

Initiation à la ligne de commandes

Mercredi 19 mars 2008

Intervenant

- Thomas Petazzoni
- Dans la communauté...
 - « responsable légal » de Toulibre
 - Membre du conseil d'administration de l'April
 - Coordinateur du Libre en Fête
 - Animateur de l'Agenda du Libre
- Dans la vraie vie...
 - Ingénieur formateur/consultant à Free-Electrons
 - Linux et Logiciels Libres pour l'embarqué

Shell

- Logiciel permettant de saisir et d'exécuter les commandes
- Coquillage en anglais
- Emballage autour du noyau ?
- Shell accessible de deux façons
 - Console texte (Alt+F1 à Alt+F6)
 - En mode graphique (xterm, Konsole, gnome-terminal)
- Derrière, le même logiciel fonctionne, en général *bash*
- *Bourne Again Shell*

Prompt

- Accès à une invite de commande ou *prompt*

```
thomas@crazy:~/libre/toulibre$
```

- Va permettre de saisir les commandes
- Donne des informations :
 - Nom d'utilisateur - *thomas*
 - Nom de machine - *crazy*
 - Répertoire courant - *~/libre/toulibre*
 - Symbole de terminaison - *\$* (ou *#* en root)
- Le prompt est configurable

Première commande


- Saisir le nom de la commande, suivi de la touche Entrée pour valider
- Le résultat s'affiche dans le terminal
- Exemple avec la commande « ls »
- « ls » comme « list », liste les fichiers du répertoire courant

```
thomas@crazy:/tmp/test$ ls
fichier.txt  toto.jpg
thomas@crazy:/tmp/test$
```

Options

- Les commandes ont des options, qu'on peut spécifier avec un ou deux tirets
- Un tiret: option courte
- Deux tirets: option longue
- Exemples pour ls :
 - -l pour avoir des détails (taille, etc.)
 - -s pour trier par taille
 - -h pour avoir les tailles en format lisible par un humain
 - --human-readable
- On peut les combiner: -lhs ou -l -h -s

Connaître les options

- Pour toutes les commandes 
 - `commande -help`
 - Résumé des commandes
 - `man commande`
 - Manuel complet de la commande, description, options, exemples, astuces, etc.

Exercice

- Afficher la liste des fichiers, triés par taille en ordre inverse, avec les détails

```
ls -lrS
```


Reconnaître les types de fichiers

- Avec `ls -l`, regarder la première lettre
 - Un tiret, fichier normal
 - Un « d », répertoire
 - Un « l », lien symbolique
 - Etc.
- Avec les couleurs
 - Souvent configuré par défaut
 - Sinon: `ls --color`

Se déplacer dans les répertoires

- Commande « cd »
- *Change Directory*

```
thomas@crazy:~$ cd libre  
thomas@crazy:~/libre$
```

Noms des fichiers

- Sensibles à la casse
 - blabla != Blabla != BLABLA
- Limite assez large (255 caractères ?)
- Peuvent contenir n'importe quel caractère sauf /
- Les extensions sont simplement des conventions
- Exemples
 - README
 - .bashrc
 - index.html.old

Chemin

- Désigne un fichier dans une arborescence de répertoires imbriqués
- Peut être relatif
 - blabla/machin/truc.txt
 - Par rapport au répertoire courant
 - Pas de / au début
- Peut être absolu
 - /home/thomas/blabla/machin/truc.txt
 - Par rapport à la racine de l'arborescence
 - Un / au début

Arborescence générale

/	Répertoire racine
/bin	Programmes de base
/boot	Noyau
/dev	Fichiers spéciaux pour l'accès aux périphériques
/etc	Configuration du système
/home	Répertoires personnels des utilisateurs
/lib	Bibliothèques partagées de base
/lost+found	Fichiers retrouvés après réparation
/media	Points de montage des périphériques amovibles
/mnt	Points de montage des périphériques amovibles
/opt	Applications additionnelles
/proc	Informations systèmes
/root	Répertoire personnelle de l'administrateur
/sbin	Programmes de base de l'administrateur
/sys	Informations sur les périphériques et le système
/tmp	Fichiers temporaires
/usr	Programmes, bibliothèques, documentation
/var	Journaux, répertoires temporaires

Répertoire spéciaux

- « `.` » est le répertoire courant
 - `readme.txt` identique à `./readme.txt`
- « `..` » est le répertoire parent
- `cd ..`

```
thomas@crazy:~/libre$ cd .  
thomas@crazy:~/libre$ cd ..  
thomas@crazy:~$
```

- « `~` » est le répertoire de l'utilisateur

```
thomas@crazy:~/libre/toulibre$ cd ~/src/perl  
thomas@crazy:~/src/perl$
```

Où suis-je ?

