

# Liste alphabétique des principales commandes

## Bourne shell (sh)

## Bourne again shell (bash)

## Korn Shell (ksh)

**awk** <instructions ...> [-f] prog.awk fichier ...

Permet de travailler sur le contenu d'un fichier.

-f prog.awk

indique le nom du fichier programme awk au lieu des instructions.

**case** variable in  
expression1)

bloc de commandes

::

expression2)

bloc de commandes

::

\*)

bloc de commandes

::

esac

**cat** fichier ...

Concatène et affiche des fichiers.

**cp** [-ifRPsvu] source ... destination

CoPy

Copie des fichiers

-i interroge l'utilisateur avant d'écraser des fichiers.

-f efface les fichiers cibles existants.

-R copie récursivement répertoires et sous-répertoires en préservant les types de fichiers rencontrés.

-P crée les répertoires parents s'ils n'existent pas.

-s crée des liens symboliques plutôt que des copies.

-v affiche le nom du fichier avant la copie.

-u ne copie pas si le fichier destination est plus récent.

**cut** [-dcfs] fichier

Supprime une partie de chaque ligne d'un fichier (découpage vertical).

-c <colonnes>

affiche les colonnes spécifiées.

-f <champs>

n'affiche que les champs indiqués, par défaut le séparateur de champ est la tabulation.

-d <séparateur>

avec -f, champs délimités par le séparateur au lieu de la tabulation.

-s avec -f, ne pas afficher les lignes ne contenant pas le séparateur de champs.

**echo** [-n] chaîne

Affiche la chaîne de caractères sur le canal de sortie standard. Certains caractères non affichables peuvent être utilisés via des codes précédés d'un \.

-n Si n est précisé, la ligne ne se termine pas par un retour à la ligne.

**exit** [n]

Permet de sortir d'un script Shell avec le statut n. n peut valoir 0 (succès) ou toute autre valeur non nulle (échec). Si n n'est pas précisé, exit renvoie le statut de la dernière commande exécutée.

**find** chemin [options] [tests] <expression recherchée> [actions]

Recherche des fichiers dans une hiérarchie de répertoire.

**options :**

-maxdepth N

parcourt N niveaux maximum dans les sous-répertoires.

-mindepth N

parcourt N niveaux minimum dans les sous-répertoires.

-mount ne parcourt pas les répertoires situés sur d'autres systèmes de fichiers.

**tests :**

-amin N dernier accès au fichier il y a N minutes.

-anewer fichier

dernier accès au fichier recherché plus récent que l'accès au fichier spécifié.

-atime N dernier accès au fichiers il y a N jours.

-empty fichier vide.

-fstype fichier se trouvant sur un système de fichiers du type indiqué.

-name fichier

spécifie le nom du fichier à rechercher.

-iname fichier

identique à -name sans différencier majuscules et minuscules.

-inum N fichier dont le numéro d'i-node est N.

-lname lien

identique à -name pour les liens symboliques.

-newer fichier

fichier modifié plus récemment que celui spécifié.

-group groupe

fichier appartenant au groupe spécifié.

-perm mode

fichier dont les autorisations sont fixées exactement au mode indiqué (en symbolique ou octal).

-size N[cbkw]

fichier occupant N unités d'espace (c : octets, b : blocs de 512 octets,

-type X

k : kilo-octets, w : mots de 2 octets).

b fichier spécial en mode bloc

c fichier spécial en mode caractère

d répertoire

p tube nommé

f fichier régulier

l lien symbolique

s socket

-user utilisateur

fichier appartenant à l'utilisateur.

**actions :**

exec commande {} \ ;

exécute la commande indiquée si la recherche est positive.

-ok commande {} \ ;

identique à -exec avec une demande de confirmation.

par l'utilisateur.

-print affiche le nom complet du fichier sur la sortie standard.

-fprint fichier

inscrit le nom complet dans le fichier spécifié.

**for** variable in liste de valeurs

do

bloc de commandes

done

Crée une structure de type boucle.

**grep** [-vcnli] fichier ...

Recherche les lignes correspondant à une expression donnée.

-v affiche les lignes ne contenant pas l'expression.

-c affiche le nombre de lignes contenant l'expression.

-n ajoute à chaque ligne en sortie son numéro dans le fichier.

