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).

apropos 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 Compresse 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 propiétaire
```
chmod 755 script
```
lecture et exécution pour tous, écriture pour propiétaire
```
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 entres 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 Compresse 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
```

Aide-mémoire des commandes Unix de M à Z

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`)

make Construction d'application, avec gestion des dépendances.

man Affiche une page du manuel Unix.
- ***numéro*** recherche dans la section indiquée,
- **-a** affiche toutes les pages correspondant,
- **-t** écrit la page Postscript sur la sortie standard,
- **-k** équivalent à la commande `apropos`.

```
man 1 c
man -k socket
```

md5sum Calcule et affiche un compte-rendu MD5.

mkdir Crée des répertoires.
- **-p** crée récursivement les répertoires parents,
- **-m *mode*** fixe les autorisations d'accès.

```
mkdir -p /var/lib/new-app/font/big
```

mkfifo Crée des FIFOs (tubes nommés).
- **-m *mode*** fixe les autorisations d'accès.

```
mkfifo -m 666 /tmp/fifo_serveur
```

mknod Crée des fichiers spéciaux.
- **b** ou **c** fichier spécial bloc ou caractère

```
mknod /dev/hda1 b 3 1
```
crée le noeud de numéros majeur/mineur 3/1.

more Consulte un fichier page par page (voir `less`)

mv Déplace ou renomme des fichiers.
- **-f** force l'écrasement du fichier destination.

```
for i in *.JPG; do mv $i ${i%JPG}jpg;
 done
```
renomme tous les fichiers .JPG en .jpg

nice Exécute un programme avec une courtoisie d'ordonnancement modifiée.
- **-n *valeur*** augmente la courtoisie de la valeur.

nl Numérote les lignes d'un fichier.
- **-f a** numérote aussi les lignes vides.

nohup Exécute un programme en le rendant insensible aux déconnexions.

```
$ nohup ~/bin/calcul &
[1] 17300
$ exit
```

od Affiche le contenu d'un fichier en octal ou sous d'autres formats.
- **-c** affiche les caractères imprimables en Ascii,
- **-x** affiche les codes hexadécimaux.

passwd Change le mot de passe.

```
$ passwd
# passwd utilisateur
```

patch Applique une série de modifications à un fichier.
- **-p*n*** enlève *n* répertoires au début des noms de fichiers

```
$ patch –p1 ../new_version/patch_1
```

pathchk Vérifie la validité d'un nom de fichier.
- **-p** vérification stricte de la portabilité.

```
if pathchk "$rep/$fic" ; then...
```

perl Interpréteur du langage Perl.
- **-w** mode de vérification stricte

ping Test de liaison entre machines.
- **-c** nombre de tentatives
- **-w** délai maximal en secondes

pr Prépare des fichiers de texte pour l'impression.
- **-h *texte*** indique l'en-tête de chaque page,
- **-l *n*** affiche n lignes par pages,
- **-t** supprime les en-têtes et pieds de pages.

printf Affiche des données numériques formatées.

```
printf "%05d %4.2f" $x $y
```

ps Affiche l'état des processus en cours.
- **-ax** tous les processus (BSD)
- **-u** informations complètes (BSD)
- **-e** tous les processus (SysV)
- **-f** informations complètes (SysV)
- **-w** lignes larges.

```
ps -aux   (BSD)
ps -ef    (SysV)
```

pwd Affiche le nom du répertoire de travail.

quota Affiche les quotas d'utilisation du disque.

rcp Copîe de fichiers entre systèmes différents.

renice Modifie la priorité d'un processus en cours.

```
renice +20 14210
```

rev Inverse les lignes d'un fichier (voir aussi `tac`).

rlogin Connexion sur un système distant
(préférer `ssh`)

rm Efface des fichiers.
- **-f** pas de confirmation,
- **-i** confirmation avant chaque effacement,
- **-r** efface récursivement les sous-répertoires.

```
rm -rf /home/usera/tmp
```

rmdir Suppression de répertoires vides.

rsh Exécution de commande sur système distant.
(préférer `ssh`).

script Enregistre une session de travail .
- **-a *fic*** ajoute le résultat dans le fichier.

sed Éditeur non-interactif.
- **-e "..."** commandes fournies sur la ligne,
- **-f *fic*** commandes dans un fichier,
- **-n** supprime l'affichage des lignes traitées.

