



*Nous remercions Gilles Goubet de sa contribution  
à la rédaction des annexes de cet ouvrage.*

Annexe A : Les commandes POSIX 2 .....	15-3
Annexe B : bash – Bourne Another SHell .....	15-15
Annexe C : sh - Le shell POSIX.....	15-27
Annexe D : ksh – Le Korn shell .....	15-31
Annexe E : awk – Le processeur de texte .....	15-39
Annexe F : Sed – Editeur en mode flot .....	15-43
Annexe G : Les principaux fichiers d’administration .....	15-45



## Annexe A : Les commandes POSIX 2

**awk** - *Langage qui traite du texte.*

awk [-Fx] [-v affec] program fichier ...  
awk [-Fx] [-v affec] -f prg fichier ...

-Fx            Spécifie le séparateur de champs.  
-v affec       Permet d'affecter des variables, variable=valeur.  
program       Le programme AWK en argument, généralement quoté.  
-f prg         Le programme AWK stocké dans le fichier *prg*.

**basename** - *Extrait le nom d'un fichier à partir d'un chemin.*

basename chemin [suffixe]

suffixe    Le suffixe sera retiré du nom du fichier.

**bc** - *Calculatrice.*

bc [-l] [fichier ...]

-l            Inclut la bibliothèque mathématique.  
fichier      Fichier qui contient des fonctions.

**cat** - *Concatène et affiche le contenu de fichiers texte.*

cat [-u] [fichier ...]

-u            Les caractères sont affichés sans attendre (pas de tampons).

**cd** - *Change de répertoire.*

cd [chemin]

**chgrp** - *Change le groupe d'un fichier.*

chgrp [-R] groupe fichier ...

-R            Agit sur une arborescence de fichiers (action récursive).

**chmod** - *Change les droits d'un fichier.*

chmod [-R] mode fichier ...

-R            Agit sur une arborescence de fichiers (action récursive).  
mode         Indique la modification des droits.

**chown** - *Change le propriétaire d'un fichier.*

chown [-R] util[:groupe] fichier ...

-R            Agit sur une arborescence de fichiers (action récursive).  
util          Le propriétaire du fichier.  
groupe       Le groupe du fichier.

**cksum** - *Calcule un total de contrôle.*

cksum [fichier ...]

**cmp** - *Compare deux fichiers octet par octet.*

cmp [-l | -s] fichier1 fichier2

-l            Affiche toutes les différences rencontrées.  
-s            Mode silencieux, affecte seulement le code retour.

**comm** - *Compare deux fichiers ligne à ligne.*

comm [-123] fichier1 fichier2

- 1 Supprime la colonne 1 (lignes trouvées seulement dans fichier1).
- 2 Supprime la colonne 2 (lignes trouvées seulement dans fichier2).
- 3 Supprime la colonne 3 (lignes communes aux deux fichiers).

**command** - *Exécute une commande.*

command [-p] cmd [arg ...]

- p Utilise la valeur par défaut de la variable PATH, pour rechercher une commande standard.
- cmd Le nom du fichier qui contient la commande à exécuter, surcharge éventuellement le nom d'une commande interne ou d'une fonction.

**cp** - *Copie de fichier.*

cp [-fip] fichier1 fichier2

cp [-R | -r] [-fip] fichier ... rep

- f Force la destruction du fichier cible si ce dernier est protégé en écriture.
- i Demande confirmation de l'éventuel remplacement d'un fichier.
- p Conserve les droits, le propriétaire, le groupe, la date de dernière modification et de dernier accès.
- R Copie d'une arborescence de fichiers.
- r Idem.

**cut** - *Sélectionne des caractères ou des champs.*

cut -b liste [-n] [fichier ...]

cut -c liste [fichier ...]

cut -f liste [-d del [-s]] [fichier ...]

- b liste La liste qui suit contient les octets sélectionnés.
- n N'affiche pas les octets spécifiés (avec l'option -b).
- c liste La liste qui suit contient les caractères sélectionnés.
- f liste La liste qui suit contient les champs sélectionnés.
- d del Spécifie le délimiteur de champs.
- s Supprime les lignes qui ne possèdent pas le séparateur de champs.

**date** - *Affiche la date et l'heure.*

date [-u] [+format]

- u N'utilise pas la variable TZ, et renvoie donc l'heure universelle au lieu de l'heure locale.
- +format Spécifie un format d'affichage. Ex : +"%H".

**dd** - *Sauvegarde physique de périphérique à périphérique.*

dd [opérande ...]

- opérande Opérande de la forme nom=valeur, ex : bs=100k.  
Liste des opérands supportés : if, of, ibs, obs, bs, cbs, skip, seek, count, conv.  
Liste des valeurs de l'opérande conv : block, unblock, lcase, ucase, swab, noerror, notrunc, sync.

**diff** - Compare deux fichiers.

diff [-c | -e | -Cn ] [-br] fichier1 fichier2

- c Affiche trois lignes du contexte pour chaque différence.
- Cn Affiche « n » lignes du contexte pour chaque différence.
- e Produit un fichier de commande pour ed.
- b Ignore les espaces supplémentaires.
- r Exécute la commande récursivement sur les répertoires de même nom, fichier1 et fichier2.

**dirname** - Renvoie le répertoire d'un chemin.

dirname chemin

**echo** - Affiche un message à l'écran.

echo [chaîne ...]

**ed** - Edite un texte.

ed [-p ch] [-s] [fichier]

- p ch Fixe la chaîne d'invite à la valeur ch.
- s Supprime l'affichage du nombre d'octets lors de l'utilisation des commandes d'écriture.

**env** - Affiche l'environnement, exécute une commande avec un environnement spécifique.

env [-i] [nom=valeur ...] [cmd [arg ...]]

- i Utilise l'environnement spécifié pour exécuter la commande **cmd** au lieu d'utiliser l'environnement hérité.

**expr** - Calcule des expressions.

expr opérande ...

Liste des symboles utilisés : (, ), |, &, =, >, >=, <, <=, !=, +, -, \*, /, %, :.

**false** - Renvoie la valeur fausse.

false

**find** - Recherche de fichiers dans une arborescence.

find chemin ... [ opérande ... ]

- name fichier Vrai si le fichier a le nom indiqué.
- perm droits Vrai si les droits correspondent.
- type x Vrai si le type (b,c,d,f) correspond.
- links n Vrai si le fichier a exactement n liens.
- user nom Vrai si le nom de l'utilisateur correspond.
- group nom Vrai si le nom du groupe correspond.
- size n Vrai si le fichier a une taille de n blocs.
- atime n Vrai si le dernier accès remonte à n jours.
- mtime n Vrai si la dernière modification remonte à n jours.
- ctime n Vrai si la date de création remonte à n jours.
- nouser Vrai si le compte utilisateur du fichier n'existe pas.
- nogroup Vrai si le groupe du fichier n'existe pas.
- depth Cherche d'abord dans les sous-répertoires.
- newer f Vrai si le fichier a été modifié plus récemment que le fichier *f*.
- prune Explore les répertoires, la recherche n'est pas arborescente.
- xdev Arrête l'exploration quand on change de périphérique.
- print Affiche le nom du fichier.
- exec cmd {} \; Exécute la commande **cmd**.
- ok cmd {} \; Exécute la commande **cmd** avec demande de confirmation.

**fold** - *Affiche les lignes d'un fichier avec une limite de la longueur de ligne.*

fold [-bs] [ -w larg ] [ fichier ...]

- w larg        Fixe la largeur maximale d'une ligne à larg caractères, par défaut 80.
- b        L'unité de larg est en octet (« byte »).
- s        Si la coupure a lieu au niveau d'un blanc, les blancs suivants sont ignorés.

**getconf** - *Renvoie les valeurs de configuration.*

getconf var  
getconf var chemin

- var        Variable système dont on veut obtenir la valeur.
- chemin    La valeur de la variable dépend du répertoire ; on le spécifie en conséquence.

**getopts** - *Utilitaire d'analyse des options d'une commande.*

getopts options cmd [arg ...]

**grep** - *Recherche de chaînes dans un fichier.*

grep [-E|-F] [-c|-l|-q] [-insvx] [[-e] list | [-f fic]] ... [fichier ...]

- E        Utilise les expressions régulières étendues (anciennement celles de egrep).
- F        Utilise des chaînes de recherche au lieu d'expressions régulières.
- c        Compte le nombre de lignes qui correspondent.
- l        Affiche le nom des fichiers qui contiennent la chaîne.
- q        N'affiche rien, retourne le code 0 si des lignes ont été sélectionnées.
- i        Ignore les différences majuscules, minuscules.
- n        Affiche les numéros et les lignes qui correspondent.
- s        Supprime les messages d'erreur associés aux fichiers non accessibles.
- v        Affiche toutes les lignes qui ne correspondent pas.
- x        Affiche toutes les lignes qui correspondent exactement.
- e list    Utilise une liste d'expressions régulières.
- f fic    Utilise les expressions régulières stockées dans le fichier *fic*.

**head** - *Affiche le début d'un fichier.*

head [-n nb] [fichier ...]

- n nb    Affiche les n premières lignes, par défaut n est égal à 10.

**id** - *Donne l'identité d'un utilisateur.*

id [util]  
id -G [-n] [util]  
id -g [-nr] [util]  
id -u [-nr] [util]

- G        Affiche tous les groupes de l'utilisateur util.
- g        Affiche seulement le groupe effectif.
- n        Affiche les noms de groupes au lieu des valeurs numériques.
- r        Affiche l'UID réel, au lieu de l'UID effectif de l'utilisateur.
- u        Affiche seulement l'UID effectif.

**join** - *Opérateur de jointure de base de données.*

join [-a nb | -v nb] [-e ch] [-o liste] [-t car] [-1 ch] [-2 ch] fichier1 fichiers

- a nb    Produit une ligne pour chaque ligne non appariée du fichier numéro nb.
- v nb    Produit seulement une sortie pour les lignes non appariées du fichier numéro 1.
- e ch    Remplace une sortie vide par la chaîne ch.
- o liste Spécifie les champs présents en sortie.
- t car    Spécifie le délimiteur de champ.
- 1 ch    Spécifie le champ de jointure du fichier 1.
- 2 ch    Spécifie le champ de jointure du fichier 2.

**kill** - *Supprime un processus ou lui envoie un signal.*

kill -s sig pid ...

kill -l [code]

- s sig    Spécifie le nom sig du signal envoyé.
- l        Liste les signaux supportés par l'implémentation.
- pid      Numéro du processus recevant le signal.
- code     Valeur numérique d'un signal, la commande renvoie le nom du signal.

**ln** - *Crée un lien.*

ln [-f] fichier1 fichier2

ln [-f] fichier ... chemin

- f        Force la destruction de la cible, si elle existe.

**locale** - *Affiche les paramètres POSIX locaux.*

locale [-a | -m]

locale [-ck] nom ...

- a        Affiche la liste de tous les paramètres.
- m        Écrit les noms des « charmaps » disponibles.
- c        Affiche les paramètres des catégories spécifiées.
- k        Affiche les valeurs des paramètres spécifiés.

**localedef** - *Définit un environnement de paramètres POSIX local.*

localedef [-c][-f fic] [-i fic] nom

- c        Force la création du résultat.
- f fic    Spécifie le fichier contenant le codage des caractères.
- i fic    Spécifie le fichier contenant les paramètres.

**logger** - *Enregistre un message dans le journal des événements.*

logger chaîne ...

**logname** - *Affiche le nom de l'utilisateur.*

logname

**lp** - *Imprime un fichier.*

lp [-c] [-d dest] [-n nb] [fichier ...]

- c        Fait une copie du fichier avant de le mettre dans la file d'attente d'impression.
- d dest    Envoie l'impression sur dest.
- n nb     Impression en nb copies.

**ls** - *Liste les caractéristiques des fichiers.*

ls [-CFRacdilqrut1] [fichier ...]

- C Affiche les noms sur plusieurs colonnes.
- F Affiche un « / » après chaque répertoire, un « \* » après chaque fichier exécutable et un « | » après chaque tube nommé.
- R Affiche le contenu des arborescences dont on a donné les chemins.
- a Affiche toutes les entrées.
- c Affiche (-l) les dates de création ou trie (-t) par date de création.
- d Affiche les attributs des répertoires, non leur contenu.
- i Affiche les numéros d'inode.
- l Affiche les principaux attributs, dont les droits.
- q Affiche les caractères non imprimables sous forme de « ? ».
- r Inverse l'ordre du tri.
- t Entrées triées, par défaut selon la date de dernière modification.
- u Affiche (-l) les dates de dernier accès ou trie (-t) par date de dernier accès.
- 1 (1=un) Provoque une sortie par ligne.