-l affiche le nom des fichiers où l'expression a été trouvée.

-i ignore la casse.

**if** condition

then bloc de commandes

fi

Crée une structure de test.

**if** condition

then bloc de commandes

elif condition

then bloc de commandes

else bloc de commandes

fi

Crée une structure de test.

**fonction\_personnalisée** ()

{

bloc de commandes

}

Crée une fonction personnalisée.

## local variable

Déclare une variable locale à la fonction.

## return valeur

Termine la fonction en renvoyant la valeur en retour.

## ls [-laidRrtF] répertoire | fichier ...

Affiche le contenu d'un répertoire, le répertoire courant si aucun argument.

- l affichage au format détaillé (long).
- a affiche tous les fichiers, dont les fichiers commençant par un . (cachés).
- i affiche le numéro d'i-node.
- d affiche les répertoires uniquement.
- R liste récursive avec les sous-répertoires.
- r affiche dans l'ordre inversé.
- t tri par date de dernière modification.
- F ajoute un caractère à chaque nom de fichier pour indiquer son type (/ pour un répertoire, @ pour un lien, \* pour un exécutable).

## mv [-fiuv] source ... destination

MoVe

Déplace ou renomme des fichiers.

- f efface les fichiers cibles existants.
- i interroge l'utilisateur avant d'écraser des fichiers.
- u ne copie pas si le fichier destination est plus récent.
- v affiche le nom du fichier avant la copie.

## rm [-firvd] fichier ...

ReMove

Supprime des fichiers.

- f efface les fichiers cibles sans confirmation.
- i interroge l'utilisateur avant de supprimer des fichiers.
- r efface récursivement les sous-répertoires.
- v affiche le nom du fichier avant la suppression.
- d efface un répertoire (y compris non vide - commande ROOT uniquement).

## sed <instructions ...> [-f] instr.sed fichier ...

Permet d'effectuer des transformations basiques de texte.

## sort [-cm] [-t séparateur] [-o fichier] [-bdfnr] fichier ...

Trie les lignes d'un fichier texte.

- c vérifie si les fichiers sont déjà triés.
- m regroupe les fichiers indiqués en les triant, chaque fichier doit être trié individuellement.
- t séparateur  
utilise le caractère séparateur qui distinguera les champs pour rechercher les clés sur chaque ligne.
- o fichier  
écrit dans un fichier plutôt que sur la sortie standard.
- b ignore les blancs en début de ligne pendant la recherche de la clé de tri.

- d trie dans l'ordre des répertoires téléphoniques (ignore les caractères autres que les lettres, les chiffres, les blancs).
- f considère les majuscules comme des minuscules.
- n compare suivant la valeur numérique d'une chaîne initiale.
- r inverse l'ordre de tri.

## tar [-] [crtuxv] [f sauvegarde] fichiers | répertoires ...

Gère un ensemble de fichiers associés dans un fichier de sauvegarde. La liste des fichiers peut contenir des répertoires ; les fichiers et répertoires contenus dans ces derniers seront archivés en conservant l'arborescence initiale.

- c crée une sauvegarde.
- r ajoute des fichiers à une sauvegarde.
- t liste le contenu d'une sauvegarde.
- u ajoute des fichiers nouveaux ou plus récents à une sauvegarde.
- x extrait des fichiers d'une sauvegarde. Par défaut, tar extrait tous les fichiers contenus dans la sauvegarde.
- v affiche des informations concernant les fichiers.
- f sauvegarde  
donne un nom à la sauvegarde.

## test condition

ou

[ condition ]

Condition :

- a fichier vrai si le fichier existe.
- b fichier vrai si le fichier existe et est de type bloc.
- c fichier vrai si le fichier existe et est de type caractère.
- f fichier vrai si le fichier existe et est de type ordinaire.
- d fichier vrai si le fichier existe et est de type répertoire.
- r fichier vrai si le fichier existe et est accessible en lecture (R).
- w fichier vrai si le fichier existe et est accessible en écriture (W).
- G fichier vrai si le fichier existe et appartient au groupe de l'utilisateur.
- o fichier vrai si le fichier existe et appartient à l'utilisateur.
- x fichier vrai si le fichier existe et est exécutable (X).
- S fichier vrai si le fichier existe et est un socket.
- k fichier vrai si le fichier existe et a le sticky bit activé.
- s fichier vrai si le fichier existe et a une taille non nulle.
- L fichier vrai si le fichier existe et est un lien symbolique.
- n chaîne vrai si la longueur de la chaîne est non nulle.
- z chaîne vrai si la longueur de la chaîne est nulle.

