

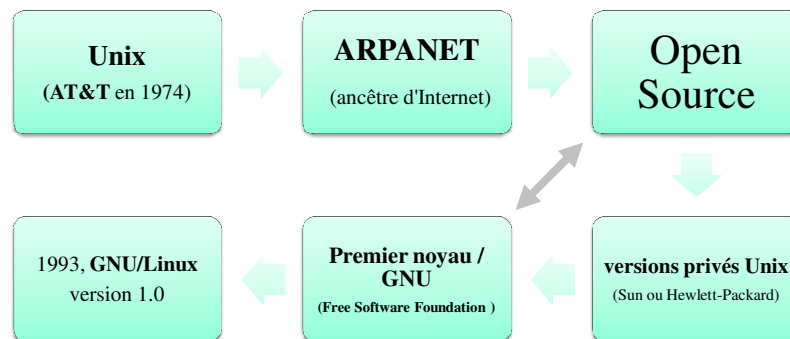
Systèmes d'exploitation :

5. UNIX



Systèmes d'exploitation: Unix - Notions de base (1)

1 Histoire



Systèmes d'exploitation: Unix - Notions de base (2)

2 Propriétés

Quelles que soient les versions :

UNIX = noyau + interpréteur de commandes + utilitaires

UNIX est construit autour de deux notions essentielles :

- **Fichiers** organisés en **structure arborescente**
 - **Processus** : un processus correspond à l'exécution d'un programme (système ou utilisateur) et à son environnement (descripteurs de fichiers, registres, compteur ordinal, ...).
- La gestion de la mémoire centrale constitue un aspect important du contrôle des processus.

Systèmes d'exploitation: Unix - Notions de base (3)

3 Les entrées / Sorties

Dans UNIX, toute unité d'E/S est désignée par un **fichier spécial**.

Il existe **deux modes d'échange** pour l'E/S :

- **Bloc** : les mémoires de masse (disque, disquettes) échangent des blocs d'information avec la mémoire centrale en utilisant des tampons
- **Caractère** : les terminaux, les imprimantes et les réseaux échangent des caractères avec la mémoire centrale, sans tampon.

On peut rediriger les E/S standard :

- **<** : pour redéfinir **stdin**
- **>** : pour redéfinir **stdout**
- **>>** : idem, mais les données sont ajoutées en fin de fichier

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (1)

1 Introduction

Le S.G.F. gère une structure d'arbre :

- la racine est désignée par /
- les nœuds sont les répertoires non vides
- les feuilles sont les répertoires "vides" et les fichiers.

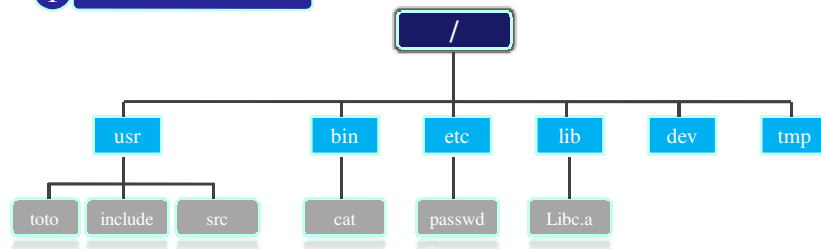


Les chemins sont décrits avec le séparateur /

⚠ Les noms de fichiers et répertoires sont limités à 14 caractères (/ est interdit et - ne peut être le premier caractère).

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (1-1)

1 Introduction



- ❑ Le répertoire **/usr** contient les répertoires du système et ceux des utilisateurs.
- ❑ Le répertoire **/etc** contient les fichiers de contrôle
- ❑ Le répertoire **/bin** contient les commandes systèmes
- ❑ Le répertoire **/lib** contient les bibliothèques du système
- ❑ Le répertoire **/dev** contient les fichiers des périphériques
- ❑ Le répertoire **/tmp** contient les fichiers temporaires

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (1-2)

1 Introduction

- / Le répertoire racine (root). Point de départ de toute l'arborescence.
- /bin Contient les programmes exécutables nécessaires en mode mono-utilisateur pour démarrer ou réparer le système.
- /boot Contient les fichiers statiques utilisés par le chargeur du système. Ce répertoire ne contient que les fichiers nécessaires pour le chargeur.
- /dev Fichiers spéciaux se rapportant à des périphériques physiques.
- /etc Contient les fichiers de configuration spécifiques à la machine.
- /home Répertoires pour les comptes utilisateurs locaux à la machine
- /lib Ce répertoire doit contenir les bibliothèques partagées nécessaires pour démarrer le système et pour utiliser les commandes de /bin.
- /mnt Ce répertoire sert de point de montage temporaire pour d'autres systèmes de fichiers. Il est parfois lui-même séparé en plusieurs sous-répertoires (/mnt/cdrom, /mnt/disque_a, /mnt/disque_b...).
- /sbin Comme /bin, ce répertoire contient les commandes nécessaires au démarrage
- /tmp Ce répertoire sert à contenir des fichiers temporaires que l'on peut détruire régulièrement, par un script périodique, ou au démarrage du système.
- /usr Ce répertoire est généralement monté depuis une partition séparée. Il ne devrait contenir que des données partageables, en lecture-seule, afin d'être monté par plusieurs machines utilisant Linux. (/usr/X11R6/ /usr/doc etc)

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (1-3)

1 Introduction

- Le **nom de chemin** ou **path name** d'un fichier est la concaténation, depuis la racine, de tous les répertoires qu'il est nécessaire de traverser pour y accéder, chacun étant séparé par le caractère « / ». C'est un **chemin absolu** : **/home/toto/Docs/Backup/fic.bak**
- Un nom de chemin peut aussi être **relatif** à sa position courante dans le répertoire : **Source/fic.c**
- Le système (ou le Shell) mémorise la position actuelle d'un utilisateur dans le système de fichier, le répertoire actif. On peut donc accéder à un autre répertoire de l'arborescence depuis l'emplacement actuel sans taper le chemin complet.
- Pour se déplacer dans les répertoires, on utilise la commande **cd**. Le « .. » permet d'accéder au répertoire de niveau supérieur. Le « . » définit le répertoire actif (répertoire courant). La commande **ls** permet de lister le contenu du répertoire. La commande **pwd** (print working directory) affiche le chemin complet du répertoire actif.

» Ces commandes seront étudiées avec plus de détails ultérieurement

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (2)

2 Les i-nodes

A tout fichier est attaché un **nœud** d'index, ou **i-node** contenant des informations sur ce fichier.

Cette structure est décrite dans **/usr/include/sys/inode.h** et contient les champs suivants :

- le type (fichier ordinaire, spécial, catalogue ,...).
- les protections.
- les dates de création, accès et mise à jour.
- la taille du fichier en octets.
- le nombre de liens.
- les éléments d'identification du propriétaire et de son groupe.
- l'adresse physique d'implantation sur disque.

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (2-1)

2 Les i-nodes

Tout fichier est référencé de manière unique par son numéro d'**i-node** ,

L'**i-node** de n° **0** n'est jamais utilisée.

L'**i-node** de n° **1** permet la gestion des blocs défectueux.

L'**i-node** de n° **2** est attribué à la racine de l'arborescence.

Dans chaque répertoire, '**.**' est un lien au répertoire lui-même
(dans le répertoire, en regard de ce "nom", figure le n° d'i-node du répertoire).

Dans chaque répertoire, '**..**' est un lien au répertoire père
(dans le répertoire, en regard de ce nom, figure le n° d'i-node du répertoire père).

Un répertoire **UNIX** n'est donc jamais totalement vide.

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (3)

3 Types de fichiers

On distingue :

- les **fichiers ordinaires** : données, programmes, textes.
- les **fichiers spéciaux** : ils correspondent à des ressources. Ils n'ont pas de taille.
(Les opérations de lecture/écriture sur ces fichiers activent les dispositifs physiques associés).
- les **(fichiers) répertoires** : ils contiennent des informations sur d'autres fichiers et permettent la structuration arborescente.
- les **fichiers de liens symboliques**.

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4)

4 Protection des fichiers

A la création d'un compte d'utilisateur, l'administrateur affecte celui-ci à un groupe.

Chaque groupe possède un nom.

On peut donner des droits d'accès particuliers à certains fichiers pour les membres d'un même groupe.