**mailx** - *Envoie un courrier électronique à un utilisateur.*

mailx [-s sujet] adresse ...

- s sujet Précise le sujet du message.

**mkdir** - *Crée un répertoire.*

mkdir [-p] [-m mode] chemin ...

- p Crée éventuellement les répertoires intermédiaires.
- m mode Fixe les droits des répertoires créés.

**mkfifo** - *Crée un tube nommé.*

mkfifo [-m mode] fichier ...

- m mode Fixe les droits des tubes nommés créés.

**mv** - *Déplace un lien.*

mv [-fi] fichier1 fichier2  
mv [-fi] fichier ... chemin

- f Force la destruction du fichier cible si ce dernier est protégé en écriture.
- i Demande confirmation de l'éventuel remplacement d'un fichier.

**nohup** - *Empêche la mort d'un processus à la déconnexion.*

nohup cmd [arg ...]

**od** - *Affiche le contenu d'un fichier en octal, en hexadécimal.*

od [-v] [-A form] [-j saut] [-N nb] [-t form] [fichier ...]

- v Affiche toutes les données, y compris les données dupliquées.
- A form Spécifie le format des adresses (d,o,x).
- j saut Saute « saut » octets en entrée.
- N nb Traite seulement nb octets.
- t form Spécifie le format des données, notamment l'utilisation de l'octal, de l'hexadécimal, ... (a,c,d,f,o,u,x).



**paste** - Fusionne des fichiers par ligne.

paste [-s] [-d liste] fichier ...

-s Utilise le caractère tab au lieu du séparateur de ligne, comme délimiteur des éléments à fusionner.

-d liste Utilise les caractères de la liste, au lieu du séparateur de ligne, comme délimiteur des éléments à fusionner.

**pathchk** - Vérifie un chemin, en termes d'accessibilité et de portabilité.

pathchk [-p] chemin ...

-p Effectue des tests de portabilité plus approfondis.

**pax** - Sauvegarde un fichier.

(1) pax -w [-dituvX] [-b bs] [-a [-f ar]] [-o opt] ... [-s sub][[-x form] ... [fichier ...]

(2) pax [-cdnv] [-f ar] [-s sub] [modèle ...]

(3) pax -r [-cdiknuv] [-f ar] [-o opt] [-p ch] ... [-s sub]... [modèle ...]

(4) pax -r -w [-dikltuvX] [-p ch] ... [-s sub] ... [fichier ...] chemin

(1) Sauvegarde.

(2) Liste le contenu de l'archive.

(3) Restauration.

(4) Copie d'arborescence.

-d Les répertoires sont sauvegardés ou restaurés.

-i Permet de renommer interactivement les fichiers.

-u Restauration inconditionnelle (-r), sauvegarde inconditionnelle (-w -a).

-v («verbose») Une ligne est affichée pour chaque fichier sauvé/restauré.

-X L'exploration arborescente s'arrête lorsque l'on change de périphérique.

-b bs Spécifie la taille des blocs.

-a Ajout des fichiers en fin d'archive.

-f ar Le fichier *ar* est le support d'archive, par défaut, l'entrées ou la sortie standard.

-o opt Permet de spécifier des options non standard.

-s sub Modifie le nom des fichiers en se basant sur la chaîne sub.

-x form Spécifie le format de l'archive : cpio ou ustar, par défaut ustar.

-c Prend en compte tous les fichiers, exceptés ceux spécifiés par les modèles.

-n Ne prend en compte que le premier fichier correspondant au modèle.

-k Empêche la restauration d'effacer des fichiers.

-p ch Spécifie grâce à la chaîne ch, les caractéristiques conservées ou non des fichiers restaurés.

-l Des liens sont créés si possible.

**pr** - Produit une sortie formatée.

pr [+page] [-col] [-adFmrt] [-e[car][gap]][-h entete] [-i[car][gap]][-l lig][-n[car][larg]] [-o decal] [-s[car]][-w larg] [fichier ...]

+page Numéro de la première page.

-col Nombre de colonnes.

-a Avec l'option précédente, modifie l'ordre de remplissage des colonnes.

-d Sortie avec espacement double.

-F Utilise des « Form Feed » au lieu de « New Line » pour sauter des pages.

-m Produit une sortie multi-colonne, avec une colonne par fichier.

-r Ne produit pas de messages d'erreur si les fichiers ne sont pas accessibles.

-t Supprime l'en-tête.

-e[car][gap] Transforme les tabulations en une suite d'espaces. Les tabulations sont par défaut toutes les huit colonnes, gap remplace la valeur par défaut. Le caractère car précise le caractère de tabulation.

-h entête Spécifie le message d'en-tête.

-i[car][gap] Option inverse de l'option « e », remplace une suite d'espaces par des tabulations.

- l lig Spécifie le nombre de lignes par page, par défaut soixante-six.
- n[car][larg] Les lignes sont numérotées, la numérotation occupe larg caractères et est séparée du texte par une tabulation ou par le caractère car.
- o decal Chaque ligne est précédée de « decal » espaces, par défaut decal vaut 0.
- s car Spécifie le caractère car comme séparateur de colonnes.
- w larg Spécifie la largeur de la sortie dans le cas de l'utilisation du multi-colonne.

**printf** - *Ecrit une sortie formatée.*

printf format [arg ...]

**pwd** - *Affiche le répertoire courant.*

pwd

**read** - *Lit une ligne sur l'entrée standard.*

read [-r] var ...

- r Le caractère \ ne joue pas son rôle d'échappement.
- var Le nom d'une variable du shell, existante ou non.

**rm** - *Détruit un fichier.*

rm [-fiRr] fichier ...

- f Force la destruction des fichiers.
- i Demande confirmation de la destruction de chaque fichier.
- R Destruction récursive des fichiers et des répertoires.
- r Idem.

**rmdir** - *Supprime un répertoire vide.*

rmdir [-p] chemin ...

- p Supprime toutes les composantes du chemin.

**sed** - *Editeur en mode flot.*

sed [-n] script [fichier ...]  
sed [-n] [-e script] ... [-f prg] ... [fichier ...]

- n Supprime la sortie par défaut : normalement chaque ligne non modifiée est recopiée sur la sortie standard.
- script Une instruction sed.
- e script Une instruction sed.
- f prg Le fichier *prg* contient une suite d'instructions sed.

**sh** - *Exécute un shell interactif ou un script.*

sh [-aCefinuvx] [script [ arg ...]]  
sh -c [-aCefinuvx] [cmd [ arg ...]]  
sh -s [-aCefinuvx] [ arg ...]

- a,-C,-e,-f,-i,-n,-u,-v,-x Voir la description de la sous-commande **set** de la commande **sh**.
- script Le script exécuté par le shell.
- c Le shell exécute la commande **cmd**.
- s Lit les commandes à partir de l'entrée standard.

**sleep** - *Temporisation.*

sleep secondes

**sort** - *Tri, fusion de fichiers.*

sort [-m][-o fic][-bdfinru][-t car][-k clef] ... [fichier ...]  
 sort -c [-bdfinru][-t car][-k clef] ... [fichier]

- m On réalise la fusion des fichiers.
- o fic Le résultat du tri est dans le fichier *fic*.
- b Ignore les blancs pour déterminer le début et la fin d'une clef de tri.
- f On ignore la différence majuscule et minuscule.
- i Ignore les caractères non imprimables.
- n Tri numérique et non lexicographique.
- r Tri inverse.
- u Ne produit qu'une ligne pour un ensemble de lignes identiques.
- t car Le caractère car est le séparateur de champs.
- k clef Le critère de tri : début[type][,fin[type]].  
 Le critère de tri commence au champ « début » (les champs sont numérotés à partir de 1).  
 Le reste de la ligne après le champ « fin » n'est plus utilisé comme critère de tri.  
 « type » correspond à une des options b, d, f, i ou n.
- c Ne fait que vérifier si les fichiers sont déjà triés, ne produit pas de sortie, simplement renvoie un code retour.

**stty** - *Configure le terminal.*

stty [-a | -g]  
 stty opérande ...

- a Affiche toutes les caractéristiques du terminal.
- g Affiche les caractéristiques dans le format utilisé pour modifier ces caractéristiques.

**tail** - *Affiche la fin d'un fichier.*

tail [-f] [-c nb] [-n nb] [fichier]

- f Reste en lecture après la fin du fichier.
- n nb Indique le point de départ de la copie, exprimé en caractères.
- +nb Par rapport au début du fichier.
- nb Par rapport à la fin du fichier (par défaut).
- c nb Idem, mais exprimé en octets.

**tee** - *Duplique la sortie standard.*

tee [-ai] [fichier ...]

- a Ajout de la sortie à la fin des fichiers, au lieu de les écraser.
- i Ignore le signal SIGINT.

**test** - *Evalue une expression.*

test [exp]  
 « [ » [exp] « ] »

- b Fichier Fichier spécial par blocs.
- c Fichier Fichier spécial par caractères.
- d Fichier C'est un répertoire.
- e Fichier Vrai si Fichier existe.
- f Fichier Fichier ordinaire.
- g Fichier Bit setgid est activé.
- n Chaîne La longueur de la chaîne est différente de zéro.
- p Fichier C'est un fichier spécial, FIFO ou un tube.
- r Fichier Accessible en lecture.
- s Fichier La taille est supérieure à zéro.
- t [fd] fd est un descripteur associé à un terminal.

-u Fichier	Le bit setuid est activé.
-w Fichier	Accessible en écriture.
-x Fichier	Fichier exécutable.
-z Chaîne	La longueur de Chaîne est égale à zéro.
Chaîne	Vrai si Chaîne n'est pas la chaîne nulle.
Ch1 = Ch2	Vrai si Ch1 correspond à Ch2.
Ch1 != Ch2	Vrai si Ch1 ne correspond pas à Ch2.
E1 -eq E2	E1 est égale à E2 (numériquement parlant).
E1 -ne E2	E1 est différente de E2.
E1 -lt E2	E1 est inférieure à E2.
E1 -gt E2	E1 est supérieure à E2.
E1 -le E2	E1 est inférieure ou égale à E2.
E1 -ge E2	E1 est supérieure ou égale à E2.
(E)	Vrai si E est vraie.
! E	Vrai si E est fausse.
E1 -a E2	Vrai si E1 et E2 sont vraies.
E1 -o E2	Vrai si E1 ou E2 est vraie.

**touch** - *Crée un fichier vide ou met à jour la date de modification d'un fichier.*

touch [-acm] [-r fic | -t tmp] fichier ...

- a Modifie la date de dernier accès du fichier.
- c Ne crée pas le fichier s'il n'existe pas.
- m Modifie la date de dernière modification du fichier.
- r fic Utilise les dates du fichier *fic* pour les options « -a » et « -m ».
- t tmp Spécifie la date tmp pour les options « -a » et « -m ».

**tr** - *Modifie les caractères d'un fichier.*

tr [-cs] chaîne1 chaîne2  
tr -s [-c] chaîne1  
tr -d [-c] chaîne1  
tr -ds [-c] chaîne1 chaîne2

- c Tous les caractères qui ne se trouvent pas dans la première chaîne sont remplacés par le dernier caractère de la deuxième chaîne.
- s Remplace la répétition d'un caractère par un seul.
- d Détruit tous les caractères qui se trouvent dans la chaîne 1.

**true** - *Retourne la valeur vrai.*

true

**tty** - *Affiche le terminal courant.*

tty

**umask** - *Positionne ou affiche les droits par défaut.*

umask [-S] [masque]

- S Utilise un format symbolique pour afficher les droits par défaut.

**uname** - *Affiche le nom et les caractéristiques du système.*

uname [-amnrsv]

- a Affiche toutes les caractéristiques du système.
- m Affiche le type de matériel.
- n Affiche le nom réseau du système.
- r Affiche la « Release » du système d'exploitation.
- s Affiche le nom du système d'exploitation.
- v Affiche la version du système d'exploitation.

**uniq** - *Elimine les doublons d'un fichier trié.*

uniq [-c|-d|-u] [-f nb][-s nb] [fichier1 [fichier2]]

- c Affiche chaque ligne précédée du nombre d'occurrences.
- d Supprime les lignes qui ne sont pas répétées.
- u Supprime les lignes répétées.
- f nb Ignore les nb premiers champs lors des comparaisons.
- s nb Ignore les nb premiers caractères lors des comparaisons.
- fichier1 Le fichier d'entrée, par défaut l'entrée standard.
- fichier2 Le fichier de sortie, par défaut la sortie standard.

