Système d'exploitation
Les Commandes

IUT G.T.R

1ère Année

Sylvain MERCHEZ

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc …

Syntaxe générale

La syntaxe générale des commandes :

nom_cmd [options] [[argument1][argument2] …]

- nom_cmd : nom de la commande (en minuscules)
- options : elles permettent des variantes de la commande. En général une option est précédée du caractère tiret ('-').
- arguments : en général les noms des objets cibles de la commande. L'espace sert de séparateur d'arguments

Principes

Il existe 2 types de commandes :

- **les commandes internes** : sous-programmes de l'interpréteur de commande (shell). Elles sont directement exécutables sans création d'un shell-fils. Exemples : alias, cd, echo, ...
- **les commandes externes** : fichiers exécutables. Leur exécution nécessite la création d'un processus fils. Exemples : mkdir, mv, chmod, ...

Les commandes simples

- Consulter le manuel : man
- Identifier un utilisateur : who,whoami,id
- Changer de mot de passe : passwd
- Afficher une chaîne de caractères : echo, banner

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc …

Le Système de Gestion des Fichiers (SGF)

Pour Le système Unix, tout est fichier.

On distingue 3 types de fichier :

- Les fichiers ordinaires :
  - Fichiers de données : texte ou au format propriétaire
  - Fichiers programmes : instructions exécutables (binaire ou script)
- Les répertoires
- Les fichiers spéciaux : imprimantes, disques, cédéroms, ... etc

Le Système de Gestion des Fichiers (SGF)

- Un fichier est désigné par son nom
- Accès au fichier se fait par :
  - Référence absolue
  - Référence relative
- L 'ensemble des caractéristiques d 'un fichier est contenu dans un **i-nœud** (nœud d 'information)

Chemin absolu versus relatif

- 2 éléments particuliers dans tous les répertoires
  - le répertoire courant : .
  - le répertoire père : ..
- Chemin absolu
  - Il faut indiquer le **chemin complet** depuis la racine (il débute obligatoirement par ' / ')
    « */usr/local/Acrobat/bin/acroread toto.pdf* »
- Chemin relatif
  - On démarre du répertoire **courant**
    « *essai/ma_commande mes_paramètres* »
    « *../repertoire_pere/commande* »

Le système de gestion des fichiers

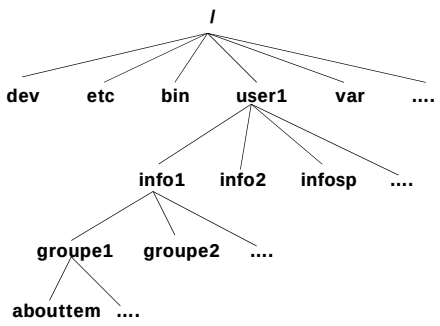
Ces éléments sont regroupés dans une structure hiérarchisée représentée sous forme d'un arbre inversé.

L'arbre comprend :

- Les nœuds : les répertoires
- Les feuilles : les fichiers ordinaires ou spéciaux

Le sommet de l'arbre est appelé racine (**root**)

Exemple d'un système de fichier



Les commandes Unix : répertoires

- Changement de répertoire : cd
- Affichage du nom du répertoire courant : pwd
- Affichage du contenu du répertoire : ls
- Création d'un répertoire : mkdir
- Suppression d'un répertoire : rmdir
- Renommage/déplacement un répertoire : mv
- Comparaison de 2 répertoires : dircmp

Les commandes Unix : répertoires

- Exemple : changer de répertoire

Syntaxe : $ cd nom_répertoire

$ cd /usr/bin
$ cd
$ cd ../etc
$ cd ..
$ cd ~/230

Les caractères spéciaux

- \ banalise le caractère suivant
- « ... » banalise les caractères sauf \, $, et `
- ' ... ' banalise tous les caractères
- ‘ ... ‘ substitution de commande
- * n 'importe quelle chaîne de caractères
- ? n 'importe quel caractère
- [...] n 'importe quel caractère décrit entre les crochets

Une seule arborescence (1)

Tout support physique (disque local ou réseau, cédérom, disquette) peut être découpé en un ensemble de partitions logiques

Toute partition organisable en SGF contient une arborescence de fichiers et de répertoires.

