

MODULE 4

Introduction à la console et la commande « man »

Objectifs de ce module :

- ✓ *Utiliser l'invite de commandes (prompt)*
- ✓ *Utiliser les caractères d'échappement.*
- ✓ *Utiliser les trucs et astuces de la console.*
- ✓ *Utiliser la commande man pour la recherche d'informations.*

Table des matières

<i>Sujets</i>	<i>Page</i>
MODULE 4.....	1
Introduction à la console et la commande « man ».....	1
Introduction.....	3
L'invite de commandes (« Prompt » en anglais).....	4
L'interpréteur de commande.....	4
Le “prompt”.....	5
Exercice 1 (Sauvegarde du contenu du prompt courant).....	5
Les caractères de séquence d'échappement.....	6
Exercice 2 (Changer le prompt avec les caractères d'échappement).....	7
La gestion des couleurs.....	8
Exercice 3 (Changement de couleur du « prompt »).....	8
Quelques trucs et astuces.....	10
La répétition des commandes.....	10
L'historique des commandes.....	10
La reprise d'une commande.....	11
La complétion d'une commande.....	11
Le fichier de l'historique des commandes.....	12
Recherche d'information avec la commande « man ».....	13
Syntaxe de man.....	13
Sections du manuel.....	13
Recherche par mot-clés.....	15
Site Web des pages de « man » en français.....	15

Introduction

Dans la plupart des distributions, vous avez la possibilité de vous connecter avec un écran graphique. C'est-à-dire, que le branchement et votre mot de passe seront demandés par l'intermédiaire d'une fenêtre graphique. Cette option est automatiquement configurée si, à l'installation, votre carte vidéo a été reconnue et que le système s'est mis en mode graphique lors de l'installation.

Vous pouvez cependant passer du mode graphique au mode console (non graphique) en utilisant des combinaisons de touches du clavier telle que discuté ci-dessous.

Passer du mode graphique au mode texte

On peut passer au mode texte à partir du mode graphique en effectuant la séquence de touche suivante:

CTRL-ALT-F2 à F6

En effet, il existe 6 écrans consoles indépendants les uns des autres. Par exemple, pour passer à la console texte #2, vous pouvez taper la séquence suivante :

CTRL-ALT-F2

Vous obtenez un écran similaire à celui-ci :

Passer du mode texte au mode graphique

Avec Fedora, on peut passer du mode texte au mode graphique en tapant la séquence de touche suivante:

CTRL-ALT-F1

En effet, il existe 6 écrans graphiques virtuels indépendant les uns des autres. Ceci peut permettre de tester différentes configurations de votre interface graphique sans toutefois être obligé de vous débrancher. Par exemple, pour passer à l'écran graphique #1, vous pouvez taper la séquence suivante :

CTRL-ALT-F1

L'invite de commandes (« Prompt » en anglais)

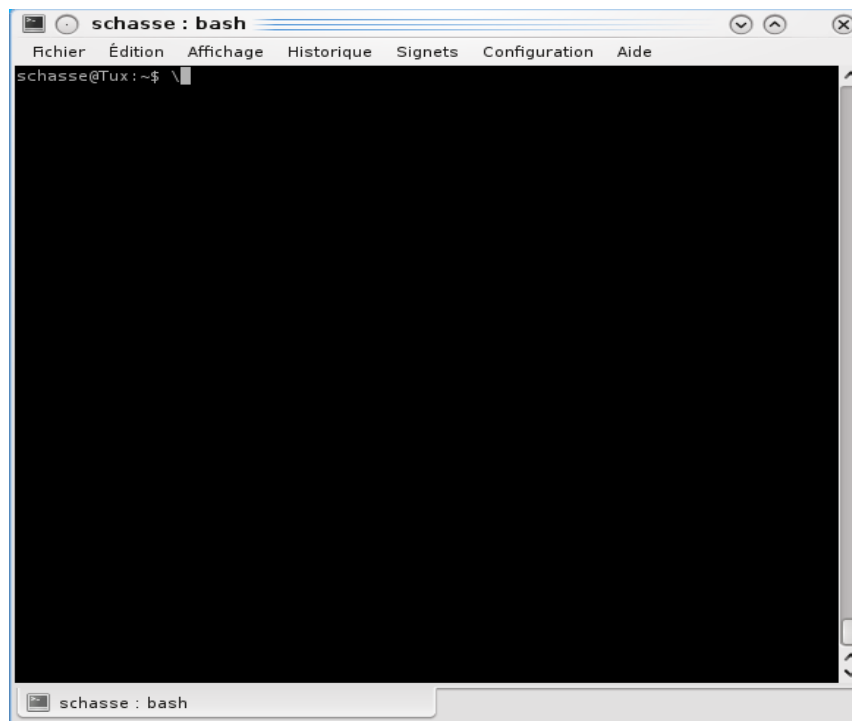
L'interpréteur de commande

Qu'est-ce que le BASH ?

