



Unix

Petit manuel de référence

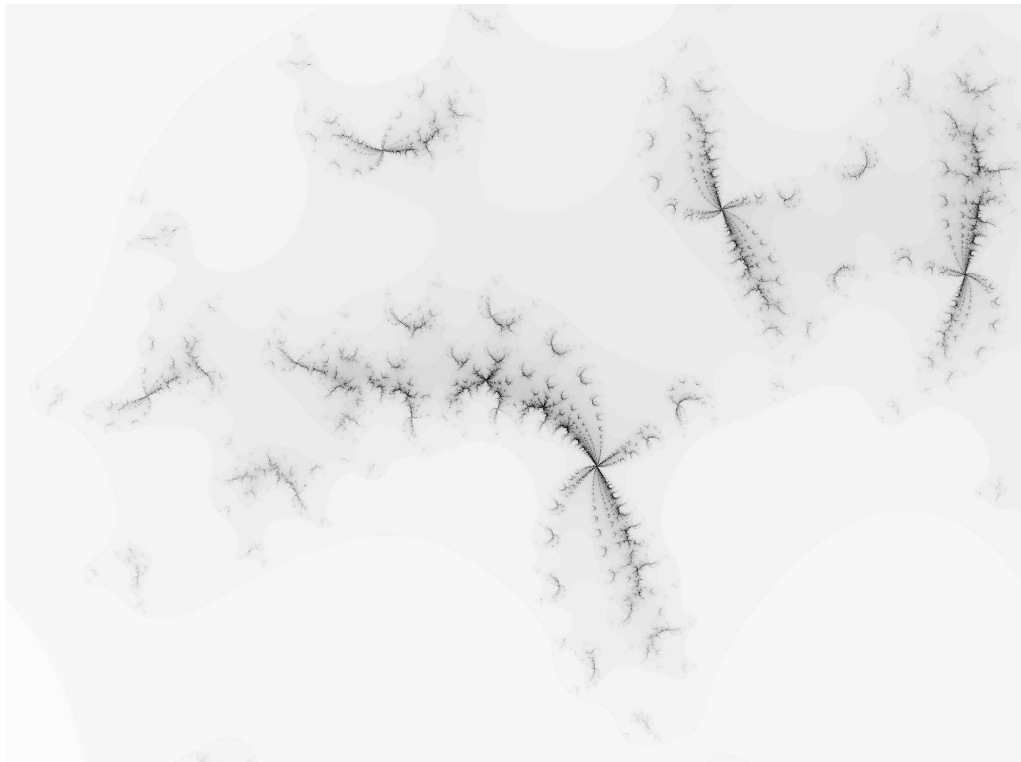


Table des matières

1	Avant propos	1
1.1	Introduction	1
1.2	Manuel de référence	1
1.3	Conventions	1
2	Commandes UNIX	3
3	Le Shell et ses commandes	27
3.1	Introduction au Shell	27
3.2	Les méta-caractères	28
3.3	Les guillemets	28
3.4	Les séparateurs	28
3.5	Les redirections	29
3.6	Les variables	29
3.7	Les variables d’environnement du shell	29
3.8	Les fonctions shell	30
3.9	Le dièse et le nombre magique	30
3.10	Les commandes shell	31
A	Références et licence	39
	Indexe	41

Chapitre 1

Avant propos

1.1 Introduction

Le système UNIX fut conçu au début des années 70 par AT&T (actuellement USL). Du fait de la variété de machines sur lesquelles il tournait, il s'est enrichi de commandes créées par des générations successives de développeurs. Sous l'impulsion de différents organismes (comme USL ou l'université californienne Berkeley), plusieurs standards UNIX ont ainsi vu le jour et ont évolué en parallèle. Actuellement, la normalisation POSIX a apporté une homogénéisation des principaux UNIX (System V et BSD) ainsi, quelque soit leur marque, il existe un ensemble de commandes communes à tous les UNIX actuellement distribués.

1.2 Manuel de référence

Ce manuel présente les commandes qui ont le plus d'intérêt pour les utilisateurs de systèmes Unix sous forme de « manuel de références ».

Chaque entrée est indexée par le nom de la commande. La syntaxe est suivie d'une brève description, de la liste des principales options disponibles et, souvent, d'exemples d'utilisation.

1.3 Conventions

Les conventions typographiques sont les suivantes :

Gras	est utilisé pour les répertoires, les fichiers, les commandes ainsi que les options. Ce sont les mots clés qu'il faut utiliser tel quel (sans changer un seul caractère).
<i>Italique</i>	est utilisé pour les expressions génériques d'arguments et d'options.
Gras italique	est utilisé pour les titres.
Courrier	est utilisé pour les exemples.
\$	présente le prompt du shell.
[.]	encadre les listes de paramètres optionnels.
EOF	indique le caractère « fin de fichier ». Classiquement CONTROL-D
	sépare les paramètres dont un seul à la fois est nécessaire.

Chapitre 2

Commandes UNIX

ar	ar [-V] [-]clé[options] [fichpos] archive [fichiers] Gère un ensemble de <i>fichiers</i> associés dans un fichier <i>archive</i>. Ceci est le plus souvent utilisé pour créer et mettre à jour des bibliothèques utilisées par l'éditeur de liens (ld). Seul une lettre clé peut être utilisée mais elles peuvent être associées à un nombre quelconque d'<i>options</i>. <i>fichpos</i> est le nom d'un fichier inclus dans l'<i>archive</i>. Lors du déplacement ou du remplacement de <i>fichiers</i>, il est possible de préciser qu'ils soient placés avant ou après <i>fichpos</i>. -V affiche le numéro de version de ar sur le canal d'erreur. Clé <table><tr><td>d</td><td>Efface les <i>fichiers</i> de l'<i>archive</i>.</td></tr><tr><td>m</td><td>Déplace les <i>fichiers</i> en fin d'<i>archive</i>.</td></tr><tr><td>p</td><td>Affiche les <i>fichiers</i> de l'<i>archive</i>.</td></tr><tr><td>q</td><td>Ajoute les <i>fichiers</i> dans l'<i>archive</i>.</td></tr><tr><td>r</td><td>Remplace les <i>fichiers</i> dans l'<i>archive</i>.</td></tr><tr><td>t</td><td>Liste le contenu de l'<i>archive</i> ou simplement les <i>fichiers</i> nommés.</td></tr><tr><td>q</td><td>Extrait les fichiers contenu de l'<i>archive</i> ou simplement les <i>fichiers</i> nommés.</td></tr></table> Options <table><tr><td>a</td><td>S'utilise avec r et m pour placer les <i>fichiers</i> après <i>fichpos</i>.</td></tr><tr><td>b</td><td>S'utilise avec r et m pour placer les <i>fichiers</i> avant <i>fichpos</i>.</td></tr><tr><td>c</td><td>Crée une <i>archive</i> sans demander de confirmation.</td></tr><tr><td>i</td><td>Identique à b.</td></tr><tr><td>s</td><td>Impose la remise à jour de la table des symboles de l'<i>archive</i> (utile après avoir lancé strip).</td></tr><tr><td>u</td><td>S'utilise avec r pour ne remplacer que les <i>fichiers</i> ayant changés après avoir été inclus dans l'<i>archive</i>.</td></tr><tr><td>v</td><td>Affiche une description sommaire de la commande.</td></tr></table> Exemple Changement des objets du répertoire courant dans l'archive mabiblio.a : <pre>ar r mabiblio.a 'ls *.o'</pre>	d	Efface les <i>fichiers</i> de l' <i>archive</i> .	m	Déplace les <i>fichiers</i> en fin d' <i>archive</i> .	p	Affiche les <i>fichiers</i> de l' <i>archive</i> .	q	Ajoute les <i>fichiers</i> dans l' <i>archive</i> .	r	Remplace les <i>fichiers</i> dans l' <i>archive</i> .	t	Liste le contenu de l' <i>archive</i> ou simplement les <i>fichiers</i> nommés.	q	Extrait les fichiers contenu de l' <i>archive</i> ou simplement les <i>fichiers</i> nommés.	a	S'utilise avec r et m pour placer les <i>fichiers</i> après <i>fichpos</i> .	b	S'utilise avec r et m pour placer les <i>fichiers</i> avant <i>fichpos</i> .	c	Crée une <i>archive</i> sans demander de confirmation.	i	Identique à b .	s	Impose la remise à jour de la table des symboles de l' <i>archive</i> (utile après avoir lancé strip).	u	S'utilise avec r pour ne remplacer que les <i>fichiers</i> ayant changés après avoir été inclus dans l' <i>archive</i> .	v	Affiche une description sommaire de la commande.
d	Efface les <i>fichiers</i> de l' <i>archive</i> .																												
m	Déplace les <i>fichiers</i> en fin d' <i>archive</i> .																												
p	Affiche les <i>fichiers</i> de l' <i>archive</i> .																												
q	Ajoute les <i>fichiers</i> dans l' <i>archive</i> .																												
r	Remplace les <i>fichiers</i> dans l' <i>archive</i> .																												
t	Liste le contenu de l' <i>archive</i> ou simplement les <i>fichiers</i> nommés.																												
q	Extrait les fichiers contenu de l' <i>archive</i> ou simplement les <i>fichiers</i> nommés.																												
a	S'utilise avec r et m pour placer les <i>fichiers</i> après <i>fichpos</i> .																												
b	S'utilise avec r et m pour placer les <i>fichiers</i> avant <i>fichpos</i> .																												
c	Crée une <i>archive</i> sans demander de confirmation.																												
i	Identique à b .																												
s	Impose la remise à jour de la table des symboles de l' <i>archive</i> (utile après avoir lancé strip).																												
u	S'utilise avec r pour ne remplacer que les <i>fichiers</i> ayant changés après avoir été inclus dans l' <i>archive</i> .																												
v	Affiche une description sommaire de la commande.																												