**wait** - *Attend la fin des tâches d'arrière-plan.*

wait [pid ...]

**wc** - *Compte les lignes, les mots et les caractères d'un fichier.*

wc [-clw] [fichier ...]

- l Affiche le nombre de lignes.
- w Affiche le nombre de mots.
- c Affiche le nombre de caractères.

**xargs** - *Génère une commande et l'exécute.*

xargs [-t] [-n nb [-x]] [-s taille] [cmd [arg ...]]

- t Active le mode trace.
- n nb Limite le nombre d'arguments de la commande générée.
- x Termine la commande si la valeur nb est atteinte.
- s taille Limite la longueur de la commande générée en termes d'octets.



# Annexe B : bash – Bourne Another SHell

Le shell bash est développé par la « *Free Software Foundation* ». Il est couvert par la licence GNU, et c'est le shell standard du système Linux. Le bash, comme le Korn shell, dérive du shell Bourne dont il tire son nom « Bourne Another SHell ». Le bash est compatible avec le shell POSIX sh et il est d'ailleurs fréquemment lancé sous ce nom. Le bash intègre la majorité des fonctionnalités du Korn shell.

*Principales différences du bash par rapport au Korn shell (éléments manquants)*

- Pas de « tracked aliases ».
- Absence des variables suivantes : ERRNO, FPATH, COLUMNS, LINES, EDITOR, VISUAL.
- Pas de coprocessus (|&, >&p, <&p).
- L'historique des commandes est en mémoire et non dans un fichier.
- Absence des commandes internes suivantes : alias -x, newgrp, print, set -A, whence.

## Fonction

bash - Appelle le Bourne Another Shell.

## Syntaxe

bash [option...] [script...]

- |        |  |
|--------|--|
| script | Le script exécuté par le shell.                  |
| -c cmd | Le shell exécute la commande cmd.                |
| -i     | Shell interactif.                                |
| -s     | Lit les commandes à partir de l'entrée standard. |

*Voir la commande set pour les autres options d'une seule lettre.*

- |                   |   |
|-------------------|---|
| -norc             | Ne lit pas le fichier ~/.bashrc.  |
| -rcfile           | Fichier<br>Spécifie le fichier de remplacement de ~/.bashrc.              |
| -noprofile        | Ne lit pas les fichiers de démarrage.                                     |
| -version          | Affiche la version du shell.  |
| -login            | Exécute le shell comme un shell de connexion.                             |
| -nobraceexpansion | Ne réalise pas l'expansion des accolades ( <i>cf. Expansion des {}</i> ). |
| -nolineediting    | N'utilise pas la bibliothèque GNU pour la lecture de commande.            |
| -posix            | Le shell se conforme au standard POSIX 2.                                 |

## La commande interne set

- a Les variables sont exportées.
- b Les comptes rendus des travaux en arrière-plan sont donnés immédiatement.
- d Désactive la recherche des commandes en mémoire (table de Hash).
- e Sort du shell si la commande se termine en erreur.
- f Désactive la substitution de nom de fichier.
- h Mémorise les fonctions.
- k Met dans l'environnement d'une commande chaque variable affectée dans la ligne de commande.
- m Les travaux en arrière-plan sont exécutés dans un groupe de processus distincts.
- n Lit les commandes sans les exécuter.
- o (*cf. paragraphe suivant*).

- p Mode privilégié.  
Le fichier \$ENV n'est pas exécuté, les fonctions ne sont pas héritées.
- t Sort après l'exécution d'une commande.
- u Traite les paramètres non définis en tant qu'erreurs.
- v Mode verbose.
- x Trace les commandes.
- C Equivalent au drapeau noclobber.
- H Active l'utilisation de l'historique des commandes de style C-Shell (!! , ...).
- P Utilise les répertoires physiques au lieu des liens symboliques correspondants.
- Désactive les options « -x » et « -v ». Signale la fin des options.
- Débute la liste d'arguments, même si le premier argument commence par « + » ou « - ».

*Remplacer le signe - par + désactive l'option.*

## La commande set -o

- allexport Equivalent à -a.
- braceexpand Réalise les substitutions des accolades (par défaut).
- errexit Equivalent à -e.
- emacs Utilise emacs pour l'édition des commandes (par défaut).
- histexpand Equivalent à -H.
- ignoreeof Ignore la fin de fichier.
- interactive-comments Le # débute un commentaire, y compris dans un shell interactif.
- monitor Equivalent à -m.
- noclobber Empêche la redirection > de détruire les fichiers.
- noexec Equivalent à -n.
- noglob Equivalent à -f.
- nohash Equivalent à -f.
- notify Equivalent à -d.
- nounset Equivalent à -u.
- physical Equivalent à -P.
- posix Respect du standard POSIX 2.
- privileged Equivalent à -p.
- verbose Equivalent à -v.
- vi Active l'édition vi des commandes.
- xtrace Equivalent à -x.

## Les commentaires

# ....<New Line>

## Le groupement de commandes

- ! cmd Inverse le code retour.
- cmd1 newline cmd2
- cmd1 ; cmd2
- cmd1 | cmd2
- cmd &
- cmd1 || cmd2
- cmd1 && cmd2
- (Liste)
- { Liste; }



## Les structures de contrôle

```
if Liste ;then Liste [elif Liste ;then Liste] ... [;else Liste] ;fi
for Identificateur [in Mot ...] ;do Liste ;done
while Liste ;do Liste ;done
until Liste ;do Liste ;done
case Mot in [[ ( ] Forme [ | Forme] ... ) Liste ;;] ... esac
select Identificateur [in Mot ...] ;do Liste ;done
```

## Les fonctions

```
[function] Identificateur {Liste ;}
```

## Substitution de répertoires

```
~          Le répertoire de connexion.
~nom       Idem mais de l'utilisateur nom.
~-         Répertoire précédent.
~+         Chemin absolu du répertoire courant.
```

## Les caractères de protection (d'échappement)

```
\          Annule la signification du caractère suivant.
'...'     Annule tous les caractères.
"..."    Annule tous les caractères, sauf ` , \, et $.
```

## Remplacement de noms de fichiers

```
*          Correspond à une chaîne quelconque, même vide.
?          Correspond à un caractère quelconque.
[[domaine]...]    Un des caractères du domaine.
[![domaine]...]   Aucun des caractères du domaine.
```

## Les redirections

```
[n]<mot     Redirection de l'entrée standard.
[n]>mot     Redirection de la sortie standard.
[n]>| mot   Idem, mais ne tient pas compte de noclobber.
[n]>> mot   Redirection de la sortie standard en ajout à mot.
&>mot     Redirige la sortie standard et l'erreur standard (équivalent à >mot 2>&1).
>&mot     Idem.
[n]<> mot   Mot devient l'entrée et la sortie standard.
[n]<<[-] ch L'entrée du shell est lue jusqu'à la ligne correspondant à ch. Si - est
ajouté à <<, les tabulations de début sont supprimées.
[n]<&m     L'entrée standard est dupliquée dans le descripteur m.
[n]>&m     La sortie standard est dupliquée dans le descripteur m.
[n]<&-    L'entrée standard est fermée.
[n]>&-    La sortie standard est fermée.
```

*Dans toutes les redirections qui précèdent, n représente le descripteur du fichier redirigé. Par défaut les redirections portent sur l'entrée ou la sortie standard. Mot représente un fichier ou bien un descripteur dans le cas où mot est numérique.*

## Substitution de commandes

```
`commande` ou $(commande)
```

## Substitution d'accolades

La substitution des accolades permet de générer des chaînes arbitraires. Ce mécanisme est voisin du remplacement des noms de fichiers via les jokers.

```
a{d,c,b}e          est remplacé par          ade ace abe
```

Les accolades peuvent être imbriquées et elles sont interprétées avant les autres caractères spéciaux.

*Exemple complet :* `chown root /usr/{ucb/{ex,edit},lib/{ex?.?*,how_ex}}`

## Les expressions de variables

<code>\$nom</code>	La valeur de la variable.
<code>\${nom}</code>	Idem.
<code>\${#nom}</code>	Le nombre de caractères de la variable.
<code>\${nom:-mot}</code>	Mot si nom est nulle ou renvoie la variable.
<code>\${nom:=mot}</code>	Affecte mot à la variable si elle est nulle et renvoie la variable.
<code>\${nom:?mot}</code>	Affiche mot et réalise un exit si la variable est non définie.
<code>\${nom:+mot}</code>	Mot si non nulle.
<code>\${nom#modèle}</code>	Supprime le petit modèle à gauche.
<code>\${nom##modèle}</code>	Supprime le grand modèle à gauche.
<code>\${nom%modèle}</code>	Supprime le petit modèle à droite.
<code>\${nom%%modèle}</code>	Supprime le grand modèle à droite.

## Les variables internes du shell

<code>\$0</code>	Nom du script.
<code>\$1,\$2, ...</code>	Paramètres du shell
<code>\$*</code>	Tous les paramètres.
<code>\$@</code>	Idem (mais "\$@" eq. à "\$1" "\$2" ...).
<code>\$#</code>	Nombre de paramètres.
<code>\$-</code>	Options du shell.
<code>\$?</code>	Code retour de la dernière commande.
<code>\$\$</code>	PID du shell.
<code>!</code>	PID du dernier processus shell lancé en arrière-plan.
<code>_</code>	Le dernier argument de la commande précédente. Cette variable est également mise dans l'environnement de chaque commande exécutée et elle contient le chemin complet de la commande.

## Les variables - les variables prédéfinies

<code>BASH</code>	Le chemin complet du shell.
<code>BASH_VERSION</code>	La version du shell.
<code>CDPATH</code>	Répertoires explorés par cd.
<code>ENV</code>	(cf. <i>Fichiers de démarrage</i> ).
<code>EUID</code>	EUID de l'utilisateur courant.
<code>FCEDIT</code>	Editeur d'édition de commandes.
<code>IGNORE</code>	Liste de suffixes séparés par « : » indiquant les fichiers qui ne doivent pas apparaître dans l'expansion des noms de fichiers.
<code>HISTCMD</code>	Le numéro de la commande courante dans l'historique.
<code>HISTCONTROL</code>	Si sa valeur est « ignorespace », les lignes commençant par des blancs ne sont pas mises dans l'historique. Si sa valeur est « ingoredups », la dernière ligne n'est pas mise dans l'historique si elle est identique à la ligne précédente. Si sa valeur est « ignoreboth », c'est équivalent aux deux options précédentes.
<code>HISTFILE</code>	Le fichier historique.
<code>HISTFILESIZE</code>	Taille du fichier historique.
<code>HISTSIZ</code>	Nombre de commandes mémorisées par la commande <b>history</b> .
<code>HOSTFILE</code>	Nom d'un fichier au format de /etc/hosts utilisé quand le shell utilise des hostname.
<code>HOSTTYPE</code>	Le type de machine.
<code>HOME</code>	Le répertoire de connexion.
<code>IFS</code>	Séparateur de champs.
<code>IGNOREEOF</code>	Nombre de caractères EOF provoquant la fin du shell.
<code>INPUTRC</code>	(cf. <i>Fichiers</i> ).
<code>LINENO</code>	Ligne courante du script.
<code>MAIL</code>	Fichier contenant le courrier.

MAILCHECK	Fréquence de vérification du courrier.
MAILPATH	Liste des fichiers de courrier.
OLDPWD	Répertoire précédent.
OPTARG	(cf. <i>getopts</i> ).
OPTERR	(cf. <i>getopts</i> ).
OPTIND	(cf. <i>getopts</i> ).
OSTYPE	Chaîne qui décrit le système d'exploitation.
PATH	Chemin de recherche des commandes.
PPID	PID du processus père.
PROMPT_COMMAND	Commande exécutée avant chaque affichage du prompt.
PS1	Invite de commande, par défaut « \$ » (cf. <i>Le prompt</i> ).
PS2	Invite secondaire, par défaut « > ».
PS3	Invite de boucle select, par défaut « #? ».
PS4	Invite de trace, par défaut « + ».
PWD	Répertoire courant.
RANDOM	Nombre aléatoire.
REPLY	Réponse à un select.
SECONDS	Temps écoulé depuis le lancement du shell.
SHLVL	Le nombre d'instance de shell.
TMOUT	Temps maximum d'inactivité.
UID	UID de l'utilisateur courant.
auto_resume	Permet le rappel de job stoppé.
allow_null_glob_expansion	Les fichiers qui ne correspondent à rien dans une expansion de noms de fichiers sont remplacés par une chaîne vide.
cdable_vars	Drapeau qui implique que les arguments de cd qui ne sont pas des répertoires soient interprétés comme des variables contenant un nom de répertoire.
command_oriented_history	Mémorise dans l'historique en une seule ligne une commande entrée sur plusieurs lignes.
glob_dot_filenames	Drapeau qui implique la présence des noms commençant par « . » dans l'expansion des noms de fichiers.
history_control	(cf. <i>HISTCONTROL</i> ).
histchars	Les caractères qui contrôlent l'expansion de l'historique, par défaut « !^# » (cf. <i>Historique</i> ).
hostname_completion_file	(cf. <i>HOSTFILE</i> ).
noclobber	Equivalent à set -C.
no_exit_on_failed_exec	Un script ne se termine pas si une commande <code>exec</code> n'aboutit pas.
nolinks	Equivalent à set -P.
notify	Equivalent à set -b.