Commandes essentielles de Sed :
- **p** affiche la ligne sélectionnée
- **d** ignore la ligne sélectionnée
- **n** affiche la ligne et passe à la suivante
- **s** recherche un motif et le remplace

```
sed -ne '1,/^$/p' < mail.txt
```
extrait l'en-tête d'un mail.

sh Shell Bourne.

sleep Attend une durée déterminée.

```
sleep 14
```
(14 secondes)

```
sleep 3m
```
(3 minutes)

sort Trie les lignes d'un fichier texte.
- **-b** ignorer les blancs en début de ligne,
- **-f** ignorer les différences majuscules/minuscules,
- **-r** inverser l'ordre du tri.

split Découpe un fichier en différentes partie.
- **-l *n*** en fichiers de *n* lignes,
- **-b *n*** en fichiers de taille *n*.

```
split -b 1440k gros_fichier disquette_
```

ssh Connexion sécurisée sur un système distant.

```
ssh user@hostname
ssh –l user hostname /usr/bin/commande
```

strings Cherche les chaînes Ascii dans un fichier

stty Configuration du terminal.
- **-a** affiche la configuration en cours,
- **sane** revient en configuration normale,
- **-echo** pas d'écho des caractères frappés,
- **-icanon min 0 time 1**
  lecture des caractères à la volée.

su Exécute un shell avec un UID et un GID différents.
- exécute un shell de login.

sum Somme de contrôle, et nombre de blocs.

tac Concatène et affiche des fichiers à l'envers.

tail Affiche la fin d'un fichier.
-*num* affiche le nombre de lignes indiqué,
-f affiche en continu les modifications.
```
tail -f /var/log/messages
```

tar Utilitaire de gestion d'archives.
-f *fic* nom de l'archive,
-c crée une archive,
-t affiche le contenu d'une archive,
-x extrait le contenu d'une archive,
-z invoque Gnu gzip pour les (dé)compressions,
-j invoque bzip2 pour les (dé)compressions,
-v mode volubile.
```
tar -czf save.tar.gz /home/usera/*
tar -xzf appli-1.10.tgz
```

tcl Interpréteur du langage Tcl (voir aussi `wish`).

tee Copie entrée sur sortie standard et dans un fichier.
-a ajout en fin de fichier sans écrasement.
```
...| tee hublot_1.log | ...
```

telnet Connexion sur un système distant.
(préférer `ssh`)

test Type d'un fichier, ou comparaison de valeurs.
Synonyme de la commande shell `[...]`.
```
if test –f $fichier ; then
```

time Chronométre une commande simple.

touch Modifie l'horodatage d'un fichier.
-t *MMJJhhmm* utilise l'horodatage indiqué,
-r *fichier* utilise l'horodatage du fichier,

tr Transpose ou élimine des caractères.
```
tr ‘àâäçéèêîïôöùû’ ‘aaaceeeiioouu’|...
```

true Réussit à ne rien faire...
```
while true; do ...
```

tty Affiche le nom du terminal de l'entrée standard.

uname Affiche des informations sur le système.
-m type de matériel,
-n nom d'hôte,
-r version du système,
-s système d'exploitation,
-a toutes les informations.

uncompress Décompression de fichier `.Z`.

unexpand Convertit les espaces en tabulation.
(voir aussi `expand`)

uniq Ôte les lignes dupliquées d'un fichier trié.
-u n'affiche que les lignes uniques,
-d n'affiche que les lignes dupliquées,
-c affiche le nombre d'occurrences des lignes.

unix2dos Conversion de textes du format Unix au format Dos.

unzip Décompresse un fichier `.zip`.

uptime Temps fonctionnement et charge système.

users Nom des utilisateurs connectés.

uudecode Décode un fichier `.uu`.

uuencode Code un fichier binaire en Ascii.

