitrc 175
fichiers cachés
 .xinitrc 175
fichiers exécutables 28

fichiers point 47
fichiers texte 27
file d'attente 140

G

guillemets 62

H

historique des commandes 20–21
horloge système 159

I

identificateurs de processus, *Voir* PID
impression
 Voir aussi commande *lp*
 annulation des requêtes 142–144
 courrier 111
 demande de notification de fin
 d'impression 135
 en plusieurs exemplaires 136
 fichiers *vi* 77
 numéro d'ID d'impression 143
 sur l'imprimante par défaut 134
 sur une imprimante donnée 134–135
 vérification de l'état 137–140
 vérification de la disponibilité des
 imprimantes 138
 vérification du type d'imprimante 140
imprimante par défaut 134, 158
imprimantes
 port
 parallèle ou série 208
interréseau 146

K

Korn Shell
 fichiers d'initialisation 155–156

L

lancement

- options spéciales 12
- plusieurs écrans, mode d'exécution
 - exemples 14–15
- langue locale 158
- ligne d'état 73
- limitation
 - de la recherche via `grep` 56
 - résultat de la commande `du` à l'aide
 - de la commande `sort` 69
 - résultat de la commande `ps` à l'aide
 - de la commande `grep` 67

M

- fichier `.mailrc` 123
- mémoires tampon nommées 87
- messagerie
 - affichage de la liste de lettres 109
 - aide en ligne 131
 - alias
 - dans `.mailrc` 123–125
 - dans `/etc/aliases` 125–128
 - définition 123
 - tableau comparatif 128
 - annulation des lettres non envoyées 114
 - commande `copy` 118
 - commandes tilde 129
 - contrôle de la version 107
 - copies aveugles 114
 - copies carbone 114
 - détermination de l'adresse d'un utilisateur 112
 - dossiers 118–121
 - envoi 112–114
 - à plusieurs destinataires 112
 - et boîte à lettres 104
 - fichier `mbox` 104
 - impression 111
 - insertion d'un fichier 116
 - insertion d'une autre lettre 116
 - lancement 104
 - lecture 107, 108–110
 - lettres non transmises 114
 - quitter 107

- réponse 116–117
- sauvegarde 117–120
- suppression de lettres 110–111
- utilisation de `vi` 121
- métacaractères 59
- mode commande 73, 74
- mode entrée 73
- modem
 - (définition) 205
- mot de passe 1
 - choix 63
 - expiration 65–66
 - nouveau 64
- Mots de passe, processus et mémoire disque 63

N

- option `-noauth` d'`openwin` 192
- option `-noauth` d'`openwin` 13
- nom de connexion 1
 - définition 158
- nom utilisateur 1
- numéro d'ID d'impression 143

O

- `openwin`
 - `-dev` option 14, 15, 16
 - option `-noauth` 13
 - option `-noauth` 192
 - options_périphérique 14
- OpenWindows
 - avec le mécanisme de contrôle d'accès MIT-MAGIC-COOKIE 200
 - connexion et déconnexion 12
 - et messagerie 103
 - fichiers d'initialisation 156
 - lancement avec plusieurs moniteurs 14
- options 21
- options de commande 21

P

pages manuel 3, 158
pages manuel, *Voir* pages manuel
par défaut
 répertoire 28
paramètres
 sélection pour `vi` 97
PID 66–68
plusieurs écrans, lancement 14
polices
 liste 169
 modification de la police par défaut 166
 pour applications en réseau 202
port parallèle 208
port série 208
processus
 en cours d'exécution 66–67
 interruption 67–68
profil système 156
profil utilisateur 156
fichier `.profile` 156–165
programme `convert_to_xdefaults` 174
programme `ditroff` 71
programme `mailx`
 quitter 107
programme `mailx`
 Voir aussi messagerie
programme `nroff` 71
programme `troff` 71
programme `xauth` 195
prompt de commande 17
 définition 159, 162
 par défaut 17
protocoles d'attribution de droits, *Voir* sécurité