at	at <i>options</i> <i>1 heure</i> [<i>date</i>] [+<i>incrément</i>] at <i>options</i> <i>1</i> [<i>tâches</i>] Exécute des commandes entrées sur le canal d'entrée standard à une <i>heure</i> précise et, éventuellement, à une certaine <i>date</i>. La fin de l'entrée est marquée par <i>EOF</i>. L'<i>heure</i> peut être écrite sous forme numérique (avec accessoirement les minutes et les codes horaires) ou à l'aide d'un mot clé. La <i>date</i> peut être de la forme mois/jour ou un jour de la semaine ou un mot clé spécial. L'<i>incrément</i> est un entier positif suivi d'un mot clé. Option1 -f <i>fichier</i> les commandes contenues dans le <i>fichier</i>. -m Envoie un mél à l'utilisateur une fois la tâche effectuée. Option2 -l Affiche la liste de toutes les tâches (ou seulement les <i>tâches</i> précisées) par l'utilisateur. Voir aussi atq. -r Retire les <i>tâches</i> précédemment programmées. Vous devez avoir des droits particuliers ou être propriétaire de la tâche. Utilisez -l d'abord pour connaître la liste des tâches programmées. Voir aussi atrm. Heure <i>hh:mm</i> [<i>codes</i>] L'heure peut être un nombre à un ou deux chiffres (par défaut, l'horloge est supposée de 24 heures); les minutes optionnelles peuvent être écrites sur un ou deux chiffres; les deux points peuvent être omis : le format est <i>h</i>, <i>hh</i> ou <i>hhmm</i> c'est-à-dire que 5, 5:30, 0530 et 1945 sont des heures valides. Si les <i>codes</i> horaires am ou pm sont ajoutés, l'horloge est alors de 12 heures. Le code zulu indique une référence au méridien de Greenwich. midnight noon now Ces mots clés peuvent être utilisés en remplacement d'une heure numérique. now doit être suivi d'<i>incrément</i>. Date <i>mois nombre</i> [, <i>année</i>] Le <i>mois</i> (en anglais) doit être écrit en toute lettre ou simplement à l'aide de son abréviation en trois lettres; <i>nombre</i> représente un jour du mois en chiffre; l'<i>année</i> s'écrit quand à elle à l'aide d'un nombre à quatre chiffres. Si le <i>mois</i> précisé dans la ligne de commande est antérieur au mois en cours, <i>at</i> comprend automatiquement que cela se réfère à l'année prochaine. <i>jour</i> Le <i>jour</i> (en anglais) doit être écrit en toute lettre ou simplement à l'aide de son abréviation en trois lettres. now tomorrow Référence le jour en cours ou le suivant. Si la <i>date</i> n'est pas précisée, at programme la tâche au jour en cours si l'<i>heure</i> n'est pas déjà passée, sinon elle est programmée pour le lendemain.
-----------	---

	Incrément Ajoute un incrément numérique si vous voulez programmer le jour ou l'heure d'exécution d'une tâche <i>relativement</i> à la date actuelle. Le nombre doit être suivi d'un des mots clé suivant : minute, hour (heure), day (jour), week (semaine), month (mois) ou year (année) (ou leurs formes plurielles). Le mot clé qui peut être utilisé, est un équivalent de +1. Exemples Remarquez que les deux premiers exemples sont identiques. <pre>at 1305 August 3 at 1:05pm Aug 3 at 11 am Friday at now + 7 hours at noon next month</pre>
atq	atq [<i>options</i>] [<i>utilisateurs</i>] Affiche les tâches créées par la commande at qui sont encore en attente. Habituellement les tâches sont affichées dans l'ordre où elles vont être exécutées. Si aucun <i>utilisateur</i> n'est précisé, ou que vous n'avez pas de droit particulier, seules vos tâches sont listées. Options -c Classe les tâches par ordre de programmation via la commande at. -o Affiche uniquement le nombre de tâches en attente.
atrm	atrm [<i>options</i>] [<i>utilisateurs</i> <i>idTâches</i>] Enlève les tâches programmées par at correspondantes aux identificateurs « <i>id-Tâches</i> ». Si vous avez des droits particuliers, vous pouvez aussi préciser le nom de l'<i>utilisateur</i> qui a programmé des tâches. Options -a Enlève toutes les tâches programmées appartenant à un <i>utilisateur</i> courant. -f Enlève les tâches sans envoyer d'information d'effacement. -i Demande une confirmation (y ou n).
banner	banner <i>phrase</i> Affiche sous forme de poster une <i>phrase</i> sur le canal de sortie standard. Tous les mots doivent contenir moins de 10 caractères.
c++	c++ [<i>options</i>] <i>fichiers</i> Compilateur C++. Voir g++

cal	cal <i>[[mois] année]</i> Sans argument, cal affiche le calendrier du mois en cours. Sinon elle affiche les douze mois de l'<i>année</i> précisée (par un nombre de 1 à 9999) ou le <i>mois</i> de l'<i>année</i> précisé (par un nombre de 1 à 12). <i>Exemples</i> <pre>cal 8 1973 cal 1998 > fichier_annee</pre>
cat	cat <i>[options] [fichiers]</i> Lit un ou plusieurs <i>fichiers</i> et les affiche sur le canal de sortie standard. cat lit les données provenant du canal d'entrée standard si aucun fichier n'est précisé ou si « - » fait partie de la liste des <i>fichiers</i> ; la fin est marquée par <i>EOF</i>. Vous pouvez utiliser l'opérateur « > » pour concaténer plusieurs <i>fichiers</i> en un nouveau fichier. De même, l'opérateur « > > » vous permet d'ajouter des <i>fichiers</i> en fin d'un fichier existant. <i>Options</i> -A Equivalent à -vTE. -b Numérote les lignes non vides. -E Affiche un \$ en fin de chaque ligne. -e Equivalent à -vE. -n Numérote toutes les lignes. -T Affiche un ^I pour chaque tabulation. -t Equivalent à -vT. -v Affiche tous les caractères « invisibles » via les notations ^ et M- sauf pour les retours à la ligne et les tabulation. <i>Exemples</i> <pre>cat fich1 Affiche le contenu du fichier fich1 sur la sortie standard. cat f1 f2 > f3 Concatène les fichiers f1 et f2 en le fichier f3. cat f4 >> f5 Ajoute le fichier f4 a la fin du fichier f5. cat > f6 << STOP Crée un fichier à partir du terminal jusqu'à STOP.</pre>
chmod	chmod <i>[options] mode fichiers</i> Change les droits d'accès (suivant un certain <i>mode</i>) de un ou plusieurs <i>fichiers</i>. Seul le propriétaire des <i>fichiers</i> ou un utilisateur ayant des droits particuliers peut changer le <i>mode</i> d'accès. Le format du <i>mode</i> d'accès se décompose en trois parties : <i>Niveau(x)OpérationDroit(s)</i>. Si aucun <i>niveau</i> n'est précisé, le changement est effectué à tous les niveaux. Une seule <i>opération</i> n'est possible à la fois.

Options	
-c	N'affiche que les opérations au cours desquelles les droits des fichiers ont changés.
-f	N'affiche aucun message d'erreur.
-R	Effectue le changement de droit de façon récursive. C'est-à-dire que si la liste des <i>fichiers</i> contient des répertoires, leurs contenus voient aussi leurs droits changer.
-v	Affiche les opérations effectuées fichier par fichier.
Niveaux	
u	Propriétaire
g	Groupe
o	Autre
a	Tout niveau (c'est-à-dire ugo)
Opération	
+	Ajoute des droits.
-	Supprime des droits.
=	Fixe des droits en enlevant les permission non spécifiées.
Droits	
r	Droits de lecture
w	Droits d'écriture
x	Droits d'exécution
X	Droits d'exécution pour des fichiers exécutables ou des répertoires (propagation du x aux niveaux qui ne l'ont pas encore)
s	Récupération des droits du propriétaire (ou du groupe) lors de l'exécution.
t	« Sticky bit »
u	Droits actuels du propriétaire
g	Droits actuels du groupe
o	Droits actuels des autres
Il est aussi possible d'écrire le <i>mode</i> sous forme d'un nombre octal de 3 chiffres. Le premier désigne les droits du propriétaire, le deuxième ceux du groupe et le troisième ceux des autres. Ces chiffres sont calculés par addition de codes fonction des droits (4 ↔ lecture, 2 ↔ écriture et 1 ↔ exécution). Un quatrième chiffre peut précéder cette séquence pour compléter des droits exceptionnels (4 ↔ u=s, 2 ↔ g=s et 1 ↔ =t).	

	Exemples Ajoute les droits d'exécution au propriétaire et au groupe à prog : <pre>chmod ug+x prog</pre> Les deux opérations suivantes sont équivalentes : <pre>chmod 751 prog chmod u=rwx,g=rx,o=x prog</pre> Elles fixent les droits de l'utilisateur en lecture, écriture et exécution, les droits du groupe en lecture et écriture et les droits des autres uniquement en exécution.
cc	cc [<i>options</i>] <i>fichiers</i> Compilateur C. Voir gcc
chfn	chfn [<i>options</i>] [<i>utilisateur</i>] Permet de changer les informations utilisées par finger. Seul l'administrateur peut changer les informations d'un autre utilisateur que lui même. Si aucune option n'est précisée, chfn demande champ par champ à entrer les modifications. Options -f noms Change le nom et le prénom. -o bureau Change le nom du bureau. -p btel Change le numéro de téléphone du bureau. -h mtel Change le numéro de téléphone personnel.
chsh	chsh [<i>options</i>] [<i>utilisateur</i>] Permet de changer de shell. Seul l'administrateur peut changer le shell d'un autre utilisateur que lui même. Si aucune option n'est précisée, chsh demande quel est le nouveau shell. Options -s shell Précise le nouveau <i>shell</i>. -l Affiche la liste des shells disponibles (fichier <i>/etc/shells</i>).
compress	compress [<i>options</i>] [<i>fichiers</i>] Réduit la taille d'un ou plusieurs <i>fichiers</i> en utilisant l'algorithme de compression Lempel-Ziv et les renomme en <i>fichier.Z</i>. uncompress et zcat permettent de revenir aux anciens fichiers. Ce compresseur est moins efficace que gzip. Options -c Ecrit sur le canal standard de sortie (sans changer les <i>fichiers</i>). -f Force l'opération sans demander de confirmation. -r Décompresse le contenu de répertoires. -v Affiche le taux de compression des <i>fichiers</i>.