C'est l'interface de commande sur la plupart des systèmes Linux. Le BASH est un descendant du Bourne Shell, Bash est un diminutif de "Bourne Again Shell.". Il est excellent au niveau de l'interactivité et de l'édition de la ligne de commande. Il peut également compléter les commandes et il peut aussi rappeler les dernières commandes exécutées. Il supporte aussi des "prompt" configurables.

Un interface de commande est une sorte de petit programme entre vous et le système d'exploitation et qui permet d'interpréter vos commandes pour les envoyer au système d'exploitation.

La console prend habituellement l'allure suivante:



Les prochains exercices ne sont que quelques possibilités que le "prompt" peut vous offrir. En effet, on peut faire afficher presque n'importe quoi avec le "prompt" comme le numéro de votre terminal, l'heure, la date, les ressources utilisées par le processeur, etc. Le prompt peut aussi utiliser des couleurs pour rehausser l'affichage.

Le “prompt”

L'apparence du “prompt” est gérée par la variable d'environnement nommée “PS1”. Ainsi, pour changer l'allure du “prompt”, on change le contenu de la variable “PS1”.

Afin d'expérimenter un peu avec le prompt, nous allons changer le contenu de la variable “PS1” directement à la ligne de commande.

N'oubliez pas:

Avant de jouer avec le “prompt”, il est important de retenir que si le contenu de la chaîne “PS1” change, le “prompt” changera aussitôt. De façon à ne pas perdre votre “prompt” courant, il est parfois utile de le sauvegarder dans une autre variable avant de changer son contenu.

Exercice 1 (Sauvegarde du contenu du prompt courant)

- Ouvrez une fenê tre console. (application konsole sur KDE)
- Pour sauvegarder le contenu du « prompt » courant, tapez :

```
SAVE=$PS1
```

Ceci permet de sauvegarder dans la variable « SAVE » le contenu du prompt stocké dans PS1.

- Faites afficher le contenu de la variable « SAVE ».

```
Tapez : echo $SAVE
```

Remarquez que vous avez un prompt ressemblant à ceci:

```
[\u@\h \W]\$
```

- Le “prompt” le plus simple serait le signe plus grand que (>). Ainsi, vous pouvez taper:

```
PS1=">"
```

Comme vous pouvez le constater, le « prompt » a changé pour prendre

l'allure d'un signe plus grand que (>).

Vous noterez que le texte de la commande apparaît directement après le "prompt" ce qui rend la lecture plus difficile.

- Ajoutez maintenant un espace à la chaîne contenant le « prompt ».
Tapez ce qui suit :

```
PS1="> "
```

- Pour revenir au prompt original, remettez dans la variable "PS1" le contenu de la variable sauvee précédemment. Ainsi, si je tape:

```
PS1=$SAVE
```

Vous devriez récupérer le "prompt" original . Dans mon cas :
[schasse@Hawking:~\\$](#).

Les caractères de séquence d'échappement

Il existe une multitude de séquence d'échappement pour pouvoir donner une allure différente au prompt. Voici une liste partielle:

Séquence	Description
\a	Le caractère qui sonne la cloche
\d	La date en format "Jour Mois " (Ex: Wed 18 Sep)
\h	Le nom de la machine jusqu'au premier ".". Ex: Si la machine a pour nom: "Trinity.domaine.com" alors ce caractère fera affiché uniquement "Trinity".
\H	Le nom de la machine au complet.
\n	Le caractère de saut de ligne.
\r	La caractère de retour à la ligne (carriage return).
\t	L'heure courante en format 24 hres. HH:MM:SS

Séquence	Description
\@	L'heure courante en format 12 hres am/pm.
\u	Le nom d'utilisateur de l'utilisateur présentement branché.
\w	La version courante du BASH.
\W	Le nom du répertoire courant. Ex.: Si vous êtes dans le répertoire « /home/Travail » alors ce caractère affichera: « Travail » seulement.
\w	Le nom du répertoire plus détaillé. Ex.: Si vous êtes dans le répertoire « /home/Travail » alors ce caractère affichera: « /home/Travail ».
\[Commence une séquence de caractère non-imprimable. Par exemple, une couleur.
\]	Termine une séquence de caractère non-imprimable.