R

racine, répertoire 34
recherche

 à l'aide de la commande `find` 42–45
 à l'aide de la commande `grep` 55–62
reconfiguration des boutons de la souris 179
relais 146
relatif
 chemin d'accès 37
renvoi 64
répertoire
 arborescence 34
 changement 35–37
 changement de nom 38
 copie 38
 création d'un nouveau 37
 `dead.letter` 114
 définition 28, 119
 définition de droits d'accès par défaut 164–??
 définition des droits par défaut 165
 déplacement 38
 en cours, affichage 35
 et chemin d'accès 33
 modification des droits d'accès 48–51
 par défaut 28
 racine 34
 sécurité 45–54
 suppression 39
 utilisateur 35
 vérification de l'utilisation 68
répertoire père 36
répertoire utilisateur 28, 35
 définition 158
 et commande `rlogin` 147
 et tilde de raccourci 35
répétition des commandes 20
réseau
 définition 145
 protocole 146
réseau campus 146
réseau local 146
réseau longue distance 146
réseau, étendu xiii

S

sécurité

- fichier .Xauthority 193–194
- mécanismes de contrôle d'accès
 - définition 190
- option-noauth 190
- programme xauth 195
- protocole d'attribution de droits MIT-MAGIC-COOKIE-1 191
- protocoles d'attribution de droits 191
- protocoles d'autorisation ??–197

sécurité MIT-MAGIC-COOKIE 200

SELECT xv

service d'impression, voir impression

session de travail standard 1

Shell de connexion 2–3, 155

Shell par défaut 155

souris

- reconfiguration des boutons 179
- utilisation par les gauchers 179

sous-système d'impression en ligne, voir impression

SunLink, architecture en réseau 201

superutilisateur 126

suppression

- de répertoires 39

suppression de

- texte vi 84
- texte vi à l'aide des commandes ex 90

suppression de lettres 110–111

symbole pipe 22

T

temps d'unité centrale 66

tilde, commandes 129

touche Compose,

- activation/désactivation 179

transmission par pipe

- du courrier à la commande lp 111
- du résultat des commandes 22

travaux en arrière plan 23

U

utilisation du clavier pour les gauchers 180

utilisation du disque 68

Utilitaire de commande

- sur une machine distante 188

V

variable

- CDPATH 158
- HISTORY 158
- HOME 158
- LANG 158
- LOGNAME 158
- LPDEST 158
- MAIL 158
- MANSECTS 158
- PATH 158
- PS1 159, 162
- SHELL 159
- TERM 159
- TERMINFO 159
- TZ 159

variable d'environnement DISPLAY 196

variable PATH 159–160

variables d'environnement

- dans les fichiers d'initialisation 157–160

VAX, en tant qu'hôte de l'application client X11 199

vi

- utilisation avec la messagerie 121

vi

- annulation de modifications 84
- commande d'ajout 81
- commande d'insertion 81
- commande d'ouverture de lignes 81
- commande de copie ex 89
- commande de déplacement ex 90
- commande de suppression ex 90
- commande put 86
- commande yank 86

- copie de texte 86
- coupure de lignes 83
- création de fichier 72
- curseur 72
 - déplacement 78
- défilement dans 80
- définition 71
- déplacement dans 78–81
- déplacement de texte 86
- distinction majuscules-minuscules
 - dans une recherche 91
- édition de plusieurs fichiers 95–96
- et commandes `ex` 75, 88–90
- impression de fichiers 77
- insertion de fichiers 94
- insertion de texte 81–82
- insertions répétées 87
- inversion de caractères 83
- jointure de lignes 83
- ligne d'état 72–73
- mémoires tampon nommées 87
- mode commande 73, 74
- mode insertion 73–74
- numérotation des lignes 88
- parcours des fichiers 80
- passage à une ligne donnée 94
- récapitulatif des principales commandes 98
- recherche et remplacement ??–94
- remplacement d'un mot 82
- remplacement d'une ligne 82
- remplacement de texte dans 82–83
- répétition des commandes 87
- remplacement d'un caractère 83
- reprise après une panne 98
- sélection des paramètres 97
- sortie
 - description 76–98
- substitution de caractères 82
- suppression d'un caractère 84
- suppression d'un mot 85
- suppression d'une ligne 85
- suppression de texte 84–85
- utilisation d'un comptage 87

X

- fichier `.Xauthority` 193–194
- `.Xdefaults`
 - et applications SunView 173
- `.xinitrc` 175
- `xmodmap`
 - commande 179
 - et reconfiguration du clavier 180