## Le prompt

Avant l'affichage du prompt primaire (spécifié par la variable PS1), le shell exécute la commande contenue dans la variable \$PROMPT\_COMMAND.

Dans le cas d'un shell non interactif, la variable PS1 est vide.

La variable PS1 peut contenir les caractères spéciaux suivants :

- \a Active la sonnerie.
- \e Le caractère Escape.
- \t Affiche l'heure au format HH :MM :SS, basé sur 24 heures.
- \T Idem, mais basé sur 12 heures.
- \@ Idem, mais avec indication am/pm.
- \d Affiche la date au format « Jour\_de\_la\_semaine mois jour\_du\_mois ».

\n Saut de ligne.  
\r Retour-chariot.  
\s Affiche le nom du shell.  
\w Affiche le chemin complet du répertoire courant.  
\W Affiche seulement le nom du répertoire courant (basename).  
\u Affiche le nom de l'utilisateur.  
\v Affiche la version du shell.  
\V Idem, mais plus complet.  
\h Affiche le nom réseau de la machine.  
\H Idem, mais sous la forme du nom complet.  
\# Affiche le numéro de la commande.  
\! Affiche le numéro de la commande dans l'historique.  
\nnn  
Affiche un caractère spécifié en octal.  
\s Affiche # si l'EUID est 0 (root), et \$ autrement.  
\ \ Affiche un backslash.  
\[ Début d'une séquence d'échappement.  
\] Termine une séquence d'échappement.

*Exemple :* PS1="[t \W] "

## Les expressions arithmétiques

Les expressions arithmétiques peuvent apparaître dans l'instruction `let` et l'expansion arithmétique.

*L'instruction `let`*  
`let expression ...`

*L'expansion arithmétique*  
`$( expression )`  
`$( (expression) )`

*Opérateurs unaires*  
- moins  
+ plus  
! non  
~ complément à un

*Opérateurs binaires*  
\* multiplication  
/ division entière  
% modulo  
+ addition  
- soustraction  
<< décalage à gauche  
>> décalage à droite  
< inférieur à  
<= inférieur ou égal à  
> supérieur à  
>= supérieur ou égal à  
== égal à  
!= différent  
& et binaire  
^ ou binaire exclusif  
| ou binaire  
&& et logique  
|| ou logique  
= affectation

op= exp = exp op exp  
 \*= /= %= ~= <<= >>= &= ^= |=

*Les constantes commençant par 0x ou 0X sont en hexadécimal. On peut exprimer un nombre dans une base comprise entre 2 et 36 en préfixant la constante par BASE#. Par défaut les constantes sont en base 10.*

## Les tests

[ Expression ] ou test expression

- b Fichier Fichier spécial par blocs.
- c Fichier Fichier spécial par caractères.
- d Fichier C'est un répertoire.
- e Fichier Vrai si Fichier existe.
- f Fichier Fichier ordinaire.
- g Fichier Bit setgid est activé.
- k Fichier Le sticky bit est activé.

Chaîne

- n Chaîne La longueur de la chaîne est différente de zéro.
- p Fichier C'est un fichier spécial, FIFO ou un tube.
- r Fichier Accessible en lecture.
- s Fichier La taille est supérieure à zéro.
- t [fd] fd est un descripteur associé à un terminal.
- u Fichier Le bit setuid est activé.
- w Fichier Accessible en écriture.
- x Fichier Fichier exécutable.
- z Chaîne La longueur de Chaîne est égale à zéro.
- L Fichier Lien symbolique.
- O Fichier Appartient à l'ID utilisateur effectif du processus.
- G Fichier Le groupe auquel il appartient correspond à l'ID de groupe du processus.
- S Fichier C'est une socket.

f1 -nt f2 Fichier f1 est plus récent que f2.  
 f1 -ot f2 Fichier f1 est moins récent que f2.  
 f1 -ef f2 Fichiers f1 et f2 font référence au même fichier.

Chaîne = Forme Vrai si Chaîne correspond à Forme.  
 Chaîne != Forme Vrai si Chaîne ne correspond pas à Forme.

E1 -eq E2 E1 est égale à E2.  
 E1 -ne E2 E1 est différente de E2.  
 E1 -lt E2 E1 est inférieure à E2.  
 E1 -gt E2 E1 est supérieure à E2.  
 E1 -le E2 E1 est inférieure ou égale à E2.  
 E1 -ge E2 E1 est supérieure ou égale à E2.

### *Expression composée*

(E) Vrai si E est vraie.  
 ! E Vrai si E est fausse.  
 E1 -a E2 Vrai si E1 et E2 sont vraies.  
 E1 -o E2 Vrai si E1 ou E2 est vraie.

## getopts

getopts Chaîne\_options Nom [Argument ...]

Vérifie les options d'un argument. Si aucun argument n'est spécifié, le paramètre Chaîne\_options contient les lettres reconnues par la commande **getopts**. Si une lettre est suivie d'un signe deux-points (:), l'option doit comporter un argument. Un espace peut séparer l'option et l'argument. La commande `getopts` place la lettre correspondant à l'option suivante dans la variable Nom lorsque le signe + précède l'argument. L'index de l'argument suivant est enregistré dans OPTIND. L'argument d'option, s'il existe, est placé dans OPTARG. Si un signe deux-points (:) est placé au

début du paramètre Chaîne\_options, la commande `getopts` enregistre la lettre correspondant à une option incorrecte dans `OPTARG`, et attribue la valeur `?` à la variable `Nom` d'une option inconnue, ou la valeur `:` lorsqu'une option requise n'est pas spécifiée. L'état de sortie est différent de zéro lorsqu'il n'y a plus d'option.

## Les commandes internes

`:` [Arg ...] Etend les arguments, retourne vrai.  
`.` Fichier [Arg...]  
`source` Fichier [Arg...] Le shell courant exécute le fichier.  
`alias` [Nom[=valeur]...] Crée ou liste les alias.  
`bg` [Travail ...] Place les travaux en arrière-plan.  
`break` [n] Sort de n niveaux de boucle.  
`builtin` [Arg...] Exécute la commande interne au lieu d'une fonction qui porte le même nom.  
`bind`  
`bind` [-m KEYMAP] [-lvd] [-q NAME]  
`bind` [-m KEYMAP] -f FILENAME  
`bind` [-m KEYMAP] KEYSEQ:FUNCTION-NAME Affiche les associations courantes ou réalise des associations.  
`-m KEYMAP` Utilise KEYMAP pour les associations. Les valeurs possibles sont `emacs`, `emacs-standard`, `emacs-meta`, `emacs-ctlx`, `vi`, `vi-move`, `vi-command`, `vi-insert`.  
`-l` Liste le nom des fonctions readline.  
`-v` Liste le nom des fonctions et leur association.  
`-d` Sortie des associations pour une relecture ultérieure.  
`-f FILENAME` Lit les associations à partir d'un fichier.  
`-q NAME` Spécifie les touches associées à NAME.  
`command` [-pVv] cmd [arg ...] Exécute la commande `cmd`, surcharge éventuellement le nom d'une commande interne ou d'une fonction.  
`-p` Utilise la valeur par défaut de la variable `PATH`, pour rechercher une commande standard.  
`-v,-V` Affiche des informations sur la commande.  
`continue` [n] Passe à l'itération suivante, n indique le nombre d'imbrications.  
`declare` [-frxi] [NAME[=VALUE]]  
`typeset` [-frxi] [NAME[=VALUE]] Déclare des variables ou leur donne des attributs.  
`-f` Les noms correspondent à des fonctions.  
`-r` Les variables sont en lecture seule.  
`-x` Les variables sont exportées.  
`-i` Les variables correspondent à des entiers.  
`dirs` [-l] [+/-n] Affiche la liste des répertoires mémorisés.  
`-l` Produit un listing commenté.  
`-n` Liste les n premiers répertoires.  
`+n` Liste les n derniers répertoires.  
`echo` [-neE] [Arg ...] Affiche les arguments.  
`-n` Supprime le saut de ligne final.  
`-e` Active l'interprétation des séquences d'échappement.  
`-E` Désactive l'interprétation des séquences d'échappement.  
`\a` alert (bell)  
`\b` backspace  
`\c` Supprime le saut de ligne final  
`\f` form feed  
`\n` new line  
`\r` carriage return

	\t	horizontal tab
	\v	vertical tab
	\\	backslash
	\nnn	Code ASCII en octal
enable [-n] [-all] [name ...]		Autorise ou non les scripts équivalant aux commandes internes.
-n		Interdit (par défaut autorise).
-all		Agit sur l'ensemble des commandes internes.
eval [Arg ...]		Les arguments sont lus comme entrée du shell et ensuite exécutés.
exec [[-] Commande [Arg ...]]		La commande donnée en argument se substitue au shell. Si un moins est présent, l'argument 0 est précédé de moins.
exit [n]		Provoque la sortie du shell avec l'état de sortie n.
export [Nom[=Valeur]] ...		
export -p		Les noms spécifiés sont exportés, l'option « -p » permet de lister les variables exportées.
fc [-e Nom_éditeur] [-nlr] [Première [Dernière]]		
fc -s [pat=rep] [cmd]		Liste les commandes ou les édite.
fg [travail ...]		Place le travail spécifié au premier plan.
getopts Chaîne_options Nom [Arg ...]		Extrait les options ( <i>cf. Paragraphe getopts</i> ).
hash [-r] [Nom ...]		Mémorise le chemin complet des commandes spécifiées.
-r		Retire les noms spécifiés de la mémoire (table de Hash).
help [Modèle]		Affiche une aide concernant les commandes internes.
history [n]		
history -rwan [Fichier]		Affiche l'historique des commandes. Avec l'argument n, affiche les n dernières commandes.
-a		Ajoute l'historique courant au fichier historique.
-n		Le fichier historique s'ajoute à l'historique courant.
-r		Lecture du fichier historique, qui devient l'historique courant.
-w		Ecriture de l'historique courant dans le fichier historique.
jobs [-lnp] [travail ...]		
jobs -x command [ args ... ]		Liste les travaux spécifiés.
-l		Liste également le PID des travaux.
-n		Affiche seulement les travaux dont l'état a changé.
-p		Liste le PID du leader.
-x		Exécute la commande, un travail indique le groupe de processus .
kill -l [Signal]		Liste des noms des signaux.
kill [-s Signal   -Signal] Travail ...		Transmet un signal (par défaut TERM) aux travaux spécifiés. <i>Désigne un travail :</i>
	PID	
	%Numéro_Travail	
	%Chaîne	Travail qui commence par Chaîne.
	%?Chaîne	Travail qui contient Chaîne.
	%%	Travail en cours.
	%+	Equivalent à %%.
	%-	Travail précédent.
let expression ...		Evalue les expressions ( <i>cf. Expressions arithmétiques</i> ).
local Nom[=Valeur]		Crée une variable locale (obligatoirement dans une fonction).
logout		Met fin à un shell de connexion.
popd [+/-n]		Retire des entrées de la pile des répertoires ( <i>cf. pushd</i> ).
pushd [Répertoire]		