Comparaison :

- entier1 -eq entier2  
vrai si entier1 et entier2 sont algébriquement égaux.
- entier1 -gt entier2  
vrai si entier1 est plus grand que entier2.
- entier1 -lt entier2  
vrai si entier1 est plus petit que entier2.

entier1 -ne entier2

vrai si entier1 est différent de entier2.

entier1 -ge entier2

vrai si entier1 est supérieur ou égal à entier2.

entier1 -le entier2

vrai si entier1 est inférieur ou égal à entier2.

chaîne1 = chaîne2

vrai si les deux chaînes sont égales.

chaîne1 != chaîne2

vrai si les deux chaînes sont différentes.

fichier1 ef fichier2

vrai si les fichiers existent et sont identiques.

fichier1 nt fichier2

vrai si fichier1 existe et est plus récent que fichier2.

fichier1 ot fichier2

vrai si fichier1 existe et est plus ancien que fichier2.

## until condition

do

bloc de commandes

done

Crée une structure de type boucle.

## wc [-lwmc] fichier ...

Word Count

Affiche le nombre de lignes, de mots, de caractères d'un fichier.

- l affiche le nombre de lignes.
- w affiche le nombre de mots.
- m affiche le nombre de caractères.
- c affiche le nombre d'octets.

## while condition

do

bloc de commandes

done

Crée une structure de type boucle.

## who [-mqjuHw] [am i]

Montre qui est connecté. Sans argument who affiche le nom de connexion, le terminal, l'heure de connexion, le nom de l'hôte distant ou de terminal X.

- m identique à who am i.
- q affiche uniquement les noms d'utilisateurs connectés et le nombre d'utilisateurs connectés.
- i -u  
affiche le nombre d'heures et de minutes d'inactivité.
- H affiche une ligne de libellé pour les colonnes.
- w ajoute après le nom de connexion un caractère indiquant le statut de l'utilisateur vis-à-vis des messages.
- ami affiche le nom de l'utilisateur actuellement connecté et le terminal de connexion.

## Regroupement thématique

### Répertoires et fichiers

cp, ls, mv, rm, wc

### Traitement de fichiers et filtres

awk, cat, cut, sed, sort, tar

### Recherche

find, grep

### Gestion des comptes utilisateurs

who

### Divers

echo, exit

### Test

test ... [ condition ]

### Structures de contrôle

if ... if, while, until

### Analyse de chaîne

case ... esac

### Boucle

for ... done

### Fonction

fonction\_personnalisée

## Exécution de commandes

# ! /bin/shell shell peut prendre la valeur sh, ksh, bash...

Ligne shebang : sert à indiquer au système d'exploitation le chemin d'accès de l'exécutable à lancer pour interpréter le fichier.

commande1 ; commande2 Exécution successive de commandes.

commande1 | commande2 Sortie standard de la commande injectée dans le flux d'entrée de la suivante.

commande1 && commande2 Enchaînement conditionnel de commandes. La deuxième n'est exécutée que si la première s'est correctement déroulée (ET).

commande1 || commande2 Enchaînement conditionnel de commandes. La deuxième n'est exécutée que si la première ne s'est pas bien passée (OU).

{ liste de commandes } Regroupement de commandes sans création de sous shell.

( liste de commandes ) Création de sous shell dans une commande.

Retrouvez nos ouvrages sur le même sujet dans la rubrique "Système et réseau" sur notre site internet.



ISBN : 2-7460-3119-1

Prix : 6 €

www.editions-eni.com



9 782746 031197

## Redirection des commandes

Commande > fichier La sortie standard est redirigée dans le fichier indiqué.  
Commande >> fichier La sortie standard est redirigée à la fin du fichier indiqué.  
Commande < fichier L'entrée standard est lue depuis le fichier-indiqué.

