NFA083 – Réseaux et Administration Web
TP 1 – Introduction à la ligne de commande sous GNU/Linux

2015

Les TPs seront réalisés sous GNU/Linux. Pour commencer, démarrez une machine sous GNU/Linux et connectez-vous en utilisant vos login et mot de passe habituels ou en utilisant les login/mot de passe génériques : `licencep` / `7002n*`

Partie 1 : Introduction

a) Le système de fichiers

Un système de fichier est organisé sous forme *arborescente* où les fichiers sont contenus dans des *répertoires* (ou *dossiers*).

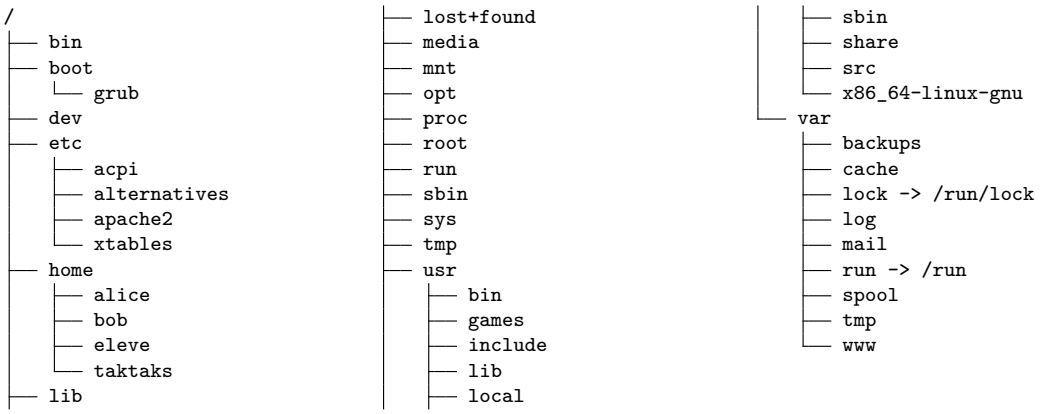

FIGURE 1 – Linux Root File System

Sous UNIX, il existe *une seule* arborescence pour accéder à tous les fichiers indépendamment du nombre d'unité de stockage (partitions, Disques dur, Clés USB, ...).

Sous MS Windows, il existe une arborescence par unité de stockage. Chaque arborescence est désignés par une lettre suivis de ':' (`C:`, `D:`, ...).

Pour la suite du TP, nous allons considérer les systèmes de fichiers de type UNIX.

b) Les Fichiers

Le système de fichiers est donc organisé en une seule structure arborescente contenant des fichiers. Les nœuds sont appelés *répertoires* et les feuilles *fichiers*.

On distingue trois types de fichiers :

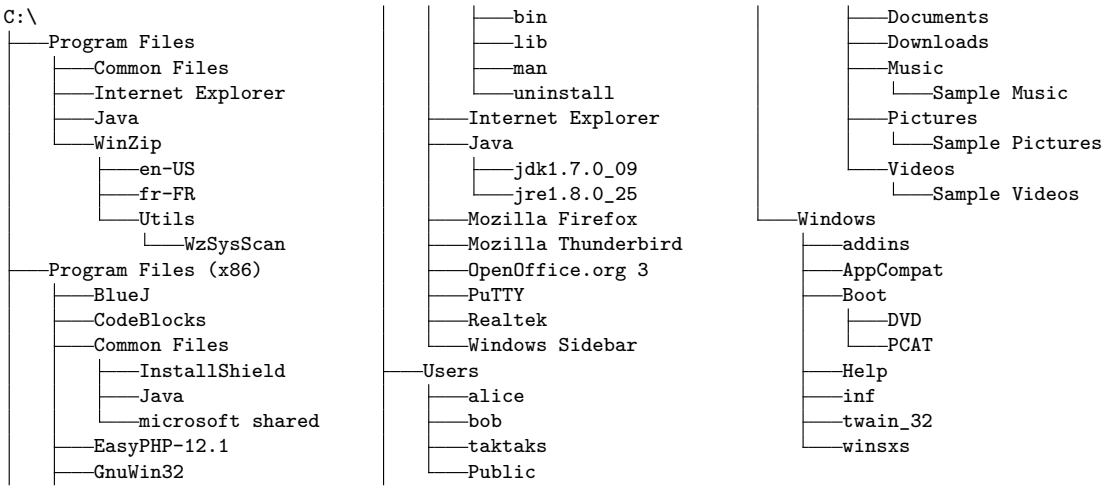

FIGURE 2 – Windows C Root File System

— Les fichiers ordinaires : ils contiennent les données, les programmes et les données des utilisateurs et du système ;
— Les fichiers répertoires ou simplement répertoires : un répertoire contient une liste de fichiers inclue dans le répertoire ;
— Les fichiers spéciaux : ils désignent des moyens de communication soit avec les périphériques, soit avec le système, soit entre processus.