Exercice 2 (Changer le prompt avec les caractères d'échappement)

Tapez la ligne qui commence par « PS1 » ci-dessous :

```
PS1="\u@ \h \W> "
```

Ceci est un prompt que l'on rencontre souvent dans la plupart des distributions.

La commande PS1 précédente permet d'obtenir le prompt suivant:

```
schasse@Hawking ~ >
```

Dans l'exemple précédent :

- \u Fait afficher le nom de l'utilisateur.
- \h Fait afficher le nom de l'ordinateur.
- \W Fait afficher le nom du répertoire courant. (Le tilde est le symbole pour vous indiquer que vous êtes dans le répertoire de votre compte.)

Si vous voulez afficher l'heure en format 24 hres, tapez alors:

```
PS1="\t[\u@\h:\w]> "
```

Ceci donne le prompt suivant: [16:20:49][[schasse@Hawking:~](#)] >

La gestion des couleurs

Comme mentionné précédemment, les couleurs font parties des séquences de caractères "non-imprimables" et doivent être circonscrite par les caractères suivants: `\[\033[et \]`. Pour les couleurs, cette séquence doit également être suivie par la lettre `m`.

Exercice 3 (Changement de couleur du « prompt »)

Tapez ce qui suit :

```
PS1="\[\033[1;34m\][\u@\h:\w]> "
```

Si vous remarquez, la couleur est bel et bien ajoutée mais les commandes que vous tapez sont aussi de cette couleur. Autrement dit, la couleur originale n'est pas rétablie. Pour rétablir le tout, on reprend la couleur "nulle", c'est-à-dire, la valeur 0.

Tapez ce qui suit :

```
PS1="\[\033[1;34m\][\u@\h:\w]>\[\033[0m\] "
```

Quelle est la couleur du prompt ?

Quelle est la couleur des commandes tapées après le prompt ?

Voici les codes de couleur disponible:

Black	0;30	Dark Gray	1;30
Blue	0;34	Light Blue	1;34
Green	0;32	Light Green	1;32
Cyan	0;36	Light Cyan	1;36
Red	0;31	Light Red	1;31
Purple	0;35	Light Purple	1;35
Brown	0;33	Yellow	1;33
Light Gray	0;37	White	1;37

Voici la liste des codes de couleur pour le fond d'écran:

Couleur	Code
Noir	40
Rouge	41
Vert	42
Orange	43
Bleu	44
Magenta	45
Cyan	46
Gris	47

Tapez ce qui suit :

```
PS1="\[\033[44;1;31m\][\u@\h:\w]>"
```

Il existe évidemment plusieurs autres possibilités que vous pouvez explorer notamment en examinant les fichiers et les textes d'aide en ligne. Vous pouvez consulter:

<http://www.tldp.org/HOWTO/Bash-Prompt-HOWTO/>

Quelques trucs et astuces

La répétition des commandes

Il existe plusieurs petits trucs utiles pour vous simplifier la vie lorsque vient le temps de taper les commandes à la console. Par exemple, vous devez taper une commande qui comporte plusieurs paramètres et comble de malheur, vous faites une erreur dans l'orthographe de la commande. Que fait-on ? On retape au complet ?

Il existe un moyen simple de retaper les commandes passées. Il suffit d'utiliser les flèches de curseur haut ou bas pour se déplacer au travers toutes les commandes précédemment tapées.

L'historique des commandes

On peut facilement obtenir l'historique des commandes précédemment entrées en tapant la commande: *"history"*. Voici un exemple:

```
2 kill -9 2223
3 logout
4 cd /etc
5 joe inittab
6 reboot
7 ls -al
8 df
9 ifconfig
10 ping 205.237.246.66
11 ping 205.237.246.4
```

La reprise d'une commande

Une autre série de caractères possible pour reprendre la dernière commande est la séquence “!”. En tapant 2 points d'exclamation d'affilé, on répète la dernière commande tapée. De même, on peut reprendre n'importe quelle commande de l'historique en tapant “! suivi_du_numéro_de_la_commande_dans_l'historique”.