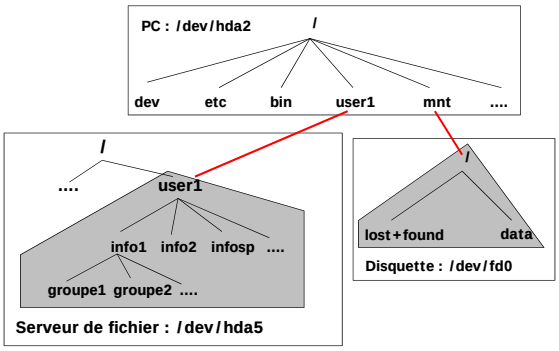
La partition support de la racine est la partition principale.

Une seule arborescence (2)

Les autres partitions peuvent être attachées à un sous répertoire de la partition principale à l'aide du mécanisme de montage (**mount**). Ces partitions peuvent être montées ou démontées suivant le besoin de l'utilisateur.

L'ensemble des fichiers est vu par l'utilisateur comme appartenant à **une arborescence unique**

Une seule arborescence (3)



Une seule arborescence (4)

```
[merchez@wilson /soft]$ df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda2              7448704   1467324   5603004  21% /
iut-gtr2:/user1/info1  6048320   5020756    720324  87% /user1/info1
/dev/hda1              6048320   5020756    720324  87% /dos/disk_c
iut-gtr2:/soft         3028080   2528516    345744  88% /mnt/soft
[merchez@wilson /soft]$
```

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc ...

Les caractéristiques d'un fichier

- Un fichier est désigné par son nom
- Accès au fichier se fait par :
  - Référence absolue
  - Référence relative
- L'ensemble des caractéristiques d'un fichier est contenues dans un **i-nœud** (nœud d'information).

Le nœud d'information

Un i-nœud contient les informations suivantes :

- Le type de fichier : fichier ordinaire, répertoire, ...
- Les droits d'accès
- Le nombre de liens
- Le propriétaire
- Le groupe du propriétaire
- La taille du fichier en octets et en blocs
- Les dates d'accès et de modification
- ... et le contenu du fichier

Les types de fichier

On distingue plusieurs types de fichier sous unix :

- Les fichiers ordinaires (-) : fichiers textes dont chaque ligne est terminée par <LF> et non <CR><LF>
- Les fichiers spéciaux de gestion des périphériques :
  - Les fichiers blocs (b) : disque dur, cédérom, ... etc
  - Les fichiers caractères (c) : écran, clavier, souris, ... etc
- Les fichiers FIFO ou pipes (p) : destinés à la communication entre les processus
- Les répertoires (d)
- Les liens symboliques (l)

Les commandes Unix : fichiers (1)

- Suppression d'un fichier : rm
- Copie des fichiers : cp
- Déplacement d'un fichier : mv
- Création ou modification de la date de modification d'un fichier : touch
- Afficher le contenu d'un fichier page par page : more
- Afficher le contenu d'un fichier : cat
- Afficher les N premières lignes : head
- Afficher les N dernières lignes : tail
- Découper un fichier : cut

Les commandes Unix : fichiers (2)

- Comparaison de 2 fichiers : diff, cmp
- Comptage du nombre de caractères, de mots et de lignes contenus dans un fichier : wc
- Impression d'un fichier : lpr, a2ps, lp, ...
- Création d'un lien physique ou symbolique : ln
- Dimensionner un fichier : split
- Archivage des fichiers : tar, cpio
- Compression d'un fichier : compress, gzip, bzip2
- Décompression d'un fichier : uncompress, gunzip, bzip2

Les commandes Unix : fichiers (3)

- Identifier le type de fichiers : file
- Recherche d'un fichier : find
- Recherche d'un motif dans un fichier : grep
- Tri d'un fichier : sort
- Suppression des lignes adjacentes identiques : uniq
- Masque de création des fichiers : umask
- Fusion de 2 fichiers : paste
- Extraction des lignes communes de 2 fichiers : comm