Vi Éditeur interactif
Version X-Window : `gvim`

wait Attend la fin d'un processus.
```
$ ./commande &
[1] 2927    ./commande
$ wait 2927
[1]+ Done   ./commande
$
```

wc Nombres de caractères, mots et lignes d'un fichier.

whereis Recherche les fichiers exécutables, les sources et les pages de manuel d'une commande.

which Affiche le chemin d'accès des commandes.

who Montre qui est connecté.

whoami Affiche notre UID effectif.

wish Interpréteur Tcl avec bibliothèque Tk.

xargs Construit et exécute une ligne de commande.
```
find . -name "*.c" | xargs grep "init()"
```

yacc Générateur d'analyseur syntaxique.

yes Affiche indéfiniment une chaîne
(par défaut ‘y’)
```
yes | rm -r /var/old-backup/
```

zcat Affiche le contenu d'un fichier compressé.

zip Compresse un fichier.

Expressions rationnelles

`grep`, `sed`, `find –regex`
utilisent des expressions rationnelles *simples*.
`grep –e`, `awk`, `perl`
utilisent des expressions rationnelles *étendues*.

Éléments communs

**** supprime la signification des caractères spéciaux,
```
prix=25\$
```
. remplace n'importe quel caractère,
```
g.n.rique
```
^ représente le début de chaîne,
$ représente la fin de chaîne,
```
^$      (ligne vide)
```
***** indique zéro, une ou plusieurs occurrences,
[ ] représente une liste, un intervalle ou une classe,
```
[eéêèë]
[0-9]
[[:upper:]]
```
i contenu du $i^{ème}$ regroupement entre parenthèses.

Classes de caractères

alpha caractères alphabétiques,
digit chiffres décimaux,
xdigit chiffres hexadécimaux,
alnum caractères alphanumériques,
lower minuscules,
upper majuscules,
blank caractères blancs,
space caractères séparateurs,
punct signes de ponctuation,
graph symboles visibles,
print symboles visibles ou blancs,
cntrl caractères de contrôle d'impression.

Éléments des expressions rationnelles étendues

| représente une alternative,
```
Y|y
```
+ réclame une ou plusieurs occurrences,
? réclame zéro ou une occurrence,
```
[+-]?[[:digit:]]+
```
{ } réclament un certain nombre de répétitions,
( ) regroupent des éléments.

Équivalences pour les expressions rationnelles simples

\| correspond au **|** des expressions étendues,
\+ correspond au **+** des expressions étendues,
\? correspond au **?** des expressions étendues,
\{ \} correspondent aux **{}** des expressions étendues,
\(\) correspondent aux **()** des expressions étendues.

Aide-mémoire administrateur Linux

*Ce document rappelle les commandes et les options les plus utilisées par l'administrateur d'un système Linux. Il s'agit surtout de commandes assez générales, dont tout administrateur aura besoin un jour ou l'autre. Les commandes pour l'utilisateur courant se trouvent dans un autre aide-mémoire. Pour plus d'informations, on consultera les pages de manuel ou les publications du Linux Documentation Project (*http://www.tldp.org*)*

Informations système

uname – Identification du système.
- **-a** : toutes les informations.

dmesg – Messages du noyau (et ceux du boot).
uptime – Durée et charge du système.
free – Occupation de la mémoire.
vmstat – Détails sur l'utilisation de la mémoire.
ipcs – Utilisation des ressources IPC System V.
ipcrm – Suppression de ressources IPC System V.
ldconfig – Valider les bibliothèques dynamiques.
init – Changement de niveau de fonctionnement :
- **0** : arrêt.
- **1** : mono-utilisateur,
- **3** : multi-utilisateurs mode texte,
- **5** : multi-utilisateurs mode graphique,
- **6** : redémarrer.

Utilisateurs

useradd – Ajout d'un utilisateur :
```
useradd -m -p "" linus
```
crée un compte *linus*, avec répertoire personnel et mot de passe vide.

userdel – Suppression d'un compte utilisateur :
```
userdel -r linus
```
supprime le compte et le contenu de son répertoire.

passwd – Modification d'un mot de passe :
```
passwd linus
```

Partitions et systèmes de fichiers

fdisk – Édition de la table des partitions :
```
fdisk /dev/hda
```

mkswap – Création d'une zone de swap :
```
mkswap /dev/hda2
mkswap /boot/swap_file
```

swapon – Activation d'une zone de swap :
```
swapon /dev/hda2
```
-a active toutes les zones de swap de `/etc/fstab`.

swapoff – Désactivation d'une zone de swap :
```
swapoff /dev/hda2
```

mkfs – Création d'un système de fichiers :
```
mkfs.ext2 /dev/hda3
mkfs.ext3 /dev/hda4
mkfs.vfat /dev/hda5
```

fsck – Vérification d'un système de fichiers :
```
fsck.ext2 -p /dev/hda3
```
réparation automatique d'un système ext2 / ext3,
```
fsck.vfat /dev/hda4
```
vérification d'une partition Windows.

mount – Insertion de partition dans le système :
```
mount -t vfat /dev/hda4 /mnt/dos/
```
monter une partition Windows,
```
mount -a
```
monter toutes les partitions de `/etc/fstab`,
```
mount 192.1.1.254:/home /home/users/
```
Montage d'un répertoire distant par NFS.