Par exemple, je veux retaper la commande “ping 205.237.246.66” de l'historique précédent, je peux alors tout simplement taper:

```
!!0
```

La complétion d'une commande

Vous pouvez également vous aider à taper les longs noms de fichier ou de répertoire en utilisant la touche “TAB”. Cette touche permet de compléter le paramètre en cours. Par exemple, vous avez la liste de fichier suivante:

```
anaconda2-version2.2.3-i386-rpm.build02032005  
anaconda2-version2.2.4-i386.rpm  
mestravaux/  
calc.java
```

Supposons que vous voulez copier le premier fichier de cette liste sous un autre nom, on peut alors taper:

```
cp ana (suivi de la touche  
TAB)
```

Dans le cas où il y a plusieurs fichiers débutant avec la même série de caractères, vous pourrez alors taper 2 fois la touche « tab » et le système affichera toutes les entrées commençant par cette série de caractères. Vous pourrez alors continuer à taper le texte et la touche « tab » pour permettre de compléter à nouveau.

Le fichier de l'historique des commandes

Ce fichier est habituellement situé dans le compte "maison" de l'utilisateur (son home directory). Le fichier se nomme ".bash_history" par défaut. Il conserve les commandes tapées (les 500 dernières commandes par défaut) depuis le début de l'ouverture de la session.

ATTENTION:

L'historique est sauvegardé dans ce fichier uniquement lorsque l'utilisateur quitte la session. On peut cependant forcer l'écriture de l'historique dans le fichier ou détruire le contenu du fichier avec les paramètres spécifiques telle que montré ci-contre.

Voici l'aide de la commande « history » :

history [-c] [_n]

history -anrw [fichier]

history -p arg [argument ...]

history -s arg [argument ...]

Les options ont les significations suivantes :

- a Ajouter les "nouvelles" lignes d'historiques (celles créées depuis le début de la session actuelle de bash) dans le fichier.
- n Lire dans le fichier, et ajouter à la liste d'historique, les lignes non encore lues. Il s'agit des lignes qui ont été ajoutées dans le fichier depuis le début de la session actuelle de bash.
- r Lire le contenu du fichier, et l'utiliser comme historique courant.
- w Écrire dans le fichier l'historique actuel, en écrasant le contenu précédent.
- c Effacer le fichier d'historique en détruisant toutes ses entrées.

Recherche d'information avec la commande « man »

La commande `man` est l'abréviation de *manual* et sert à obtenir l'aide sur les commandes en console. Il est possible d'obtenir de l'aide sur les commandes console comme sur les fichiers de configuration ou les fonctions C intégrées à Linux.

Syntaxe de man

La syntaxe de base est toute simple:

```
man nom_de_la_commande
```

Par exemple

```
man ls ou man grep ou man yum
```

Sections du manuel

Les pages `man` sont regroupées en sections, chaque section ayant un thème central. Par exemple:

- La section 1 contient l'aide sur les commandes Linux
- La section 2 contient l'aide sur les appels systèmes qui peuvent être réalisés par des programmes C
- La section 3 contient l'aide sur les appels systèmes de Linux en langage C. Ce sont ces fonctions que le système utilise lorsqu'il répond à une de vos commandes.
- La section 4 contient l'aide sur les périphériques et les cartes réseaux
- La section 5 traite des formats des différents fichiers de configuration de Linux

On peut, si désiré, spécifier la section dans laquelle on veut aller fouiller, ainsi:

```
man <section> <item>
```

Ceci nous retournera l'aide sur un item, telle que trouvée dans la section <section>. Par exemple:

```
man 5 crontab
```

La plupart du temps, ce n'est pas nécessaire de spécifier la section. Si on ne le fait pas, man nous retourne la première page trouvée qui correspond à l'item demandé (si on cherche de l'aide sur une commande, qui sera dans la section 1, ce comportement sera idéal pour nous). Il arrive toutefois qu'une commande ait le même nom qu'un fichier de configuration, par exemple. Dans ce cas, si on veut l'aide sur le fichier, il faudra spécifier d'utiliser la section 5 pour ne pas obtenir l'aide sur la commande elle-même mais bien celle sur son fichier de configuration.

On peut découvrir le contenu d'une section en faisant un man sur "intro", dans la section voulue. Par exemple :

```
man 5 intro nous donnera un résumé du contenu de la section 5.
```

Recherche par mot-clés

Il est également utile de savoir que:

- `man -k <mot-clé>` retourne une **liste de pages** ayant ce mot-clé dans leur description.
- `man -f <commande>` retourne une **liste de sections** contenant une page man pour cette commande.

Ces deux usages ont besoin qu'une base de données de mot-clés soit construite au préalable. La commande `makewhatis` est nécessaire pour créer la base de données sur laquelle se basera `man -k` et `man -f`. Cette base de données est déjà créée à l'installation du système. Cependant, si on avait à bâtir la base de données, on pourrait taper la commande suivante :

```
makewhatis
```

Site Web des pages de « man » en français

<http://manpagesfr.free.fr/consulter.html>