

## 1 Généralités

### Shells

On peut démarrer un nouveau sous-shell en entrant son nom sur une ligne de commande : `bash`, `csh`, `tcsh`, `zsh`.

### Redirections

✓ `> commande1 | commande2`

Passes le résultat de `commande1` à `commande2`.

✓ `> commande > fichier`

Redirige le résultat de `commande` vers `fichier`. Si `fichier` existe déjà, son contenu est écrasé.

✓ `> commande >> fichier`

Redirige le résultat de `commande` vers `fichier`. Si `fichier` existe déjà, son contenu est ajouté à la fin du fichier.

`> commande < fichier`

La `commande` lit le contenu du fichier `fichier`.

## 2 Principales commandes

Les commandes importantes sont signalées par un ✓. Les commandes qui sont spécifiques à un Unix sont également signalées.

Il va de soi que toutes les options des commandes ne sont pas listées ici, pour en savoir plus sur chacune de ces fonctions, il suffit de se référer à leurs pages de `man` respectives.

`> alias [nom=valeur]`

Créé un alias que l'on pourra utiliser à la place de sa valeur. Utilisé sans arguments, la commande renvoie la liste des alias définis. (En `csh`, l'alias se passe sans le signe égal).

`> apropos commande`

Renvoie une description succincte de `commande`.

`> bg [numero_de_tâche]`

Place le processus désigné en tâche de fond.

`> cal [mm aaaa]`

Affiche un calendrier succinct du mois `mm` de l'année `aaaa`. Sans argument, le calendrier retourné est celui du mois en cours.

`> cancel num_job`

Annule l'impression désignée par `num_job`. (HP-UX, Solaris)

`> cat fichier(s)`

Affiche sur la sortie standard le contenu des fichiers réguliers listés.

✓ `> cd [répertoire]`

Vous déplace dans le répertoire donné. La commande `cd ..` vous amène dans le répertoire parent et la saisie de `cd` sans arguments vous ramène sur le racine l'utilisateur.

`> chmod droits fichiers`

Modifie les droits d'accès aux fichiers listés. Deux écritures des droits sont possibles :

– l'écriture symbolique en entrant `chmod [ugoa] [+ -] [rwx] fichier`, le premier paquet de paramètre désignant les utilisateurs (`u` pour utilisateur, `g` pour groupe, `o` pour other, `a` pour all), `+` ou `-` servant respectivement à dire si on rajoute ou si on enlève les droits. Enfin, le dernier paquet décrit les droits : `r` pour lecture, `w` pour écriture, `x` pour exécution.

– en entrant sous forme d'entiers, 1 désignant l'accès en exécution, le 2 l'accès en écriture et le 4 l'accès en lecture. Le premier chiffre sert pour l'utilisateur, le deuxième chiffre pour le groupe et le troisième pour le reste du monde. Ainsi `chmod 644` donne accès en écriture et en lecture pour l'utilisateur et en lecture seule pour le groupe et le reste du monde.

`> chown login fichiers`

Modifie le propriétaire d'un fichier (seul le root peut effectuer une telle opération).

`> clear`

Efface les écritures sur la console ou sur le terminal.

✓ `> cp [-i] fichier1 fichier2/répertoire`

Copie le `fichier1` vers le `fichier2` ou dans le `répertoire` indiqué en deuxième argument. L'option `-i` permet demande confirmation avant d'écraser un fichier portant le même nom que le fichier destination.

`> cut -c[liste [fichier]]`

Extrait des colonnes de données listées. Par exemple `c1,8-14` extrait la première colonne et les colonnes de 8 à 14.

`> date [-s heure]`

Affiche l'heure courant. Avec l'option `[-s]`, l'heure est

modifiée.

> dmesg

Affiche l'ensemble des messages affichés lors du démarrage du noyau.

> df [répertoire]

Affiche l'espace libre sur le volume contenant le répertoire. Si ce répertoire est omis, l'espace libre sur tous les disques du système sont affichés. L'option -k permet d'obtenir un affichage en kilo octets.