Options avec `-o` ou dans `/etc/fstab` :
- **default** : rw,suid,dev,exec,auto,nouser,async,
- **remount** : changer les attributs d'un système monté,
- **rw** : lecture écriture,
- **ro** : lecture seule,
- **noauto** : ne pas monter automatiquement avec -a,
- **nodev** : interdire les fichiers spéciaux,
- **noexec** : pas de fichiers exécutables,
- **nosuid** ; ignorer les bits Set-UID/GID,
- **sync** : écriture synchrones,
- **user** : peut être monté par un utilisateur.

Types de systèmes de fichiers courants :
minix, ext2, ext3, msdos, vfat, proc, iso9660, smb.

umount – Démontage d'un système de fichiers :
- **-a** : démonte tous les systèmes dans `/etc/mtab`.

```
umount /dev/hda4
umount /mnt/dos
umount -a
```

df – Occupation des systèmes de fichiers montés.

Distribution / installation de logiciel

tar – Gestion d'archives :
- **-c** : création d'archive,
- **-x** : extraction d'archive,
- **-t** : consultation d'archive,
- **-f** : nom du fichier archive,
- **-v** : mode volubile,
- **-z** : (dé)compresser avec g(un)zip,
- **-j** : (dé)compresser avec b(un)zip2.

```
tar -czf archive.tar.gz distrib/
```
crée une archive compressée du répertoire `distrib/`,
```
tar -tvf archive.tar
```
liste le contenu de l'archive,
```
tar -xjf archive.tar.bz2
```
extrait le contenu d'une archive compressée.

installation classique
```
tar -xzf application-1.01.tar.gz
cd application-1.01
./configure
make && make install
```

rpm – Gestion des paquetages RedHat :
- **-h** affichage de la progression du travail.

```
rpm -ivh paquet.rpm
```
installation d'un paquetage,
```
rpm -Uvh paquet.rpm
```
mise à jour / installation d'un paquetage,
```
rpm -Fvh paquet.rpm
```
mise à jour d'un paquetage déjà installé,
```
rpm -e paquet
```
désinstallation d'un paquetage,
```
rpm -qa
```
liste de tous les paquetages installés,
```
rpm -qf /chemin/fichier
```
recherche du paquetage auquel appartient le fichier,
```
rpm -qip paquet.rpm
```
informations sur un paquetage,
```
rpm -qlp paquet.rpm
```
liste des fichiers contenus dans le paquetage.

apt – Gestion des paquetages Debian :
```
apt-get install application
```
installation de l'application et ressources éventuelles,
```
apt-get remove application
```
suppression application et dépendances éventuelles,
```
apt-get update
```
mise à jour de la base de données interne,
```
apt-get upgrade
```
mise à jour du système.

Gestion des processus

`application &`
lance l'application à l'arrière-plan,
`fg 1`
ramène à l'avant-plan le job numéro 1,
`(Ctrl-Z)`
endort l'application à l'avant-plan,
`bg`
relance à l'arrière-plan un job endormi.

ps – État des processus :
`ps -ef`
ou
`ps -aux`
affichage long de tous les processus du système.

top – Affichage continu des processus du système.
-d délai de rafraîchissement.

renice – Changer la courtoisie d'un processus :
`renice +5 12857`
augmente la courtoisie du processus 12857 de 5 unités,
`renice -5 -u root`
diminue de 5 la courtoisie de tous les processus de root.

kill – Envoyer un signal à un processus :
`kill -15 12857`
-l (lettre l) : liste des signaux disponibles.

killall – Tuer tous les processus du même nom :
`killall -9 boucle_fork`

fuser – Liste des processus accédant à un fichier :
`fuser -k -m /dev/hda5`
tue tous les processus accédant à la partition indiquée.

