

Minima de Linux en cours de C

Eric Berthomier Laurent Signac
eric.berthomier@free.fr signac@sic.sp2mi.univ-poitiers.fr

16 novembre 2008

1 Démarrage

Au démarrage de la machine, une petite ligne indiquant Lilo apparaît. Lilo est un petit programme qui se loge dans le MBR (Master Boot Record) et qui vous permet de choisir sur quelle partition (quel morceau de disque et, de ce fait, quel système d'exploitation) vous allez travailler. Si la ligne boot : n'apparaît pas, appuyez sur la touche Maj, Ctrl ou Alt. Pour accéder aux différents choix proposés appuyez sur Tab. Une fois les choix affichés, tapez votre sélection, dans notre cas, **linux** puis **Entrée** (Return).

2 Connexion

Au moment de se connecter, le message suivant apparaît :

login :

tapez **cours**. Il s'agit de votre nom d'utilisateur, il vous donne certains droits et des accès au système d'exploitation Linux. Dans un environnement de travail réel, chacun aurait son nom d'utilisateur et son mot de passe. Dans le cas de notre initiation, nous utiliserons tous le même login.

passwd :

tapez **emf2000**. Il s'agit du mot de passe associé au compte.

3 Déconnexion

Pour vous déconnecter, fermez toutes vos applications puis tapez **exit** ou **logout**. Répétez cette opération sur toutes les vues utilisées.

Une fois ceci fait vous pouvez rebooter la machine avec Ctrl-Alt-Suppr.

Il vous est impossible d'éteindre la machine, seul le super utilisateur (*le superman de votre machine*) que l'on appelle root peut le faire mais pour ceci il faut un mot de passe qui est « vous ne le saurez pas »...

4 Complétion automatique

Lors de la frappe d'une commande, il vous est possible de demander à linux de compléter votre commande grâce à la touche Tab. Si plusieurs choix sont possibles, il complètera au maximum de ses possibilités et affichera les différents choix possibles.

5 Rappel des commandes

A l'aide des flèches de direction Haut (↑) et bas (↓), il vous est possible de naviguer à travers les différentes commandes précédemment saisies.

6 Exécution

Pour exécuter un programme que vous avez réalisé vous devez faire précéder le nom du programme à exécuter par `./` Pourquoi ? me direz vous puisque `joe` s'exécute tout seul. En résumé un peu erroné, tout ce qui lié au système s'exécute en direct (sans `./`) le reste en a besoin...

7 Consoles

Linux est un système comportant plusieurs consoles, c'est à dire qu'il vous est possible de travailler sur différents écrans virtuels (vous n'avez qu'un seul écran réel!). Pour ce faire appuyez simultanément sur `Ctrl Alt F<no de la vue>` (touches de fonctions F1 à F8). Si `login` : apparaît, loggez vous comme décrit dans la section connexion.

8 man

La commande `man` permet d'afficher une page de manuel.

La syntaxe en est simple `man [commande]` .

Pour quitter une page de manuel appuyez sur la touche `q`. Pour en savoir plus, `man man` évidemment !

9 ls

9.1 Signification

`ls` = list (équivalent dos : dir)

9.2 Syntaxe

`ls [-options] [argument]`

9.3 Utilisation

Permet de connaître le contenu d'un répertoire. Si vous tapez la commande `ls` seule, vous obtiendrez tout le contenu du répertoire courant sur une ou plusieurs lignes sans pouvoir faire la distinction entre les répertoires, les fichiers, les liens...

Nous pouvons donc adjoindre à la commande diverses options. L'option ou la série d'options doit toujours être précédée d'un trait d'union. Elle suit la commande `ls` suivi d'un espace. Une autre possibilité d'ajout des options est de mettre deux traits d'union suivi de la commande en clair. Les options peuvent bien entendu être panachées. Dans ce cas, on commence par un trait d'union et on met les options à la suite. Exemple : `ls -a1B`. Ne perdez pas de vue que, sous Unix/Linux, les minuscules et les majuscules ne sont jamais confondues, ainsi la lettre `a` n'équivaut pas à la lettre `A`. Par conséquent, lors d'un tri alphabétique, vous aurez d'abord le fichiers de `A` à `Z` puis de `a` à `z`, les lettre accentuées venant en fin de voyelle... C'est aussi pourquoi si vous tapez la commande `LS` vous n'obtiendrez qu'un message d'erreur !