Commande << [-] délimiteur  
ligne 1  
ligne 2  
délimiteur  
Les deux lignes seront envoyées dans l'entrée standard de la commande.

Commande 2> fichier La sortie d'erreur est redirigée dans le fichier indiqué.  
Commande 2>> fichier La sortie d'erreur est redirigée à la fin du fichier indiqué.  
Commande 2>&1 La sortie d'erreur est redirigée de la même façon que la sortie standard.

## Métacaractères

\* Remplace 0 à n caractères quelconques. Peut être utilisé seul ou pour remplacer n caractères dans une chaîne.

? Remplace un seul caractère quelconque.

[liste caractères]

Remplace un seul caractère parmi ceux de la liste entre crochets, bornes incluses.

[Caractère Début-Caractère Fin]

Remplace un seul caractère compris dans l'intervalle défini entre crochets, bornes incluses.

[! ...] négation, remplace un seul caractère différent de ceux cités dans la liste entre crochets.

exemples:

\*.c Tous les noms qui se terminent par .c.

ab?h Tous les noms de 4 caractères commençant par ab, se terminant par h, la valeur du troisième caractère est quelconque.

ab[rthn]24 Tous les noms de cinq caractères commençant par ab se terminant par 24, la valeur du troisième caractère est r, t, h ou n.

ab[a-h1-5]4 Tous les noms de quatre caractères commençant par ab se terminant par 4, la valeur du troisième caractère doit être comprise entre a et h ou entre 1 et 5.

ab[!otu]h Tous les noms de quatre caractères commençant par ab, se terminant par h, la valeur du troisième caractère est différente de o, t ou u.

[A-Z] Un caractère parmi les lettres majuscules.

[A-Za-z] Un caractère parmi toutes les lettres minuscules ou majuscules.

## Caractères spéciaux

Ces caractères fonctionnent par paire.

"..." Empêche l'extension des métacaractères.

'...' Empêche l'interprétation des caractères spéciaux et des métacaractères.

`...` Récupère l'évaluation d'une commande et fournit son résultat.

\$(...) Identique à `...` (valide uniquement en Korn Shell).

\$... Préfixe un nom de variable et permet de spécifier son contenu.

\ Empêche l'interprétation du caractère spécial qui suit.

## Variables Korn shell

Les variables précédées par ► sont également disponibles en Bourne shell (sh).

### Affectation d'une variable

Variable chaîne de caractères

► variable="valeur"

Variable entière

integer variable=valeur déclaration de variable entière.

typeset -i variable=valeur commande identique à integer.

Tableau de variables

tableau[n]="valeur" déclaration de la valeur au rang n du tableau de variables.

### Protection d'une variable (création d'une constante)

► readonly variable mise en lecture seule de la variable.

typeset -r variable=valeur création et mise en lecture seule de la variable.

### Suppression de la protection d'une variable

typeset +r variable suppression de la protection de la variable.

### Utilisation d'une variable stockée

► \$variable

### Destruction d'une variable

► unset variable

### Substitution de variables

#{variable-valeur}

remplace la variable par valeur si elle n'est pas définie.

#{variable=valeur}

affecte la valeur à la variable si elle n'est pas définie.

#{variable?message}

affiche le message et termine le shell si la variable n'est pas définie.

#{#variable}

donne la longueur du contenu de la variable.

#{#tableau[\*]}

donne le nombre d'éléments du tableau de variables.

### Variables réservées

\$\$ numéro du processus en cours.

\$? code de retour (code erreur par exemple).

\$! numéro du dernier processus lancé en arrière-plan.

### Variables réservées accessibles à partir d'un script

\$# nombre de paramètres reçus par le script.

\$0 nom de la commande ou du script.

\$n arguments d'un script (\$1 : premier argument ... \${10} dixième argument).

\$\* valeurs concaténées des paramètres reçus sans séparateurs.

@\$ valeurs des paramètres reçus encadrés par " ".

### Visualisation des variables

► set affiche les variables du shell et de l'environnement.

► env affiche les variables d'environnement.

typeset affiche les variables.

### Exportation des variables

► export variable exporte une variable dans les shell fils du shell courant (permet de créer des variables d'environnement).

# Variables

## Bourne again shell (bash)

Les variables précédées par **>** sont également disponibles en Bourne shell (sh).