pushd +/-n	Dans la première forme, le répertoire est mis au sommet de la pile et devient le répertoire courant. Les options « + » ou « - » un nombre permettent de faire une rotation de la pile.
pwd	Affiche le répertoire courant.
read [-r] [Nom ...]	Lit des variables sur l'entrée standard.
-r	Les backslash (\) ne sont pas ignorés.
readonly [Nom[=Valeur]] ...	
readonly -p	Les noms sont marqués en lecture seule, l'option « -p » liste ces variables.
return [n]	Retour d'une fonction shell, avec le code n.
set	(cf. <i>Paragraphe set</i> ).
shift [n]	Décale les paramètres.
suspend [-f]	Suspend l'exécution du shell jusqu'à la réception du signal SIGCONT.
-f	Force la suspension même si le shell est le shell de connexion.
test expression	
[ expression ]	Evalue une expression (cf. <i>Paragraphe test</i> ).
times	Affiche les temps cumulés CPU utilisateur et système du shell et des processus lancés par le shell.
trap [-l] [[Commande] [Signal ...]]	Lit et exécute la commande spécifiée lorsque le shell reçoit le ou les signaux spécifiés.
-l	Liste les noms des signaux et le numéro correspondant.
type [-all] [-type   -path ] [Nom ...]	Identifie une commande.
-all	Affiche l'ensemble des références du nom.
-type	Affiche la nature de la commande : alias, builtin, file, keyword ...
-path	Si la commande est un fichier, affiche son chemin.
ulimit [-acdmstfpnuvSH] [Limite]	Gère les ressources des processus.
-S	Spécifie la limite « soft ».
-H	Spécifie la limite « hard » (par défaut).
-a	Affiche l'ensemble des limites.
-c	La taille maximale d'un core.
-d	La taille maximale de la zone de données.
-m	La taille maximale de la zone résidente.
-s	La taille maximale de la pile.
-t	La durée maximale d'exécution (temps CPU) en secondes.
-f	La taille maximale d'un fichier créée par un processus.
-p	La taille des tubes.
-n	Le nombre maximum de fichiers ouverts.
-u	Le nombre maximum de processus simultanés.
-v	La taille maximale de l'espace virtuel.
umask [-S] [Masque]	Spécifie le masque, ou l'affiche.
-S	Utilise la notation symbolique.
unalias [-a] [Nom ... ]	Supprime les alias spécifiés.
-a	Supprime l'ensemble des alias.
unset [-fv] Nom ...	
-v	Les variables spécifiées sont supprimées (par défaut).
-f	Les fonctions sont supprimées.
wait [n]	Attend la terminaison de l'ensemble des processus lancés en arrière-plan. Il est possible de préciser un processus.



## Historique de commandes, style C-Shell

!!	Référence la dernière commande.
!23	Référence la commande n°23 (cf. <i>La commande interne history</i> ).
!-2	L'avant-dernière commande.
!ls	La dernière commande commençant par ls.
! ?profile	La dernière commande contenant la chaîne profile.
!#	La commande entière.
^1993^1999^	Substitution d'une chaîne par une autre dans la dernière commande.
! !:s/99/93/	Substitution d'une chaîne par une autre, technique générale.

## Historique de commandes, mode emacs (mode par défaut)

*Principales commandes (cf. manuel pour compléments)*

CTRL-P (↑)	Rappelle la commande précédente.
CTRL-N (↓)	Rappelle la commande suivante.
CTRL-B (←)	Déplace le curseur d'un caractère vers la gauche.
CTRL-F (→)	Déplace le curseur d'un caractère vers la droite.
Backspace	Détruit le caractère à gauche du curseur.
CTRL-D (Suppr)	Détruit le caractère sous le curseur.
Frappe de caractères	Les caractères sont insérés à partir du curseur.
CTRL-A (Début)	Déplace le curseur en début de ligne.
CTRL-E (Fin)	Déplace le curseur en fin de ligne.
CTRL-K	Supprime la fin de la ligne.
CTRL-X	Supprime le début de la ligne.
CTRL-V TAB	Insère une tabulation.

## Historique de commandes, mode vi

Le mode vi est activé par la commande **set -o vi**.

On utilise la touche Echappement pour rentrer dans le mode historique.

Les commandes **vi** d'édition peuvent alors être utilisées pour le rappel et l'édition de commandes (h,j,k,l,o,\$,a...<ESC>,i...<ESC>,x,r,/ls/,23G,etc.).

## La complétion

On peut compléter un nom de fichier en utilisant le caractère Tabulation. Ce caractère peut également être utilisé pour proposer un choix lors de la saisie d'une variable (texte commençant par \$), d'un utilisateur (texte commençant par ~), d'un nom d'ordinateur (texte commençant par @) ou d'une commande.

## Les fichiers de démarrage

Pour un shell de connexion (l'option « -noprofile » n'étant pas utilisée) :

- A la connexion

- 1) /etc/profile
- 2) ~/.bash\_profile s'il existe  
~/.bash\_login sinon  
~/.profile si aucun des précédents n'existe
- 3) ~/.bashrc s'il est activé par l'un des scripts précédents
- 4) /etc/bashrc activé par ~/.bashrc

- A la déconnexion

~/.bash\_logout

Pour un shell interactif qui n'est pas de connexion (sans les options « -norc » ou « -rcfile ») :

~/.bashrc

Pour un script (shell non interactif)

- 1) \$BASH\_ENV
- 2) \$ENV

- Pour un shell invoqué sous le nom sh
- Pour un shell de connexion (sans l'option `-noprofile`)
    - 1) `/etc/profile`
    - 2) `~/.profile`
  - Autre shell : aucun fichier de démarrage

Pour un shell invoqué avec l'option « `-posix` », un seul fichier est exécuté :  
`$ENV`

## Les fichiers

- `/bin/bash` L'exécutable
- `/etc/profile`
- `/etc/bashrc`
- `~/.bash_profile`
- `~/.bash_login`
- `~/.profile`
- `~/.bashrc`
- `$BASH_ENV`
- `$ENV` Fichiers de démarrage (*cf. Les fichiers de démarrage*).
- `~/.bash` Fichier de clôture.
- `~/.bash_logout` Script exécuté automatiquement à la déconnexion.
- `$INPUTRC`
- `~/.inputrc` Fichiers contenant la définition des touches d'édition de commande du mode emacs.
- `~/.bash_history`  
Fichier historique des commandes. C'est la valeur par défaut de la variable `HISTFILE`. Si cette variable est détruite, le shell ne sauvegarde pas l'historique des commandes quand il se termine.

# Annexe C : sh - Le shell POSIX

## Fonction

sh - Appelle le shell standard (shell POSIX).

## Syntaxe

```
sh [-aCefnuvx] [script [ arg ...]]
sh -c [-aCefnuvx] [cmd [ arg ...]]
sh -s [-aCefnuvx] [ arg ...]
```

-a,-C,-e,-f,-n,-u,-v,-x

Voir la description de la commande interne set.

script

Le script exécuté par le shell.

-c

Le shell exécute la commande cmd.

-i

Shell interactif.

-s

Lit les commandes à partir de l'entrée standard.

## La commande interne set

```
set [-aCefnuvx] [arg ...]
set [+aCefnuvx] [arg ...]
set -- [arg ...]
```

-a Les variables sont exportées.

-C Empêche la redirection > de détruire les fichiers.

-e Sort du shell si la commande se termine en erreur.

-f Désactive la substitution de nom de fichier.

-n Lit les commandes sans les exécuter.

-u Traite les paramètres non définis en tant qu'erreurs.

-v Mode verbose.

-x Trace les commandes.

-- Débute la liste d'arguments, même si le premier argument commence par « + » ou « - ».

*Remplacer le signe - par + désactive l'option.*

## Le groupement de commandes

! cmd Inverse le code retour.

cmd1 newline cmd2

cmd1 ; cmd2

cmd1 | cmd2

cmd &

cmd1 || cmd2

cmd1 && cmd2

(Liste)

{ Liste; }

## Les structures de contrôle

if Liste ;then Liste [elif Liste ;then Liste] ... [else Liste] ;fi

for Identificateur [in Mot ...] ;do Liste ;done

while Liste ;do Liste ;done

until Liste ;do Liste ;done

case Mot in [( ) Forme [ | Forme] ... ) Liste ;;] ... esac

## Les commentaires

# ....<New Line>

## Les fonctions

Identificateur () {Liste ;}

## Substitution de répertoires

~ Le répertoire de connexion.  
~jean Le répertoire de connexion de jean.

## Les caractères de protection (d'échappement)

\ Annule la signification du caractère suivant.  
'...' Annule tous les caractères.  
"..." Annule tous les caractères, sauf ` , \, et \$.

## Remplacement de noms de fichiers

\* Correspond à une chaîne quelconque, même vide.  
? Correspond à un caractère quelconque.  
[[domaine]...] Un des caractères du domaine.  
![domaine]... Aucun des caractères du domaine.

## Substitution de commandes

`commande` ou \$(commande)

## Les redirections

[n]<mot Redirection de l'entrée standard.  
[n]>mot Redirection de la sortie standard.  
[n]>| mot Idem, mais ne tient pas compte de noclobber.  
[n]>> mot Redirection de la sortie standard en ajout à mot.  
[n]<> mot Mot devient l'entrée et la sortie standard.  
[n]<<[-] ch L'entrée du shell est lue jusqu'à la ligne correspondant à ch. Si - est ajouté à <<, les tabulations de début sont supprimées.  
[n]<&m L'entrée standard est dupliquée dans le descripteur m.  
[n]>&m La sortie standard est dupliquée dans le descripteur m.  
[n]<&- L'entrée standard est fermée.  
[n]>&- La sortie standard est fermée.

*Dans toutes les redirections qui précèdent, n représente le descripteur du fichier redirigé. Par défaut les redirections portent sur l'entrée ou la sortie standard. Mot représente un fichier ou bien un descripteur dans le cas où mot est numérique.*

## Les expressions de variables

\$nom La valeur de la variable.  
\${nom} Idem.  
\${nom[:]-mot} Mot si nulle ou non définie.  
\${nom[:]=mot} Affecte mot si nulle ou non définie.  
\${nom[:]?mot} Affiche mot et exit si non définie.  
\${nom[:]+mot} Mot si non nulle.  
\${#nom} La longueur de la variable.  
\${nom#modèle} Supprime le petit modèle à gauche.  
\${nom##modèle} Supprime le grand modèle à gauche.  
\${nom%modèle} Supprime le petit modèle à droite.  
\${nom%%modèle} Supprime le grand modèle à droite.

## Les variables internes du shell

\$0	Nom du script.
\$1-\$9	Paramètres du shell (1 à 9).
\$*	Tous les paramètres.
\$@	Idem (mais "\$@" eq. à "\$1" "\$2" ...).
\$#	Nombre de paramètres.
\$-	Options du shell.
\$?	Code retour de la dernière commande.
\$\$	PID du shell.
#!	PID du dernier processus shell lancé en arrière-plan.

## Les variables - les variables prédéfinies

HOME	Le répertoire de connexion.
LANG	Spécifie la langue, et donc spécifie les valeurs par défaut des variables LC_...
LC_ALL	Interagit avec les autres variables LC ( <i>cf. le manuel</i> ).
LC_COLLATE	Interagit dans les comparaisons de chaînes ( <i>cf. le manuel</i> ).
LC_CTYPE	Détermine l'interprétation d'une suite d'octets comme caractères.
LC_MESSAGES	Indique la langue utilisée pour les messages.
IFS	Séparateur de champs.
PATH	Chemin de recherche des commandes.

## Les expressions arithmétiques

\$((expression))

### *Opérateurs unaires*

- moins  
! non  
~ complément à un

### *Opérateurs binaires*

\* multiplication  
/ division entière  
% modulo  
+ addition  
- soustraction  
<< décalage à gauche  
>> décalage à droite  
< inférieur à  
<= inférieur ou égal à  
> supérieur à  
>= supérieur ou égal à  
== égal à  
!= différent  
& et binaire  
^ ou binaire exclusif  
| ou binaire  
&& et logique  
|| ou logique  
= affectation  
op= exp = exp op exp  
\*= /= %= ~= <<= >>= &= ^= |=

## Les commandes internes

:	[Arg ...]	Etend les arguments, retourne vrai.
.	Fichier [Arg...]	Le shell courant exécute le fichier.
break	[n]	Sort de n niveaux de boucle.
continue	[n]	Passe à l'itération suivante, n indique le nombre d'imbrications.
eval	[Arg ...]	Les arguments sont lus comme entrée du shell et ensuite exécutés.
exec	[Arg ...]	La commande donnée en argument se substitue au shell.
exit	[n]	Provoque la sortie du shell avec l'état de sortie n.
export	[Nom[=Valeur]] ...	
export	-p	Les noms spécifiés sont exportés, l'option « -p » permet de lister les variables exportées.
readonly	[Nom[=Valeur]] ...	
readonly	-p	Les noms sont marqués en lecture seule, l'option « -p » liste ces variables.
return	[n]	Retour d'une fonction shell, avec le code n.
set		(cf. début du chapitre)
shift	[n]	Décale les paramètres.
trap	[[Commande] [Signal ...]]	
		Lit et exécute la commande spécifiée lorsque le shell reçoit le ou les signaux spécifiés.
unset	[-fv] Nom ...	
	-v	Les variables spécifiées sont supprimées (par défaut).
	-f	Les fonctions sont supprimées.

