

Système d'exploitation

Jean-Sébastien Coron

Université du Luxembourg

September 26, 2009

- Les commandes usuelles.
- Redirections et pipes.

- Un shell est un programme qui lit et exécute les commandes de l'utilisateur.
- Un shell permet aussi :
 - De contrôler les processus.
 - De rediriger l'entrée et/ou la sortie.
 - D'écrire des scripts.
- Script shell:
 - Fichier texte contenant une série de commandes.
 - Comme fichier .bat sous MS-DOS.
- De nombreux shell existent:
 - sh, bash, csh, tcsh.

- Les commandes de manipulation de fichier.
 - `cd`: change le répertoire courant.
 - `ls`: affiche la liste des fichiers.
 - `mkdir`: création d'un répertoire.
 - `rmdir`: suppression d'un répertoire.
 - `cp source destination`: copie d'un fichier.
 - `mv source destination`: renommer ou déplacer un fichier.
 - `rm`: supprimer un fichier.
 - `cat`: affiche le contenu d'un fichier.
 - `more`, `less`: affiche le contenu d'un fichier avec des pauses.

- Manipulation de fichier (suite)
 - `chmod mode fichier`: modifie les droits d'accès au fichier.
 - `pwd`: affiche le répertoire courant.
- Autre commandes:
 - `which`: affiche le chemin exécutable correspondant à une commande.
 - `which ls` donne `/usr/bin/ls`.
 - `date`: affiche la date et l'heure.
 - `man commande`: affiche le manuel pour commande.

- Utilitaires UNIX.

- `head -n fichier`: affiche les n premières lignes d'un fichier.
- `diff fich1 fich2`: compare les fichiers `fich1` et `fich2`.
- `tar`: permet d'archiver des fichiers.
- `wc [-cwl] [fichier]`: affiche le nombre de caractère, mot ou ligne d'un fichier.
 - `-c`: caractère, `-w`: mot, `-l`: ligne.

- Permet de chercher dans un répertoire et ses sous-répertoires des fichiers présentant certaines caractéristiques.
- `find chemin expression`
- Principales options:
 - `-name fich`: recherche le fichier `fich`.
 - `-print`: écrit le nom du fichier.
 - `-type: d`: répertoire, `f`: fichier symbolique.
 - `-exec`: permet l'exécution d'une commande sur le fichier représenté par `{}`.

Exemple avec find

- Rechercher tous les fichiers nommés `hello.txt` à partir du répertoire racine.
 - `find / -name hello.txt -print.`
- Afficher tous les fichiers `.h` à partir du répertoire courant.
 - `find . -name '*.h' -print`
- Affiche à l'écran le contenu de tous les fichiers `.c`.
 - `find . -name '*.c' -exec cat '{}' \;`

- Permet de rechercher un certain motif dans un fichier.
 - Le motif est décrit par une *expression régulière*.
- `grep [option] motif fichier` Affiche les lignes de fichier qui contiennent le motif `motif`.
 - `-l`: n'affiche que le nom des fichiers.
- Exemple:
 - `grep 'define' stdio.h`: recherche le mot "define" dans le fichier `stdio.h`.
 - `grep 'hello' *`: recherche le mot "hello" dans tous les fichiers du répertoire.

- Par défaut, une commande écrit dans la sortie standard (l'écran) et prend en entrée l'entrée standard (le clavier).
- On peut rediriger la sortie d'une commande dans un fichier.
 - Exemple: `ls -l > liste`. Stocke dans le fichier `liste` la liste des fichiers.
- Pour écrire à la fin d'un fichier, on utilise `>>`
 - `ls >> liste`.

- On peut rediriger l'entrée d'une commande à partir d'un fichier:
 - `cat < input`: la commande lit le fichier `input` et l'imprime, au lieu de lire le clavier.
- On peut rediriger l'entrée et la sortie en même temps:
 - `cat < input > output`

- On peut rediriger la sortie d'un processus vers l'entrée d'un autre avec l'opérateur pipe '|'.
 - Les processus s'exécutent en parallèles.
- Permet de combiner plusieurs commandes pour obtenir de fonctions plus élaborées.
 - `cat hello.txt | sort | uniq`. Afficher les lignes de `hello.txt` triées en éliminant les lignes identiques.
 - `ls | sort -r`. Affiche les fichier d'un répertoire trié à l'envers.