cp	cp [<i>options</i>] <i>fichier1</i> <i>fichier2</i> cp [<i>options</i>] <i>fichiers</i> <i>répertoire</i> Copie <i>fichier1</i> en <i>fichier2</i> ou copie un ou plusieurs <i>fichiers</i> sous le même nom dans un autre <i>répertoire</i>. Si la destination est un fichier qui existe déjà, le fichier est écrasé. Options -i Demande une confirmation avant d'écraser un fichier existant. -p Garde la date de dernière modification et les droits lors de la copie. -r Effectue une copie récursive (par un parcours des sous-répertoires) dans un autre <i>répertoire</i>. Exemple Copie deux fichiers dans leur répertoire père. <pre>cp unix.txt comp.c ..</pre>
diff	diff <i>fichier1</i> <i>fichier2</i> Compare deux fichiers et indique les différences. Les lignes provenant de <i>fichier1</i> sont préfixées par > et celles provenant de <i>fichier2</i> par <. Des lignes de contexte indiquent les blocs de lignes qui ont été modifiés (c), ajoutés (a) ou détruits (d); le principe étant d'indiquer les modifications à effectuer pour convertir <i>fichier1</i> en <i>fichier2</i>.
f77	f77 [<i>options</i>] <i>fichiers</i> Compilateur Fortran 77. Voir g77
find	find <i>répertoires</i> <i>conditions</i> <i>operations</i> Recherche (à partir d'un ou plusieurs <i>répertoires</i>) en fonction d'une ou plusieurs <i>conditions</i> des noms de fichiers et y effectue des opérations. Cette commande est extrêmement puissante. Il ne faut pas oublier de préciser au minimum une <i>condition</i> et un <i>répertoire</i>. Le point d'exclamation (« ! ») permet de nier une condition. Conditions -group g Recherche tout fichier appartenant au groupe <i>g</i>. -name m Recherche tout nom correspondant à <i>m</i>. -perm [-]ooo Recherche tout fichier dont les droits d'accès correspondent exactement à <i>ooo</i> (code octal). Avec Le « - », find recherche tout fichier dont les droits correspondent au moins à <i>ooo</i>. -type t Recherche tout fichier de type <i>t</i> (b ↔ périf. bloc, c ↔ périf. caractère, d ↔ répertoire, l ↔ lien symbolique ou f ↔ fichier normal). -size s Recherche tout fichier dont la taille est de <i>s</i>. -user u Recherche tout fichier appartenant à l'utilisateur <i>u</i>.

	Commandes -exec <i>c</i> {} \; Exécute une commande <i>c</i> sur chacun des noms trouvés. {} est remplacé par le nom trouvé. -follow Suit les liens symboliques. -ok <i>c</i> {} \; Demande une confirmation avant d'exécuter la commande <i>c</i>. -print Affiche chaque nom trouvé. Exemples Recherche dans /trash de tous les fichiers appartenant à bill. <pre>find /trash -user bill -print</pre> Compression de tous les fichiers non déjà compressés (depuis le répertoire courant). <pre>find ./ ! -name '*.gz' -exec gzip {} \;</pre> Efface les fichiers vides appartenant à porte dans son répertoire principal. <pre>find ~porte/ -size 0 -exec rm -f {} \;</pre>
finger	finger [<i>options</i>] <i>utilisateurs</i> Affiche des informations concernant un ou plusieurs <i>utilisateurs</i> ainsi que les fichiers .plan, .project et .forward contenus dans les racines des <i>utilisateurs</i>. Vous pouvez préciser le nom de login (exact), le prénom ou le nom de famille (finger renverra tous les utilisateurs dont l'un des noms correspond). Il est possible d'interroger une machine distante avec la syntaxe <i>utilisateur@machine</i>. Options -l Permet d'utiliser le format long (par défaut). -m Les nom donnés ne doivent correspondre qu'à des noms de login. -p N'affiche pas les fichiers .plan et .project. -s Permet d'utiliser le format court.
ftp	ftp <i>machine</i> Permet de communiquer avec une <i>machine</i> distante en utilisant le protocole FTP (File Transfer Protocol) et ainsi de transférer des données entre deux machines via le réseau. Les principales commandes de ftp sont :

	Commandes acsii Indique que le transfert des fichiers s'effectue sur 7 bits (fichiers textes). binary Indique que le transfert des fichiers s'effectue sur 8 bits (tout autre que les fichiers textes). cd Change de répertoire sur la machine distante. close Ferme une session FTP. delete Efface un fichier sur la machine distante. get Copie un fichier de la machine distante vers la machine locale. hash Affiche des dièses « # » pour chaque kilo-octet transféré. help Affiche une aide sommaire pour une commande donnée. Si aucune commande n'est indiquée, help affiche la liste des commandes. ls Affiche la liste des fichiers sur la machine distante. lcd Change de répertoire sur la machine locale. mdelete Efface plusieurs fichiers sur la machine distante. mget Copie plusieurs fichiers de la machine distante vers la machine locale. mkdir Crée un répertoire sur la machine distante. mput Transfère plusieurs fichiers de la machine locale vers la machine distante. open Ouvre une session FTP. prompt Si prompt est on, une confirmation sera demandée lors de toute opération sur un ensemble de fichiers. put Transfère un fichier de la machine locale vers la machine distante. quit Quitte le logiciel FTP. rmdir Efface un répertoire sur la machine distante.
g++	g++ [<i>options</i>] <i>fichiers</i> Compile un ou plusieurs <i>fichiers</i> sources C++. g++ fonctionne exactement de la même façon que gcc ; le détail des options est donc le même (voir gcc).
g77	g77 [<i>options</i>] <i>fichiers</i> Compile un ou plusieurs <i>fichiers</i> sources Fortan 77. g77 fonctionne exactement de la même façon que gcc ; le détail des options est donc le même (voir gcc).
gcc	gcc [<i>options</i>] <i>fichiers</i> Compile un ou plusieurs <i>fichiers</i> sources C. gcc appelle automatiquement l'éditeur de liens ld (tant que l'option -c n'est pas utilisée). Lorsque gcc doit générer un fichier objet, si aucun nom n'est précisé, ce dernier prend le nom sans extension du fichier C avec comme suffixe .o. Par défaut, tout exécutable créé est nommé a.out.

	Options -c Compile sans faire l'édition de liens (génération des fichier objet .o). -Dct[=v] Définit une constante <i>ct</i> valant <i>v</i> pour le pré-processeur (par défaut <i>ct</i> prend la valeur 1). -g Effectue la compilation avec des symboles de débogage utilisés par gdb. -Irep Ajoute le répertoire <i>rep</i> dans la liste des répertoires explorés lors de la recherche des fichiers à inclure (#include <fichier.h>). -Lrep Ajoute le répertoire <i>rep</i> dans la liste des répertoires explorés lors de la recherche des bibliothèques pour l'édition de liens. -ltruc Ajoute la bibliothèque libtruc.a dans la liste des bibliothèques pour l'édition de liens. -Onum Effectue des optimisations plus ou moins performantes, fonction de <i>num</i> (de 1 à 3). -o fichier Donne un <i>nom</i> au fichier généré (objet ou exécutable). -W Affiche les messages de mise en garde les plus flagrants. -Wall Affiche tous les messages de mise en garde. Exemple Compilation puis édition de liens de prog.c. <pre>gcc -Wall -c prog.c gcc -o prog prog.o</pre>
gdb	gdb [<i>programme</i>] Débogueur C, C++ et Fortran 77. Permet d'exécuter pas à pas et de suivre l'évolution des variables d'un <i>programme</i>. Pour cela, il ne faut pas oublier de compiler le <i>programme</i> avec l'option -g du compilateur. Les commandes principales de ce débogueur sont les suivantes : Commandes break [<i>f</i>:]<i>fc</i> Précise un point d'arrêt à la fonction <i>fc</i> (dans le fichier <i>f</i>). run [<i>arg</i>] Lance un programme (avec une liste d'arguments). bt Affiche la pile du programme. print <i>exp</i> Affiche la valeur de l'expression <i>exp</i>. c Continue l'exécution d'un programme (après un arrêt). next Exécute en bloc la ligne suivante du programme. step Exécute la ligne suivante du programme et entre dans une fonction si nécessaire. help Affiche une aide pour chaque commande. quit Quitte gdb. gdb possède une sur-couche graphique plus facile d'utilisation xxgdb. Toutefois, cette dernière ne sera pas traitée dans ce manuel. Pour plus d'information, reportez-vous à la page de manuel de xxgdb.
grep	grep [<i>options</i>] <i>expression</i> [<i>fichiers</i>] Recherche dans un ou plusieurs <i>fichiers</i> la présence d'une <i>expression</i> régulière.

	Options -b Précède chaque ligne de sa référence en octet depuis le début du fichier. -c Affiche uniquement le nombre de fois qu'apparaît l'<i>expression</i>. -h Affiche les lignes sans préciser le nom des <i>fichiers</i>. -i Ne fait pas de distinction entre majuscules et minuscules. -l Affiche le nom des <i>fichiers</i> dans les quels l'<i>expression</i> apparaît sans préciser les lignes (inverse de -h). -n Précède chaque ligne de son numéro. -s Supprime l'affichage des messages d'erreur. -v Affiche les lignes où l'<i>expression</i> n'apparaît pas. Exemples Affiche le nombre d'utilisateurs ayant pour shell Zsh <pre>grep -c /bin/zsh /etc/passwd</pre> Affiche la liste des fichiers inclus dans le programme c prog.c <pre>grep '^#include' prog.c</pre>
gzip	gzip [<i>options</i>] [<i>fichiers</i>] Réduit la taille d'un ou de plusieurs <i>fichiers</i> en utilisant l'algorithme de compression Lempel-Ziv et les renomme en <i>fichier.gz</i>. gunzip et zcat permettent de revenir aux anciens fichiers. Ce compresseur est plus performant que compress. Options -c Ecrit sur le canal standard de sortie (sans changer les <i>fichiers</i>). -d Décompresse. -f Force l'opération sans demander de confirmation. -l Affiche des informations sur les <i>fichiers</i> compressés. -r Comprime le contenu de répertoires. -t Teste les <i>fichiers</i> compressés.
gunzip	gunzip [<i>options</i>] [<i>fichiers</i>] Décompresse un ou plusieurs <i>fichiers</i> traités par gzip (extension .gz) Options -c Ecrit sur le canal standard de sortie (sans changer les <i>fichiers</i>). -f Force l'opération sans demander de confirmation. -l Affiche des informations sur les <i>fichiers</i> compressés. -r Décompresse le contenu de répertoires. -t Teste les <i>fichiers</i> compressé.