La syntaxe d'un nom de fichier n'est pas très stricte, mais certains caractères sont à éviter :
— caractères ayant une signification particulière : `\ > < | $ ? & [ ] * ! " ‘ ( ) @ ~ ␣`
— caractères peu pratiques à l'usage : caractères spéciaux et accentués.

Remarque. *Le point "." joue un rôle particulier lorsqu'il se trouve en première position dans le nom d'un fichier : un fichier dont le nom commence par un point est un fichier caché (c'est-à-dire qu'il n'apparait pas par défaut lorsque l'on liste le contenu d'un répertoire).*

c) Répertoires Spécifiques

Le système de fichier étant organisé de façon arborescente, il possède un répertoire racine noté `"/"` et appelé `root`. C'est ce répertoire qui contient tous les répertoires et tous les fichiers du système.

Un répertoire peut donc contenir des sous-répertoires et/ou des fichiers ordinaires. Un fichier est repéré par son nom et sa position dans l'arborescence, son chemin d'accès :

```
/repertoire/sous-repertoire/sous-sous-repertoire/nom-du-fichier
```

Nom de répertoires spéciaux :
— Le répertoire `home` d'un utilisateur est le répertoire contenant les fichiers de cet utilisateur ; ce répertoire est désigné par `~`
— Le répertoire courant est désigné par `.`
— Le répertoire parent du répertoire courant est désigné par `..`

d) Principales Commandes de Manipulation de Fichiers et Répertoires

- **Manipulation des répertoires :**

commande	argument(s)	Description	origine du nom
`cd`	`nom_rep`	Changer de répertoire courant	call directory
`ls`	`nom_rep`	Lister le contenu du répertoire	list
`mkdir`	`nom_rep`	Créer un nouveau répertoire	make directory
`rmdir`	`nom_rep`	Supprimer un répertoire	remove directory
`pwd`	*(aucun)*	Afficher le nom du répertoire courant	print current/working directory

Exemples.

— *Obtenir le nom du répertoire courant :*

```
$ pwd
/home/eleve
```

— *Lister le contenu du répertoire courant :*

```
$ ls
Archives              Documents             public_html
bin                   Downloads             Téléchargements
bonjour.txt           mbox                  test.txt
Bureau                Music                 tmp
Desktop               Pictures              Videos
```

— *Lister le contenu d'un répertoire :*

```
$ ls /usr/bin
7z                    ab                    adb2mhc
7za                   aclocal               addftinfo
a2p                   aconnect              addpart
a2ping                acpi                  add-patch
a2x                   acpi_listen           adhocfilelist
a5booklet             acyclic               aj
a5toa4                adb                   aj5
```

— *Créer un nouveau répertoire :*

```
$ mkdir NSY103
```

— *Supprimer un répertoire (le répertoire doit être vide) :*

```
$ rmdir tmp
```

- **Manipulation des fichiers :**

commande	argument(s)	Description	origine du nom
`cat`	`nom_fichier`	Afficher le contenu d'un fichier	Con**cat**enate files
`less`	`nom_fichier`	Afficher le contenu d'un fichier page par page	pager
`cp`	`nom_src nom_dest`	Copie d'un fichier	Copy
`cp`	`-r nom_src rep_dest`	Copie d'un répertoire	Copy (recursive)
`mv`	`nom_src rep_dest`	renomer/déplacer un fichier	***mov****e*
`rm`	`nom_fichier`	Suppression d'un fichier	***rem****ove*

Exemples.

— *Afficher le contenu du fichier* `bonjour.txt`

```
$ cat bonjour.txt
Bonjour le monde !
```

— *Copie d'un fichier :*

```
$ ls
bonjour.txt
$ cp bonjour.txt bonjour2.txt
$ ls
bonjour2.txt bonjour.txt
```

— *Copie d'un répertoire :*

```
$ ls rep1
bonjour2.txt bonjour.txt
$ cp -r rep1 rep2
$ ls
rep1/ rep2/
$ ls rep2
bonjour.txt bonjour2.txt
```

— *Renommer un fichier*

```
$ mv rep2 rep_bonjour
$ ls
rep1/ rep_bonjour/
```

— *Suppression d'un fichier* `bonjour2.txt` *dans le repertoire* `rep1` *:*

```
$ rm rep1/bonjour2.txt
$ ls rep1
bonjour.txt
```

e) Obtenir de l'aide : les pages de manuels

Remarque. *Si votre terminal n'est pas en français, vous pouvez le mettre en français en entrant la commande suivante :*

```
$ export LANG="fr_FR.UTF-8"
```

Les différentes commandes présentées jusqu'à présent possèdent un certain nombre d'options. Afin d'obtenir de l'aide sur les différentes options d'une commande, il existe plusieurs possibilités :

— l'aide en ligne de commande, souvent obtenue grâce à l'option `--help` ou `-h`.
Exemple : pour afficher l'aide en ligne de la commande `rm`

```
$ rm --help
```

— en consultant les pages de manuel grâce à l'utilitaire `man`.
Exemple : pour afficher la page de manuel de la commande `rm`

```
$ man rm
```

Les pages de manuel sont organisées en sections, la section 1 concernant les commandes utilisateur. Pour obtenir de l'aide sur l'utilitaire `man`, il suffit de taper la commande "`man man`".