# Annexe D : ksh – Le Korn shell

## Fonction

ksh - Appelle le Korn shell.

## Syntaxe

ksh [-i] [+|-aefhkntuvx] [-o Option][-c Chaîne | -s | Fichier [Paramètre]]

## Options

-i	Shell interactif.
-c cmd	Exécute la commande <b>cmd</b> .
-p	Permet aux IDs réel et effectif d'être différents.
-r	Shell restreint.
-s	Commandes lues à partir de l'entrée standard.

*Pour les autres options, voir la commande **set**.*

## La commande interne set

-a	Les variables sont exportées.
-e	Sort du shell si la commande se termine en erreur.
-f	Désactive la substitution de nom de fichier.
-h	Mémorise les fonctions.
-k	Met dans l'environnement chaque variable affectée.
-m	Les travaux en arrière-plan sont exécutés dans un groupe de processus distincts.
-n	Lit les commandes sans les exécuter.
-o	( <i>cf. paragraphe suivant</i> ).
-t	Sort après l'exécution d'une commande.
-u	Traite les paramètres non définis en tant qu'erreurs.
-v	Mode verbose.
-x	Trace les commandes.
-	Désactive les options « -x » et « -v ».
--	Débute la liste d'arguments, même si le premier argument commence par « + » ou « - ».

*Remplacer le signe - par + désactive l'option.*

## La commande set -o

allexport	Equivalent à -a.
errexit	Equivalent à -e.
bgnice	Faible priorité pour les travaux en arrière-plan.
emacs	Utilise emacs pour l'édition des commandes.
gmacs	Utilise gmacs pour l'édition des commandes.
ignoreeof	Ignore la fin de fichier.
keyword	Equivalent à -k.
markdirs	Ajoute « / » à tous les noms de répertoires générés.
monitor	Equivalent à -m.
noclobber	Empêche la redirection > de détruire les fichiers.
noexec	Equivalent à -n.
noglob	Equivalent à -f.
nolog	Ne pas sauvegarder les fonctions dans l'historique.
nounset	Equivalent à -u.
privileged	Equivalent à -p.
verbose	Equivalent à -v.
trackall	Equivalent à -h.
vi	Active l'édition vi des commandes.
viraw	Traite chaque caractère tel qu'il est entré.

xtrace       Equivalent à -x.

## Le groupement de commandes

cmd1 newline cmd2  
cmd1 ; cmd2  
cmd1 | cmd2  
cmd &  
cmd |&  
cmd1 || cmd2  
cmd1 && cmd2  
(Liste)  
{ Liste; }

## Les structures de contrôle

if Liste ;then Liste [elif Liste ;then Liste] ... [;else Liste] ;fi  
for Identificateur [in Mot ...] ;do Liste ;done  
while Liste ;do Liste ;done  
until Liste ;do Liste ;done  
case Mot in [[ ( ] Forme [ | Forme] ... ) Liste ;;] ... esac  
select Identificateur [in Mot ...] ;do Liste ;done

## Les fonctions

function Identificateur {Liste ;}  
Identificateur () {Liste ;}

## Substitution de répertoires

~            Le répertoire de connexion.  
~nom        Idem mais de l'utilisateur nom.  
~+          Répertoire précédent.  
~+          Chemin absolu du répertoire courant.

## Les caractères de protection (d'échappement)

\           Annule la signification du caractère suivant.  
'...'  
"..."  
Annule tous les caractères.  
Annule tous les caractères, sauf ` , \, et \$.

## Remplacement de noms de fichiers

\*           Correspond à une chaîne quelconque, même vide.  
?           Correspond à un caractère quelconque.  
[[domaine]...]  
[![domaine]...]  
?(modèle[ |modèle]...)  
\*(modèle[ |modèle]...)  
+(modèle[ |modèle]...)  
!(modèle[ |modèle]...)  
@(modèle[ |modèle]...)  
Un des caractères du domaine.  
Aucun des caractères du domaine.  
Correspond à n'importe quel modèle.  
Correspond à zéro ou plusieurs occurrences.  
Correspond à une ou plusieurs occurrences.  
Correspond à aucun modèle.  
Correspond exactement à un modèle.

## Substitution de commandes

`commande` ou \$(commande)



## Les redirections

<Fichier	Redirection de l'entrée standard.
>Fichier	Redirection de la sortie standard.
>  Fichier	Idem, mais ne tient pas compte de noclobber.
>> Fichier	Redirection de la sortie standard en ajout au fichier.
<> Fichier	Le fichier devient l'entrée et la sortie standard.
<<[-] Mot	L'entrée du shell est lue jusqu'à la ligne correspondant à Mot. Si - est ajouté à <<, les tabulations de début sont supprimées.
<&n	L'entrée standard est dupliquée dans le descripteur n.
>&n	La sortie standard est dupliquée dans le descripteur n.
<&-	L'entrée standard est fermée.
>&-	La sortie standard est fermée.
<&p	L'entrée du coprocesseur est déplacée vers l'entrée standard.
>&p	La sortie du coprocesseur est déplacée vers la sortie standard.
n>&m	Duplique le descripteur n dans m.

## Les expressions de variables

\$nom	La valeur de la variable.
\${nom}	Idem.
\${#nom[*]}	Le nombre d'éléments du tableau.
\${nom[*]}	Tous les éléments du tableau.
\${nom[0]}	Le premier élément du tableau.
\${nom[:]-mot}	Mot si nulle ou non définie.
\${nom[:]=mot}	Affecte mot si nulle ou non définie.
\${nom[:]?mot}	Affiche mot et exit si non définie.
\${nom[:]+mot}	Mot si non nulle.
\${nom#modèle}	Supprime le petit modèle à gauche.
\${nom##modèle}	Supprime le grand modèle à gauche.
\${nom%modèle}	Supprime le petit modèle à droite.
\${nom%%modèle}	Supprime le grand modèle à droite.

## Les variables internes du shell

\$0	Nom du script.
\$1-\$9	Paramètres du shell (1 à 9).
\$*	Tous les paramètres.
\$@	Idem (mais "\$@" eq. à "\$1" "\$2" ...).
\$#	Nombre de paramètres.
\$-	Options du shell.
\$?	Code retour de la dernière commande.
\$\$	PID du shell.
\$_	PID du dernier processus shell lancé en arrière-plan.

## Les variables - les variables prédéfinies

ERRNO	Code erreur du dernier appel système.
LINENO	Ligne courante du script.
OLDPWD	Répertoire précédent.
OPTARG	(cf. <i>getopts</i> )
OPTIND	(cf. <i>getopts</i> )
PPID	PID du processus père.
PWD	Répertoire courant.
RANDOM	Nombre aléatoire.
REPLY	Réponse à un select.
SECONDS	Temps écoulé depuis le lancement du shell.
CDPATH	Répertoires explorés par cd.
COLUMNS	Largeur de l'écran.
EDITOR	Editeur préféré.
ENV	Le script d'initialisation.
FCEDIT	Editeur d'édition de commandes.

FPATH	Répertoires des fonctions.
IFS	Séparateur de champs.
HISTFILE	Le fichier historique.
HISTSIZE	Taille du fichier historique.
HOME	Le répertoire de connexion.
LINES	Utilisé par select.
MAIL	Fichier contenant le courrier.
MAILCHECK	Fréquence de vérification du courrier.
MAILPATH	Liste des fichiers de courrier.
PATH	Chemin de recherche des commandes.
PS1	Invite de commande, par défaut « \$ ».
PS2	Invite secondaire, par défaut « > ».
PS3	Invite de boucle select, par défaut « #? ».
PS4	Invite de trace, par défaut « + ».
SHELL	Le shell de préférence.
TMOUT	Temps maximum d'inactivité.
VISUAL	L'éditeur de commandes.

### Alias prédéfinis

```
autoload='typeset -fu'  
false='let 0'  
functions='typeset -f'  
hash='alias -t'  
history='fc -l'  
integer='typeset -i'  
nohup='nohup '  
r='fc -e -'  
true='.'  
type='whence -v'
```

### Les tests

```
[[Expression]]  
-a Fichier Vrai si Fichier existe.  
-b Fichier Fichier spécial par blocs.  
-c Fichier Fichier spécial par caractères.  
-d Fichier C'est un répertoire.  
-f Fichier Fichier ordinaire.  
-g Fichier Bit setgid est activé.  
-k Fichier Le sticky bit est activé.  
-n Chaîne La longueur de la chaîne est différente de zéro.  
-o Option L'option est active.  
-p Fichier C'est un fichier spécial, FIFO ou un tube.  
-r Fichier Accessible en lecture.  
-s Fichier La taille est supérieure à zéro.  
-t [fd] fd est un descripteur associé à un terminal.  
-u Fichier Le bit setuid est activé.  
-w Fichier Accessible en écriture.  
-x Fichier Fichier exécutable.  
-z Chaîne La longueur de Chaîne est égale à zéro.  
-L Fichier Lien symbolique.  
-O Fichier Appartient à l'ID utilisateur effectif du processus.  
-G Fichier Le groupe auquel il appartient correspond à l'ID de groupe du processus.  
-S Fichier C'est une socket.  
f1 -nt f2 Fichier f1 est plus récent que f2.  
f1 -ot f2 Fichier f1 est moins récent que f2.  
f1 -ef f2 Fichiers f1 et f2 font référence au même fichier.  
Chaîne = Forme Vrai si Chaîne correspond à Forme.  
Chaîne != Forme Vrai si Chaîne ne correspond pas à Forme.
```

Chaîne1 < Chaîne2 Vrai si Chaîne1 précède Chaîne2.  
 Chaîne1 > Chaîne2 Vrai si Chaîne1 suit Chaîne2.  
 E1 -eq E2 E1 est égale à E2.  
 E1 -ne E2 E1 est différente de E2.  
 E1 -lt E2 E1 est inférieure à E2.  
 E1 -gt E2 E1 est supérieure à E2.  
 E1 -le E2 E1 est inférieure ou égale à E2.  
 E1 -ge E2 E1 est supérieure ou égale à E2.

*Expression composée*

(E) Vrai si E est vraie.  
 ! E Vrai si E est fausse.  
 E1 && E2 Vrai si E1 et E2 sont vraies.  
 E1 || E2 Vrai si E1 ou E2 est vraie.

**Les expressions arithmétiques**

expression  
 terme  
 (expression)  
 exp\_avec\_opérateur\_unaire  
 exp\_avec\_opérateur\_binaire

*Terme*

variable\_entière  
 [base#]constant

*Opérateurs unaires*

- moins  
 ! non  
 ~ complément à un

*Opérateurs binaires*

\* multiplication  
 / division entière  
 % modulo  
 + addition  
 - soustraction  
 << décalage à gauche  
 >> décalage à droite  
 < inférieur à  
 <= inférieur ou égal à  
 > supérieur à  
 >= supérieur ou égal à  
 == égal à  
 != différent  
 & et binaire  
 ^ ou binaire exclusif  
 | ou binaire  
 && et logique  
 || ou logique  
 = affectation  
 op= exp = exp op exp  
 \*= /= %= ~= <<= >>= &= ^= |=