head	head [<i>options</i>] <i>fichiers</i> Affiche les premières lignes (par défaut les 10 premières) d'un ou plusieurs <i>fichiers</i> et/ou du canal d'entrée standard si « - » est précisé. Lorsque plusieurs fichiers sont traités, leur début est précédé d'une entête contenant leur nom entouré de ==> <==. Options -<i>nb</i> Identique à -n <i>nb</i>. -c <i>nb</i>[bkm] Fixe la taille des débuts de fichiers à <i>nb</i> octet, <i>nb</i> blocs de 512 octets (b), <i>nb</i> kilo-octet (k) ou <i>nb</i> méga-octet (m). -n <i>nb</i> Fixe le nombre de lignes à <i>nb</i>. -q Interdit l'affichage d'entête.
kill	kill [<i>options</i>] <i>IDs</i> Termine un ou plusieurs processus référencés par leurs <i>IDs</i>. Options -l Affiche la listes des signaux. -<i>signal</i> Le <i>signal</i> (référéncé par son numéro ou par son nom) est envoyé aux processus. Le signal 9 (KILL) tue systématiquement les processus. Exemple Terminaison du processus 5437. <pre>\$ ps PID TTY STAT TIME COMMAND 5437 p3 S N 0:02 xdvi.bin -name xdvi unix-2.dvi 5597 p1 R 0:00 ps \$ kill -9 5437 [1] + 5597 killed xdvi.bin</pre>
ld	ld [<i>options</i>] <i>fichiers</i> Associe un ensemble de <i>fichiers</i> pour créer un exécutable. Par défaut, l'exécutable créée est nommé a.out. ld est l'éditeur de liens invoqué par gcc, g++ et g77. Options -g Effectue la compilation avec des symboles de débogage utilisés par gdb. -Lrep Ajoute le répertoire <i>rep</i> dans la liste des répertoires explorés lors de la recherche des bibliothèques pour l'édition de liens. -ltruc Ajoute la bibliothèque libtruc.a dans la liste des bibliothèques pour l'édition de liens. -o fichier Donne un <i>nom</i> au fichier généré (objet ou exécutable). Cette commande ne s'utilise quasiment pas directement.

ln	ln [<i>options</i>] <i>fichier1</i> <i>fichier2</i> ln [<i>options</i>] <i>fichiers</i> <i>répertoire</i> Crée des pseudonymes (liens) pour des fichiers, permettant ainsi d'y accéder sous différents noms. La première syntaxe permet de créer un lien nommé <i>fichier2</i> pour <i>fichier1</i>. Si <i>fichier2</i> est un fichier qui existe déjà, il est écrasé. La seconde syntaxe permet de créer des liens portant le même nom dans un autre <i>répertoire</i>. Options -f Force l'opération sans demander de confirmation (en cas d'écrasement de fichiers). -n Interdit l'écrasement de fichiers. -s Crée un lien symbolique. Cela permet de garder une référence (visible à la commande ls -l) avec le fichier lié. La commande ln s'utilise en fait <i>systématiquement</i> avec l'option -s.
lpq	lpq [-Pimprimante] [<i>IDs</i>] [<i>utilisateur</i>] Affiche l'état des impressions en attente sur une <i>imprimante</i>. Il est restreint à une liste de demandes (précisées par leurs numéros d'identification (<i>IDs</i>) ou par un nom d'<i>utilisateur</i>.
lpr	lpr [<i>options</i>] [-Pimprimante] [<i>fichiers</i> -] Envoie des <i>fichiers</i> à une <i>imprimante</i>. Pour imprimer le canal d'entrée standard, il faut préciser « - » à la place des <i>fichiers</i>. Options -h Retirer la page de garde. -m Envoie un mël en fin d'impression. -r Efface le fichier après la mise en attente ou l'impression (avec l'option -s). -s Crée un lien symbolique. Habituellement les <i>fichiers</i> sont copiés dans le répertoire d'attente (<i>/var/spool/lpd/...</i>). Cette option permet de ne créer que des liens symboliques plutôt que de copier de gros <i>fichiers</i>. Cela veut dire que les <i>fichiers</i> ne doivent pas être modifier ou effacer avant d'avoir été entièrement imprimés.
lprm	lprm [-Pimprimante] [<i>IDs</i>] [<i>utilisateur</i>] Annule les demandes d'impression faites par lpr. Il est possible de préciser les demandes par leur(s) <i>IDs</i>, par l'<i>imprimante</i> pour laquelle elles sont actuellement en attente ou par le nom de l'<i>utilisateur</i> qui a effectué ces demandes d'impression (il faut avoir des droits particuliers pour cela). Utilisez lpq pour connaître les numéros d'identifications (<i>ID</i>) des demandes d'impression.
ls	ls [<i>options</i>] [<i>noms</i>] Seule, ls liste les fichiers dans le répertoire courant. Si un ou plusieurs <i>noms</i> sont précisés, ls affiche les fichiers qui appartiennent aux répertoires nommés ou dont le nom correspond à l'un de ceux spécifiés (les <i>noms</i> peuvent contenir des méta-caractères). Les <i>options</i> permettent de modifier la liste et de l'enrichir d'informations secondaires.

	Options -a Affiche tous les fichiers même ceux cachés commençant par un « . ». -b Présente les caractères non affichables en octal. -c Affiche les fichiers par date de création/modification. -C Affiche les informations en colonnes classées verticalement (par défaut). -d Affiche uniquement les répertoires mais pas leur contenu. -f Interprète tous les <i>noms</i> comme des répertoires (les fichiers sont ignorés). -F Etiquette les répertoire par un « / », les exécutables par un « * » et les liens symboliques par un « @ ». -i Liste les I-node de chaque fichier. -l Affichage long (droits, nom et groupe du propriétaire, taille, date de dernière modification). -L Affiche le contenu d'un répertoire pointé par un lien symbolique plutôt que le lien lui-même. -m Affiche les noms séparés par des virgules. -n En complément de -l, affiche le nom et le groupe des propriétaires en fonction de leur UID et de leur GID. -o Identique à -l sans afficher le groupe des propriétaires. -q Remplace les caractères non affichables par « ? ». -r Classe les noms dans l'ordre inverse. -R Affiche le contenu des sous-répertoire de la même façon que le répertoire courant. -s Affiche la taille des fichiers en blocs (classiquement 512 octets ou 1 kilo-octet suivants les systèmes). -t Classe les fichiers par date de dernière modification. -C Affiche les informations en colonnes classées horizontalement (par défaut). -1 Affiche en information une par une en ligne. Exemples Liste tous les fichiers du répertoire courant en colonnes avec des étiquettes pour les fichiers particuliers. <pre>ls -aCF</pre> Liste les fichiers C du répertoire prog par ordre de mise à jour. <pre>ls -ltr prog/*.c</pre>
mail	mail [<i>options</i>] [<i>utilisateurs</i>] Lit (si aucun <i>utilisateur</i> n'est spécifié) ou envoie à un ou plusieurs <i>utilisateurs</i> un courrier électronique. Il est possible d'obtenir une liste sommaire des commandes par « ? » en mode lecture. Pour l'envoi de courrier, le corps du mël est récupéré du canal d'entrée standard (fin marquée par <i>EOF</i>).

	Options d'envoi de mël -b u Envoie en secret une copie à <i>u</i>. -c u Envoie une copie à <i>u</i>. -s s Fixe le sujet du mël à <i>s</i> (par défaut, mail demande un sujet pour la lettre). Option de lecture de mël -f b Indique que votre boîte à mël est <i>b</i> (par défaut celle-ci porte votre nom d'utilisateur et se trouve dans le répertoire /var/spool/mail/). Exemple Envoie de prog.c à bill. <pre>cat prog.c mail -s "Programme C" bill</pre> ou <pre>mail -s "Programme C" bill < prog.c</pre>
man	man [<i>options</i>] [<i>section</i>] <i>sujet</i> Recherche et affiche la page de manuel associée à un <i>sujet</i> (souvent une commande Unix, une fonction C ou Fortran ...). La recherche s'effectue de la section 1 à la section 10 et <i>man</i> s'arrête à la première page trouvée. Il est possible de préciser à quelle <i>section</i> se rapporte le <i>sujet</i> recherché. Si le <i>sujet</i> contient un / man essaie d'afficher le fichier correspondant au nom du <i>sujet</i>. Options -a Affiche toutes les pages de manuel associé au <i>sujet</i>. -K Recherche une chaîne de caractères parmi toutes les pages. -M liste précise la <i>liste</i> des répertoires à explorer. Normalement, man récupère la liste contenu dans la variable MANPATH. -P disp Utilise <i>disp</i> pour afficher les pages. -w Affiche les fichiers contenant les pages de manuel. Sans <i>sujet</i>, man affiche la liste des répertoires explorés (variable MANPAH). Exemples Affiche la page truc contenue dans le répertoire courant. <pre>man ./truc.4</pre> Liste les pages concernant printf. <pre>man -wa printf</pre>

mesg	mesg [<i>options</i>] Autorise ou interdit l’affichage de messages dans votre terminal. Sans option, mesg affiche l’état de la permission. Options n Interdit les messages de write. y Autorise les messages de write (par défaut).
mkdir	mkdir [<i>options</i>] <i>répertoires</i> Crée un ou plusieurs <i>répertoires</i>. Vous devez avoir les droits d’écriture dans le répertoire père pour pouvoir créer un répertoire. Voir aussi rmdir. Options -m mode Fixe les droits des nouveaux <i>répertoires</i>. -p Crée les répertoires pères nécessaires s’ils n’existent pas. Exemple Création du répertoire prog. <pre>mkdir prog</pre>
more	more [<i>options</i>] [<i>fichiers</i>] Affiche les <i>fichiers</i> dans un terminal page à page. Certaines touches vont vous permettre de naviguer, et faire évoluer l’affichage. Touches RETURN Passe à la ligne suivante. ESPACE Passe à la page suivante. h Affiche un écran d’aide. q Quitte. / recherche dans un fichier. :n Passe au fichier suivant. Options -c Affiche une page entière sans défilement. -d more affiche en dernière ligne « [Press space to continue, 'q' to quit.] » et « [Press 'h' for instructions.] » au lieu de faire « sonner » le terminal lorsque l’on appuie sur une touche sans fonctionnalité. -f Compte les lignes réelles plutôt que les lignes de l’écran. -l Ignore le caractère ^L. -s Ramène plusieurs lignes vides d’affilé en une seule. -nb Utilise <i>nb</i> lignes par page (par défaut c’est le nombre total de lignes du terminal). +num Commence l’affichage à la ligne <i>num</i>.