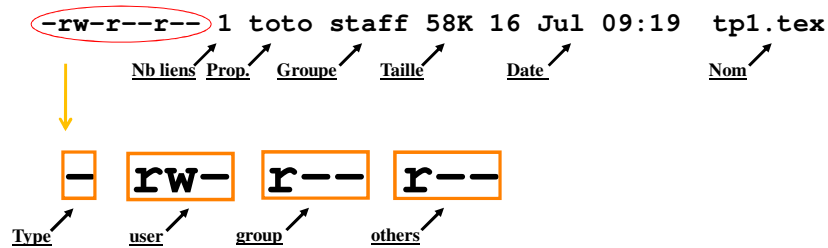
Au total, il existe 3 types d'accès :

propriétaire (u = user),
groupe (g = group),
autres (o = other).

L'**ensemble** (user+group+other) est désigné par a (**all**).

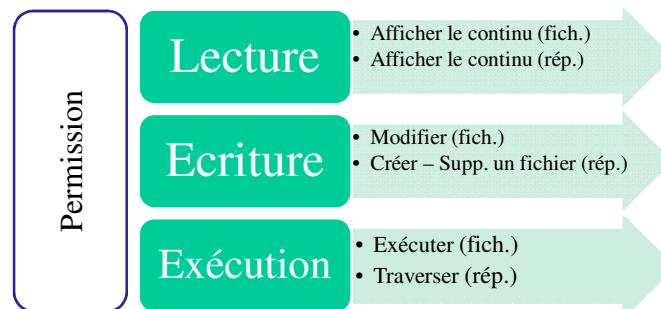
Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4-1)

4 Protection des fichiers



Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4-2)

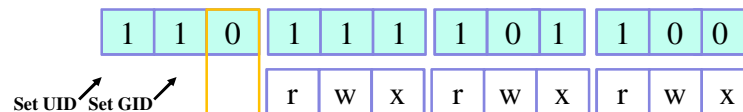
4 Protection des fichiers



Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4-3)

4 Protection des fichiers

L'expression des droits nécessite 12 bits :



Pour un fichier, r = droit de lecture, w = droit d'écriture, x = droit d'exécuter (1 = oui, 0 = non)

Pour un répertoire,

r : on peut consulter la liste des fichiers qui y sont contenus

w : on peut créer ou effacer des fichiers à l'intérieur

x : on peut utiliser ce répertoire en argument d'une commande et s'y positionner

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4-4)

4 Protection des fichiers

Les 9 bits des droits d'accès sont précédés par 3 autres bits pour compléter la description des protections :

- le bit "**set-uid**" : quand il vaut 1 pour un fichier exécutable, le processus d'exécution a les droits du propriétaire du fichier et non de l'utilisateur qui le lance.
- le bit "**set-gid**" : même rôle, mais relativement au groupe
- l'attribut spécifique : outre la valeur "-" pour les fichiers ordinaires, il peut prendre la valeur:
 - d** : fichier répertoire
 - c** : fichier spécial en **mode caractère** (cas des terminaux)
 - b** : fichier spécial en **mode bloc** (cas de lecteurs de disques ou de bandes)
 - s** : bit de collage ou "**sticky-bit**" : il maintient le fichier en zone de recouvrement (swap) à partir de laquelle le chargement est plus rapide

Systèmes d'exploitation: Unix – Système de Gestion de Fichiers (4-5)

4 Protection des fichiers

ls -l » cette commande sera étudiée avec plus de détails ultérieurement

```
-rwx----- 1 mahmoudr info      1030 oct  5 2001 asm.s~  
drwx----- 3 mahmoudr info      4096 mai 10 12:08 backcompte  
-rwx----- 1 mahmoudr info         0 fév  5 2004 boot.dd  
-rwx----- 1 mahmoudr info       466 avr  1 2003 config-reseau-esiee.txt~  
drwx----- 4 mahmoudr info      4096 mar  8 2004 DEA-eleves-ATEME  
drwx----- 4 mahmoudr info      4096 sep  7 14:09 Desktop  
-rw-rw-r-- 1 mahmoudr info        90 jun 15 10:08 dtwm.db  
drwx----- 2 mahmoudr info      4096 jan 30 2004 Enseign2004  
-rwx----- 1 mahmoudr info    1171962 mai  3 11:41 Enseignement.tar.gz
```



La commande ls -las permet l'affichage succinct des numéros d'i-nodes du répertoire.

Systèmes d'exploitation: Unix – Les commandes (1)

1 Format

Cde [- Option(s)] [arguments]



Respecter la casse et les espaces .

Systèmes d'exploitation: Unix – Les commandes (2)

2 Commande de base (Bienvenue)

man nom_de_la_commande

Afficher la description complète de la commande nommée.

La documentation est structurée selon le format suivant :

- Name : le nom est une description de ses fonctionnalités
- Synopsis : le syntaxe de la commande
- Description : la description complète de la commande
- Options: la description détaillée des options supportées
- Files : une liste des fichiers utilisés par la commande
- See also : une liste de commandes liées
- Bugs: les erreurs connues et non encore corrigées

Systèmes d'exploitation: Unix – Les commandes (2-1)

2 Commande de base (Bienvenue)

date : Afficher la date du jour, suivant le format du système

who : Afficher les utilisateurs connectés à cet instant au système

cal 2012: afficher le calendrier de l'année 2012

cal 12 2012: afficher le mois de Dec de l'année 2012

Systèmes d'exploitation: Unix – Les commandes (2-2)

2 Commande de base (Bienvenue)

passwd : modifier le mot de passe de l'utilisateur connecté

```
$ passwd  
Old password:  
New password:  
Reenter password:
```

Systèmes d'exploitation: Unix – Les commandes (2-3)

2 Commande de base (Bienvenue)

su [-] utilisateur [-c commande]

- Le système permet dans certains cas à un utilisateur connecté de changer de nom en cours de travail avec la commande **su**.
- Le mot de passe du nouvel utilisateur sera demandé.
- Si - est précisé, l'environnement du nouvel utilisateur est chargé.
- Si -c est précisé les commandes qui suivent sont exécutées.

Systèmes d'exploitation: Unix – Les commandes (2-4)

2 Commande de base (Bienvenue)

echo: elle transmet tous ses paramètres sur écran (ou canal de sortie standard).

Le texte est quelconque mais peut aussi admettre quelques caractères de formatage.

Caractère	Effet
\n	Saut de ligne (newline)
\b	Retour arrière (backslash)
\t	Tabulation
\c	Pas de retour à la ligne (carriage)
\\	Affiche \
\\$	Affiche \$
\valeur	Affiche le caractère spécial de code octal valeur

Systèmes d'exploitation: Unix – Les commandes (3)

2 Commande de base (fichier)

cp fich1 fich2: copier le contenu du fich1 dans le fichier fich2

mv ancien_nom fich1 nouveau_nom fich1: renommer fich1

mv nom fich1 nom rep1 : déplacer fich1 dans rep1

rm : supprimer un fichier

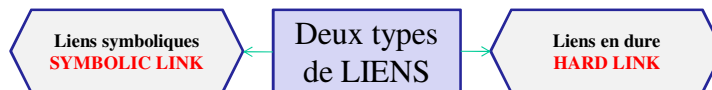
cat : affiche le contenu d'un fichier

Systèmes d'exploitation: Unix – Les commandes (3-1)

2 Commande de base (fichier)

- Un **lien** permet de donner plusieurs noms à un même fichier
- Un **lien** permet de pointer un fichier sur un autre.
- On fait appel à la commande **ln** :

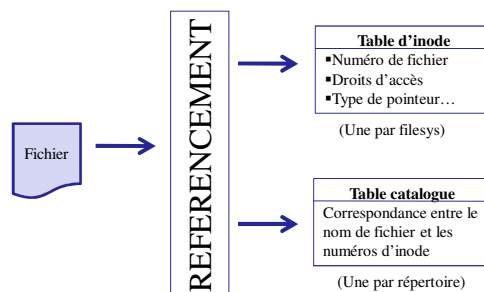
```
ln [options] fic1 fic2  
ln [options] fic1 rep1  
ln [options] rep1 fic2
```



Systèmes d'exploitation: Unix – Les commandes (3-2)

2 Commande de base (fichier)

- Un **hard link** permet d'ajouter une référence sur un inode.