- Voir le *prompt*
- Ou utiliser la commande `pwd`

```
thomas@crazy:~/libre/toulibre$ pwd
/home/thomas/libre/toulibre
thomas@crazy:~/libre/toulibre$ cd ~/src/perl
thomas@crazy:~/src/perl$
thomas@crazy:~/src/perl$ pwd
/home/thomas/src/perl
```

Combinaison

- Que fais

```
thomas@crazy:~/libre/april$ ls ~/src/../../libre/toulibre/statuts.pdf
```

?

Fichiers: autres commandes

- Copie: `cp`
 - `cp tata.txt toto.txt` copie un fichier
 - `cp -r tata toto` copie un répertoire
- Déplacement: `mv`
 - `mv tata.txt ~/libre/toulibre/`
 - `mv tata ~/libre/april`
 - `mv tata.txt toto.txt titi.txt ~/libre/april`
- Suppression: `rm`
 - `rm tata.txt` supprime un fichier
 - `rm -r tata` supprime un répertoire et son contenu!
- Nombreuses options, voir `man`

Wildcards

- Comment supprimer tous les fichiers .jpg d'un répertoire ?
- Solution: `rm *.jpg`
- « `*` » remplace n'importe quel nombre de n'importe quel caractère
- « `?` » remplace un seul caractère, n'importe lequel
- `ls a?to*.jpg`
 - `a1topouet.jpg` -> oui
 - `a12tomeuh.jpg` -> non

Répertoires

- Création: `mkdir`
 - `mkdir toto`
- Suppression: `rmdir`
 - `rmdir toto`
 - Le répertoire doit être vide
 - Autre solution: `rm -r toto`

Contenu d'un fichier

- **Commande de base: cat**
 - Affiche l'ensemble du contenu d'un fichier, sans s'arrêter
- **Mieux: more**
 - S'arrête à chaque page
- **Encore mieux: less**
 - S'arrête à chaque page
 - Permet de revenir en arrière
 - Permet de faire des recherches
 - Etc.
- **Marche bien sur les fichiers texte brut !**

Droits et propriétaires

- Chaque fichier possède
 - Des droits d'accès, lecture, écriture, exécution
 - Un utilisateur propriétaire
 - Un groupe propriétaire

```
$ ls -l formation-cyberbase.pdf  
-rw-r--r-- 1 thomas users 252348 jan 21 22:46 formation-cyberbase.pdf
```

Groupe propriétaire

Propriétaire

Droits des autres

Droits du groupe propriétaire

Droits du propriétaire

Type de fichier

Droits et propriétaire

- Pour les fichiers
 - `r` pour la lecture
 - `w` pour l'écriture
 - `x` pour l'exécution
- Pour les répertoires
 - `r` pour lire le contenu du répertoire
 - `w` pour modifier le répertoire (créer, supprimer des fichiers, modifier leurs attributs)
 - `x` pour entrer dans un répertoire

Modifier les droits

- Commande `chmod`
- `chmod mode fichiers`
- Mode
 - Qui s'applique la modification: `u` pour le propriétaire (`u` pour user), `g` pour le groupe, `o` pour les autres (`o` pour others)
 - Quelle modification: `+` pour ajouter des droits, `-` pour en retirer
 - Quels droits: `r`, `w`, `x`
- Exemple: `chmod o+rx README`
- Option `-R` pour récursif

Modifier le propriétaire

- Le propriétaire
 - Commande chown
 - `chown nomutilisateur fichier`
- Le groupe
 - Commande chgrp
 - `chgrp nomgroupe fichier`
- Les deux en même temps
 - `chown nomutilisateur:nomgroupe fichier`
- Option -R pour récursif

Variables d'environnement

- Variables qui affectent le comportement du shell ou des commandes exécutées
- Affiche leur valeur avec `echo $VARIABLE`
- Positionne leur valeur avec `export`
- `PATH`
 - Variable d'environnement listant les chemins dans lesquels le shell cherche les commandes à exécuter (plus les *builtins*)
- `PS1`
 - Variable d'environnement décrivant le format de l'invite de commande

VARIABLES D'ENVIRONNEMENT

- Exemples

```
export PS1="\w | "
```

```
export PS1="\e[34m\d \e[33m\t\e[m \w \e[31m|\e[m "
```

- Avoir la liste des variables d'environnement : `export sans argument`

Manipuler du texte : grep

- Chercher un motif : `grep`
- `grep motif fichier`
- Le motif peut être un simple mot, ou une expression plus complexe
- Va retourner seulement les lignes du fichier contenant le motif

Manipuler du texte : grep

```
$ cat fichier
abcde
abc
azerty
blabla
abcd
awkje
$ grep abc fichier
abc
abcd
abcde
$ grep a...e fichier
abcde
awkje
$ grep ^b fichier
blabla
$ grep a.*d.* fichier
abcd
abcde
```