	Exemples <pre>more prog.c cat prog.c more</pre>																		
mv	mv [<i>options</i>] <i>sources destination</i> Permet de déplacer ou de renommer des fichiers et des répertoires. mv fonctionne de la façon suivante : <table><tr><th><i>Sources</i></th><th><i>Destination</i></th><th><i>Résultat</i></th></tr><tr><td>Fichier</td><td><i>nom</i></td><td>Renomme le fichier en <i>nom</i>.</td></tr><tr><td>Fichier</td><td>Fichier existant</td><td>Ecrase le fichier destination par le fichier source.</td></tr><tr><td>Répertoire</td><td><i>nom</i></td><td>Renomme le Répertoire en <i>nom</i>.</td></tr><tr><td>Répertoire</td><td>Répertoire existant</td><td>Déplace le répertoire source dans le répertoire destination.</td></tr><tr><td>Un ou plusieurs fichiers</td><td>Répertoire existant</td><td>Déplace les sources dans le répertoire destination.</td></tr></table> Options -f Force l'opération sans demander de confirmation. -i Demande une confirmation avant d'écraser un fichier existant. -- Indique la fin des options de mv. Ceci est utile pour pouvoir agir sur un fichier commençant par « - ».	<i>Sources</i>	<i>Destination</i>	<i>Résultat</i>	Fichier	<i>nom</i>	Renomme le fichier en <i>nom</i> .	Fichier	Fichier existant	Ecrase le fichier destination par le fichier source.	Répertoire	<i>nom</i>	Renomme le Répertoire en <i>nom</i> .	Répertoire	Répertoire existant	Déplace le répertoire source dans le répertoire destination.	Un ou plusieurs fichiers	Répertoire existant	Déplace les sources dans le répertoire destination.
<i>Sources</i>	<i>Destination</i>	<i>Résultat</i>																	
Fichier	<i>nom</i>	Renomme le fichier en <i>nom</i> .																	
Fichier	Fichier existant	Ecrase le fichier destination par le fichier source.																	
Répertoire	<i>nom</i>	Renomme le Répertoire en <i>nom</i> .																	
Répertoire	Répertoire existant	Déplace le répertoire source dans le répertoire destination.																	
Un ou plusieurs fichiers	Répertoire existant	Déplace les sources dans le répertoire destination.																	
passwd	passwd [<i>utilisateur</i>] Permet de changer le mot de passe d'un <i>utilisateur</i>. Seul l'administrateur peut changer le mot de passe d'un autre utilisateur que lui même.																		
printenv	printenv [<i>variables</i>] Affiche les <i>variables</i> d'environnement spécifiée. Si aucune variable n'est précisée, printenv affiche toutes les variables d'environnement.																		
ps	ps [-][<i>options</i>] [<i>PIDs</i>] Affiche la liste des processus actif ou simplement ceux précisés par leur <i>PIDs</i> (numéro d'identification de processus). Sans option, ps affiche dans un format court la liste des processus lancés par l'utilisateur. Le « - » est un héritage de la normalisation BDS ; il ne devrait plus apparaître dans la normalisation Unix-98. Les informations sont affichées en colonnes. Suivant les options spécifiées un certain nombre de champs sont affichés dont les entêtes sont détaillées ci-dessous.																		

	Options l Format long. u Format utilisateur. j Format tâche. s Format signal. v Format mémoire virtuelle. m Format mémoire. f Affichage en « arbre ». a Affichage des processus des autres utilisateurs. x Affichage de tous les processus (même ceux non liés à un terminal). c Raccourcit le nom de la commande associée au processus. e Affiche les variables d'environnement. w Allonge d'une ligne le format d'affichage. Vous pouvez mettre jusqu'à 100 w. h Retire l'entête. r Affiche uniquement les tâches en action. n Affiche les champs USER et WCHAN sous forme de nombre. Champs COMMAND Commande associée. CPU Pourcentage de puissance de calcul utilisée. MEM Pourcentage de mémoire utilisée. NI Niveau de priorité. Plus il est élevé, moins le processus est prioritaire. PAGEIN Nombre de pages mémoire. PID Numéro d'identification du processus. PPID Numéro d'identification du père du processus. PRI Ordre de priorité. RSS Taille du programme en mémoire. SHARE Mémoire partagée. SIZE Taille de mémoire virtuelle (texte+données+pile). START Heure de lancement de la tâche. STAT Information d'état : R ↔ actif, S ↔ en sommeil, D ↔ en sommeil non interrompable, T ↔ arrêté, Z ↔ zombie, W ↔ sans page résidente et N ↔ si le processus est moins prioritaire. SWAP Mémoire secondaire (en Ko). TIME Temps d'inactivité. TRS Taille résidente de texte. TTY Nom du terminal. UID Numéro d'identification du propriétaire. USER Nom du propriétaire. WCHAN Nom de la fonction du noyau dont le processus attend une réponse.

rm	rm [<i>options</i>] <i>fichiers</i> Efface un ou plusieurs <i>fichiers</i>. Vous devez avoir les droits d'écriture dans le répertoire sans pour autant avoir des droits sur les <i>fichiers</i> eux-mêmes. Si vous n'avez pas de droits d'écriture sur les <i>fichiers</i>, un message vous demandera si vous confirmez l'opération ou non. Options -f Efface les fichiers protégés en écriture sans demander de confirmation. -i Demande une confirmation pour tout effacement. -r Efface des répertoires ainsi que leurs contenus. A utiliser avec précaution. -- Indique la fin des options de rm. Ceci est utile pour pouvoir effacer un fichier commençant par « - ».
rmdir	rmdir [<i>options</i>] <i>répertoires</i> Efface les <i>répertoires</i> (les noms seulement pas leurs contenus). Pour cela les répertoires doivent être vides (sinon, rm -r peut être utilisé à la place). Voir aussi mkdir. Options -p Efface les répertoires pères précisés s'ils sont vides. Ceci est utile pour effacer des arbres de répertoires. -s Supprime les messages d'erreurs générés par -p.
tar	tar [-<i>clé</i>][<i>options</i>] [<i>fichiers</i>] Gère un ensemble de <i>fichiers</i> associés dans un fichier de sauvegarde. Ceci est le plus souvent utilisé pour faciliter le transport de plusieurs <i>fichiers</i>. La liste des <i>fichiers</i> peut contenir des répertoires ; les fichiers et répertoires contenus dans ces derniers seront archivés en conservant l'arborescence initiale. Si le nom de la Clé c Crée une sauvegarde. r Ajoute des <i>fichiers</i> à une sauvegarde. t Liste le contenu d'une sauvegarde. u Ajoute des <i>fichiers</i> nouveaux ou plus récents à une sauvegarde. x Extrait des <i>fichiers</i> d'une sauvegarde. Par défaut, tar extrait tous les fichiers contenus dans la sauvegarde. Options f <i>sauv</i> Donne le nom <i>sauv</i> à la sauvegarde. Si <i>sauv</i> vaut « - », l'archive est envoyée sur le canal de sortie standard. Par défaut, la sauvegarde est /dev/rmt0 (c'est-à-dire le lecteur de bande). Ce paramètre est donc essentiel. Par convention, l'extension d'une sauvegarde est .tar. v Affiche des informations concernant les <i>fichiers</i>.

	Exemple Création de la sauvegarde prog.tar contenant les fichiers du répertoire c/. <pre>tar cfv prog.tar c/</pre> Extraction du fichier c/td1.c de la sauvegarde. <pre>tar xfv prog.tar c/td1.c</pre>
sort	sort [<i>options</i>] [<i>fichiers</i>] Classe et affiche les lignes d'un ou plusieurs <i>fichiers</i> classiquement par ordre alphabétique. Options -b Supprime les espaces et les tabulations. -c Vérifie si les fichiers sont triés ou pas. -d Trie comme pour un répertoire téléphonique ne tenant compte que des lettres et des chiffres. -f Ignore la différence entre lettres minuscule et majuscule. -i Ignore les caractères non imprimables. -M Compare les 3 premiers caractères en tant que mois (JAN < FEB < ... < DEC). -m Lie les fichiers en entrée. -n Trie suivant l'ordre arithmétique. -o<i>fic</i> Redirige le canal de sortie standard vers le fichier <i>fic</i>. -r Trie dans l'ordre inverse. -tc Indique que le caractère de séparation entre les champs est le caractère <i>c</i>. -u Enlève les lignes identiques. +<i>n</i>[-<i>m</i>] Les champs pris en compte pour le classement sont alors les champs <i>n</i> à <i>m</i>-1 (par défaut, la fin de la ligne). Exemples Classe la liste des fichiers par taille décroissante. <pre>ls -ls sort +1n -r</pre> Classe la liste des comptes utilisateurs par rapport à leur UID (3^e champ). <pre>sort +2n -t: /etc/passwd</pre>
split	split [<i>options</i>] <i>fichier</i> [<i>debutN</i>] Découpe un <i>fichier</i> en plusieurs fichiers de taille identiques (soit en ligne soit en octet) dont les noms commencent par <i>debutN</i> (par défaut x). Par défaut split découpe les fichiers par 1000 lignes. Si <i>fichier</i> vaut « - », split récupère le canal d'entrée standard.

	Options -nb Identique à -l nb. -b nb[bkm] Fixe la taille des fichiers de sortie à <i>nb</i> octet, <i>nb</i> blocs de 512 octets (b), <i>nb</i> kilo-octet (k) ou <i>nb</i> méga-octet (m). -l nb Fixe le nombre de lignes à <i>nb</i>.
strings	strings fichiers Affiche les séquences de caractères imprimables trouvées dans un ou plusieurs <i>fichiers</i>. Options - Identique à -a. -a Recherche dans tout le fichier et non simplement dans la zone de données des fichiers objets. -n n Fixe la taille minimale des chaînes à <i>n</i> caractères (par défaut 4). -o Identique à -t o. -t m Affixe la position de la chaîne dans le fichier dans la base fixée par <i>m</i> : octal ↔ o, hexadécimal ↔ x ou décimal ↔ d.
strip	strip fichier Supprime les symboles ajoutés par le compilateur. Cela permet de réduire la taille d'un exécutable, d'un fichier objet ou d'une archive (créée par ar).
tail	tail [options] fichiers Affiche les dernières lignes (par défaut les 10 dernières) d'un ou plusieurs <i>fichiers</i> et/ou du canal d'entrée standard si « - » est précisé. Lorsque plusieurs fichiers sont traités, leurs fins sont précédées d'une entête contenant leur nom entouré de ==> <==. Options -nb Identique à -n nb. -c nb[bkm] Fixe la taille des fins de fichiers à <i>nb</i> octet, <i>nb</i> blocs de 512 octets (b), <i>nb</i> kilo-octet (k) ou <i>nb</i> méga-octet (m). -f Continue d'attendre de nouveaux caractères. Ceci est très utile pour suivre l'évolution d'un fichier de log. -n nb Fixe le nombre de lignes à <i>nb</i>. -q Interdit l'affichage d'entête. Exemple Affiche des 10 dernières lignes du fichier /var/log/message avec poursuite. <pre>tail -10 -f /var/log/message</pre>

telnet	telnet <i>machine</i> [<i>port</i>] Permet de communiquer avec une <i>machine</i> distante en utilisant le protocole TELNET. Il est possible de préciser sur quel <i>port</i> doit s'établir la communication (par défaut 23). Vous pouvez obtenir le mode de commande de telnet en tapant CONTROL-]. Exemple <pre>\$ telnet marval Trying 193.55.62.80... Connected to marval.univ-mlv.fr. Escape character is '^]'. ...</pre>
tee	tee [-a] <i>fichiers</i> Duplique le canal d'entrée standard vers un ou plusieurs <i>fichiers</i> et vers le canal de sortie standard. L'option -a permet d'ajouter le canal d'entrée standard à des <i>fichiers</i> sans les écraser.
uncompress	uncompress [<i>options</i>] [<i>fichiers</i>] Décompresse un ou plusieurs <i>fichiers</i> traités par compress (extension .Z) Options -c Ecrit sur le canal standard de sortie (sans changer les <i>fichiers</i>). -f Force l'opération sans demander de confirmation. -l Affiche des informations sur des <i>fichiers</i> compressés.
uptime	uptime Donne une ligne d'information sur la machine : l'heure, depuis combien de temps la machine fonctionne, combien d'utilisateurs sont connectés et la charge de calcul moyenné sur 1, 5 et 15 minutes. Exemple <pre>\$ uptime 1:05pm up 4:35, 1 user, load average: 0.09, 0.20, 0.22</pre>
which	which <i>programme</i> Indique le chemin d'accès exact d'un <i>programme</i> Unix.