Systèmes d'exploitation: Unix – Les commandes (3-3)

2 Commande de base (fichier)

- Un **hard link** rajoute donc une association dans la table catalogue entre un nom et un inode.
- Un **hard link** ne modifie pas les droits du fichier .
- Un **hard link** ne permet pas d'affecter **plusieurs nom** à un **même répertoire**.
- Un **hard link** ne permet pas d'effectuer des liens depuis ou vers un autre **filesystem**.

Systèmes d'exploitation: Unix – Les commandes (3-4)

2 Commande de base (fichier)

- L'exemple suivant montre que les hard links n'ont pas de type particulier et sont considérés comme des fichiers ordinaires.
- On constate que chacun a deux liens. Et c'est logique puisque deux fichiers pointent sur **le même inode**.
- On constate aussi en résultat du **ls -i** que **fic1** et **fic2** ont le **même inode**, à savoir **484**.

```
$ cd
$ touch fic1
$ ln fic1 fic2
$ ls
fic1 fic2
$ ls -l
-rw-r--r-- 2 oracle system 0 Jul 25 11:59 fic1
-rw-r--r-- 2 oracle system 0 Jul 25 11:59 fic2
$ ls -i
484 fic1 484 fic2
```

Systèmes d'exploitation: Unix – Les commandes (3-5)

2 Commande de base (fichier)

- Un **lien symbolique** crée une sorte d'alias : un fichier spécial contenant une donnée pointant vers un autre chemin.
- Un **lien symbolique** ne possède pas les limitations du hard link.
- Il est possible d'effectuer des liens entre plusieurs FileSystems, et vers des répertoires.
- La suppression de tous les liens symboliques **n'entraîne que** la suppression de ces liens, pas du fichier pointé.
- La suppression du fichier pointé **n'entraîne pas** la suppression des liens symboliques associés.

Systèmes d'exploitation: Unix – Les commandes (3-6)

2 Commande de base (fichier)

- Sur le même exemple précédent, on commence par supprimer **fic2** puis on crée un lien symbolique.
- Avec **ls -l** on remarque bien que le lien symbolique est en fait un fichier spécial de type « l » pointant vers un autre fichier.

```
$ rm fic2
$ ln -s fic1 fic2
$ ls -l
-rw-r--r-- 1 oracle system 0 Jul 25 11:59 fic1
lrwxrwxrwx 1 oracle system 4 Jul 25 12:03 fic2 -> fic1
$ ls -l
484 fic1 635 fic2
$ ls -F
fic1 fic2@
```

Systèmes d'exploitation: Unix – Les commandes (3-7)

2 Commande de base (fichier)

- Attention : les droits indiqués sont ceux du fichier spécial.
- Lors de son utilisation, ce sont les droits du fichier ou du dossier pointés qui prennent le dessus.
- On distingue (**ls -F**) le caractère « @ » indiquant qu'il s'agit d'un lien symbolique.
- On remarque aussi que les inodes sont différents et que les compteurs sont tous à 1.

```
$ rm fic2
$ ln -s fic1 fic2
$ ls -l
-rw-r--r-- 1 oracle system 0 Jul 25 11:59 fic1
lrwxrwxrwx 1 oracle system 4 Jul 25 12:03 fic2 -> fic1
$ ls -li
484 fic1 635 fic2
$ ls -F
fic1 fic2@
```

Systèmes d'exploitation: Unix – Les commandes (4-1)

2 Commande de base (répertoire)

mkdir rep1 : créer un nouveau répertoire rep1

rmdir rep1: supprimer rep1 (répertoire vide)

mv nom_rep1 nom_rep2 : renommer ou déplacer un répertoire

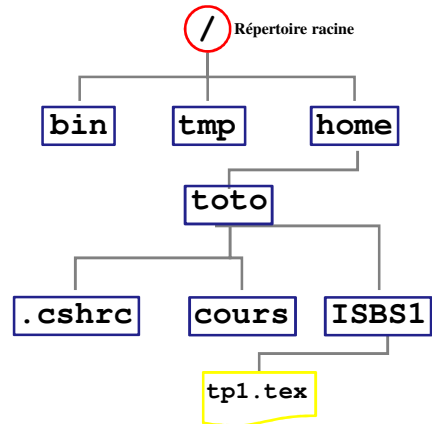
cd .. : changer le répertoire courant en son répertoire parent

find rep -name fich -print: afficher les noms complets de tous les fichiers de nom fich se trouvant dans rep

Systèmes d'exploitation: Unix – Les commandes (4-2)

2 Commande de base (répertoire)

- le répertoire de login: **~**
- le répertoire courant: **.**
- le répertoire supérieur: **..**
- connaître le rép. courant: **pwd**
- lister le contenu: **ls**
- chemin d'accès au fichier **tp1.tex**:
 - /home/toto/ISBS1/tp1.tex
 - ou bien: ~/ISBS1/tp1.tex



A.U 2012/2013

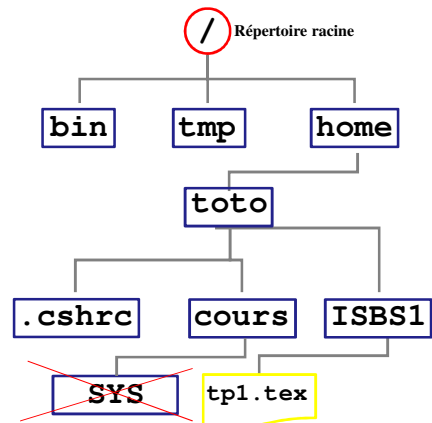
Ramzi Mahmoudi

107

Systèmes d'exploitation: Unix – Les commandes (4-3)

2 Commande de base (répertoire)

- **pwd** retourne:
`/home/toto/ISBS1`
- se déplacer: **cd**
`[/home/toto/ISBS1]$ cd ..`
`[/home/toto]$`
`[/home/toto]$ cd /cours`
`[/cours]$`
- créer un répertoire: **mkdir**
`[/cours]$ mkdir SYS`
- supprimer un répertoire: **rmdir**
`[/cours]$ rmdir SYS`



A.U 2012/2013

Ramzi Mahmoudi

108

Systèmes d'exploitation: Unix – Les commandes (5-1)

2 Commande de base (ls)

ls : afficher le contenu d'un répertoire

- La commande **ls** permet de lister le contenu d'un répertoire (catalogue) en lignes ou colonnes.
- Elle supporte plusieurs options:

Option	Signification
-l	Sortie de chaque information des fichiers
-F	Ajoute un « / » au nom d'un répertoire, un « * » au nom d'un exécutable, un « l » pour un tube nommé et « @ » pour un lien symbolique.
-a	Affiche toutes les entrées, y compris « . », « .. » et les fichiers cachés (qui commencent par un .)
-d	affiche le nom (et les attributs) des répertoires et pas leur contenu.
-l	Affiche les numéros d'inode.

A.U 2012/2013

Ramzi Mahmoudi

109

Systèmes d'exploitation: Unix – Les commandes (5-2)

2 Commande de base (ls)

ls : afficher le contenu d'un répertoire

- La commande **ls** permet de lister le contenu d'un répertoire (catalogue) en lignes ou colonnes.
- Elle supporte plusieurs options:

Option	Signification
-R	Mode récursif. Rentre dans les répertoires et affiche leur contenu.
-r	Inverse l'ordre du tri (à l'envers)
-t	Tri par date de modification
-c	Affiche la date de création (si -l) ou tri par date de création (si -t)
-C	Les noms sont affichés sur plusieurs colonnes
-u	Affiche la date d'accès (-l) ou tri par date d'accès (-t)
-l	Liste sur une seule colonne.

A.U 2012/2013

Ramzi Mahmoudi

110

Systèmes d'exploitation: Unix – Les commandes (6-1)

2 Commande de base (utilitaires)

le disque est “découpé” en **partitions** :

- commandes & applications
- comptes utilisateurs
- swap
- fichiers temp
- périphériques (disques, ...)
- ...

