

Commandes et paramètres

On travaille dans la console en tapant ce qu'on appelle des **commandes**. Ces dernières étant nombreuses, vous ne pourrez jamais toutes les connaître... et ce n'est pas le but : le but, c'est que vous sachiez vous servir par cœur de la plupart des commandes « courantes » et, pour les moins courantes, que vous soyez capables d'apprendre à vous en servir en lisant leur **manuel d'utilisation**.

Le manuel d'utilisation est la **véritable bible de tous les linuxiens**. Vous verrez rapidement qu'ils ne jurent que par ça. Pourquoi ? Parce que c'est tout simplement un outil de référence, là où l'on peut trouver la réponse à TOUTES ses questions pour peu qu'on sache lire le manuel et qu'on prenne la peine de le faire. Un chapitre entier vous apprendra à lire le manuel : c'est vraiment très important.

Pour information, il est courant de voir un linuxien dire « RTFM » aux débutants qui posent des questions qu'il estime « simples ». RTFM est l'abréviation de « Read the fucking manual! », ce qui veut dire grosso modo « Lis le p*** de manuel ! ». Ce n'est pas vraiment une insulte en fait, mais plutôt une sorte de... soupir d'exaspération. Pour éviter qu'on vous dise ça un jour, je vous conseille donc de bien ouvrir vos oreilles quand je vous expliquerai comment utiliser le manuel. ;-)

Une commande simple

Bon : trêve de bavardages, on va rentrer une commande ! Par exemple, tapez `date` puis appuyez sur la touche Entrée du clavier.

Le résultat devrait ressembler à cela :

```
mateo21@mateo21-desktop:~$ date
lundi 20 septembre 2010, 15:39:51 (UTC+0200)
```

La première ligne contient l'invite de commandes suivie de la commande que j'ai tapée.

La seconde ligne est la réponse de l'ordinateur à cette commande.

Je suppose que vous avez deviné comme des grands ce que l'on vient de faire : on a demandé quelles étaient la date et l'heure !

Vous en voulez encore ? O.K., alors essayons une toute autre commande : tapez `ls`. C'est l'abréviation de « list », qui signifie « **lister les fichiers et dossiers du répertoire actuel** ».

```
mateo21@mateo21-desktop:~$ ls
```

```
Desktop Examples Images
```

Cela signifie que le répertoire actuel est constitué de trois dossiers : `Desktop`, `Examples` et `Images`. En général, le système colore les éléments pour que l'on puisse distinguer facilement les dossiers des fichiers.

Si vous n'avez aucune réponse, c'est que vous êtes dans un dossier qui ne contient ni fichier ni dossier.

Voilà, c'est aussi simple que cela. Une commande est constituée d'un mot et ne contient aucun espace. Dans des cas très simples comme ceux que l'on vient de voir, il suffit juste de taper la commande pour avoir une réponse ; mais dans la quasi-totalité des cas on peut (et parfois on DOIT) rentrer des options, qu'on appelle **paramètres**.

Les paramètres

Les paramètres sont des options que l'on écrit à la suite de la commande. La commande et les paramètres sont séparés par un espace, comme ceci :

```
mateo21@mateo21-desktop:~$ commande parametres
```

Les paramètres peuvent eux-mêmes contenir des espaces, des lettres, des chiffres... un peu de tout, en fait. Il n'y a pas de règle véritable sur la forme des paramètres, mais heureusement les programmeurs ont adopté une sorte de « convention » pour que l'on puisse reconnaître les différents types de paramètres.

Les paramètres courts (une lettre)

Les paramètres les plus courants sont constitués d'une seule lettre précédée d'un tiret. Par exemple :

```
commande -d
```

Si on doit donner plusieurs paramètres, on peut faire comme ceci :

```
commande -d -a -U -h
```

Ou, plus court :

```
commande -daUh
```

Attention à la casse des paramètres (majuscules / minuscules) ! Si vous écrivez `-u`,

cela n'a en général pas du tout le même sens que -U !

Faisons un essai avec la commande `ls` et rajoutons-lui le paramètre « a » (en minuscule) :

```
mateo21@mateo21-desktop:~$ ls -a
.          .gconfd          .mozilla-thunderbird
..         .gimp-2.2        .nautilus
.bash_history .gksu.lock       .profile
.bash_logout .gnome           .recently-used
.bashrc     .gnome2          .recently-used.xbel
.config     .gnome2_private .ssh
Desktop    .gstreamer-0.10 .sudo_as_admin_successful
.dmrc      .gtkrc-1.2-gnome2 .themes
.esd_auth  .ICEauthority    .thumbnails
.evolution .icons           .Trash
Examples   .lessht         tutos
.face      .local          .update-manager-core
.fontconfig .macromedia     .update-notifier
.gaim      .metacity       .Xauthority
.gconf     .mozilla        .xsession-errors
```

Cela affiche tout le contenu du dossier, même les fichiers cachés.