Manipuler du texte: trier

- Commande sort
- `sort fichier`
- Ordre inverse: option `-r`
- Ordre numérique: option `-n`

```
$ cat fichier
abcde
abc
azerty
blabla
abcd
awkje
$ sort fichier
abc
abcd
abcde
awkje
azerty
blabla
```

```
$ cat fichier
abcde
abc
azerty
blabla
abcd
awkje
$ sort -r fichier
blabla
azerty
awkje
abcde
abcd
abc
```

Remplacer: sed

- Une commande pour remplacer du texte et bien plus : sed
- `sed s/motif/remplacement/ fichier`
- Motif peut être une *expression régulière*

```
$ cat fichier
abcde
abc
azerty
blabla
abcd
awkje
$ sed s/abc/machin/ fichier
machin
machind
machinde
awkje
azerty
blabla
```

```
$ cat fichier
abcde
abc
azerty
blabla
abcd
awkje
$ sed s/^a.*e$/machin/ fichier
abc
abcd
machin
machin
azerty
blabla
```

Combiner

- Les commandes Unix effectuent leur affichage sur la *sortie standard* et peuvent prendre leur contenu en entrée sur l'*entrée standard*
- Le tube ou *pipe* est un moyen de rediriger la sortie d'une commande vers l'entrée d'une autre
- `cat fichier | grep abc | sort -r | sed s/a/z/`

Rediriger

- Pour rediriger la sortie standard d'une commande vers un fichier, utiliser l'opérateur `>` ou `>>`
- `grep abc fichier > nouveaufichier`
- Pour rediriger le contenu d'un fichier vers l'entrée standard d'une commande, utiliser l'opérateur `<`
- `grep abc < fichier`
- Tout ensemble
- `grep abc < fichier | sort -r > nouveaufichier`

Compression

- Compresser un fichier
 - `gzip fichier.txt`
 - `bzip2 fichier.txt`
- Décompresser un fichier
 - `gunzip fichier.txt.gz`
 - `bunzip2 fichier.txt.bz2`
- Comment faire pour compresser plusieurs fichiers en un seul ?
- Solution: archiver

Archivage

- Archivage, commande tar
- Créer une archive
 - `tar cf archive.tar monrepertoire`
- Décompacter une archive
 - `tar xf archive.tar`
- Créer et compresser en même temps
 - `tar czf archive.tar.gz monrepertoire`
 - `tar cjf archive.tar.bz2 monrepertoire`
- Décompresser et décompacter
 - `tar xzf archive.tar.gz`
 - `tar xjf archive.tar.bz2`

Occupation disque

- Commande `df` pour connaître l'espace utilisé et disponible sur les différents systèmes de fichiers
 - Option `-h` pour quelque chose de lisible
- Commande `du` pour connaître la place utilisée
 - Option `-s` pour faire la somme et `-h` pour quelque chose de lisible

Occupation disque

- Commande `df` pour connaître l'espace utilisé et disponible sur les différents systèmes de fichiers
 - Option `-h` pour quelque chose de lisible
- Commande `du` pour connaître la place utilisée
 - Option `-s` pour faire la somme et `-h` pour quelque chose de lisible

Changer d'utilisateur

- Commande `su`
 - Pour « Switch User »
- `su` sans argument pour passer root
- `su` avec un nom d'utilisateur
- Pour exécuter des commandes en root rapidement : `sudo`
 - Doit être configuré au préalable
 - Configuré par défaut sous Ubuntu pour le premier utilisateur du système

Initiation à vi

- Éditeur de texte à la fois simple et très puissant
- Lancer: `vi fichier`
- Deux modes
 - Mode commande
 - Mode édition
- Commande `i` permet de passer en mode édition, Echap permet d'en sortir
- `:wq` pour enregistrer et quitter

Boucles

- Faire des boucles pour réaliser un traitement répétitif, avec `for`
- ```
for i in tata titi toto ; do gzip $i.txt ; done
```
- Va compresser `tata.txt`, `titi.txt` et `toto.txt`

# Un exemple plus compliqué

```
n=0 ; for img in *.JPG ; do
 NAME=$(printf "photo%02d.jpg" $n) ;
 NAMEMINI=$(printf "photo%02dt.jpg" $n) ;
 echo "Conversion de $img en $NAME et $NAMEMINI" ;
 mv $img $NAME ;
 convert -resize 200 $NAME $NAMEMINI ;
 n=$((n+1)) ;
done
```



# Combinons tout ça

- Créer répertoire ~/bin
- Créer un fichier genthumbs
- Y mettre le contenu du script précédent
  - Ajouter #!/bin/bash en première ligne
- Donner les droits d'exécution au fichier
- Ajouter ~/bin à la variable d'environnement PATH
- Pour le faire de manière permanente
  - Éditer ~/.bashrc
  - Y ajouter la commande pour ajouter ~/bin au PATH