	Exemple <pre>\$ which gcc /usr/bin/gcc</pre>
who	who [<i>options</i>] [am i] Affiche des informations au sujet des utilisateurs actuellement logués. En l'absence de paramètre, who affiche les informations suivantes : nom de l'utilisateur, nom du terminal, date et heure de connection et nom de la machine distante ou du terminal X. Options -H Affiche une ligne d'entête. -i Ajoute le temps d'inactivité de chaque utilisateur. -m N'affiche que les informations concernant l'utilisateur associé au canal d'entrée standard. -q Affiche uniquement le nom des utilisateurs et leur nombre. -T Ajoute l'état de réception des messages : +, - ou ?. -u Identique à -i. -w Identique à -T. La commande who am i (équivalente à whoami) indique uniquement le nom de l'utilisateur propriétaire du canal d'entrée standard. Exemple <pre>\$ who mazet ttyt1 Oct 15 10:10 (:0.0)</pre>
write	write <i>utilisateur</i> [<i>terminal</i>] <i>message</i> Ecrit un <i>message</i> dans le terminal d'un <i>utilisateur</i> logué. Un message se termine par <i>EOF</i>. Si un <i>utilisateur</i> est logué sur plusieurs terminaux, il faut préciser le numéro du <i>terminal</i>. Le message s'affichera si l'utilisateur l'a permis (grâce à la commande mesg).
yppasswd	yppasswd [<i>options</i>] [<i>utilisateur</i>] Permet de changer le mot de passe d'un <i>utilisateur</i> à travers le réseau (lorsque la table des mots de passe est distribuée via NIS). Seul l'administrateur peut changer le mot de passe d'un autre utilisateur que lui même. Options -f Equivalent à ypchfn. -l Equivalent à ypchsh. -p Equivalent à yppasswd.

ypchfn	ypchfn [<i>options</i>] [<i>utilisateur</i>] Permet de changer les informations utilisées par finger. Seul l'administrateur peut changer le informations d'un autre utilisateur que lui même. <i>Options</i> <table><tr><td>-f</td><td>Equivalent à ypchfn.</td></tr><tr><td>-l</td><td>Equivalent à ypchsh.</td></tr><tr><td>-p</td><td>Equivalent à yppasswd.</td></tr></table>	-f	Equivalent à ypchfn .	-l	Equivalent à ypchsh .	-p	Equivalent à yppasswd .
-f	Equivalent à ypchfn .						
-l	Equivalent à ypchsh .						
-p	Equivalent à yppasswd .						
ypchsh	ypchsh [<i>options</i>] [<i>utilisateur</i>] Permet de changer de shell. Seul l'administrateur peut changer le shell d'un autre utilisateur que lui même. <i>Options</i> <table><tr><td>-f</td><td>Equivalent à ypchfn.</td></tr><tr><td>-l</td><td>Equivalent à ypchsh.</td></tr><tr><td>-p</td><td>Equivalent à yppasswd.</td></tr></table>	-f	Equivalent à ypchfn .	-l	Equivalent à ypchsh .	-p	Equivalent à yppasswd .
-f	Equivalent à ypchfn .						
-l	Equivalent à ypchsh .						
-p	Equivalent à yppasswd .						
zcat	zcat [<i>options</i>] <i>fichiers</i> Décompresse et affiche le contenu un ou plusieurs <i>fichiers</i> compressés par compress ou gzip. Cette commande est équivalente à uncompress -c ou à gunzip -c.						

Chapitre 3

Le Shell et ses commandes

3.1 Introduction au Shell

Commençons par une petite métaphore : supposons qu'un système Unix puisse s'assimiler à une voiture. Pour conduire une voiture, vous appuyez sur l'embrayage, vous actionnez le levier de vitesse, vous passez une vitesse et vous lâchez l'embrayage et appuyez sur l'accélérateur. Ce dont vous venez de vous servir ne sont en fait que des leviers de commandes, vous ne savez pas comment vos ordres ont été transmis au moteur de la voiture et, la plupart d'entre vous n'en ont rien à faire.

Un Shell a à peu près la même fonction vis-à-vis d'un système Unix. Il vous permet de naviguer dans le système, exécuter des commandes sans avoir à vous soucier de la façon dont réagit le système. C'est essentiellement un interpréteur de commande mais il permet aussi de gérer des processus, un ensemble de variables paramétrisant votre environnement et de programmer des tâches relativement complexes (fichier contenant un ensemble de commande Unix).

Suivant le système sur lequel vous travaillez, un ou plusieurs shells sont disponibles ; ils n'ont pas exactement les mêmes fonctions ni la même syntaxe. La figure 3.1 vous permettra de mieux comprendre les différentes évolutions des shells Unix.

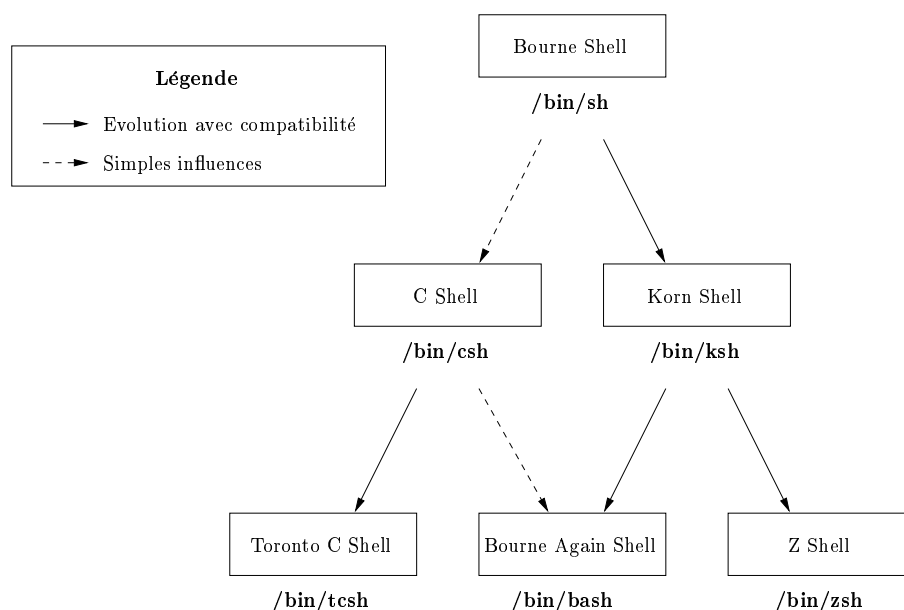


FIG. 3.1 – Evolution des Shells Unix.

Le Bourne Shell (**sh**) est le premier shell créé et il reste celui (ou un de ses clones) dont tout système Unix a besoin pour lancer les outils de configuration au démarrage de la machine. Les shell **ksh**, **bash** et **zsh** utilisent des syntaxes plus riches mais qui restent compatibles avec **sh**. Le **csh** et le **tcsh** ont par contre des

syntaxes beaucoup plus différentes (de **sh**) et très proches de celle du langage C.

La seule chose dont vous pouvez être sûr, c'est qu'au moins un shell compatible **sh** existe sur toute machine Unix. C'est pourquoi, ce manuel se limitera à décrire les fonctions essentielles du Bourne Shell bien que je vous recommande d'utiliser **zsh**.

3.2 Les méta-caractères

Ils permettent de créer des noms génériques.

*	Correspond à une chaîne de caractères quelconque (même vide).
?	Correspond à un caractère quelconque.
[abc...]	Correspond à un des caractères spécifiés dans la liste. Il est possible de décrire des listes de caractères en ne précisant que le premier caractère et le dernier caractère de la liste séparé par un moins (par exemple : a-z, I-M, 0-9).
[!abc...]	Correspond à tout caractère non spécifié dans la liste.

Exemples

\$ ls toto*	Liste les fichiers toto , toto.c et toto.txt .
\$ cat prog?.c	Affiche le contenu du fichier prog2.c .
\$ ls [A-D]*	Liste les fichiers commençant par A , B , C ou D .

3.3 Les guillemets

Ils servent à délimiter des chaînes des caractères mais permettent aussi de créer de nouvelles chaînes par concaténation.

' '	Tous les caractères compris entre les apostrophes (<i>quote</i> en anglais) sont pris en compte tel quel.
' '	La chaîne comprise entre les anti-apostrophes (<i>back-quote</i> en anglais) est évaluée en tant que commande et son résultat vient se substituer à la chaîne initiale.
\	Tout caractère précédé d'un \ est pris en compte tel quel. Ceci est très utile pour afficher tous les caractères interprétés par le shell.
" "	Seul les caractères non interprétables (espace compris) sont pris en compte tel quel, les autres sont interprétés.