Utilitaires réseau

ifconfig – Configuration des interfaces réseau :
`ifconfig -a`
affiche la configuration de toutes les interfaces réseau,
`ifconfig eth0 192.1.1.50`
configure la première interface ethernet.

route – Gestion de la table de routage du noyau :
`route add -net 192.1.1.0 eth0`
ajoute une route statique via l'interface `eth0`,
`route add -net 172.1.1.0 gw 192.1.1.5`
ajoute un réseau accessible par une passerelle,
`route add default eth1`
ajoute une route par défaut,
`route del default`
supprime la route par défaut.

socklist – Liste des sockets actives.

netstat – Statistiques réseau :
`netstat -r`
affiche la table de routage du noyau,
`netstat -i`
affiche l'état des différentes interfaces,
`netstat -a`
affiche l'état des sockets du système.

arp – Gestion de la table ARP du noyau :
-a affiche toutes les entrées dans le cache ARP,
`arp -d hote`
supprime les entrées concernant l'hôte indiqué.

ping – demande d'écho vers d'autres hôtes :
`ping -c 1 -w 2 192.1.1.53`
une seule requête et attend au plus 2 secondes,
`ping -b 192.1.1.255`
requête diffusée en broadcast à tous les hôtes du réseau.

traceroute – Chemin pour joindre un hôte :
`traceroute www.destination.com`
-n ne pas résoudre les adresses numériques en noms.

tcpdump – Examen du trafic réseau :
`tcpdump -i eth0`
affiche tout ce qui circule sur `eth0`,
`tcpdump -i eth0 port telnet`
affiche les message depuis / vers le port 23 (*telnet*).

telnet – Connexion TCP/IP :
`telnet mail.isp.com pop-3`
connexion sur port 110 (*Pop/3*) du serveur de courrier.

rsh – Exécution d'un shell distant.

ssh – Exécution sécurisée d'un shell distant.

ftp – Transferts de fichiers :
Commandes usuelles :
open ftp.serveur.org
cd /chemin/distant/
lcd /chemin/local/
get fichier
put fichier
prompt
mget *.c
mput *.h

wget – Rapatrier le contenu d'une URL :
`wget http://www.site.com/repertoire/`
-c reprendre un transfert déjà entamé,
-r charger récursivement les liens,
-l niveau maximal de récursion,
-k convertir les liens en pointeurs locaux.

Signaux fréquemment utilisés

0 : pseudo signal vérifiant la présence d'un processus,
1 (**SIGHUP**) : fin de connexion,
2 (**SIGINT**, Ctrl-C) : fin immédiate du programme,
3 (**SIGQUIT**, Ctrl-\) : fin immédiate avec fichier core,
9 (**SIGKILL**) : fin obligatoire et immédiate,
15 (**SIGTERM**) : fin normale.

Gestion des modules du noyau

lsmod – Liste des modules chargés.

modinfo – Informations sur un fichier module.

insmod – Insertion d'un module dans le noyau :
`insmod module.o`

rmmod – Suppression d'un modulé chargé :
`rmmod module`

depmod – Vérification des dépendances :
`depmod -an`

modprobe – Chargement gérant les dépendances :
`modprobe module.o`

Compilation d'un noyau Linux

`ftp ftp.kernel.org`
récupérer le noyau désiré (connexion *anonymous*) depuis le répertoire `/pub/linux/kernel/`,

```
tar -xjf linux-XXXX.tar.bz2
cd linux-XXXX
make mrproper
make menuconfig
```

choisir et sauver la configuration désirée, puis
`make dep clean bzImage` (noyau ≤ 2.4)
ou :
`make` (noyau ≥ 2.6)
Puis, sous compte *root* :

```
make modules && make modules_install
cp System.map /boot/System.map-XXXX
cd arch/i386/boot/
cp bzImage /boot/vmlinuz-XXXX
vi /etc/lilo.conf
```

ajouter l'entrée pour le nouveau noyau,

```
/sbin/lilo
/sbin/init 6
```

Aide-mémoire de la programmation shell

Ce document d'accompagnement du stage « Programmation Shell et Langages de Scripts » rappelle les points principaux à retenir concernant la programmation pour shells Bourne et Korn.