A.U 2012/2013

Ramzi Mahmoudi

111

Systèmes d'exploitation: Unix – Les commandes (6-2)

2 Commande de base (utilitaires)

Tous les disques amovibles (disquette, cdrom, clé usb) => **/mnt**

- Exemple pour utiliser une disquette:
 - Montage: **mount** /mnt/floppy
 - Lire/Ecrire dans /mnt/floppy
 - Démontage: **umount** /mnt/floppy
- idem pour clés usb

A.U 2012/2013

Ramzi Mahmoudi

112

Systèmes d'exploitation: Unix – Les commandes (6-3)

2 Commande de base (utilitaires)

- **gzip/gunzip** : compresse et décompresse
 - gzip tp.tar → transforme tp.tar en tp.tar.gz
 - gunzip tp.tar.gz → transformation inverse

Systèmes d'exploitation: Unix – Les commandes (6-3)

2 Commande de base (utilitaires)

- **split** : découper un fichier en petits fichiers

```
-rwx----- 1 mahmoudr info 1171962 mai 3 11:41 Enseignement.tar.gz
split -b700k Enseignement.tar.gz
-rw-r--r-- 1 mahmoudr info 716800 sep 24 13:53 xaa
-rw-r--r-- 1 mahmoudr info 455162 sep 24 13:53 xab
```
- **cat** : réassembler les morceaux

```
cat xaa xab > g_enseignement
-rw-r--r-- 1 grandpit info 1171962 sep 24 13:58 g_enseignement
```

Systemes d'exploitation: Unix – Les commandes (7)

2 Commande de base

Cde1 [- Option(s)] [arguments] ; Cde2 [- Option(s)] [arguments]



Pour lier les commandes, on sépare les commandes par le caractère « ; ».

Systemes d'exploitation: Unix – Les commandes (8)

3 Méta-caractères

Caractères spéciaux:

! ^ * ? [] \ ; & < > | >>

Systèmes d'exploitation: Unix – Les commandes (8-1)

3 Méta-caractères

- L'astérisque ou étoile: *****
 - interprété comme toute suite de caractères alphanumériques
 - utiliser **avec précaution** (commande **rm** par ex...)
- Le point d'interrogation: **?**
 - remplace **1** seul caractère alphanumérique
- Le point-virgule: **;**
 - Séparateur de commandes
- Les crochets: **[]**
 - Remplace un caractère choisi parmi ceux énumérés entre les crochets
- L'anti-slash: ****
 - Inhibe la signification du méta-caractère qui suit

A.U 2012/2013

Ramzi Mahmoudi

117

Systèmes d'exploitation: Unix – Les commandes (8-2)

3 Méta-caractères

Interprétation des chaînes de caractères :

- ☐ Texte entre **' '** (simples quotes):
Le texte n'est pas interprété mais considéré comme un mot
- ☐ Texte entre **" "** (doubles quotes):
Seuls sont interprétés les métacaractères **\$**, **** et **`**
- ☐ Texte entre **` `** (anti quotes):
Considérez comme une commande à interpréter, et c'est le résultat qui sera utilisé.

A.U 2012/2013

Ramzi Mahmoudi

118

Systèmes d'exploitation: Unix – Les commandes (8-3)

3 Méta-caractères



```
...inal — zsh (ttyp2) — %2
[lewandow:~] echo 'date'
date
[lewandow:~]
[lewandow:~]
[lewandow:~] echo `date`
Mon Jul 19 17:59:10 CEST 2004
[lewandow:~]
[lewandow:~]
[lewandow:~] echo "date = `date`"
date = Mon Jul 19 17:59:19 CEST 2004
[lewandow:~]
[lewandow:~]
[lewandow:~]
```

A.U 2012/2013

Ramzi Mahmoudi

119

Systèmes d'exploitation: Unix – Les commandes (8-4)

3 Méta-caractères

Exemples:

- ❑ **echo ***
Tous les fichiers sauf ceux dont le nom commence par un point
- ❑ **echo *c**
Tous les fichiers dont le nom se termine par un 'c'
- ❑ **echo .***
Tous les fichiers dont le nom commence par un point
- ❑ **echo [0-9]***
Tous les fichiers dont le nom commence par un chiffre

A.U 2012/2013

Ramzi Mahmoudi

120

Systèmes d'exploitation: Unix – Les commandes (8-5)

3 Méta-caractères

Exemples:

```
❑$ ls -d b*  
    backcompte backup back_win bin boot.dd  
  
❑$ ls *.html  
    index.html som.html  
  
❑$ ls -d *ack*  
    backcompte backup back_win  
  
❑$ ls ???l  
    pgml tstl igml
```

Systèmes d'exploitation: Unix – Les commandes (8-6)

3 Méta-caractères

Exemples:

```
❑$ ls [a-c]*  
    agm atft bxmm cmpo cpro  
  
❑$ [0-9]*  
    1mp 6mmh  
  
❑$ .x[nm]*  
    test.xml
```


Systèmes d'exploitation: Unix – Les commandes (9)

4 Commandes de gestion des processus

Processus = objet dynamique qui représente un programme en cours d'exécution et son contexte

Caractéristiques:

1. identification (pid)
2. identification du proc. parent (ppid)
3. propriétaire
4. priorité
5. ...

Systèmes d'exploitation: Unix – Les commandes (9-1)

4 Commandes de gestion des processus

Pour voir les processus en cours: **ps**. Les informations sont sous la forme :

PID	numéro d'identification du processus
TT	numéro du terminal qui a lancé le processus
STAT	Etat du processus
TIME	Temps utilisé par le processus en mm:ss
COMMAND	le nom de la commande avec ses arguments

Systèmes d'exploitation: Unix – Les commandes (9-2)

4 Commandes de gestion des processus

[lewandow:~] ps

PID	TT	STAT	TIME	COMMAND
3899	p1	S	0:00.08	-zsh
4743	p1	S+	0:00.14	emacs
4180	std	S	0:00.04	-zsh

Numéro processus → Etat de processus → Temps CPU utilisé → Commande exécuté

Systèmes d'exploitation: Unix – Les commandes (9-3)

4 Commandes de gestion des processus

[lewandow:~] ps

PID	TT	STAT	TIME	COMMAND
3899	p1	S	0:00.08	-zsh

Etats des processus

- R actif
- T bloqué
- P en attente de page
- D en attente de disque
- S endormi
- IW swappé
- Z tué

Systèmes d'exploitation: Unix – Les commandes (9-4)

4 Commandes de gestion des processus

```
[lewandow:~] kill PID
```

Terminer le processus indiqué

Systèmes d'exploitation: Unix – Les commandes (10)

5 Lancement en arrière plan d'une commande

<code>cp fic1 fic2&</code>	Lance en arrière plan la commande cp avec fic1 et fic2 en arguments, affiche le numéro de processus créé et rend la main à l'utilisateur.
<code>nohup longue_cmd&</code>	Pour éviter qu'une commande ne soit détruite par une interruption (coupure de ligne par exemple), on utilise la commande nohup : interdit les interruptions sur la commande longue_cmd
<code>nice cmd_non_urgente&</code>	Pour ne pas gêner les autres utilisateurs par l'exécution d'une commande non urgente, on utilise la commande nice : lance la commande en arrière plan avec une faible priorité

Systèmes d'exploitation: Unix – Les commandes (11-1)

6 Redirection des entrées/sorties standard

Redirection : c'est la possibilité de rediriger l'affichage de l'écran vers un fichier ou remplacer la saisie clavier par le contenu d'un fichier.

Unix utilise des **canaux d'entrées/sorties** pour lire et écrire ses données.

Par défaut le **canal d'entrée** est le **clavier**, et le **canal de sortie** est l'**écran**.

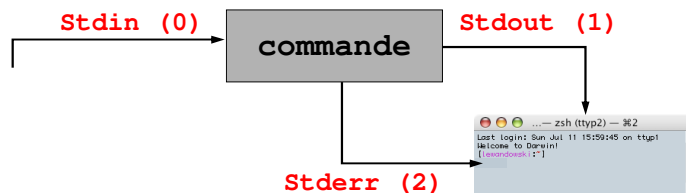
Un troisième canal, le **canal d'erreur**, est aussi redirigé vers l'**écran**.

Il est donc possible de rediriger ces canaux vers des fichiers, ou du flux texte de manière transparente pour les commandes Unix.