3.4 Les séparateurs

Ils jouent un rôle particulier en shell, puisque, la plus part du temps, ils ne sont pas utilisés.

cmd &	Lance la commande <i>cmd</i> en arrière plan.
cmd1 ; cmd2	Séparateur de commandes. Permet d'écrire deux commandes sur la même ligne.
(cmd1 ; cmd2)	Ensemble de commandes groupées.
cmd1 cmd2	<i>Pipe</i> . Permet de rediriger le canal de sortie de la commande <i>cmd1</i> vers le canal d'entrée de <i>cmd2</i> . Les informations vont être traitées d'abord par <i>cmd1</i> puis par <i>cmd2</i> .
cmd1 && cmd2	Exécute la commande <i>cmd1</i> puis, si <i>cmd1</i> s'est bien déroulée, exécute <i>cmd2</i> .
cmd1 cmd2	Exécute la commande <i>cmd1</i> puis, si <i>cmd1</i> s'est mal déroulée, exécute <i>cmd2</i> .
{ cmd1 ; cmd2 }	Permet d'exécuter un ensemble de commandes dans un sous-shell.

Exemples

\$ gcc -c prog.c &	Lance une compilation en tâche de fond.
\$ cd ; ls	Revient à la racine perso. et liste les fichiers.
\$ (date; who) > log	Les résultats sont envoyés vers un fichier.
\$ ls -ls sort -n	Liste les fichiers par taille croissante.

3.5 Les redirections

Il est souvent utile de pouvoir changer les canaux d'entrée, de sortie ou d'erreur de commande.

<code>cmd [n]> fich</code>	Envoie le canal <i>n</i> de la commande <i>cmd</i> vers le fichier <i>fich</i> (en écrasant <i>fich</i> s'il existe). Si <i>n</i> n'est spécifié ou s'il vaut 1, c'est le canal de sortie, s'il vaut 2 c'est le canal d'erreur.
<code>cmd [n]>> fich</code>	Envoie le canal <i>n</i> de la commande <i>cmd</i> vers fichier <i>fich</i> (en ajoutant à <i>fich</i> s'il existe). Si <i>n</i> n'est spécifié c'est le canal de sortie.
<code>cmd < fich</code>	Renvoie dans le canal d'entrée de la commande <i>cmd</i> le fichier <i>fich</i> .
<code>cmd << [texte]</code>	Renvoie dans le canal d'entrée de la commande <i>cmd</i> toutes les chaînes de caractères provenant du canal d'entrée standard jusqu'à qu'apparaisse la phrase <i>texte</i> . Si aucun <i>texte</i> n'est spécifié, la lecture s'arrête sur <i>EOF</i> .
<code>cmd [m]>&n</code>	Envoie le canal <i>n</i> vers le canal <i>n</i> . Si <i>m</i> n'est spécifié c'est le canal de sortie.

Exemples

```
$ cat prog.c | grep ^# Affiche les lignes de prog.c commençant par #.
$ cat func.c >> prog.c Ajout de func.c en fin de prog.c.
$ gcc -c prog.c 2>g.log Récupère tous les messages d'erreurs de la compilation dans g.log
```

3.6 Les variables

Elles permettent de stocker des données qui sont utilisées soit par le shell lui-même, soit par les commandes et logiciels Unix (récupération de fichiers de ressources, liste d'options par défaut, ...).

<code>var=val</code>	Met <i>val</i> dans la variable <i>var</i> .
<code>\${var}</code>	Utilise le contenu de la variable <i>var</i> . Les accolades sont facultatives si le nom de la variable est séparable.
<code>\${var}:-val</code>	Utilise le contenu de la variable <i>var</i> si elle existe sinon utilise la valeur <i>val</i> .
<code>\${var}:=val</code>	Utilise le contenu de la variable <i>var</i> si elle existe sinon utilise la valeur <i>val</i> et assigne <i>val</i> à <i>var</i> .
<code>\${var}?:val</code>	Utilise le contenu de la variable <i>var</i> si elle existe sinon affiche « <i>var : val</i> » ou « <i>var : parameter not set</i> » si <i>val</i> est la chaîne nulle.
<code>\${var}:+val</code>	Utilise la valeur <i>val</i> si la variable <i>var</i> existe sinon ne fait rien.

Exemples

```
$ t=chose La variable t reçoit chose
$ t2=$t La variable t2 reçoit le contenu de t.
$ t3="pas grand $t" La variable t3 reçoit pas grand chose.
$ t4='$t' La variable t4 reçoit $t.
```

3.7 Les variables d'environnement du shell

Elles contiennent les informations nécessaires au bon fonctionnement du shell.

\$#	Nombre d'arguments de la ligne de commande.
\$-	Options actuellement utilisées.
\$?	Retour de la dernière commande exécutée.
\$\$	Numéro du processus courant.
#!	Numéro du dernier processus lancé en arrière plan.
\$0	Premier argument (c'est-à-dire, la commande elle-même).
\$n	<i>n</i> ^e argument (n est compris entre 1 et 9).
\$*	Tous les arguments (" \$1 \$2 ... ").
CDPATH	Liste de répertoires utilisée par cd en tant que raccourcis. Les répertoires sont séparés par des deux points. Par défaut, cette variable est vide.
HOME	Contient le nom du répertoire principal de l'utilisateur du shell.
IFS	Liste de caractères marquant la séparation entre deux mots. Par défaut, cette variable contient les caractères « espace », « tabulation » et « fin de ligne ».
MAIL	Contient le nom du fichier boîte aux lettres de l'utilisateur.
MAILCHECK	Temps (en seconde) entre deux vérification de l'état de la boîte aux lettres.
MAILPATH	Liste des boîtes aux lettres séparées par des deux points. Par défaut le shell affiche, « You have new mail » lors de la réception de nouveaux messages.
PATH	Liste des chemins d'accès des commandes Unix (les répertoires sont séparés par des deux points). Par défaut, cette variable contient /usr/bin .
PS1	Prompt standard. Par défaut \$.
PS2	Prompt de second niveau (utilisé pour des commandes tapées sur plusieurs lignes). Par défaut > .
SHELL	Nom du shell utilisé. Si vous utilisez un Bourne Shell, ce doit être /bin/sh .
TERM	Type du terminal. Si vous avez un problème, n'hésitez pas à revenir à un terminal basique tel que vt100 .

3.8 Les fonctions shell

La définition d'une fonction s'écrit comme ceci :

```
nom () { liste de commandes ; }
```

En Korn Shell, on peut ajouter le mot clé **function** mais cela n'ajoute qu'un peu de clarté à la programmation. Les fonctions s'utilisent alors comme de simples commandes.

Exemple

```
$ count () {
> ls $@ | wc -l
> }
```

La fonction **count** permet de compter le nombre de fichiers d'un répertoire ou plusieurs répertoires.

3.9 Le dièse et le nombre magique

Le dièse permet d'écrire des commentaires dans un fichier de commandes shell. Par contre, en début de tout fichier de commande shell, on trouve :

```
#!shell
```

C'est le « nombre magique » qui indique que le fichier est un programme qu'il faut exécuter via le *shell* spécifié.

Exemple

Le programme **count** s'écrit donc :

```
#!/bin/sh

# Message
echo -n " Nombre de fichier dans le(s) repertoire $@ : "
# Commande comptant les fichiers
ls | wc -l
```

3.10 Les commandes shell

alias	alias [<i>nom</i>[='commande']] C'est en fait une commande Korn Shell mais comme la plupart des shells l'inclue, vous pouvez l'utiliser pour vos fichiers de configurations. alias permet d'assigner une <i>commande</i> (avec ou sans paramètre) à un <i>nom</i>. Si la commande (et le signe égal) n'est pas précisé, alias affiche (si il existe) la définition de l'alias <i>nom</i>. Sans paramètre, alias affiche la liste de tous les alias précédemment définis. Voir unalias. <i>Exemples</i> <pre>allps='/bin/ps aux grep -v "^root"' clean='/bin/rm -rf *.o *~' copy=cp del='/bin/rm -i' dir='ls -la' la='/bin/ls -la' ll='ls -lg' ls='/bin/ls -CF' m=less nps='/bin/ps ux' xtar='tar -xvf'</pre>
bg	bg [<i>tâche</i>] Passe en arrière plan une <i>tâche</i> référencée par son numéro (voir la commande jobs). Celle-ci a été le plus souvent interrompue par un CONTROL-Z. Sans paramètre, bg agit sur le dernier processus interrompu. Pour un exemple, reportez-vous à la commande jobs.
break	break [<i>n</i>] Permet de sortir de boucles for, while ou until. Le paramètre permet de sortir du <i>n^e</i> niveau de boucles.

case	case <i>valeur</i> in <i>mot1</i>) <i>cmd1</i> ;; <i>mot2</i>) <i>cmd2</i> ;; . . . esac Exécute la première liste de commandes <i>cmd1</i> si la <i>valeur</i> correspond au <i>mot1</i> et sort (c'est-à-dire, l'exécution reprend après le mot clé esac). Exécute la seconde liste de commandes <i>cmd2</i> si la <i>valeur</i> correspond au <i>mot2</i> et sort. Et ainsi de suite. Les mots peuvent comporter des méta-caractères et il est possible de faire des choix multiples (séparés par des). Cette structure est très utile pour la gestion des paramètres d'un programme shell. Exemple Ce petit programme analyse le premier argument entré et initialise les variables adéquates. <pre>#!/bin/sh case \$1 in oui non) reponse=presente ;; -[lL]) liste=1 ;; *) echo Usage: \$0 [oui non] [-lL] exit 1 ;; esac</pre>
cd	cd [<i>rép</i>] cd [-] cd <i>ancien nouveau</i> Sans argument, cd change le répertoire de travail en la racine de l'utilisateur. Sinon, elle change le répertoire de travail en <i>rép</i>. Si <i>rép</i> est un chemin relatif (ne commençant pas par /) ne se trouvant pas dans le répertoire courant, le shell recherche parmi les chemins indiqués dans la variable CDPATH. cd - renvoie dans le précédent répertoire de travail. La dernière syntaxe permet de remplacer dans le nom du répertoire de travail courant <i>ancien</i> par <i>nouveau</i>.

