

## Aide-mémoire des commandes Unix de A à L

Ce document regroupe les principales commandes susceptibles d'être employées par les stagiaires de la formation « Programmation Shell et Langages de Scripts », en rappelant leurs options les plus utilisées. Pour avoir plus de détail sur une commande particulière, on consultera le manuel Unix (commande man).

© Christophe BLAESS 2004

**a propos** Liste les pages du manuel concernant un sujet  
| apropos shell

**arch** Affiche l'architecture de la machine.

**at, atq, atrm** Mémorise, examine ou supprime des jobs à exécuter ultérieurement.

-f lire les commandes dans le fichier indiqué.

```
$ at now + 10 minutes < script.sh
$ at 20:55 -f demarre_magnetoscope.sh
```

**awk** Interpréteur du langage Awk.

**NF** nombre de champs sur la ligne

**FS** séparateur de champs

**NR** numéro d'enregistrement (de ligne)

**\$1, \$2...\$NF** champs successifs

```
ls -l | awk '{somme+= $5} END{print
somme}'
```

additionne les tailles des fichiers du répertoire courant

**basename** Élimine le chemin d'accès et le suffixe éventuel d'un nom de fichier.

```
$ basename /usr/src/linux/signal.c
signal.c
$ basename /usr/src/linux/signal.c .c
signal
$
```

**bash** Shell Gnu Bash

**batch** Lance un job en différé suivant la charge système

```
batch << FIN
    tri_des_enregistrements
FIN
```

**bc** Calculateur en précision arbitraire.

-l bibliothèque mathématique complète.

```
$ pi=$(echo "a(1)*4" | bc -l)
$ echo $pi
3.14159265358979323844
$
```

**bg** Relance à l'arrière-plan un job arrêté.

```
$ application
(Ctrl-Z)
[1]+ Stopped application
$ bg
[1]+ application &
$
```

**bunzip2** Décompresse un fichier .bz2.

**bzip2** Comprime un fichier.

**cal** Affiche un calendrier.

```
cal 5 2003
```

**cat** Concatène des fichiers sur la sortie standard.

-n numérote les lignes en sortie,

-v caractères spéciaux sous forme symbolique.

```
cat entete.txt corps.txt signature.txt
mail.txt
```

**cc** Compilateur C.

**cd** Change de répertoire de travail.

```
cd /usr/src/linux
```

```
cd -
```

revient au répertoire précédent

```
cd
```

revient dans le répertoire personnel.

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

-R modifie récursivement les sous-répertoires.

```
chgrp equipe_2 fichier
```

**chmod** Modifie les permissions d'un fichier.

-R modifie récursivement les sous-répertoires.

```
chmod 644 texte
```

lecture pour tous, écriture seulement pour propriétaire

```
chmod 755 script
```

lecture et exécution pour tous, écriture pour propriétaire

```
chmod u+s executable
```

Activation du bit Set-UID du fichier.

**chown** Modifie propriétaire et groupe d'un fichier.

-R modifie récursivement les sous-répertoires.

```
chown user1.equipe1 fichier
```

**chsh** Change le shell appelé à la connexion.

-l liste des shells disponibles,

-s utilise le shell de connexion indiqué.

```
chsh -s /bin/ksh
```

**cksum** Nombre d'octets et somme de contrôle.

**clear** Efface l'écran.

**cmp** Compare deux fichiers.

-l affiche le rang de chaque octet différent,

-s n'affiche rien, renvoie vrai ou faux.

```
if cmp -s fichier1 fichier2; then ...
```

**col** Élimine les retours et sauts-de-ligne en arrière.

-b enlève tous les retours en arrière.

```
man col | col -b > col.man.txt
```

**compress** Compression simple de fichier.

**cp** Copie de fichiers.

-R copie récursive des sous-répertoires,

-p garde horodatage, propriétaire, permissions,

-d copie les liens symboliques en tant que tels.

```
cp fichier fichier.sauve
```

```
cp -Rdp fichier_* /autre/repertoire
```

**crontab** Édite le fichier crontab personnel.

-l affiche le contenu actuel,

-e édite le fichier crontab,

-r supprime le fichier crontab.

**csh** Shell C.

**csplit** Découpe un fichier suivant des lignes de contexte.

-f préfixe pour nommer les nouveaux fichiers.

```
csplit -f prefixe fichier '/^$/' {*}
crée prefixe00 prefixe01... en découpant le fichier à
chaque ligne vierge
```

**cut** Supprime une partie de chaque ligne.

-b affiche les caractères indiqués,

-f affiche les champs indiqués,

-d caractère séparateur de champ

```
ls -l | cut -b 20-28
```

affiche uniquement les caractères 20 à 28, c'est-à-dire le groupe des fichiers.

**date** Affiche la date et l'heure du système.

-d indique la date à afficher

+ chaîne de format pour l'affichage.

```
date +"Date = %D, Heure = %X"
```

```
date -d 20030401 +"%A"
```

affiche "mardi" (1er avril 2003).

**dd** Copie générique et conversion de fichiers.

if= nom du fichier d'entrée,

of= nom du fichier de sortie,

bs= taille des blocs à copier,

count= nombre maximal de blocs à copier,

skip= position de début de lecture,

seek= position de début d'écriture.

```
dd if=/dev/fd0 of=disquette.img bs=1024
count=1440
```

**df** Place occupée sur les systèmes de fichiers.

-k affiche les tailles en kilo-octets,

-P affiche une ligne d'en-tête.

```
df -k /tmp/sauvegarde
```

**diff** Trouve les différences entre des fichiers.

- i ignore les différences majuscule/minuscule,
- b ignore les différences d'espaces blancs,
- u utilise un format compatible avec patch,
- r étudie récursivement les sous-répertoires.

```
| diff -u original copie > modifs.patch
```

**dirname** Affiche le répertoire d'un chemin d'accès.

```
| $ dirname /usr/src/linux/signal.c
| /usr/src/linux
| $
```

**dos2unix** Conversion de texte du format Dos vers Unix.

**du** Statistiques sur l'utilisation du disque.

- a affiche les statistiques pour les fichiers,
- s affiche seulement le total,
- x ignore les sous-répertoires sur autre système de fichiers,
- k affiche les tailles en kilo-octets.