## Les commandes internes

: [Arg ...]	Etend les arguments, retourne vrai.
. Fichier [Arg...]	Le shell courant exécute le fichier.
alias [-tx] Nom[=Valeur] ...	Crée ou liste les alias. -x Exporte l'alias. -t Crée un alias « repéré » par le chemin absolu.
bg [Travail ...]	Place les travaux en arrière-plan.
break [n]	Sort de n niveaux de boucle.
cd [Arg]	Change de répertoire.
continue [n]	Passes à l'itération suivante, n indique le nombre d'imbrications.
echo [Arg ...]	(cf. La commande <i>echo</i> )
eval [Arg ...]	Les arguments sont lus comme entrée du shell et ensuite exécutés.
exec [Arg ...]	La commande donnée en argument se substitue au shell.
exit [n]	Provoque la sortie du shell avec l'état de sortie n.
export [Nom[=Valeur]] ...	Les noms spécifiés sont exportés.
fc [-e Nom_éditeur] [-nlr] [Première [Dernière]]	Liste les commandes ou les édite.
fg [travail ...]	Place le travail spécifié au premier plan.
getopts Chaîne_options Nom [Arg ...]	Extrait les options (cf. paragraphe suivant).
jobs [-lnp] [travail ...]	Liste les travaux spécifiés.
kill -l	Liste des signaux.
kill [-Signal] Travail ...	Transmet un signal (par défaut TERM) aux travaux spécifiés. <i>Désigne un travail :</i> PID %Numéro_Travail %Chaîne Travail qui commence par Chaîne. %?Chaîne Travail qui contient Chaîne. %% Travail en cours. %+ Equivalent à %%. %- Travail précédent.
let Exp ...	Evalue les expressions.
newgrp [-] [group]	Change le groupe courant.
print [-Rnpreu[n]] [Arg...]	Affiche les arguments. -R Ecrit en mode brut. -n Empêche d'ajouter une ligne sur la sortie. -p Envoie les arguments vers le tube généré par l'intermédiaire de  &. -r Ecrit en mode brut. -e Envoie les arguments vers le fichier historique. -u Ecrit la sortie vers le fichier de descripteur u.
pwd	Affiche le répertoire courant.
read [-preu[n]] [Nom?Invite] [Nom ...]	Une ligne est lue et divisée en plusieurs zones, les caractères de IFS étant utilisés comme séparateurs. -p Extrait la ligne d'entrée du tube d'entrée d'un processus généré par le shell en utilisant  &. -r Sauvegarde l'entrée dans l'historique. -u Lecture à partir du fichier de descripteur u.
readonly [Nom[=Valeur]] ...	Les noms sont marqués en lecture seule.
return [n]	Retour d'une fonction shell, avec le code n.
set [+ -aefhkmnopstuvx] [+ -o Option]...[+ -A Nom] [Argument ...]	-A Affectation de tableau.

shift [n]	Décale les paramètres.
test exp ou [ exp ]	L'équivalent de [[ exp ]] sauf les opérateurs && et   .
time Séquence_commandes	
times	Ecrit le temps utilisateur et le temps système du shell et des processus lancés par le shell.
trap [Commande] [Signal] ...	Lit et exécute la commande spécifiée lorsque le shell reçoit le ou les signaux spécifiés.
type Nom	Identifie une commande.
typeset [+HLRZfiltux[n]] [Nom[=Valeur]] ...	Définit les attributs et les valeurs des paramètres du shell.
-L	Justifier à gauche.
-R	Justifier à droite.
-Z	Justifier à droite avec des zéros en début de ligne.
-f	Les noms sont des fonctions.
-x	Active la trace pour cette fonction.
-i	Le paramètre est un entier.
-l	Convertir les majuscules en minuscules.
-r	Noms en lecture seule.
-t	Marque les paramètres nommés.
-u	Convertit les minuscules en majuscules.
	Remplacer le signe - par + désactive ces options.
ulimit [-HSacdfmst] [Limite]	Définit ou affiche une limite.
-H	Spécifie la limite permanente.
-S	Spécifie la limite temporaire.
-a	Les limites en cours.
-c	Nombre de blocs pour les fichiers « core ».
-d	Nombre de blocs de la zone de données.
-f	Nombre de blocs des fichiers écrits.
-m	Nombre de Ko de la mémoire physique.
-p	Nombre de Ko pour la pile.
-t	Nombre de secondes attribuées à chaque processus.
umask [Masque]	Spécifie le masque, ou l'affiche.
unalias Nom ...	Supprime des alias.
unset [-f] Nom ...	Les variables sont supprimées.
	-f Les fonctions sont supprimées.
wait [Travail]	Suspend le shell, et attend le travail spécifié ou par défaut tous les travaux.
whence [-v] Nom ...	Décrit l'interprétation d'une commande.
	-v Mode verbose.

## getopts

getopts Chaîne\_options Nom [Argument ...]

Vérifie les options d'un argument. Si aucun argument n'est spécifié, le paramètre Chaîne\_options contient les lettres reconnues par la commande `getopts`. Si une lettre est suivie d'un signe deux-points (:), l'option doit comporter un argument. Un espace peut séparer l'option et l'argument. La commande `getopts` place la lettre correspondant à l'option suivante dans la variable Nom lorsque le signe + précède l'argument. L'index de l'argument suivant est enregistré dans OPTIND. L'argument d'option, s'il existe, est placé dans OPTARG. Si un signe deux-points (:) est placé au début du paramètre Chaîne\_options, la commande `getopts` enregistre la lettre correspondant à une option incorrecte dans OPTARG, et attribue la valeur ? à la variable Nom d'une option inconnue, ou la valeur : lorsqu'une option requise n'est pas spécifiée. L'état de sortie est différent de zéro lorsqu'il n'y a plus d'option.



# Annexe E : awk – Le processeur de texte

## Syntaxe

```
awk [-Fx] [-v affec] program fichier ...
awk [-Fx] [-v affec] -f prg fichier ...
```

-Fx            Spécifie le séparateur de champs.  
 -v affec      Permet d'affecter des variables, variable=valeur.  
 program      Le programme AWK en argument, généralement quoté.  
 -f prg        Le programme AWK stocké dans le fichier *prg*.

## Structure d'un programme AWK

```
[ section BEGIN ]            # exécuté avant la lecture du fichier de données
[ section principale ]      # exécuté pour chaque ligne du fichier
[ section END ]              # exécuté après la lecture du fichier
```

### Remarque :

Les commentaires sont précédés du caractère #.

#### Structure de la section BEGIN

```
BEGIN action
action ::= { énoncé ... }
```

Les énoncés sont des ordres AWK séparés par des « ; » ou des sauts de ligne.

#### Structure de la section END

```
END action
```

#### Structure de la section principale

```
clause ...
clause ::= [ critère ] [ action ]
```

## Les expressions régulières

.	Un caractère quelconque.
^	Le début de la ligne.
\$	La fin de la ligne.
\x	Le caractère x, x étant un caractère spécial qui perd ainsi sa signification.
[liste]	Un caractère de la liste, ex : [abc].
[!liste]	Un caractère qui n'est pas dans la liste.
[x-y]	Un caractère appartenant au domaine précisé.
exp1 exp2	Correspond à l'une ou l'autre des expressions régulières.
exp ?	Zéro ou une occurrence de exp.
exp*	Zéro ou plusieurs occurrences de exp.
exp+	Au moins une occurrence de exp.

## Les caractères spéciaux

\b	Le retour arrière.
\f	Un saut de page.
\n	Un saut de ligne.
\r	Un retour chariot.
\t	Une tabulation.
\0xxx	Un caractère ASCII exprimé en octal.

## Les critères

Les critères sont utilisés par AWK pour sélectionner les lignes d'entrée qui déclenchent l'action qui suit.

/exp/	Contient l'expression régulière exp. Ex : /sh/ { print \$1 }
ch ~ exp	La chaîne ch contient l'expression régulière exp. Ex : \$2 ~ /^\$/ { print \$1 }
ch !~ exp	La chaîne ch ne contient pas l'expression régulière exp. Ex : \$2 !~ /^\$/ { print \$1 }
v1 == v2	La valeur v1 est égale à v2. Ex : \$2 == "" { print \$1 }
v1 != v2	La valeur v1 est différente de v2. Ex : \$2 != "oui" { print \$1 }
v1 < v2	La valeur v1 est inférieure à v2, numériquement ou lexicographiquement parlant. Ex : \$5 < "50" { print \$1 }
v1 <= v2	La valeur v1 est inférieure ou égale à v2. Ex : \$5 <= "M" { print \$1 }
v1 > v2	La valeur v1 est supérieure à v2. Ex : \$5 > "50" { print \$1 }
v1 >= v2	La valeur v1 est supérieure ou égale à v2. Ex : \$5 >= "M" { print \$1 }
c1 && c2	Le critère c1 et le critère c2. Ex : \$2 ~ /^\$/ && \$7 ~ /sh/ { print \$1 }
c1    c2	Le critère c1 ou le critère c2. Ex : \$5 == "50"    \$5 == "100" { print \$1 }
( critère )	Groupement de critères. Ex : ( \$5 == "50"    \$5 == "100" ) && \$7 ~ /sh/ { print \$1 }
! critère	La négation d'un critère. Ex : !( \$2 == "" ) { print \$1 }
/exp1/,/exp2/	Un domaine de lignes. Ex : /^root/,/^bin/ { print \$1 }

## Les variables

\$0	L'enregistrement (ligne) courant.
\$1, \$2, ...	Les champs de l'enregistrement courant.
ARGC	Le nombre d'éléments du tableau des arguments de la ligne de commandes.
ARGV	Le tableau des arguments de la ligne de commandes.
CONVFMT	Le format pour convertir des nombres en chaînes par la commande <b>printf</b> .
ENVIRON	Tableau associatif contenant l'environnement, le nom d'une variable d'environnement est utilisé comme indice.



FILENAME	Le nom du fichier d'entrée courant.
FNR	Nombre d'enregistrements lus (du fichier courant).
FS	Séparateur de champs en entrée.
NR	Nombre d'enregistrements lus (tous les fichiers confondus).
NF	Nombre de champs de l'enregistrement courant.
OFMT	Le format pour les nombres de la commande <b>printf</b> dans les instructions de sortie.
OFS	Le séparateur de champs de la commande <b>print</b> , l'espace par défaut.
ORS	Le séparateur d'enregistrements de la commande <b>print</b> , le « new line » par défaut.
RLENGTH	Longueur de la chaîne traitée par la fonction match.
RS	Séparateur d'enregistrements en entrée.
RSTART	Le point de départ de la chaîne trouvée par la fonction match.
SUBSEP	Séparateur d'indices d'un élément de tableaux.

## Les expressions

### *Les opérateurs mathématiques*

Les opérateurs unaires - (moins) et + (plus).

Les opérateurs binaires :

+, -, /, ^, \*, %

**Remarque :**

^ correspond à l'exponentiation et % au modulo.

### *Les opérateurs de pré- ou post-incrémentation*

++Variable ou --Variable

Variable++ ou Variable--

### *Les opérateurs d'affectation*

=, +=, -=, /=, \*=, ^=, %=

### *La concaténation de chaînes*

Il suffit d'indiquer les chaînes les unes derrière les autres, ex : \$3 = \$1\$2.

### *Les opérateurs de comparaison*

<, <=, ==, !=, >=, >

### *Les opérateurs booléens*

&&, ||, !

## Les tableaux

Un élément d'un tableau

tb[i]

Un élément d'un tableau à deux dimensions

tab[i,j]

Un élément d'un tableau associatif

table["pierre"]

Supprimer un élément de tableau

delete tb[i]

Tester l'existence d'un élément de tableau (*indice in tableau*)

"pierre" in table

## Les fonctions définies par l'utilisateur

fonction *nom*(arg1,arg2,...) { énoncé ... }

## Les instructions

if ( condition ) énoncé [ else énoncé ]

while ( condition ) énoncé

do énoncé while ( condition )

```
for ( exp ; condition ; exp ) énoncé
for ( variable in tableau ) énoncé
delete tableau[ domaine ]
print [ liste_d_expression ] [ > exp ]
printf format [, liste_d_expression ]
break
continue
next
exit [ expression ]
return [ expression ]
expression
```

## Les fonctions prédéfinies

atan2(y,x)	La tangente de y/x.
cos( x )	Le cosinus de x, x étant exprimé en radian.
sin( x )	Le sinus de x, x étant exprimé en radian.
exp( x )	La fonction exponentielle de x.
log( x )	Le logarithme de x.
sqrt(x)	La racine carrée de x.
int( x )	Renvoie la partie entière de x.
rand( )	Renvoie un nombre aléatoire n, avec $0 \leq n < 1$ .
srand( [exp] )	Utilise exp pour initialiser la génération des nombres aléatoires, par défaut utilise l'heure.
gsub(exp,ch,[In])	Identique à la fonction sub, mais toutes les occurrences de l'expression régulière exp sont remplacées.
sub(exp,ch,[In])	Remplace la première occurrence de l'expression régulière exp, par la chaîne ch dans la chaîne In. Si In n'est pas précisé, on utilise \$0.
length(x)	Renvoie la longueur de la chaîne x.
index(x,y)	Renvoie la position de la chaîne x dans la chaîne y, sinon renvoie 0.
match(ch1,ch2)	Recherche la chaîne ch1 dans la chaîne ch2, renvoie la position de départ ou 0 si la chaîne n'a pas été trouvée.
substr(x,i,l)	Renvoie la sous-chaîne de x, qui commence au ième caractère et de longueur l.
split(x,tb)	Met chaque champ de x dans un élément du tableau tb.
sprintf(f,exp,...)	Renvoie les expressions exp données en argument en utilisant le format f.
tolower(ch)	Renvoie la chaîne ch convertie en minuscules.
toupper(ch)	Renvoie la chaîne ch convertie en majuscules.
system(cmd)	Exécute la commande cmd et renvoie son code retour.
close(exp)	Ferme un fichier ouvert par une instruction print, printf ou getline.
exp getline [var]	Lit un enregistrement dans la variable var à partir d'un tube créé par une commande, par défaut var correspond à \$0.
getline	Lit le prochain enregistrement dans la variable \$0.
getline var	Idem, mais dans la variable var.
getline [var]<exp	Idem, mais lit l'enregistrement à partir du fichier indiqué par exp.