	Exemple <pre>\$ pwd /var/spool/mail \$ cd mail lpd /var/spool/lpd</pre>																
do	do Mot clé indiquant le début d'une liste de commandes se terminant par done pour les boucles for, while et until.																
done	done Mot clé indiquant la fin d'une liste de commandes se commençant par do pour les boucles for, while et until.																
echo	echo [-n] [chaîne] Affiche la <i>chaîne</i> de caractères sur le canal de sortie standard. Si l'option -n est précisée, la ligne ne se termine pas par un retour à la ligne. Certains caractères non affichables peuvent être utilisés via des codes précédés d'un \. Codes <table> <tr> <td>\b</td> <td>Effacement.</td> </tr> <tr> <td>\c</td> <td>Omission de la fin de ligne (identique à l'option -n).</td> </tr> <tr> <td>\f</td> <td>Formfeed.</td> </tr> <tr> <td>\n</td> <td>Nouvelle ligne.</td> </tr> <tr> <td>\r</td> <td>Retour chariot.</td> </tr> <tr> <td>\t</td> <td>Tabulation.</td> </tr> <tr> <td>\</td> <td>Backslash.</td> </tr> <tr> <td>\0nnn</td> <td>Code ASCII <i>nnn</i> en octal.</td> </tr> </table>	\b	Effacement.	\c	Omission de la fin de ligne (identique à l'option -n).	\f	Formfeed.	\n	Nouvelle ligne.	\r	Retour chariot.	\t	Tabulation.	\	Backslash.	\0nnn	Code ASCII <i>nnn</i> en octal.
\b	Effacement.																
\c	Omission de la fin de ligne (identique à l'option -n).																
\f	Formfeed.																
\n	Nouvelle ligne.																
\r	Retour chariot.																
\t	Tabulation.																
\	Backslash.																
\0nnn	Code ASCII <i>nnn</i> en octal.																
esac	esac Voir case.																
exec	exec commande Permet d'exécuter une commande à la place du shell courant. Ceci est utile pour changer de shell sans conserver le précédent.																
exit	exit [n] Permet de sortir d'un script shell avec le status <i>n</i>. <i>n</i> peut valoir 0 (succès) ou tout autre valeur non nulle (échec). Si <i>n</i> n'est pas précisé, exit renvoie le status de la dernière commande exécutée.																
export	export [variables] Permet de rendre global (visible pour des shells secondaires) des <i>variables</i> qui, par défaut, sont locales au shell courant. Sans paramètre, export affiche la liste des variables actuellement exportées.																

	Exemple <pre>TERM=vt100 export TERM</pre>
fg	fg [<i>tâche</i>] Relance en premier plan une <i>tâche</i> référencée par son numéro (voir la commande jobs). Celle-ci a été le plus souvent interrompue par un CONTROL-Z. Sans paramètre, fg agit sur le dernier processus interrompu.
fi	fi Mot clé indiquant la fin d'une structure if
for	for <i>x</i> in <i>liste</i> do <i>commandes</i> done Pour la variable <i>x</i> appartenant à <i>liste</i> effectue l'ensemble de <i>commandes</i>. Exemple Ce petit script permet de compresser tous les fichiers qui ne le sont pas encore. <pre>for i in `ls *. [^Z]` do compress \$i done</pre>
if	if <i>cond1</i> then <i>cmd1</i> [elif <i>cond2</i> then <i>cmd2</i>] . . . [else <i>cmd3</i>] fi Si la condition <i>cond1</i> est atteinte, effectue la commande <i>cmd1</i> ; sinon, si la condition <i>cond2</i> est atteinte, effectue la commande <i>cmd2</i> ; sinon effectue la commande <i>cmd3</i>. Les conditions s'écrivent grâce à la commande shell test.

	Exemple Ce petit morceau de programme ajoute un 0 à gauche au nombre inférieur à 10. <pre>if [\$n -lt 10] then n="0"\$n fi</pre>
jobs	jobs [<i>option</i>] [<i>tâches</i>] Affiche la liste des tâches (ou simplement celles précisées par <i>tâches</i>) lancées dans le shell ainsi que leur état (« running » lancée en arrière plan ou « suspended » interrompue). Vous pouvez intervenir sur ces tâches via les commandes bg, fg et kill en spécifiant leur numéro précédé d'un %. Option -l Affiche en plus le numéro du processus. Exemples <pre>\$ jobs [1] running xload [2] + suspended xroger [3] - running xhextris \$ kill %1 [1] 671 terminated xload \$ jobs [2] + suspended xroger [3] - running xhextris \$ fg %2 [2] + continued xroger</pre>
nohup	nohup <i>commande</i> & Permet de lancer une commande en arrière plan qui ne s'achèvera pas après que l'utilisateur soit délogué. Exemple <pre>nohup octave < calcul.m > resultat.txt &</pre>
pwd	pwd Affiche le chemin complet du répertoire de travail courant.

read	read <i>var1</i> [<i>var2</i> ...] Lit une ligne du canal d'entrée standard et la stocke dans les variables <i>var1</i> , <i>var2</i> ... La dernière variable contient la fin de la ligne.
return	return [<i>n</i>] Permet de sortir d'une fonction shell avec le statut <i>n</i> . <i>n</i> peut valoir 0 (succès) ou tout autre valeur non nulle (échec). Si <i>n</i> n'est pas précisé, exit renvoie le statut de la dernière commande exécutée.
shift	shift [<i>n</i>] Passe en première position l'argument <i>n</i> (par défaut 2). Ceci est utile pour les boucles while .
stop	stop [<i>tâche</i>] Stoppe une <i>tâche</i> référencée par son numéro (voir la commande jobs) exécutée en arrière plan. C'est le complément de CONTROL-Z et de suspend . Sans paramètre, stop agit sur le dernier processus lancé en arrière plan.
suspend	suspend Stoppe la tâche en cour. Identique au CONTROL-Z.
test	test <i>condition</i> [<i>condition</i>] Evalue une <i>condition</i> et renvoie un statut 0 si le résultat est vrai sinon renvoie une valeur non nulle. Ces conditions permettent de tester des fichiers, des chaînes de caractères ou des nombres entiers. Condition sur les fichiers - b <i>f</i> <i>f</i> existe et est un périphérique bloc. - c <i>f</i> <i>f</i> existe et est un périphérique caractère. - d <i>f</i> <i>f</i> existe et est un répertoire. - f <i>f</i> <i>f</i> existe et est un fichier normal. - g <i>f</i> <i>f</i> existe et possède les droits g=s . - k <i>f</i> <i>f</i> existe et possède le « sticky bit ». - p <i>f</i> <i>f</i> existe et est un « pipe ». - r <i>f</i> <i>f</i> existe et est lisible. - s <i>f</i> <i>f</i> existe et n'est pas vide. - u <i>f</i> <i>f</i> existe et possède les droits u=s . - w <i>f</i> <i>f</i> existe et est écrasable. - x <i>f</i> <i>f</i> existe et est exécutable. Condition sur les chaînes de caractères - n <i>s1</i> La chaîne <i>s1</i> est non vide. - z <i>s1</i> La chaîne <i>s1</i> est vide. <i>s1</i> = <i>s2</i> <i>s1</i> et <i>s2</i> sont identiques. <i>s1</i> != <i>s2</i> <i>s1</i> et <i>s2</i> sont différentes.

	Condition sur les nombres entiers <i>n1 -eq n2</i> <i>n1</i> est égal à <i>n2</i>. <i>n1 -ge n2</i> <i>n1</i> est supérieur ou égal à <i>n2</i>. <i>n1 -gt n2</i> <i>n1</i> est strictement supérieur à <i>n2</i>. <i>n1 -le n2</i> <i>n1</i> est inférieur ou égal à <i>n2</i>. <i>n1 -lt n2</i> <i>n1</i> est strictement inférieur à <i>n2</i>. <i>n1 -ne n2</i> <i>n1</i> n'est pas égal à <i>n2</i>. Couplage de conditions <i>(cmd1)</i> Vraie si <i>cmd1</i> est vraie. Ceci est utile pour grouper des conditions. ! <i>cmd1</i> Vraie si <i>cmd1</i> est fausse. <i>cmd1 -a cmd2</i> Vraie si <i>cmd1</i> et <i>cmd2</i> sont vraies. <i>cmd1 -o cmd2</i> Vraie si <i>cmd1</i> ou <i>cmd2</i> sont vraies. Exemples <pre>while test \$# -gt 0 Tant qu'il y a des arguments... while [-n "\$1"] Tant qu'il n'y a pas d'argument vide... if [-d Td2] Si Td2 est un répertoire... if ["\$t" != "n"] si t n'est pas n...</pre>
type	type commande Indique si une <i>commande</i> est une commande shell, une commande Unix, un alias ou une fonction définie par l'utilisateur. Exemples <pre>\$ type mv echo mv is /bin/mv echo is a shell builtin</pre>
unalias	unalias nom Efface l'alias <i>nom</i> (fonction Korn Shell). Voir alias.
unset	unset var Efface la définition de la variable <i>var</i>.
until	until condition do <i>commandes</i> done Juqu'à ce que la <i>condition</i> soit vraie, exécute l'ensemble des <i>commandes</i>. Les conditions s'écrivent grâce à la commande shell test.

while	while <i>condition</i> do <i>commandes</i> done Tant que la <i>condition</i> est vraie, exécute l'ensemble des <i>commandes</i> . Les conditions s'écrivent grâce à la commande shell test .
--------------	--

Annexe A

Références et licence

La dernière version est disponible à l'adresse <http://www.softndesign.org/?page=manuels/unix-2.php>. Des versions Postscript et PDF y sont aussi disponibles.

Ce document est distribué sous licence GPL. Vous êtes autorisé à le copier et/ou le redistribuer intégralement ou en partie, à la seule condition que cette mention y reste présente et conformément aux dispositions de la Licence de Documentation Libre GNU telle que publiée par la Free Software Foundation ; version 1.2 de la licence, ou encore (à votre choix) toute version ultérieure.

Index

alias, 31
ar, 3
at, 4
atq, 5
atrm, 5

banner, 5
bg, 31
break, 31

c++, 5
cal, 6
case, 32
cat, 6
cc, 8
cd, 32
chfn, 8
chmod, 6
chsh, 8
compress, 8
cp, 9

diff, 9
do, 33
done, 33

echo, 33
esac, 33
exec, 33
exit, 33
export, 33

f77, 9
fg, 34
fi, 34
find, 9
finger, 10
for, 34
ftp, 10

g++, 11
g77, 11
gcc, 11
gdb, 12
grep, 12
gunzip, 13
gzip, 13

head, 14

if, 34

jobs, 35

kill, 14

ld, 14
ln, 15
lpq, 15
lpr, 15
lprm, 15
ls, 15

mail, 16
man, 17
mesg, 18
mkdir, 18
more, 18
mv, 19

nohup, 35

passwd, 19
printenv, 19
ps, 19
pwd, 35

read, 36
return, 36
rm, 21
rmdir, 21

shift, 36
sort, 22
split, 22
stop, 36
strings, 23
strip, 23
suspend, 36

tail, 23
tar, 21
tee, 24
telnet, 24
test, 36
type, 37

- unalias, 37
- uncompress, 24
- unset, 37
- until, 37
- uptime, 24

- which, 24
- while, 38
- who, 25
- write, 25

- ypchfn, 26
- ypchsh, 26
- yppasswd, 25

- zcat, 26