9.4 Les options

-a	Affiche tous les types de fichiers y compris les fichiers cachés (.fichier)
-A	Affiche tout sauf les deux répertoires . et ..
-B	N'affiche pas les fichiers finissant par
-c	Affiche et trie suivant la date de dernière modification
-l	Affiche la liste détaillée
-S	Effectue le tri par la taille des fichiers
-t	Effectue le tri par la date de modification
--help	Affiche un tableau des commandes
--version	Affiche le numéro de version

9.5 Les arguments

Ce peut être le nom d'un répertoire (relatif ou absolu). Par exemple, si l'on tape `ls -al /home`, nous obtiendrons le contenu détaillé du répertoire `home` se trouvant à la racine du disque, y compris les fichiers cachés. Si l'on tape `ls ../bin`, la commande remontera d'un niveau par rapport à l'endroit où vous vous trouvez et ira rechercher un répertoire "bin" à ce niveau. Si le répertoire existe, vous en obtiendrez le contenu. S'il n'existe pas, vous obtiendrez un message d'erreur : `sh :bin not found`. Ce peut être aussi le nom d'un fichier ou le début suivi d'un caractère joker... On peut très bien taper la commande `ls -l /usr/bin/k*` afin de lister le contenu du répertoire `/usr/bin` et connaître tous les fichiers dont le nom commence par `k...`

9.6 Pour en savoir plus...

N'hésitez pas à appeler l'aide rapide par `ls --help` ou le manuel en ligne par `man ls` qui vous donnera alors tout le guide d'utilisation de cette commande!

10 rm

10.1 Signification

`rm` = remove (équivalent dos : del)

10.2 Syntaxe

`rm [-options] [argument]`

10.3 Utilisation

Permet de supprimer, des fichiers, des répertoires, tout un disque... Commande très puissante et qui peut, sous certaines conditions, se révéler dangereuse!

10.4 Les options

<code>-d</code>	Supprime le répertoire, même si ce dernier n'est pas vide (root uniquement)
<code>-f</code>	Passer outre les confirmations, force la suppression des fichiers inexistantes ou verrouillés
<code>-i</code>	Mode par défaut, demande confirmation à chaque fichier
<code>-r</code>	Supprime récursivement le contenu des répertoires contenus dans le répertoire à supprimer
<code>-v</code>	Mode "bavard" établit un rapport pour chaque opération effectuée
<code>--help</code>	Affiche ce tableau
<code>--version</code>	Affiche la version et le nom de la commande

10.5 Les arguments

Le nom du ou des fichiers à supprimer. Ce peut être aussi le nom d'un répertoire, d'un groupe de fichiers incluant des caractères joker (* et ?). Exemples : `rm lib*` supprimera tous les fichiers commençant par `lib` contenus dans le répertoire courant. `rm -d /home/toto` supprimera, en tant que `root`, le répertoire `toto` présent dans `/home` y compris tout ce qu'il contient. Exemple de commande dangereuse (voire fatale) : en tant que `root`, si vous tapez la commande `rm -rf /*`, vous supprimerez tout votre disque dur en partant du répertoire racine `/`, sans sommation !

Soyez donc très prudent avec l'utilisation de cette commande !

10.6 Pour en savoir plus...

N'hésitez pas à appeler l'aide rapide par `rm --help` ou le manuel en ligne par `man rm` qui vous donnera alors tout le guide d'utilisation de cette commande !

11 mv

11.1 Signification

`mv` = Déplacer ou renommer des fichiers. (move)

11.2 Syntaxe

```
mv [options] source dest
mv [options] source... répertoire
```

11.3 Utilisation

Si le dernier argument est le nom d'un répertoire existant, `mv` placera tous les autres fichiers à l'intérieur de ce répertoire, en conservant leurs noms.

Sinon, s'il n'y a que deux fichiers indiqués, il déplacera le premier pour remplacer le second.

Une erreur se produit s'il y a plus de deux fichiers indiqués et si le dernier argument n'est pas un répertoire.

11.4 Pour en savoir plus...

On le répètera jamais assez souvent : `man mv`

12 Joe

12.1 Syntaxe

`joe` ou `joe nom_fichier` : si le fichier n'existe pas il sera créé avec le nom spécifié au moment de la sauvegarde de celui-ci.