# Annexe F : sed – Editeur en mode flot

## Syntaxe

```
sed [-n] script [fichier ...]
sed [-n] [-e script] ... [-f prg] ... [fichier ...]
```

-n            Supprime la sortie par défaut : normalement chaque ligne non modifiée est recopiée sur la sortie standard.

script        Une instruction sed.

-e script     Une instruction sed.

-f prg        Le fichier *prg* contient une suite d'instructions sed.

## Syntaxe d'une commande sed

```
[adresse[,adresse]]commande[argument ...]
[adresse[,adresse]]{
    commande ...
}
```

Les adresses correspondent à des lignes du texte. Elles peuvent s'exprimer selon les expressions suivantes :

num    Le numéro num de la ligne.

\$        La dernière ligne.

/exp/    La première ligne qui satisfait l'expression régulière exp. Le caractère / peut être remplacé par un caractère quelconque.

Si aucune adresse n'est précisée, la commande agit sur chaque ligne du texte.

Dans la description des commandes, la syntaxe [2] signifie que la commande peut être précédée d'un domaine d'adresse : adresse,adresse ; [1] signifie que la commande ne peut être précédée que d'une seule adresse.

Le caractère # débute un commentaire qui se termine par la fin de ligne.

## Les tampons

La commande **sed** dispose de deux tampons permettant de stocker les lignes en cours de modification : l'espace des modèles (« *pattern space* »), où les lignes sélectionnées sont maintenues pendant la durée du traitement, et l'espace annexe (« *hold space* »), où une ligne peut être stockée temporairement.

## Les commandes

[1]a\  
text        Ajoute le texte text, et le place sur la sortie avant la lecture de la prochaine ligne. « text » correspond à une seule ligne de texte qui peut contenir des sauts de ligne précédés d'un backslash (« \ »).

[2]b [étiq]    On se branche à l'étiquette etiq ; si l'on ne précise pas d'étiquette, on se branche à la fin du programme.

[2]c\  
text        Substitue le texte courant par text.

[2]d        Détruit la ligne.

[2]D        Détruit la première ligne du domaine sélectionné.

[2]g        Remplace le contenu de l'espace des modèles par le contenu de l'espace annexe.

- [2]G Ajoute le contenu de l'espace annexe à celui des modèles.
- [2]h Remplace le contenu de l'espace annexe par celui des modèles.
- [2]H Ajoute le contenu de l'espace annexe à celui des modèles.
- [1]i\  
text Insère le texte text avant la prochaine sortie.
- [2]l Imprime la ligne, en visualisant tous les caractères.
- [2]n Imprime la ligne.
- [2]N Ajoute les lignes sélectionnées à l'espace des modèles en les séparant d'un saut de ligne. On peut utiliser cette commande pour rechercher des modèles s'étendant sur plusieurs lignes.
- [2]p Imprime la ligne.
- [2]P Imprime la première ligne du texte sélectionné.
- [1]q Quitte sed.
- [1]r fichier Lit le fichier et copie son contenu sur la sortie.
- [2]s/exp\_reg/ch\_sub/[drapeaux]  
La commande substitue la chaîne correspondant à l'expression régulière exp\_reg par la chaîne ch\_sub.  
Le séparateur / peut être remplacé par un séparateur quelconque.  
Dans la chaîne ch\_sub, le caractère & correspond à la chaîne trouvée.  
Dans la chaîne ch\_sub, l'expression \n correspond à la nième sous-expression \(...\) de l'expression régulière exp\_reg.  
Les drapeaux suivants sont supportés :  
n Le remplacement porte seulement sur la nième occurrence.  
g Remplace toutes les occurrences de la ligne.  
p Toutes les occurrences sont affichées.  
w fic Écrit dans le fichier fic les lignes substituées.
- [2]t [etiq] Saut à l'étiquette etiq si une substitution a été faite dans la ligne courante.
- [2]w fic Écrit la ligne dans le fichier fic.
- [2]x Permute l'espace des modèles et l'espace annexe.
- y/ch1/ch2/ Remplace chaque caractère de ch1 par le caractère correspondant de ch2.
- [1]= Affiche le numéro de la ligne courante.
- [2]!cmd Exécute cmd si la ligne n'est pas sélectionnée.
- :etiq Définit une étiquette.

# Annexe G : Les principaux fichiers d'administration

Le tableau qui suit donne la liste des principaux fichiers d'administration du système Linux. Pour chacun, une brève description est fournie. L'ordre, choisi arbitrairement, est l'ordre alphabétique.

<i>/.profile</i>	Script exécuté automatiquement en début d'une session d'administration.
<i>/boot/grub/grub.conf</i>	Le fichier de configuration du chargeur GRUB.
<i>/etc/at.allow, /etc/at.deny</i>	La configuration de l'accès à la commande <code>at</code> .
<i>/etc/bashrc</i>	Script exécuté en début de chaque session d'utilisateur du shell <code>bash</code> .
<i>/etc/modprobe.conf</i> ou <i>/etc/modules.conf</i> ou <i>/etc/conf.modules</i>	Configure le démon <code>kernel.d</code> .
<i>/etc/cron.allow,</i> <i>/etc/cron.deny</i>	La configuration de l'accès à la commande <code>crontab</code> .
<i>/etc/crontab</i>	Une table au format <code>cron</code> , gérée par <code>root</code> , mais exécutée sous diverses identités.
<i>/etc/dhcpd.conf</i>	Le fichier de configuration du service DHCP.
<i>/etc/dumpdates</i>	L'enregistrement des sauvegardes réalisées par la commande <code>dump</code> .
<i>/etc/exports</i>	Les arbres partagés par un serveur NFS.
<i>/etc/fstab</i>	Liste des systèmes de fichiers qui doivent être montés et des espaces de swap qui doivent être activés au démarrage.
<i>/etc/ftpusers</i>	Liste des utilisateurs interdits de FTP.
<i>/etc/gettydefs</i>	Fichier qui paramètre les terminaux.
<i>/etc/group</i>	Description des comptes de groupes d'utilisateurs.
<i>/etc/httpd/*</i>	Les fichiers de configuration du service Apache installé à partir d'un paquetage RPM.
<i>/etc/host.lpd</i>	Fichier de gestion de la sécurité d'accès au service d'impression à travers le réseau.
<i>/etc/hosts</i>	Table de correspondance entre un nom et une adresse IP.
<i>/etc/hosts.allow,</i> <i>/etc/hosts.deny</i>	fichiers de gestion de la sécurité d'accès du service « wrapper » TCP.
<i>/etc/hosts.equiv</i>	Liste des ordinateurs en relation de confiance pour l'exécution des « r-commande ».
<i>/etc/init.d/*</i>	Les scripts de démarrage ou d'arrêt des services, activés via les liens <code>/etc/rc ?.d/*</code> .
<i>/etc/inittab</i>	Liste des applications lancées au démarrage par <code>init</code> .

<i>/etc/issue</i>	Fichier affiché avant une connexion.
<i>/etc/lilo.conf</i>	Le fichier de configuration de LILO.
<i>/etc/login.defs</i>	Paramètre la durée de vie du mot de passe.
<i>/etc/man.config</i>	Configure la documentation en ligne.
<i>/etc/motd</i>	Fichier affiché après une connexion.
<i>/etc/named.conf</i> ou <i>/etc/named.boot</i>	Le principal fichier de configuration du démon <code>named</code> , le serveur DNS.
<i>/etc/nologin</i>	Le verrou qui empêche la connexion des utilisateurs.
<i>/etc/nsswitch.conf</i>	Le fichier qui donne l'ordre d'utilisation des services (fichiers locaux, LDAP, DNS...).
<i>/etc/pam.d/*</i>	Les fichiers de configuration de PAM.
<i>/etc/passwd</i>	Description des comptes utilisateurs.
<i>/etc/printcap</i>	Fichier de description des files d'attente d'impression.
<i>/etc/profile</i>	Script exécuté en début de chaque session d'utilisateur.
<i>/etc/rc.d</i>	Script lancé au démarrage par <code>init</code> via <i>/etc/inittab</i> . Il active les scripts des répertoires <i>/etc/rc.d/rc ?.*</i> .
<i>/etc/rc.d/init.d</i>	Le répertoire des scripts de démarrage et d'arrêt des services, appelés par les scripts <i>/etc/rc.d/*rc*</i> .
<i>/etc/rc.d/rc?.*</i>	Des liens sur les scripts de <i>/etc/init.d</i> .
<i>/etc/samba/*</i>	Les fichiers de configuration de Samba installé à partir d'un paquetage RPM.
<i>/etc/securetty</i>	Les terminaux où <code>root</code> peut se connecter.
<i>/etc/sendmail.cf</i>	Le fichier de configuration de <code>sendmail</code> , le serveur Email.
<i>/etc/services</i> , <i>/etc/inetd.conf</i>	Fichiers de configuration du super démon réseau <code>inetd</code> .
<i>/etc/shadow</i>	Paramètre le « password aging », contient les mots de passe.
<i>/etc/shutdown.allow</i>	Les utilisateurs autorisés à arrêter le système par CTRL-ALT-DEL.
<i>/etc/skel/*</i>	Les fichiers installés dans un nouveau compte.
<i>/etc/smb.conf</i>	Le fichier de configuration de Samba.
<i>/etc/termcap</i>	Les séquences de contrôle des terminaux.
<i>/etc/X0.host</i>	Définit la sécurité d'accès d'un écran X.
<i>/etc/X11/fvwm2/system.fvwm2rc</i>	Fichier de configuration, par défaut, de <code>fvwm2</code> .
<i>/etc/X11/prefdm</i>	Lien sur le gestionnaire de connexion graphique.
<i>/etc/X11/twm/system.twmrc</i>	Fichier de configuration, par défaut, de <code>twm</code> .
<i>/etc/X11/xdm/*</i>	Fichiers de configuration de <code>xdm</code> .

<i>/etc/X11/xorg.conf</i> ou <i>/etc/X11/XF86Config</i>	Le fichier de configuration du serveur X.
<i>/etc/X11/xinit/Xclients</i>	La configuration par défaut de la commande <code>xinitrc</code> .
<i>/tmp/install.log</i>	Compte rendu de l'installation.
<i>/usr/lib/X11/app-defaults/*</i>	Fichiers de configuration des ressources.
<i>/usr/lib/X11/xinitrc</i>	La configuration par défaut de la commande <code>startx</code> .
<i>/usr/share/magic</i>	Les « <i>magic numbers</i> » utilisés par la commande <code>file</code> .
<i>/usr/share/terminfo/*</i>	Alternative à <i>/etc/termcap</i> .
<i>/var/spool/cron/*</i>	Les fichiers de requêtes des travaux périodiques.
<i>/var/log/cron</i>	Le fichier <i>log</i> de cron.
<i>/var/log/messages</i>	Les messages affichés normalement sur la console maîtresse.
<i>/var/log/secure</i>	L'historique des connexions qui ont échoué.
<i>~/.bash_logout</i>	Le script exécuté à la déconnexion.
<i>~/.bash_profile</i>	Script exécuté automatiquement au début d'une session d'un utilisateur qui utilise le shell <code>bash</code> .
<i>~/.bashrc</i>	Script exécuté automatiquement au début d'une session d'un utilisateur qui utilise le shell <code>bash</code> .
<i>~/.fvwm2rc</i>	Fichier de configuration de <code>fvwm2</code> .
<i>~/.profile</i>	Script exécuté automatiquement au début d'une session d'un utilisateur qui utilise le shell ISO ou le Korn shell.
<i>~/.twmrc</i>	Fichier de configuration de <code>twm</code> .
<i>~/.xinitrc</i> , <i>~/.xsession</i> , <i>~/.Xclients</i>	Les fichiers de configuration d'une session X.
<i>~/.Xresources</i> , <i>~/.Xdefaults</i>	Fichiers de configuration des ressources.