Évaluation des expressions

`variable=valeur`
affectation de *variable* avec la *valeur*
Pas d'espace autour du signe égal !

`tableau[rang]=valeur`
affectation d'un *rang* du *tableau* avec la *valeur*.

`${variable}`
remplacé par le contenu de la *variable*,

`${tableau[rang]}`
remplacé par le contenu du *rang* du *tableau*,

`${variable-valeur}`
remplacé par la *valeur* si la *variable* n'est pas définie,

`${variable=valeur}`
affectation de la *variable* si elle n'est pas définie,

`${variable?valeur}`
afficher le *message* et fin du shell si *variable* indéfinie.

`${#variable}`
est remplacé par la longueur du contenu de la *variable*,

`${variable#motif}`
est remplacé par le contenu de la *variable* privé du plus court préfixe correspondant au *motif*,

`${variable%motif}`
est remplacé par le contenu de la *variable* privé du plus court suffixe correspondant au *motif*,

`${variable##motif}` `${variable%%motif}`
suppression du préfixe ou suffixe le plus long possible.

`~utilisateur/`
remplacé par le répertoire personnel de l'*utilisateur*,

`ab{c,d,e}fg`
est développé en `abcfg abdfg abefg`

`$(commande)`
remplacé par la sortie standard de la *commande*,

`$((expression))`
remplacé par le résultat de l'évaluation arithmétique entière de *l'expression*.

Protection des caractères spéciaux

`"$var1 $var2"`
garde la cohésion de la chaîne en remplaçant les variables par leurs valeurs,

`'$var1 $var2'`
garde la chaîne inchangée (pas de remplacement),

`\$var`
le *backslash* protège le `$` qui n'est pas interprété comme caractère spécial (pas de remplacement).

Structures de contrôle

Boucles

```
while cmd_1 ; do
  commandes
done
```
Répète les *commandes* tant que *cmd_1* renvoie vrai (0).

```
until cmd_1 ; do
  commandes
done
```
Répète les *commandes* tant que *cmd_1* renvoie faux.

```
for variable in liste ; do
  commandes
done
```
Répète les *commandes* en remplissant la *variable* avec les éléments successifs de la *liste*.

`break`
sort directement d'une boucle `for`, `while` ou `until`.

`continue`
passe à l'itération suivante de la boucle.

Tests

```
if cmd_1 ; then
  cmd_2
elif cmd_3 ; then
  cmd_4
else
  cmd_5
fi
```
Si *cmd_1* renvoie vrai exécute *cmd_2*. Sinon si *cmd_3* renvoie vrai, exécute *cmd_4*, sinon exécute *cmd_5*.

```
case expression in
            motif_1 ) cmd_1 ;;
  motif_2 | motif_3 ) cmd_2 ;;
                  * ) cmd_3 ;;
esac
```
Si l'*expression* peut correspondre au *motif_1*, exécute *cmd_1*, sinon si elle correspond au *motif_2* ou *motif_3*, exécute *cmd_2*, sinon exécute *cmd_3*.

Fonctions

```
fonction_1 ()
{
    commandes…
}
```
définit la *fonction_1*.

`fonction_1 valeur_1 valeur_2...`
invocation de *fonction_1* ; dans la fonction les arguments sont dans `$1`, `$2`… et leur nombre dans `$#`.

`local variable`
déclare une variable locale à la fonction

`return valeur`
termine la fonction en revoyant la valeur en retour.

Motifs du shell

`*`
n'importe quelle chaîne de caractères (même vide),

`?`
n'importe quel caractère,

`\*  \?  \\`
Caractères `*`, `?`, `\`,

`[liste]`
Caractères `l`, `i`, `s`, `t`, `e`

`[b-e]`
Caractères `b`, `c`, `d`, `e`

`[^liste]`
N'importe quel caractère hors de la liste

Redirections

`commande < fichier`
entrée standard depuis *fichier*,

`commande > fichier`
sortie standard vers *fichier*,

`commande >> fichier`
sortie standard ajoutée en fin de *fichier*,

`commande 2> fichier`
sortie d'erreur vers *fichier*,

`commande 2>> fichier`
sortie d'erreur ajoutée en fin de *fichier*,

`commande 2>&1`
sortie d'erreur identique à sortie standard,

```
commande <<- ETIQUETTE
  lignes à envoyer
  vers l'entrée standard
  de la commande
ETIQUETTE
```
document en ligne envoyé vers l'entrée standard.