Systèmes d'exploitation: Unix – Les commandes (11-2)

6 Redirection des entrées/sorties standard

Une commande ouvre **3 descripteurs (0,1,2)** de fichiers; par défaut:



Systèmes d'exploitation: Unix – Les commandes (11-3)

6 Redirection des entrées/sorties standard

<	redirige l'entrée standard
>	redirige la sortie standard
>>	concatène la sortie standard
2>	redirige la sortie d'erreur
>&	redirige la sortie standard et la sortie d'erreur

Systèmes d'exploitation: Unix – Les commandes (11-4)

6 Redirection des entrées/sorties standard

En sortie :

On se sert du caractère « > » pour rediriger la sortie standard (celle qui va normalement sur écran).

On indique ensuite le nom du fichier où seront placés les résultats de sortie.

```
$ ls -l > resultat.txt
$ cat resultat.txt
total 1
-rw-r--r-- 1 Administ ssh_user 0 Jul 4 12:04 TOTO
-rw-r--r-- 1 Administ ssh_user 0 Jul 25 15:13 resultat.txt
-rw-r--r-- 1 Administ ssh_user 171 Jul 25 15:13 test.txt
```

Systèmes d'exploitation: Unix – Les commandes (11-5)

6 Redirection des entrées/sorties standard

Si le fichier n'existe pas, il sera créé.
S'il existe, son contenu sera écrasé, même si la commande tapée est incorrecte.

Le shell commence d'abord par créer le fichier puis exécute ensuite la commande.

Pour rajouter des données à la suite du fichier, donc sans l'écraser, on utilise la double redirection « >> ».

Le résultat est ajouté à la fin du fichier.

A.U 2012/2013

Ramzi Mahmoudi

133

Systèmes d'exploitation: Unix – Les commandes (11-5)

6 Redirection des entrées/sorties standard

```
$ ls -l > resultat.txt
$ date >> resultat.txt
$ cat resultat.txt
total 1
-rw-r--r-- 1 Administ ssh_user 0 Jul 4 12:04 TOTO
-rw-r--r-- 1 Administ ssh_user 0 Jul 25 15:13 resultat.txt
-rw-r--r-- 1 Administ ssh_user 171 Jul 25 15:13 test.txt
Thu Jul 25 15:20:12 2012
```

A.U 2012/2013

Ramzi Mahmoudi

134

Systèmes d'exploitation: Unix – Les commandes (11-6)

6 Redirection des entrées/sorties standard

En entrée :

Les commandes qui attendent des données ou des paramètres depuis le clavier peuvent aussi en recevoir depuis un fichier, à l'aide du caractère «<».

Un exemple avec la commande « wc » (word count) qui permet de compter le nombre de lignes, de mots et de caractères d'un fichier.

```
$ wc < resultat.txt  
4 29 203
```

Systèmes d'exploitation: Unix – Les commandes (11-7)

6 Redirection des entrées/sorties standard

exemples:

<code>ls -l >liste</code>	crée/écrase le fichier liste et y dirige la sortie de 'ls'
<code>date >>liste</code>	ajoute à la fin du fichier liste la sortie de 'date'
<code>wc -l <liste</code>	envoie comme entrée à la commande 'wc' le fichier liste

Systèmes d'exploitation: Unix – Les commandes (11-8)

6 Redirection des entrées/sorties standard

On peut rediriger le canal de sortie d'erreur (descripteur 2) vers un autre fichier (2>).

```
$ rmdir rep  
rmdir: `rep`: No such file or directory
```

```
$ rmdir rep 2>error.log  
$ cat error.log  
rmdir: `rep`: No such file or directory
```

Systèmes d'exploitation: Unix – Les commandes (11-9)

6 Redirection des entrées/sorties standard

Il est possible de rediriger les deux canaux de sortie dans un seul et même fichier, en les liant.

On utilise pour cela le caractère « >& ».

Le shell interprète les redirections de cette manière: Les redirections étant en principe en fin de commande, le shell recherche d'abord les caractères « <, >, >> » en fin de ligne.

Systèmes d'exploitation: Unix – Les commandes (11-9)

6 Redirection des entrées/sorties standard

Pour grouper les deux canaux de sortie et d'erreur dans un même fichier, il faut procéder comme suit :

```
$ ls -l > resultat.txt 2>&1
```

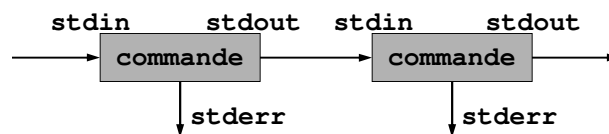
La sortie 2 est redirigée vers la sortie 1, donc les messages d'erreurs passeront par la sortie standard. Puis le résultat de la sortie standard de la commande ls est redirigé vers le fichier resultat.txt.

Ce fichier contiendra donc à la fois la sortie standard et la sortie d'erreur.

Systèmes d'exploitation: Unix – Les commandes (12)

7 Les tubes & les filtres

Tube (pipeline) : | => “connecter 2 commandes”



ex: combien de fichiers dans le rep. courant ?

sans pipe:

```
ls > temp ; wc -l < temp ; rm temp
```

avec un pipe:

```
ls | wc -l
```

Systèmes d'exploitation: Unix – Les commandes (13)

7 Les tubes & les filtres

Les filtres :

- C'est un programme sachant écrire et lire des données par les canaux standards d'entrée et de sortie. Il en modifie ou traite éventuellement le contenu.
- **wc** est un filtre.
- Les utilitaires sans être obligatoirement des filtres permettent un certain nombre d'actions sur des fichiers ou leur contenu comme le formatage ou l'impression.

Systèmes d'exploitation: Unix – Les commandes (13-1)

7 Les tubes & les filtres

Les filtres :

grep	Recherche, dans le fichier passé en paramètre, les lignes vérifiant une Expression régulière donnée
	Syntaxe : grep <i>expr_reg</i> [fichier]
	Ex: grep 'toto' essai cherche dans <i>essai</i> toutes les lignes qui contiennent le mot <i>toto</i> grep '^[A-Z]' essai cherche dans <i>essai</i> toutes les lignes qui commencent par une majuscule

Systèmes d'exploitation: Unix – Les commandes (13-2)

7 Les tubes & les filtres

Les filtres :

grep	cat essai java Architecture Système algorithme
	grep '^ [A-Z]' essai Architecture Système

Systèmes d'exploitation: Unix – Les commandes (13-3)

7 Les tubes & les filtres

Les filtres :

cut	Sélectionne uniquement certaines colonnes du fichier passé en paramètre Options: -f<liste> : liste des champs à garder -c<liste> : liste des colonnes à garder -d<char> : séparateur de champs
	Le format de sélection de colonne est le suivant : <ul style="list-style-type: none">▪ La numérotation des colonnes démarre à 1.▪ une colonne seule, (ex -c2 pour la colonne 2)▪ une plage (ex -c2-4 pour les colonnes 2, 3 et 4)▪ une liste de colonnes (ex -c1,3,6 pour les colonnes 1, 3 et 6)▪ les trois en même temps (ex -c1-3,5,6,12-)

Systèmes d'exploitation: Unix – Les commandes (13-4)

7 Les tubes & les filtres

Les filtres :

cut	<pre>\$ cut -c1-5 fic.txt Produ sour disqu ecran clavi</pre>	fic.txt
	<pre>\$ cut -c1-3,10-12,15 fic.txt Prorixu sou0010 dis05 2 ecr 20 cla5 3</pre>	<pre>Produit prix quantites souris 30001 505 disque 1005 332 ecran 300 200 clavier 45 30</pre>

⚠ Attention : Un espace est considéré comme un caractère !

A.U 2012/2013

Ramzi Mahmoudi

145

Systèmes d'exploitation: Unix – Les commandes (13-5)

7 Les tubes & les filtres

Les filtres :

cut	▪ La commande cut permet aussi de sélectionner des champs.
	▪ Ces champs doivent être par défaut délimités par une tabulation, mais l'option -d permet de sélectionner un autre caractère (espace ' ', point virgule ';').
	▪ La sélection des champs est identique à celle des colonnes.
	▪ Leur numérotation démarre à 1.
	▪ cut -dc -fChamps [fic1...]