**echo** Affiche une ligne de texte.

```
-e interprète les caractères symboliques,
-n évite le saut-de-ligne final.
| echo "Message d'avertissement" >&2
| echo -n "Votre choix :"
| echo -e "\r effectué : " $i "%"
| echo -e "\007"
```

**ed** Éditeur ligne-à-ligne

**egrep** Synonyme de `grep -E`

**emacs** Éditeur Gnu pleine page

Version X-Window : `xemacs`.

**env** Lance un programme avec environnement modifié.

- i Démarre dans un environnement vide.

```
| env
| affiche l'environnement en cours
| env -i /bin/sh
| démarre le shell dans un environnement neuf.
```

**expand** Convertit les tabulations en espaces.

- t largeur de tabulation désirée,
- i uniquement les tabulations en début de ligne.

```
| expand -i < script.sh > listing.txt
```

**export** Passe une variable dans l'environnement du shell

```
| export REP_APPLI=/usr/local/lib/appli/
| VERSION_APPLI=1.5
| export VERSION_APPLI
```

**expr** Évalue des expressions.

```
| expr 4 "*" 3 + 2 affiche 14
| (les guillemets protègent l'étoile par rapport au shell)
```

**false** Échoue en ne faisant rien.

```
| until false; do ...
```

**fc** Édite la dernière ligne de l'historique avec l'éditeur mentionné dans la variable d'environnement `FCEDIT`.

**fg** Ramène un job à l'avant-plan.

**fgrep** Synonyme de `grep -F`

**file** Affiche le type d'un fichier

**find** Recherche des fichiers dans une arborescence.

```
-name motif recherche sur le nom du fichier,
-regex expr recherche sur le nom complet,
-atime n dernier accès il y a n jours,
-ctime n dernière modif. de l'état du fichier,
-mtime n dernière modif. du contenu du fichier,
-perm mode autorisations d'accès au fichier,
-size n taille du fichier (en blocs),
-type t type du fichier,
-print affiche les noms des fichiers trouvés,
-exec ... \{ \} \; exécute l'action indiquée en remplaçant
\{ \} par le nom du fichier,
-ok ... \{ \} \; ; exec avec confirmation.
| find /tmp -ctime +30 -ok rm \{ \} \;
| find /home -name core -exec rm \{ \} \;
```

**fold** Coupe les lignes d'un fichier à une largeur donnée.

**ftp** Transfert de fichiers entre machines.

**fuser** Identifie les processus utilisant un fichier.

```
-k leur envoie le signal SIGKILL,
-i confirmation avant d'envoyer le signal,
-m tous processus accédant au système de fichiers.
| fuser -k /mnt/cdrom
```

**grep** Affiche les lignes correspondant à un motif.

```
-E le motif est une expression rationnelle étendue,
-F le motif est une chaîne pas une expression,
-i ignore différences majuscules/minuscules,
-v affiche les lignes ne correspondant pas,
-l affiche seulement le nom des fichiers.
```

```
| grep -i "MoTiF" fichiers_*
| grep -v "absent" fichier
```

**groups** Affiche les groupes d'un utilisateur.

**gunzip** Décompresse un fichier `.gz`.

**gzip** Comprime un fichier.

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

```
-c n affiche les n premiers octets,
-n n affiche les n premières lignes.
```

**hostid** Affiche l'identifiant de la machine

**hostname** Affiche le nom de la machine

**iconv** Convertit des textes d'un jeu de caractères vers un autre

```
| iconv -f LATIN1 -t UTF8 < fichier
```

**id** Affiche les UIDs et GIDs effectifs et réels.

```
-u affiche seulement l'UID,
-g affiche seulement le GID,
-r affiche les identifiants réels.
| if [ $(id -u) == 0 ]; then...
```

**jobs** Affiche la liste des jobs en cours.

**join** Fusionne les lignes de deux fichiers triés.

```
| join fichier_1 fichier_2 > fichier_3
```

**kill** Envoie un signal à un processus.

```
-numéro le signal dont le numéro est indiqué,
-l affiche la ligne des signaux disponibles.
| kill -9 30582
```

**killall** Envoie un signal aux processus de même nom.

```
-i demande confirmation individuellement,
-l affiche la liste des signaux disponibles.
| killall xterm
```

**ksh** Shell Korn

**less** Affiche un fichier page-par-page.  
(alternative libre et puissante à `more`.)

**lex** Générateur d'analyseur lexical

**ln** Crée des liens entre fichiers.

- f force l'écrasement du fichier s'il existe,
- s crée un lien symbolique.

```
| ln -sf appli-1.4.sh appli
```

**logger** Journalise un message système.

**login** Relance une connexion sur le système.

**logname** Nom de connexion de l'utilisateur.

**lp** Requête d'impression

```
-d sélection de l'imprimante
-n nombre de copies
```

```
| pr -l 66 appli.c | lp -d listing
```

**ls** Affiche l'état des fichiers et le contenu des répertoires.

- a aussi les fichiers commençant par un point,
- d noms des répertoires, pas leur contenu,
- i affiche les numéros d'i-nœud,
- l utilise un format d'affichage long,
- R affiche récursivement les sous-répertoires.

```
| ls -al /home/usera
| ls -lR /var/ftp/pub > /var/ftp/liste-
| fichiers.txt
```