Un « fichier caché » sous Linux est un fichier qui commence par un point. Normalement, si vous vous trouvez dans votre répertoire home, vous devriez avoir une bonne flopée de fichiers cachés. Ce sont en général des fichiers de configuration de programmes.

Les paramètres longs (plusieurs lettres)

Les paramètres constitués de plusieurs lettres sont précédés de deux tirets, comme ceci :

```
commande --parametre
```

Cette fois, pas le choix : si vous voulez mettre plusieurs paramètres longs, il faudra ajouter un espace entre chacun d'eux :

```
commande --parametre1 --parametre2
```

On peut aussi combiner les paramètres longs et les paramètres courts dans une

commande :

```
commande -daUh --autreparametre
```

Il y a parfois deux écritures possibles pour un paramètre de commande : une version courte et une version longue. Cela permet de vous laisser le choix selon que vous préférez l'une ou l'autre.

Notez que c'est la commande qui décide des paramètres qu'elle accepte : il arrive parfois que certaines ne proposent pas le choix entre une version courte et une longue.

Testons cela sur la commande `ls` avec le paramètre `--all`, qui signifie « tout » en anglais :

```
mateo21@mateo21-desktop:~$ ls --all
.          .gconfd          .mozilla-thunderbird
..         .gimp-2.2        .nautilus
.bash_history .gksu.lock       .profile
.bash_logout .gnome           .recently-used
.bashrc     .gnome2          .recently-used.xbel
.config     .gnome2_private .ssh
Desktop    .gstreamer-0.10 .sudo_as_admin_successful
.dmrc      .gtkrc-1.2-gnome2 .themes
.esd_auth  .ICEauthority    .thumbnails
.evolution .icons           .Trash
Examples   .lessgst         tutos
.face      .local           .update-manager-core
.fontconfig .macromedia      .update-notifier
.gaim      .metacity        .Xauthority
.gconf     .mozilla         .xsession-errors
```

Comme vous le voyez, `--all` est un synonyme de `-a`. Ceci illustre ce que je vous disais à l'instant, comme quoi parfois une commande propose deux façons d'utiliser un paramètre : une courte et une longue.

Les valeurs des paramètres

Certains paramètres nécessitent que vous les complétiez avec une valeur. Cela fonctionne différemment selon que vous travaillez avec un paramètre long ou avec un paramètre court.

Avec un paramètre court :

```
commande -p 14
```

... cela indique que l'on associe la valeur 14 au paramètre p. Avec ce genre de technique, on peut par exemple faire comprendre à l'ordinateur : « **Je veux voir la liste de tous les fichiers de plus de 14 Mo** ».

Si c'est un paramètre long, on fait en général comme ceci :

```
commande --parametre=14
```

Le résultat sera le même, il est juste plus lisible mais aussi plus long à écrire.

Les autres paramètres

Je vous l'ai dit : il n'y a pas de règle absolue au niveau des paramètres et vous en rencontrerez sûrement qui fonctionnent différemment. Heureusement, les « conventions » que je viens de vous donner sont valables dans la grande majorité des cas, ce qui devrait vous permettre de vous repérer.

Certains paramètres sont donc un peu différents et dépendent vraiment des commandes. Par exemple avec `ls`, si on ajoute le nom d'un dossier (ou sous-dossier), cela affichera le contenu de ce dossier au lieu du contenu du dossier courant :

```
mateo21@mateo21-desktop:~$ ls Examples
Experience ubuntu.ogg      logo-Ubuntu.png          oo-payment-
schedule.ods
fables_01_01_aesop.spx    oo-about-these-files.odt  oo-
presenting-kubuntu.odp
gimp-ubuntu-splash.xcf   oo-about-ubuntu-ru.rtf   oo-
presenting-ubuntu.odp
kubuntu-leaflet.png      oo-cd-cover.odg          oo-trig.xls
logo-Edubuntu.png        oo-derivatives.doc        oo-
welcome.odt
logo-Kubuntu.png         oo-
maxwell.odt              ubuntu Sax.ogg
```