A.U 2012/2013

Ramzi Mahmoudi

146

Systèmes d'exploitation: Unix – Les commandes (13-6)

7 Les tubes & les filtres

Les filtres :

cut	<pre>\$ cat fic Produit prix quantites souris 30001 505 disque 1005 332 ecran 300 200 clavier 45 30</pre>	<pre>fic.txt Produit prix quantites souris 30001 505 disque 1005 332 ecran 300 200 clavier 45 30</pre>
	<pre>\$ cut -f1 fic Produit souris disque ecran clavier</pre>	

⚠ Attention : Un espace est considéré comme séparateur de champ !

A.U 2012/2013

Ramzi Mahmoudi

147

Systèmes d'exploitation: Unix – Les commandes (13-7)

7 Les tubes & les filtres

Les filtres :

cut	<pre>\$ cat fic Produit prix quantites souris 30001 505 disque 1005 332 ecran 300 200 clavier 45 30</pre>	<pre>fic.txt Produit prix quantites souris 30001 505 disque 1005 332 ecran 300 200 clavier 45 30</pre>
	<pre>\$ cut -f1,3 fic Produit quantites souris 505 disque 332 ecran 200 clavier 30</pre>	

A.U 2012/2013

Ramzi Mahmoudi

148

Systèmes d'exploitation: Unix – Les commandes (13-8)

7 Les tubes & les filtres

Les filtres :

cut

```
$ cat /etc/group
system:*:0:root,ftp,sa_sgd,arret,ptladm,ptlrgt
daemon:*:1:daemon
uucp:*:2:uucp
kmem:*:3:root,ingres
mem:*:3:root,ingres
```

```
$ cut -d: -f1,3
system:0
daemon:1
uucp:2
kmem:3
mem:3
```

A.U 2012/2013

Ramzi Mahmoudi

149

Systèmes d'exploitation: Unix – Les commandes (13-9)

7 Les tubes & les filtres

Les filtres :

sort

- La commande **sort** permet de trier des lignes.
- Le tri s'effectue sur tout le tableau et en ordre **croissant** (par défaut)
- Le tri est possible sur un ou plusieurs champs.
- Le séparateur de champs par défaut est la tabulation ou au moins un espace.
- S'il y a plusieurs espaces, le premier est le séparateur.
- La numérotation des champs démarre à 0.

sort [options] [+pos1 [-pos2] ...] [fic1...]

Avec : **+pos1** est le premier champ, **-pos2** le dernier.

A.U 2012/2013

Ramzi Mahmoudi

150

Systèmes d'exploitation: Unix – Les commandes (13-10)

7 Les tubes & les filtres

Les filtres :

sort	■\$ cat liste			
	souris	optique	30	15
	dur	30giga	100	30
	dur	70giga	150	30
	disque	zip	12	30
	disque	souple	10	30
	ecran	15	150	20
	ecran	17	300	20
	ecran	19	500	20
	clavier	105	45	30
	clavier	115	55	30
	carte	son	45	30
	carte	video	145	30

A.U 2012/2013

Ramzi Mahmoudi

151

Systèmes d'exploitation: Unix – Les commandes (13-11)

7 Les tubes & les filtres

Les filtres :

sort	■\$ sort +0 liste			
	carte	son	45	30
	carte	video	145	30
	clavier	105	45	30
	clavier	115	55	30
	disque	souple	10	30
	disque	zip	12	30
	dur	30giga	100	30
	dur	70giga	150	30
	ecran	15	150	20
	ecran	17	300	20
	ecran	19	500	20
	souris	optique	30	15

A.U 2012/2013

Ramzi Mahmoudi

152

Systèmes d'exploitation: Unix – Les commandes (13-12)

7 Les tubes & les filtres

Les filtres :

sort	<u>Options</u>
	-n Tri numérique, idéal pour le colonnes de chiffres
	-r Reverse, tri en ordre décroissant.

A.U 2012/2013

Ramzi Mahmoudi

153

Systèmes d'exploitation: Unix – Les commandes (13-13)

7 Les tubes & les filtres

Les filtres :

sort	■sort -n -r +2 -3 liste			
	ecran	19	500	20
	ecran	17	300	20
	dur	70giga	150	30
	ecran	15	150	20
	carte	video	145	30
	dur	30giga	100	30
	clavier	115	55	30
	clavier	105	45	30
	carte	son	45	30
	souris	optique	30	15
	disque	zip	12	30
	disque	souple	10	30



Il est aussi possible de démarrer le tri à partir d'un certain caractère d'un champ. Pour cela on spécifie le « .pos » :
+0.3 commencera le tri à partir du quatrième caractère du champ 0.

A.U 2012/2013

Ramzi Mahmoudi

154

Systèmes d'exploitation: Unix – Les commandes (13-14)

7 Les tubes & les filtres

Les filtres :

comm	Sélectionne les lignes entre deux fichiers Syntaxe: comm [-123] fic1 fic2 1 = lignes de fic1 (≠ fic2) 2 = lignes de fic2 (≠ fic1) 3 = lignes communes
-------------	--

Systèmes d'exploitation: Unix – Les commandes (13-15)

7 Les tubes & les filtres

Les filtres :

uniq	Détruit les lignes consécutives dupliquées Options: -u (affiche les lignes "uniques"), -d (affiche les lignes "dupliquées") ex: uniq -u fic uniq -d fic
diff	Compare deux fichiers Options: -b (ignorer les lignes vides) ex: diff fic1 fic2

Systèmes d'exploitation: Unix – Les commandes (13-16)

7 Les tubes & les filtres

Les filtres :

tr	Recopie stdin sur stdout en Substituant des caractères.
	Syntaxe: tr [-cds] [s1 [s2]]
	Options:
	<ul style="list-style-type: none">-c (complément de s1)-d efface les car. de s1-s tte séquence dans s1 est substituée par un car. unique dans s2
	Ex: tr A-Z a-z < essai remplace les majuscules par des minuscules tr A-Z a-z < essai tr -sc a-z '\012' remplace les majuscules par des minuscules, puis remplace tout ce qui n'est pas une lettre minuscule par un retour chariot ('\012')

A.U 2012/2013

Ramzi Mahmoudi

157

Systèmes d'exploitation: Unix – Les commandes (13-17)

7 Les tubes & les filtres

Les filtres :

wc	La commande wc (word count) permet de compter les lignes, mots et caractères.
	wc [-l] [-c] [-w] fic1
	<ul style="list-style-type: none">-l : compte le nombre de lignes-c : compte le nombre d'octets-w : compte le nombre de mots
	Exemple : \$ wc liste 12 48 234 liste
	⇒ Le fichier liste contient 12 lignes, 48 mots et 234 caractères.

A.U 2012/2013

Ramzi Mahmoudi

158

Systèmes d'exploitation: Unix – Les commandes (13-18)

7 Les tubes & les filtres

Les filtres :

head	<p>Pour voir le début d'un fichier on utilise la commande head.</p> <p><code>head [-c nbcarr] [-n nblignes] [fic1...]</code></p> <p>L'option -c n'est pas disponible sous tous les Unix et permet de préciser un nombre d'octets d'en-tête à afficher. Par défaut 10 lignes sont affichées.</p> <p>L'option -n permet d'indiquer le nombre de lignes à afficher. Sur certains Unix c'est la syntaxe suivante qu'il faudra utiliser. <code>head [-nblignes] [fic1...]</code></p> <p><code>\$ head -n 3 liste</code> Produit objet prix quantites souris optique 30 15 dur 30giga 100 30</p>
-------------	---

A.U 2012/2013

Ramzi Mahmoudi

159

Systèmes d'exploitation: Unix – Les commandes (13-19)

7 Les tubes & les filtres

Les filtres :

tail	<p>Pour voir les dernières lignes d'un fichier, on utilise la commande tail.</p> <p><code>tail [+/-valeur[b/c]] [-f] [fic1...]</code></p> <p>Comme pour head, par défaut les dix dernières lignes sont affichées.</p> <p>La valeur -nblignes permet de modifier cet état. Préciser c indique un nombre de caractères. Un b indique un nombre de blocs (512 octets par bloc).</p> <p>Un + inverse l'ordre de la commande, et devient un head (<code>tail +10 <=> head -n 10</code>).</p>
-------------	--

A.U 2012/2013

Ramzi Mahmoudi

160

Systèmes d'exploitation: Unix – Les commandes (13-20)