> diff fichier1 fichier2

Affiche la liste des lignes qui diffèrent entre les fichiers *fichier1* et *fichier2*.

> du [-sk] [argument]

Renvoie l'espace disque occupé par l'argument (si c'est un fichier, la taille du fichier, si c'est un répertoire, la taille du fichier et de ses sous-répertoires). L'option -s n'affiche que la taille totale occupée par la destination (sans détails). Enfin, l'option -k permet d'afficher le résultat en Ko (le résultat par défaut pouvant être en blocs).

> e2fsck périphérique

Vérifie le système de fichiers Linux sur *périphérique*.

> echo texte/\$variable

Affiche sur la sortie standard le texte placé en argument ou le contenu de la variable *variable*.

> eject périphérique

Détache le périphérique de son attache. (Solaris)

> env

Affiche toutes les variables et leurs valeurs associées dans l'environnement de l'utilisateur.

> exit

Cette commande permet de fermer un terminal, une connexion à distance par *telnet* ou *rlogin*.

> fg [numéro\_de\_tâche]

Place au premier plan la tâche désignée par son numéro.

✓ > find répertoire [critère]

Effectue une recherche à partir du *répertoire* et selon les critères.

L'exemple le plus classique d'appel sera :

*find répertoire -name "nom" -print*

Cette ligne de commande recherche à partir de *répertoire* tous les fichiers dont le nom est *nom*.

> finger login[@hôte]

Affiche les informations relatives à l'utilisateur (connecté sur *hôte* si on utilise *hôte*).

> free

Sous Linux, affiche la quantité de mémoire vive disponible.

✓ > grep chaîne fichier

Recherche dans un ou des fichiers donnés la chaîne passée en arguments. La ligne contenant le texte est affichée sur la sortie standard. Quelques options sont fort utiles :

-i : ignore la casse

-n : affiche le numéro de la ligne où la chaîne a été trouvée

-H : affiche le nom du fichier dans lequel la chaîne a été trouvée

✓ > gunzip fichier.gz

Décompresse le fichier d'archive *fichier.gz*, ce dernier disparaissant au profit du fichier extrait.

> gzip fichier

Comprime le fichier d'archive *fichier* qui est alors remplacé par *fichier.gz*.

> head [-n nb\_lignes] fichier

Affiche les 10 premières lignes (ou *nb\_lignes* si l'option est activée) de *fichier*. *head -num* permet d'avoir les *num* premières lignes du fichier.

> history

Affiche la liste des dernières commandes exécutées précédée d'un numéro. En entrant *!numéro*, on relance cette commande.

> id

Affiche le numéro d'utilisateur et le numéro du groupe de l'utilisateur courant.

> jobs

Affiche la liste des tâches qui s'exécutent à l'arrière plan.

✓ > kill [-9] numéro\_de\_processus

Envoie un signal processus indiqué. L'option -9 « tue » ce même processus.

> less fichier

Affiche le contenu de *fichier* sur la sortie standard avec la possibilité d'aller en avant et de revenir en arrière (à l'aide des touches de direction) dans la consultation.

> ln [-s] source destination

Créé un lien (symbolique avec l'option -s) entre la destination et la source.

> logout

Permet à un utilisateur connecté sur une console de se déconnecter.

> lp fichier [-d imprimante]

Impression de *fichier* sur *imprimante* (HP-UX et Solaris).

> lpq [-P imprimante]

Affiche l'état de la file d'attente de *imprimante*. (BSD, SunOS, Linux)

> `lpr fichier [-P imprimante]`

Impression de *fichier* sur *imprimante*. (BSD, SunOS, Linux)

> `lprm [-P imprimante] num_job`

Annule l'impression désignée par *num\_job* sur l'imprimante *imprimante*. (BSD, SunOS, Linux)

> `lpstat imprimante`

Affiche l'état de la file d'attente de *imprimante*. (HP-UX, Solaris)

✓ > `ls [-alF] repertoire`

Liste les fichiers contenu dans le répertoire. L'option `-a` affiche tous les fichiers, `-F` fait suivre le nom des répertoires d'un slash et `-l` affiche en format long : sont alors listés les droits d'accès, compteur de références, le propriétaire, le groupe d'appartenance, la taille, la date de dernière modification et le nom du fichier.