Les commandes Unix : fichiers

- Exemple : copie de fichiers

  Syntaxe : $ cp [-i] fichier1 fichier2

  ou $ cp [-ir] fichier1 ... fichiern répertoire

  Le fichier1 est copié sur le fichier2 ou dans le répertoire

  OPTION : -i demande si on veut écraser un fichier2 déjà existant

  -r copie toute l 'arborescence

  $ cp test.dat test1.dat

  $ cp test1.dat test.dat tmp

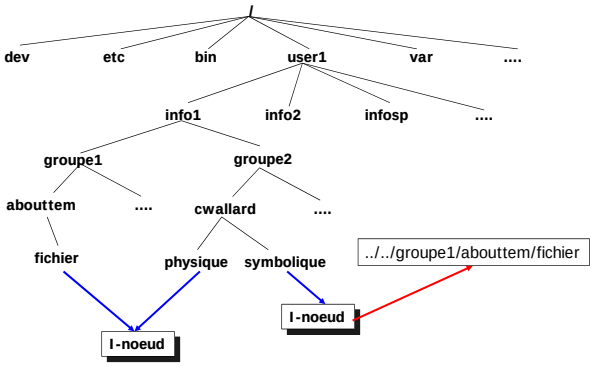
Le mécanisme de lien (1)

Le mécanisme de lien sous unix permet de désigner un même fichier par 2 références différentes.

On distingue 2 types de lien sur les fichiers sous unix :

- **Les liens symboliques** : un fichier spécial contenant le chemin d'accès vers le fichier de référence. Le fichier de référence peut ne pas exister.
- **Les liens physiques** : une entrée du répertoire pointant sur le même i-nœud que le fichier de référence. Le fichier de référence doit exister

Le mécanisme de lien (2)



Le mécanisme de lien (3)

cwallard> ls -al /user1/info1/goupe1/abouttem/fichier

-rw-r--r-- 1 abouttem info1 102 oct 12 22:26 fichier

cwallard> **ln ../../groupe1/abouttem/fichier physique**

cwallard> ls -al /user1/info1/goupe1/abouttem/fichier

-rw-r--r-- 2 abouttem info1 102 oct 12 22:26 fichier

cwallard> ls -al physique

-rw-r--r-- 2 abouttem info1 102 oct 12 22:26 physique

Le mécanisme de lien (4)

cwallard> **ln –s ../../groupe1/abouttem/fichier symbolique**

cwallard> ls -al /user1/info1/goupe1/abouttem/fichier

-rw-r--r-- 2 abouttem info1 102 oct 12 22:26 fichier

cwallard> ls -al symbolique

lrwxrwxrwx 1 cwallard info1 31 jan 23 10:15 symbolique -> ../../groupe1/abouttem/fichier

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc ...

Exécution des commandes

- Enchaînement des commandes (;)

  cd / ; pwd ; ls –al → si une des commandes n'existe pas, l'exécution s'arrête.

- Exécution en arrière plan (&)

  netscape & → l'utilisateur récupère l'interpréteur pour lancer d'autres commandes

- Exécution conditionnelle && ou ||

  cm1 && cm2 → cm2 est exécutée si l'exécution de cm1 a réussi

  cm1 || cm2 → cm2 est exécutée si l'exécution de cm1 a échoué

Exécution des commandes : redirection

- Il existe 3 fichiers associés à chaque processus :
  - L'entrée standard (stdin) : clavier
  - La sortie standard (stdout) : écran
  - L'erreur standard (stderr) : écran

Exécution des commandes : redirection

- Redirection de la sortie standard : > et >>

  cmd > fichier → le résultat de cmd est mis dans un fichier

  cmd >> fichier → le résultat de cmd est mis à la fin du fichier

  cmd >& fichier → la sortie erreur est dans le fichier

- Redirection de l'entrée standard : <

  cmd < fichier → cmd lit les données dans le fichier

- Echaînement des commandes : | (pipe)

  cmd1 | cmd2 → la sortie de cmd1 et dirigée sur l'entrée de cmd2

Les droits d'accès à un fichier

Les fichiers sont protégés au moyen d'un code binaire de protection sur 9 bits.