7 Les tubes & les filtres

Les filtres :

tail	Enfin l'option -f laisse le fichier ouvert. Ainsi si le fichier continue d'être rempli (par exemple un fichier trace), son contenu s'affichera en continu sur l'écran jusqu'à interruption volontaire de l'utilisateur (ctrl+C).
	<pre>\$ tail -5 liste ecran 19 500 20 clavier 105 45 30 clavier 115 55 30 carte son 45 30 carte video 145 30 \$ tail -10c liste eo 145 30</pre>

Systèmes d'exploitation: Unix – Les commandes (13-21)

7 Les tubes & les filtres

Les filtres :

awk	Recherche et traitement des chaînes de caractères
cat	Juxtaposition de fichiers et affichage
pr	Mise en page des fichiers
sed	Filtre d'édition

Systèmes d'exploitation: Unix – Les commandes (13-22)

7 Les tubes & les filtres

Redirection , tubes et filtres :

<pre>\$ ls > tmp \$ grep y < tmp</pre>	Affichera tous les noms de fichiers du répertoire courant contenant un « y »
<pre>\$ ls grep y</pre>	Affichera tous les noms de fichiers du répertoire courant contenant un « y » : un tube est créé par le système pour transmettre les données et synchroniser les processus. Le <u>processus producteur</u> ls sera suspendu chaque fois que le tube est plein et le <u>processus consommateur</u> grep sera suspendu chaque fois que le tube est vide.

A.U 2012/2013

Ramzi Mahmoudi

163

Systèmes d'exploitation: Unix – Les commandes (14-1)

8 Gestion des droits d'accès

- Un système d'exploitation assure la sécurité et l'accès aux données, grâce au mécanisme des **droits**.
- Chaque fichier se voit attribué des droits qui lui sont propres : des autorisations d'accès individuelles.
- Lors d'un accès le système vérifie si celui-ci est permis.
- Lors de sa création un utilisateur se voit affecté un **UID** (« User Identifier») unique par l'administrateur
- Les utilisateurs sont définis dans le fichier /etc/passwd.
- Chaque utilisateur est rattaché à au moins un groupe (groupe principal).
- Chaque groupe possédant un identifiant unique, le **GID** (« Group Identifier»).
- Les groupes sont définis dans /etc/group.

A.U 2012/2013

Ramzi Mahmoudi

164

Systèmes d'exploitation: Unix – Les commandes (14-2)

8 Gestion des droits d'accès

La commande **id** permet d'obtenir ces informations. En interne, le système travaille uniquement avec les UID et GID, et pas avec les noms par eux-mêmes.

▪ A chaque fichier (inode) sont associés un UID et un GID définissant son propriétaire et son groupe d'appartenance. On peut affecter des droits pour le propriétaire, pour le groupe d'appartenance et pour le reste du monde. On distingue de ce fait trois cas de figure

```
• Ls -l
d rwxr-xr-x 1 mahmoudr ISBS 466 Oct 1 2008 sys
- rwxrwxrwx 1 mahmoudr ISBS 14536 Oct 1 2008 tp.s
-rwxr-x--- 1 mahmoudr ISBS 1456 Oct 1 2008 prog.sh
```

Sur la première ligne du tableau, le répertoire sys appartient à l'utilisateur mahmoudr et au groupe ISBS, et possède les droits **rwxr-xr-x**.

A.U 2012/2013

Ramzi Mahmoudi

165

Systèmes d'exploitation: Unix – Les commandes (14-3)

8 Gestion des droits d'accès

<u>Droit</u>	<u>Signification</u>
r	Readable (lecture)
w	Writable (écriture)
x	Executable (exécutable comme programme)

Fichier ordinaire:

r	Le contenu du fichier peut être lu, chargé en mémoire, visualisé, recopié.
W	Le contenu du fichier peut être modifié, on peut écrire dedans. La suppression n'est pas forcément liée à ce droit.
x	Le fichier peut être exécuté depuis la ligne de commande, s'il s'agit soit d'un programme binaire (compilé), soit d'un script (shell, perl, ...)

A.U 2012/2013

Ramzi Mahmoudi

166

Systèmes d'exploitation: Unix – Les commandes (14-4)

8 Gestion des droits d'accès

Répertoire :

- r Les éléments du répertoire (catalogue) sont accessibles en lecture. Sans cette autorisation, le **ls** et les critères de filtres sur le répertoire et son contenu ne sont pas possibles. **Ce droit ne suffit pas pour entrer dans le catalogue.**
- W Les éléments du répertoire (catalogue) sont modifiables et il est possible de créer, renommer et supprimer des fichiers dans ce répertoire. On voit donc que c'est ce droit qui contrôle l'autorisation de suppression d'un fichier **même si on est pas propriétaire des fichiers du répertoire.**
- x Le catalogue peut être accédé par cd et listé. Sans cette autorisation il est impossible d'accéder et d'agir sur son contenu qui devient verrouillé. Avec uniquement ce droit les fichiers et répertoires inclus dans celui-ci peuvent être accédés mais il faut alors obligatoirement connaître leur nom.

A.U 2012/2013

Ramzi Mahmoudi

167

Systèmes d'exploitation: Unix – Les commandes (14-5)

8 Gestion des droits d'accès

- Changer les permissions: **chmod**
chmod <classe op perm>|nnn <fic>

- classe:
 - u : user
 - g : group
 - o : others
 - a : all
- op:
 - = : affectation
 - : suppr.
 - + : ajout
- perm:
 - r : lecture
 - w : écriture
 - x : exécution

chaque perm = 1 valeur:

r	4
w	2
x	1
rien	0

définition des permissions (par addition) pour chaque classe

exemples:

```
chmod u=rwx,g=rx,o=r tp1.tex
chmod u+x script.sh
chmod 755 script.sh
```

A.U 2012/2013

Ramzi Mahmoudi

168

Systèmes d'exploitation: Unix – Les commandes (15)

9 Groupement des commande

Le nom de **shell** désigne l'interpréteur de commande du système UNIX. C'est à la fois un langage de programmation et un langage de commande.

En tant que **langage de commande**, le **shell** est l'interface entre l'utilisateur et UNIX.

Les commandes peuvent venir du terminal ou d'un fichier.

Comme on peut créer des fichiers de commandes, il est possible aux utilisateurs de fabriquer leurs propres commandes, paramétrables de la même façon que celles d'UNIX.

Un utilisateur ou un groupe d'utilisateurs peut se composer un environnement personnalisé.

Systèmes d'exploitation: Unix – Les commandes (15-1)

9 Groupement des commande

En tant que **langage de programmation**, le **shell** possède des variables (de type chaîne de caractères) et des structures de contrôle (boucles, branchements).

Les arguments et les sorties des commandes sont des chaînes de caractère, ce qui permet de les transmettre facilement.

Un argument contenant des espaces ou des tabulations, mais à considérer comme un seul argument, doit être mis entre " " ou ' '.

La différence étant que s'il contient des noms de variables, ceux-ci sont remplacés par leurs valeurs dans le cas de " " et reportés tels que dans le cas de ' '.

Systèmes d'exploitation: Unix – Les commandes (15-2)

9 Groupement des commande

Il est possible de créer un fichier dans lequel l'utilisateur écrit une ou un enchaînement de commande UNIX.

Exemple :

```
$ls  
sys  
tp.s  
$ echo "ls">dir  
$cat dir  
ls -l  
$ sh dir  
sys  
tp.s
```

A.U 2012/2013

Ramzi Mahmoudi

171

Systèmes d'exploitation: Unix – Les commandes (15-3)

9 Groupement des commande

Pour rendre directement exécutable le fichier dir en modifiant ses attributs d'autorisation d'accès par la commande :

Exemple :

```
$ls  
sys  
tp.s  
$ echo "ls ">dir  
$cat dir  
ls  
$ chmod +x dir  
$ dir  
sys  
tp.s
```

A.U 2012/2013

Ramzi Mahmoudi

172

Systèmes d'exploitation: **Unix – Les commandes (16)**

8 L'éditeur de texte vi

- L'éditeur Unix par défaut se nomme **vi** (visual editor). S'il n'est pas des plus ergonomiques par rapport à des éditeurs en mode graphique, il a l'avantage d'être disponible et d'utiliser la même syntaxe de base sur tous les Unix. Chaque Unix propose généralement une syntaxe étendue au-delà de la syntaxe de base. Pour en connaître les détails : `man vi`.
- `vi [options] Fichier [Fichier2 ...]` Trois modes de fonctionnement :
- mode commande : les saisies représentent des commandes. On y accède en appuyant sur « Echap ».
- mode saisie : saisie de texte classique
- mode ligne de commande « à la ex » : utilisation de commandes spéciales saisies et se terminant par Entrée. Accès pas la touche « : ».