> `make`

Exécute le fichier `Makefile` qui se trouve dans le répertoire courant dont le but est de compiler et/ou de mettre à jour un programme.

✓ > `man commande`

Appelle la page d'aide relative à *commande*.

> `more fichier`

Affiche sur la sortie standard le contenu du fichier avec possibilité de progresser par ligne (avec les flèches de direction) ou par écran (avec la barre d'espace).

✓ > `mkdir nom`

Crée un nouveau répertoire dont le nom est donné en argument.

✓ > `mount [options] peripherique repertoire`

Monte le volume désigné par *peripherique* dans *repertoire*. L'option `-t` permet de définir le type de volume monté (`vfat` pour windows, `iso9660` pour un CD-ROM par exemples).

✓ > `mv [-if] source destination`

Déplace la source vers la destination. Si cette destination est de même nature que la source, alors il s'agit d'une opération de renommage, si la destination est un répertoire alors que la source est un fichier, alors il s'agit du déplacement d'un fichier. L'option `-i` permet d'afficher un message si le fichier destination existe déjà, alors que l'option `-f` évite l'affichage d'un tel message.

> `passwd`

Permet de modifier son mot de passe.

> `ping hôte`

Teste le routage et l'existence de l'*hôte*.

✓ > `ps [ax]`

Affiche les processus en cours d'exécution. L'option `a` permet d'afficher tous les processus (pas seulement ceux de l'utilisateur) et l'option `x` tous les processus qui ne disposent pas forcément d'un terminal de contrôle.

Attention, la liste des options est fortement liée à votre plate-forme! (Les options présentées ci-dessus sont relatives à Linux). Sous Solaris, le pendant du `ps ax` est `ps ef`.

> `pwd`

Affiche le chemin absolu du répertoire courant.

✓ > `reboot`

Redémarre la machine (seul le superutilisateur peut exécuter cette commande).

> `rlogin [-l login] hôte`

Établit une connexion par réseau sur *hôte* sous le *hôte*.

✓ > `rm [-i] fichier`

Efface le fichier. L'option `-i` permet de demander vérification avant toute opération.

✓ > `rmdir repertoire`

Efface le répertoire vide listé.

> `rusers [hôte]`

Affiche les noms des utilisateurs associés au nom de machines qui sont connectés sur les machines connectées au même réseau que votre machine. Si *hôte* est présent, la recherche se limite à cet hôte.

> `sort [options] fichier`

Trie le fichier. L'option `-o` permet de rediriger le résultat vers le même fichier. L'option `+n` permet de prendre en compte les caractères lus à la *n*<sup>e</sup> colonne pour le tri.

> `split fichier [fichiers de destination]`

Découpe *fichier* en plusieurs fichiers (*fichiers destination*). L'option `-c sizek` permet de le découper en fichiers de *size* Ko.

> `startx`

Démarre un serveur X Window.

> `su [nom_d'utilisateur]`

Lance un nouveau shell sous l'utilisateur listé (son mot de passe lui est alors demandé). Sans argument, cette commande permet d'accéder au compte superutilisateur.

> `tail [-n nb_lignes] fichier`

Affiche les 10 dernières lignes (ou *nb\_lignes* si l'option `-n nb_lignes` est activée) de *fichier*.

✓ > `tar archive fichiers`

Cette commande permet d'archiver un certains nombres de fichiers listés dans une seule et même archive. Traditionnellement, on utilise :

`tar [-cvf] archive fichiers` pour créer une nouvelle archive contenant tous les fichiers listés.

`tar [-xvf] archive` pour extraire les fichiers contenant l'archive.