Propriétaire	Groupe	Autres
3 bits	3 bits	3 bits
rwx	rwx	rwx

Une lettre (rwx) indique l'autorisation, un tiret indique l'interdiction

r : lecture

w : modification

x : droit d'exécution pour un fichier, droit de traverser pour un répertoire

Les commandes de modification des droits

Il existe 3 commandes permettant de modifier les protections d'un fichier :

- chmod : change les protections (rwx) du fichier ou répertoire
- chown : change le propriétaire du fichier
- chgrp : change le groupe propriétaire du fichier

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc ...

Les processus : caractéristiques

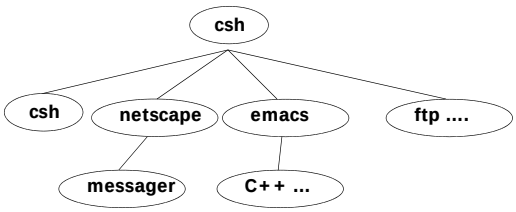
Un processus est un programme est cours d'exécution.

Il est caractérisé par :

- Son identité : pid
- L'identité de son créateur (père) : ppid
- Son propriétaire effectif et réel : uid
- Son groupe effectif et réel : gid
- Son état (élu, bloqué, prêt, zombi)
- Le programme exécutable
- Sa priorité : nice
- La date de création
- Les temps CPU consommés
- Le masque de création des fichiers : umask
- La table des descripteurs des fichiers ouverts
- .... etc

Les processus : généalogie

Tout processus (sauf init) possède un processus créateur (père). L'ensemble des processus peut être vu comme un arbre.



Gestion des processus

Un ensemble de primitives permet à l'utilisateur de gérer ses processus :

- ps : affiche la liste des processus
- pstree : affiche les processus sous forme d'un arbre → généalogie
- top : affiche la liste des processus en commençant par le processus occupant le plus la CPU

Gestion des processus

- kill : envoie un signal à un processus
  - kill –9 numpid : arrête le processus de pid numpid
- killall nomprocess : arrête un processus ayant pour nom nomprocess
- Planification des tâches : crontab
- at : exécution différée d'un processus
- nice : modification de la priorité d'un processus

La liste des processus (1)

La liste des processus peut être obtenu à l'aide de la commande ps.

```
[merchez@wilson ~]$ ps aux | more
USER     PID %CPU %MEM    VSZ  RSS TTY   STAT START   TIME COMMAND
root       1  0.0  0.3   1120  476  ?     S    Jan12   0:05 init [3]
bin      339  0.0  0.3   1216  412  ?     S    Jan12   2:41 portmap
root     354  0.0  0.0      0    0  ?     SW   Jan12   0:00 [lockd]
root     355  0.0  0.0      0    0  ?     SW   Jan12   0:00 [rpciod]
root     364  0.0  0.4   1156  512  ?     S    Jan12   0:00 rpc.statd
root     378  0.0  0.3   1104  388  ?     S    Jan12   0:00 /usr/sbin/apmd
   -p
root     408  0.0  0.3   1268  436  ?     S    Jan12   0:00 ypbind (master)
root     413  0.0  0.3   1316  484  ?     S    Jan12   0:00 ypbind (slave)
root     446  0.0  0.4   1208  512  ?     S    Jan12   0:00 /
   usr/sbin/automou
root     501  0.0  0.3   1172  472  ?     S    Jan12   0:45 syslogd -m 0
daemon 551  0.0  0.2   1144  296  ?     S    Jan12   0:00 /usr/sbin/atd
root     566  0.0  0.4   1328  564  ?     S    Jan12   0:00 crond
```

La liste des processus (2)

```
USER     PID %CPU %MEM    VSZ  RSS TTY  STAT START   TIME COMMAND
root     566  0.0  0.4   1328  564   ?       S    Jan12   0:00 crond
root     588  0.0  0.3   1156  460   ?       S    Jan12   0:00 inetd
root     602  0.0  0.2   1168  376   ?       S    Jan12   0:00 lpd
root     628  0.0  0.1   1104  144   ?       S    Jan12   0:00 rpc.rquotad
root     641  0.0  1.4   2552 1844   ?       S    Jan12   0:49 rpc.mountd --no-
    n
root     650  0.0  0.0      0    0   ?       SW   Jan12   2:47 [nfsd]
root     711  0.0  0.5   2128  668   ?       S    Jan12   0:00 sendmail:
    accepti
postgres 829  0.0  0.3   5048  400   ?       S    Jan12   0:00 /
    usr/bin/postmast
nobody  7493  0.0  3.1   7636 4008   ?       S    Jan14   0:00 httpd
root    2338  0.0  1.0   2240 1320  pts/0    S    08:08   0:00 login -- thsu
merchez2340  0.0  0.9   2112 1192  pts/0    S    08:08   0:00 –tcsh
merchez2818 15.0  0.5   2328  696  pts/1    R    08:36   0:00 ps aux
merchez2819  2.0  0.4   1336  524  pts/1    S    08:36   0:00 more
```