### Affectation d'une variable

#### Variable chaîne de caractères

**>variable="valeur"**

#### Variable entière

**declare -i variable=valeur** déclaration de variable entière.

#### Tableau de variables

**tableau[n]="valeur"** déclaration de la valeur au rang n du tableau de variables.

### Protection d'une variable (création d'une constante)

**>readonly variable** mise en lecture seule de la variable.

**declare -r variable=valeur** création et mise en lecture seule de la variable.

**declare +r variable** suppression de la protection de la variable.

### Utilisation d'une variable stockée

**>\$variable**

### Destruction d'une variable

**>unset variable**

### Substitution de variables

**\${variable-valeur}** remplace la variable par valeur si elle n'est pas définie.

**\${variable=valeur}** affecte la valeur à la variable si elle n'est pas définie.

**\${variable?message}** affiche le message et termine le shell si la variable n'est pas définie.

**\${#variable}** donne la longueur du contenu de la variable.

**\${#tableau[\*]}** donne le nombre d'éléments du tableau de variables.

### Variables réservées

**\$\$** numéro du processus en cours.

**\$?** code de retour (code erreur par exemple).

**!** numéro du dernier processus lancé en arrière-plan.

### Variables réservées accessibles à partir d'un script

**\$#** nombre de paramètres reçus par le script.

**\$0** nom de la commande ou du script.

**\$n** arguments d'un script (\$1 : premier argument ...

\$\_{10} : dixième argument).

**\$\*** valeurs concaténées des paramètres reçus sans séparateurs.

**\$@** valeurs des paramètres reçus encadrés par " ".

### Visualisation des variables

**>set** affiche les variables du shell et de l'environnement.

**>env** affiche les variables d'environnement.

**declare** affiche les variables.

### Exportation des variables

**>export variable** exporte une variable dans les shell fils du shell courant (permet de créer des variables d'environnement).

# L'éditeur de texte vi

### vi fichier

Edite le fichier.

### Quitter vi

**:q !** sortie de vi sans enregistrement (le ! force la commande).  
**:wq** sortie avec sauvegarde (Date de modification toujours changée).  
**:x** sortie avec sauvegarde (Date de modification non modifiée si aucun changement).

### Déplacements

**Flèches** déplacement du curseur d'un caractère.  
**h,j,k,l** déplacement du curseur d'un caractère.  
**H** déplacement en haut de page.  
**L** déplacement en fin de page.  
**M** déplacement en milieu de page.  
**w** déplacement en début de mot suivant.

### Insertion de texte

**i** insertion de texte avant le curseur.  
**a** insertion de texte après le curseur.  
**o** insertion d'une ligne après la ligne courante.  
**O** insertion d'une ligne avant la ligne courante.

### Recherche dans le document

**/chaîne** recherche chaîne vers le bas.  
**//** recherche la chaîne suivante.  
**?chaîne** recherche chaîne vers le haut.  
**??** recherche la chaîne précédente.

### Supprimer (couper) du texte

**x** supprime (coupe) le caractère courant.  
**X** supprime (coupe) le caractère précédent.  
**6x** supprime (coupe) les 6 caractères suivants.  
**dw** supprime (coupe) le mot courant.  
**5dw** supprime (coupe) 5 mots à partir du mot courant.  
**dd** supprime (coupe) la ligne courante.  
**5dd** supprime (coupe) 5 lignes à partir de la ligne courante.

### Copier du texte

**y5w** copie les 5 mots suivants.  
**yy** copie la ligne courante.  
**5yy** copie 5 lignes.

### Coller du texte

**p** colle après le curseur.  
**P** colle avant le curseur.

### Commandes diverses

**u** annule la dernière modification.  
**:set number** affiche la numérotation des lignes.  
**:set nonumber** supprime la numérotation des lignes.

# Shell

## Programmation sous Unix/Linux

[ Bourne shell, Bourne Again shell, Korn shell ]

[ Liste des principales commandes ]

[ Syntaxe détaillée ]

[ Regroupement thématique ]

[ Métacaractères et caractères spéciaux ]

[ Détail des variables ]

[ Éditeur de texte vi ]



Laumonier Gérard

5 la chanterie

37110 Villedomer

Email : g.laumonier@wanadoo.fr