Il existe également l'option `"z"` qui permet d'extraire directement les fichiers compactés avec `gzip` (fichier généralement suffixés en `tgz`).

```
> telnet [login] hôte
```

Établit une connexion par réseau sur *hôte* sous le *hôte*.

```
> top
```

Affiche la liste des processus en cours d'exécution ainsi que la consommation en ressources de chacun d'eux.

```
> touch fichier
```

Modifie la date de modification de *fichier*.

```
> tr ch1 ch2 [< source] [> destination]
```

Remplace un caractère de la chaîne *ch1* par le caractère correspondant dans la chaîne *ch2*.

```
✓ > umount répertoire
```

Démonte le périphérique attaché dans *répertoire*.

```
> unalias nom
```

Supprime l'alias correspondant au nom donné.

```
> uname [-a]
```

Affiche des informations relatives au système. L'option *-a* affiche toutes les informations.

```
> unzip fichier
```

Permet d'extraire l'archive zip créée par exemple avec Winzip...

```
> users
```

Affiche la liste de tous les utilisateurs connectés à la machine.

```
✓ > vi
```

Ouvre l'éditeur de texte standard sous Unix.

```
> volcheck
```

Monte les disquettes et les CD-ROM (dans */floppy/floppy0* pour la disquette). (Solaris)

```
> w
```

Affiche qui fait quoi sur le système sur lequel vous travaillez.

```
> wc [-lwc] fichier
```

Cette commande compte les lignes (avec l'option *-l*), les mots (avec l'option *-w*) ou les caractères (avec l'option *-c*) de la source.

```
> which commande
```

Affiche le chemin d'accès à la commande.

```
> who
```

Affiche la liste des personnes qui sont connectées sur le système accompagné de la liste des shells ouverts ainsi que des dates de connexion.

```
> xhost hôte
```

Permet à *hôte* d'afficher des fenêtres X sur votre écran.

### 3 Mécanismes de substitution

On récapitule ci-dessous la liste des métacaractères du shell.

?	Exactement un caractère quelconque
*	Une séquence (qui peut être de longueur nulle) de caractères
[abc]	Un des caractères spécifiés
[a-f]	Un des caractères de l'intervalle spécifié
[!abc]	Un caractère ne figurant pas dans la liste
[^abc]	Un caractère ne figurant pas dans la liste
~	Raccourci vers le répertoire utilisateur
.	Répertoire courant
..	Répertoire parent
abc{d,e,f}	Créé trois chaînes composées par la première chaîne et une des chaînes de l'ensemble.
\$(expr)	Calcule la valeur de l'expression arithmétique
'commande'	Substitue le résultat de la commande
\$var	Substitue le résultat de la variable

## 4 Expressions régulières

Une expression régulière est une manière compacte de représenter une chaîne de caractères.

.	Tout caractère à l'exception de <b>newline</b>
*	Une séquence (qui peut être de longueur nulle) de caractères
^	Début de ligne
\$	Fin de ligne
\<	Début de mot
\>	Fin de mot
[abc]	Un des caractères spécifiés
[!abc]	Un caractère ne figurant pas dans la liste
\	Caractère d'échappement : considère littéralement le symbole suivant

## 5 Variables

### Manipulation des variables

Pour afficher la valeur d'une variable, on utilise la commande :

```
echo $variable
```

Pour affecter une variable à une valeur, on utilise :

```
variable=valeur
```

Si on modifie une variable comme *PATH*, il faut penser à ajouter une nouvelle valeur et non pas à écraser l'ancienne : *variable=\$variable,valeur*

### Principales variables

- *PATH* : contient la liste des répertoires dans lesquels Unix va chercher les exécutables.
- *SHELL* : nom du shell utilisé par l'utilisateur
- *MANPATH* : contient la liste des répertoires dans lesquels Unix va chercher les pages de *man*

© 2000–2001 Lionel PORCHERON

<http://www.multimania.com/lionelp>