La liste des processus (3)

- **NI** Valeur standard Unix de gentillesse (nice). Une valeur positive signifie un accès moindre au CPU.
- **SIZE** Taille virtuelle de l'image du processus (code + données + pile).
- **RSS** Taille résidente de l'image du processus. Nombre de kilo-octets se trouvant en mémoire.
- **STAT** Etat du processus. Le premier champ correspond à **R** (runnable) prêt à être exécuté, **S** (sleeping) endormi, **D** sommeil ininterruptible, **T** (traced) arrêté ou suivi, **Z** (zombie). Le second champ contient **W** si le processus n'a pas de pages résidentes. Le troisième champ contient **N** si le processus a une valeur de gentillesse positive (nice, champ **NI**).
- **TTY** terminal de contrôle
- **TRS** Taille de code résident en mémoire.

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc ...

Gestion réseaux

- Envoie d'un message à un utilisateur : mail
- Gestionnaire de messages : pine, elm, mailx
- Envoie d'un message sur un terminal : write
- Accepte/refuse la réception des messages : mesg
- Dialogue entre 2 utilisateurs connectés : talk
- Connexion sur une machine distant : rlogin, telnet
- Exécution d'une commande à distance : rsh
- Copie des fichiers entre systèmes distants : rcp
- Transfert de fichier : ftp

Gestion d 'une session

- Changement de l'interpréteur de commandes : chsh
- Affichage de l'identité de l'utilisateur : id
- Affichage du nom de login de l'utilisateur : logname
- Changement du mot de passe local : passwd
- Changement du mot de passe réseau : yppasswd
- Affichage du répertoire courant : pwd
- Modification temporaire d'utilisateur : su
- Lancement d'un interpréteur de commandes : csh, sh, ...
- Fermeture d'une session : exit, logout

PLAN DU COURS

- Généralités
  - Forme des commandes
- Commandes de bases
  - Arborescence
  - Fichiers
  - Redirection et droits d 'accès
  - Processus
  - Réseaux
  - etc ...

Historique des commandes

- ; le point virgule permet d 'enchaîner des commandes
- !! Permet de relancer la dernière commande
- history donne la liste des commandes tapées

  le nombre de commandes conservées est initialisé par la commande set history = n
- !5 permet de relancer la commande numéro 5
- !tar permet de relancer la dernière commande commençant par tar
- !-3 va chercher l 'événement n-3

Historique des commandes

- !^ le premier argument de la dernière commande
- !* tous les arguments de la dernière commande
- !$ le dernier argument de la dernière commande
- !:n les n premiers arguments de la commande précédente
- !m:n les n premiers arguments de la commande dont le numéro est m
- :$ dernier argument
- :^ premier argument
- :n-m de l 'argument n à m
- :p affiche la commande mais ne l 'exécute pas
- :s/1/r/ substitue la chaîne r à la chaîne 1

alias

- La commande alias permet définir une commande alias d'une suite de commandes.
  - alias h history
  - alias dir 'pwd ; ls –al '
  - alias rm 'rm –i`
  - alias ls 'ls –dircolors`
  - alias copy 'cp \!:1 \!:2`
    - \!* → liste des paramètres
    - \:!n → nième paramètre
    - \!^ → premier paramètre
    - \!$ → dernier paramètre

alias suite

- Annulation de l'alias : \

  alias cp `cp –i` → définition d'un alias cp

  cp f1 f2 → équivalent à cp –i f1 f2

  \cp f1 f2 → exécute la commande initiale
- La commande unalias permet de supprimer un alias
  - unalias h