12.2 Opérations fondamentales

<code>Ctrl-k h</code>	Aide <code>Ctrl-k h</code> pour l'enlever
<code>Ctrl-k e</code>	Ouvre un fichier
<code>Ctrl-k d</code>	Sauvegarde le fichier sous ...
<code>Ctrl-k x</code>	Sauvegarde le fichier et quitte joe
<code>Ctrl-c</code>	Quitte joe sans sauvegarder les modifications
<code>Ctrl-k b</code>	Marque un début de bloc
<code>Ctrl-k k</code>	Marque une fin de bloc
<code>Ctrl-k c</code>	Copie le bloc sélectionné
<code>Ctrl-k m</code>	Déplace le bloc sélectionné (move)

13 Emacs

13.1 Syntaxe

On lance emacs avec les commandes `emacs` ou `emacs nom_fichier`. Si `nom_fichier` existe, il sera édité, sinon, un nouveau fichier de ce nom sera créé.

13.2 Raccourcis clavier

<code>Ctrl-x-f</code>	Charge un nouveau fichier si le nom donné n'existe pas, le fichier est créé
<code>Ctrl-x-s</code>	Enregistre le fichier sous le nom courant
<code>Ctrl-x-w</code>	Enregistre le fichier sous un nouveau nom
<code>C-x-k</code>	Ferme le fichier en cours d'édition
<code>Ctrl-x-c</code>	Quitte emacs
<code>Alt-g</code>	Va à la ligne spécifiée
<code>Ctrl-espace</code>	Marque un début de bloc
<code>Alt-w</code>	Ecrit une fin de bloc (copie)
<code>Ctrl-w</code>	Ecrit une fin de bloc (couper)
<code>Ctl-y</code>	Coller
<code>Ctrl-g</code>	Annule la commande en cours (très utile si on ne comprend plus rien!)

Emacs supporte la complétion automatique des commandes. Si vous êtes en train de taper un nom de fichier, une pression sur la touche **Tab** demandera à emacs de compléter ce nom pour vous...

14 Vi

14.1 Syntaxe

On lance vi avec les commandes `vi` ou `vi nom_fichier`. Si `nom_fichier` existe, il sera édité, sinon, un nouveau fichier de ce nom sera créé.

14.2 Modes

Sous vi, il existe 3 modes de fonctionnement, le mode lecture de texte, le mode modification de texte et le mode commande.

- Le mode lecture de texte est celui par défaut, il vous permet de lire le texte sans pour autant vous autoriser à y faire des modifications.
- Le mode modification/insertion de texte est obtenu en tapant l’une des lettres suivantes : **i**, **A**, **O**, **R** suivant le mode désiré. On quitte ce mode en appuyant sur la touche **Esc**.
- Le mode commande est obtenu en appuyant sur la touche **:**. Une ligne en bas de l’écran apparaît vous permettant ainsi de taper les commandes désirées. On quitte ce mode en appuyant sur **Esc**. On valide une commande par la touche **Entrée**.

14.3 Commandes

:wq sauvegarde le fichier et quitte
:w sauvegarde le fichier
:q! quitte sans sauvegarder

14.4 Changements de mode

i ajoute le texte à partir de la position du curseur
A ajoute le texte à partir de la fin de la ligne
O crée une nouvelle ligne
R écrit par dessus le texte actuel (Overwrite)
r remplace le caractère courant

14.5 Raccourcis clavier

ndd coupe n lignes à partir de la ligne courante et les place dans le tampon
nyy copie n lignes du tampon
p recopie le tampon en-dessous de la ligne courante
P recopie le tampon au-dessus de la ligne courante
x supprime le caractère courant. **nx** supprime n caractères
d\$ supprime la fin de la ligne
texte recherche la chaîne de caractères texte
n continuer la recherche
N continuer la recherche vers le haut

15 Remerciements

- Les textes concernant les commandes **ls** et **rm** sont extraits du site de César Alexanian (<http://petitjournal.org>) que je tenais à remercier, ces textes sont diffusés sous licence GPL.
- Le texte concernant la commande **mv** est extrait du manuel en ligne (**man**). La traduction en a été faite par Christophe Blaess.
- Le texte concernant **vi** est extrait de l’aide mémoire de Yves Epelboin. <http://www.lmcp.jussieu.fr/informatique/guide/vi.html>.