A.U 2012/2013

Ramzi Mahmoudi

173

Systèmes d'exploitation: **Unix – Les commandes (16-1)**

8 L'éditeur de texte vi

En mode commande :

Commande	Action
a	Ajout de texte derrière le caractère actif
A	Ajout de texte en fin de ligne
i	Insertion de texte devant le caractère actif
I	Insertion de texte en début de ligne
o	Insertion d'une nouvelle ligne sous la ligne active
O	Insertion d'une nouvelle ligne au-dessus de la ligne active

A.U 2012/2013

Ramzi Mahmoudi

174

Systèmes d'exploitation: Unix – Les commandes (16-2)

8 L'éditeur de texte vi

Pour quitter :

zz quitte et sauve le fichier
:q! quitte sans sauve
:q quitte si le fichier n'a pas été modifié
:w sauve le fichier
:wq ou **x** sauve et quitte

Systèmes d'exploitation: Unix – Les commandes (16-3)

8 L'éditeur de texte vi

Pour se déplacer :

Commande	Action
h	Vers la gauche
i	Vers la droite
k	Vers le haut
j	Vers le bas
0 (zéro)	Début de ligne (:0 première ligne)
\$	Fin de ligne (:\$ dernière ligne)
w	Mot suivant
b	Mot précédent
fc	Saut sur le caractère 'c'
Ctrl + F	Remonte d'un écran
Ctrl + B	Descend d'un écran
G	Dernière ligne du fichier
NG	Saute à la ligne 'n' (:n identique)

Systèmes d'exploitation: Unix – Les commandes (16-4)

8 L'éditeur de texte vi

Pour corriger :

Commande	Action
x	Efface le caractère sous le curseur
X	Efface le caractère devant le curseur
rc	Remplace le caractère sous le curseur par le caractère 'c'
dw	Efface le mot depuis le curseur jusqu'à la fin du mot
d\$ (ou D)	Efface tous les caractères jusqu'à la fin de la ligne
dO	Efface tous les caractères jusqu'au début de la ligne.
dfc	Efface tous les caractères de la ligne jusqu'au caractère 'c'
dG	Efface tous les caractères jusqu'à la dernière ligne, ainsi que la ligne active
DIG	Efface tous les caractères jusqu'à la première ligne, ainsi que la ligne active
dd	Efface la ligne active

A.U 2012/2013

Ramzi Mahmoudi

177

Systèmes d'exploitation: Unix – Les commandes (16-5)

8 L'éditeur de texte vi

Pour rechercher dans le texte :

- **vi** peut rechercher autre chose que des mots simples et fonctionne à l'aide de caractères spéciaux et de critères. La commande de recherche est le caractère « / ». La recherche démarre du caractère courant à la fin du fichier. Le caractère « ? » effectue la recherche en sens inverse. On indique ensuite le critère, puis Entrée.
- **/echo** recherche la chaîne 'echo' dans la suite du fichier. Quand la chaîne est trouvée, le curseur s'arrête sur le premier caractère de cette chaîne.
- La commande « **n** » permet de continuer la recherche dans le sens indiqué au début.
- La commande « **N** » effectue la recherche en sens inverse.

A.U 2012/2013

Ramzi Mahmoudi

178

Systèmes d'exploitation: Unix – Les commandes (16-6)

8 L'éditeur de texte vi

Pour rechercher dans le texte :

- ☐ /FfBb]oule : Foule, foule, Boule, boule
- ☐ /[A-Z]e : Tout ce qui commence par une majuscule avec un e en deuxième position.
- ☐ /[A-Za-Z0-9] : tout ce qui commence par une majuscule,minuscule ou un chiffre
- ☐ /^[^a-z] : plage négative : tout ce qui ne commence pas par une minuscule
- ☐ /vé.o : le point remplace un caractère, vélo, véto, véro, ...
- ☐ /Au*o : l'étoile est un caractère de répétition, de 0 à n caractères, Auo, Auto, Automoto, ...
- ☐ /.*: l'étoile devant le point, une chaîne quelconque de taille variable
- ☐ /[A-Z][A-Z]* : répétition du motif entre [] de 0 à n fois, recherche d'un mot comportant au moins une majuscule (en début de mot)
- ☐ /^Auto : le ^ indique que la chaîne recherchée devra être en début de ligne
- ☐ /Auto\$: le \$ indique que la chaîne recherchée devra être en fin de ligne

Systèmes d'exploitation: Unix – Les commandes (16-7)

8 L'éditeur de texte vi

Pour remplacer :

Pour remplacer du texte, il faut se placer au début de la chaîne à modifier, puis taper l'une des commandes suivantes.

Après cette saisie, le caractère \$ apparaît en fin de zone à modifier. Il suffit alors de taper son texte et d'appuyer sur Echap.

Commande	Action
cw	Remplacement du mot courant
c\$	Remplacement jusqu'à la fin de la ligne
cO	Remplacement jusqu'au début de la ligne
cfx	Remplacement jusqu'au prochain caractère 'x' dans la ligne courante
c/Auto (Entrée)	Remplacement jusqu'à la prochaine occurrence de la chaîne 'Auto'

Systèmes d'exploitation: **Unix – Les commandes (16-8)**

8 L'éditeur de texte vi

Pour copier / coller :

- On utilise la commande « y » (Yank) pour copier du texte, elle-même devant être combinée avec d'autres indications. Pour couper (déplacer), c'est la commande « d ». Pour coller le texte à l'endroit choisi, c'est la commande « p » (derrière le caractère) ou « P » (devant le caractère). Si c'est une ligne complète qui a été copiée, elle sera placée en-dessous de la ligne active.
- Pour copier une ligne : yy
- Pour copier cinq lignes : 5yy
- Pour placer les lignes copiées à un endroit donné : p
- L'éditeur vi dispose de 26 tampons pour y stocker les données que l'on peut nommer comme on le souhaite. On utilise pour ça le « " ».
- Pour copier cinq mots dans la mémoire m1 : m1y5w
- Pour coller le contenu de la mémoire m1 à un endroit donné : "m1p

Systèmes d'exploitation: **Unix – Les commandes (16-9)**

8 L'éditeur de texte vi

Pour substitution :

- La substitution permet de remplacer automatiquement plusieurs occurrences par une autre chaîne.
- :[1ere ligne, dernière ligne]/Modèle/Remplacement/[gc] Les numéros de lignes sont optionnels. Dans ce cas la substitution ne se fait que sur la ligne courante. En remplacement des numéros de lignes, « , » détermine la ligne courante, « 1 » la première ligne, « \$ » la dernière ligne.
- Le modèle est l'un des modèles présenté plus tôt. Remplacement est une chaîne quelconque qui remplacera le modèle.
- Par défaut seule la première occurrence est remplacée. La lettre « g » indique qu'il faut remplacer toutes les occurrences. Avec « c », vi demande une confirmation pour chacune des occurrences.
- :1,\$s/[Uu]nix/UNIX/g Cet exemple remplace, dans tout le fichier, Unix ou unix par UNIX.

Systèmes d'exploitation: **Unix – Les commandes (16-10)**

8 L'éditeur de texte vi

Pour configurer l'éditeur :

- La commande **set** permet de configurer l'éditeur.
- **set all** : affiche l'ensemble des options possibles
- **set number** (ou **nu**) / **nonumber** (ou **nonu**) : affiche / supprime les numéros de lignes.
- **set autoindent** / **noautoindent** : l'indentation est conservée lors d'un retour à la ligne.
- **set showmatch** / **noshowmatch** : lors de la saisie d'une accolade ou d'une parenthèse de fermeture, celle d'ouverture est affichée un très court instant, puis l'éditeur revient au caractère courant.
- **set showmode** / **noshowmode** : vi affichera une ligne d'état (INPUT MODE).
- **set tabstop=x** : définit le nombre de caractères pour une tabulation.

Système d'exploitation :

~ Ubuntu ~ **Travaux Pratiques (3)**