Exécution des commandes

Ligne shebang

```
#! /bin/sh
```

en tout début de script.

Pipeline

```
commande | commande | commande
```

sortie standard injectée dans l'entrée de la suivante

Liste de pipelines

```
pipeline ; pipeline
```

(exécutions séquentielles)

```
pipeline & pipeline
```

(exécutions parallèles)

```
pipeline && pipeline
```

(exécutions dépendantes)

```
pipeline || pipeline
```

(exécutions alternatives)

Commandes composées

```
{ liste de pipelines }
```

(regroupement de commandes)

```
( liste de pipelines )
```

(sous-shell)

Commandes internes essentielles

echo

```
echo arguments
```

affiche les arguments séparés par des espaces.

-n supprime le saut de ligne final

-e interprète les séquences spéciales.

read

```
read variables…
```

remplit les *variables* avec les mots successifs de la ligne lue (séparateur : contenu de la variable `IFS`).

Dernière variable reçoit tout ce qui reste. Par défaut, utilise variable `REPLY`. Renvoie faux en fin de fichier.

exec

```
exec commande
```

remplace le (script) shell en cours par la *commande*.

```
exec redirections
```

applique les *redirections* indiquées au shell courant.

source

```
source script
. script
```

interprète le *script* dans le shell en cours.

exit

```
exit valeur
```

termine le (script) shell courant en renvoyant la *valeur*.

test

```
test condition
[ condition ]
```

Laisser des espaces autour des crochets !

Renvoie une valeur vraie ou fausse suivant la condition.

Comparaisons de valeurs numériques :

-eq … égale à …

-ne … différente de …

-lt (**-le**) … inférieure (ou égale) à …

-gt (**-ge**) … supérieure (ou égale) à …

Test sur les chaînes :

-n longueur non nulle

-z longueur nulle.

Comparaisons de chaînes :

=, !=, <, >

Tests sur les fichiers :

-a existence du fichier,

-b périphérique mode bloc,

-c périphérique caractère,

-d répertoire,

-f fichier normal,

-g bit Set-GID validé,

-G appartenant au groupe de l'utilisateur,

-h lien symbolique,

-k bit Sticky validé,

-N modifié depuis la dernière lecture,

-O appartient à l'utilisateur,

-p tube nommé (fifo),

-r peut être lu,

-s taille non-nulle,

-S socket,

-u bit Set-UID validé,

-w peut être écrit,

-x peut être exécuté.

Comparaisons de fichiers :

-ef … même fichier physique que …,

-nt … modifié plus récemment que …,

-ot … modifié plus anciennement que …

Test sur les descripteurs :

-t est un terminal

cd

```
cd repertoire
```

change de répertoire de travail,

cd – revient au répertoire précédent,

cd revient au répertoire de connexion.

pwd

affiche le répertoire de travail en cours.

export

```
export variable
```

Transfère la *variable* du shell dans l'environnement qui sera transmis aux processus fils ultérieurs.

env

affiche le contenu de l'environnement

set

```
set
```

affiche les variables du shell et l'environnement,

```
set options
```

configure des paramètres du shell :

-a exporter toutes les variables

-u refuser les variables indéfinies

-v afficher les lignes de commandes avant exécution

-x afficher les développements avant exécution

unset

```
unset variable
```

efface une *variable*.

getopts

```
while getopts "ab:c" variable ; do
  case $variable in
    a) echo "opt° –a";;
    b) echo "opt° –b, arg. $OPTARG";;
    c) echo "opt° –c";;
    *) echo "opt° invalide"; exit 1;;
  esac
done

shift $((OPTIND – 1))
echo "arguments restants :"
echo "$@"
exit 0
```

Analyse la ligne de commande en fonction d'une liste d'options. Si une option prend un argument ('**:**' après sa lettre dans la liste), il est transmis dans `OPTARG`. Une fois toutes les options lues, le rang du premier argument restant est transmis dans `OPTIND`.

shift

```
shift n
```

décale les arguments en ligne de commande de *n* rangs : `$0` reste inchangé, `$n+1` passe dans `$1`, `$n+2` dans `$2`, etc.