— en consultant les pages d'information grâce à l'utilitaire `info`.
Exemple : pour afficher la page de manuel de la commande `rm`

```
$ info rm
```

Partie 2 : Manipulation de Fichiers et de Répertoires

Ouvrez un terminal et, en utilisant les informations ci-dessus, répondez aux questions suivantes :

a) Exercice 1

Essayez les exemples donnés à la section d) de l'introduction et comparez les résultats obtenus à ceux de l'énoncé.

b) Exercice 2

1. Déplacez-vous dans le répertoire racine ;
2. Listez les fichiers et répertoires présents à la racine ;
3. Déplacez-vous dans le répertoire `/etc/` ;
4. Ouvrez le fichier `protocols` qui se trouve dans le répertoire `/etc/` avec `more` puis `less` ;
5. Quelle est la différence entre la commande `more` et la commande `less` ?
6. Comment aurait-on pu ouvrir le fichier `/etc/protocols` sans se déplacer dans le répertoire `/etc/` ?
7. Retournez dans votre répertoire personnel ;
8. Affichez le nom du répertoire dans lequel vous vous trouvez ;
9. Tapez la commande "`cd .`" puis affichez le nom du répertoire courant. Que constatez-vous ?
10. Tapez la commande "`cd ..`" puis affichez le nom du répertoire courant. Retapez la commande "`cd .`" puis afficher le nom du répertoire courant. Que constatez-vous ?
11. Listez le contenu du répertoire courant. Quelle différence y a-t-il entre le résultat de la commande "`ls`" et le résultat de la commande "`ls .`" ? Que pouvez-vous en conclure sur les noms des répertoires "." et ".."
12. Retournez dans votre répertoire personnel et créez un fichier `essai` à l'aide d'un éditeur de texte (par exemple `nano` ;
13. Créez l'arborescence suivante :

```
rep1
├── fich11
├── fich12
├── rep2
│   ├── fich21
│   └── fich22
└── rep3
    ├── fich31
    └── fich32
```

14. Déplacez toute l'arborescence `rep3` dans `rep2`

c) Exercice 3

1. Trouvez les options de la commande `ls` pour afficher les informations détaillées de toute une arborescence ;
2. Trouvez l'option de la commande `rm` pour supprimer le répertoire `rep1` ainsi que tout son contenu.

Partie 3 : Droits d'accès aux fichiers

Comme tout système multi-utilisateurs, il existe sous GNU/Linux un système de gestion de droits d'accès aux fichiers et aux répertoires. Chaque fichier appartient à un utilisateur du système et à un groupe. Les droits d'accès sont définis pour l'utilisateur propriétaire, le groupe propriétaire et les autres utilisateurs (qui ne sont pas dans le groupe).

Pour chaque fichier, il existe donc trois types d'utilisateurs :
— le propriétaire du fichier ;
— les membres du groupe propriétaire du fichier ;
— les autres utilisateurs du système.

Pour chaque fichier et pour chaque type d'utilisateur, il existe trois modes principaux :
— autorisation de lecture "`r`" ;
— autorisation d'écriture "`w`" ;
— autorisation d'exécution "`x`".

Pour afficher les droits d'un fichier, on utilise la commande `ls` avec l'option "`-l`" pour afficher le détail des informations sur un fichier :

```
$ ls -l essai
-rw-r--r-- 1 taktaks ensinf 34 5 mars 16:40 essai
```

Le résultat de la commande `ls` donne les informations suivantes :
— "`- rw- r-- r--`" indique le type de fichier ainsi que les droits associés :
  — le premier caractère indique le type de fichier : "`-`" veut dire qu'il s'agit d'un fichier régulier. Un "`d`" aurait signifié qu'il s'agissait d'un répertoire ;
  — les trois groupes de caractères suivants représentent les droits d'accès au fichier pour le propriétaire (`rw-`), le groupe propriétaire (`r--`) et les autres utilisateurs (`r--`). Dans chacun des groupes, le premier caractère indique le droit d'accès en lecture "`r`", le deuxième caractère le droit d'accès en écriture "`w`", et le troisième caractère le droit d'exécution "`x`". La présence d'un tiret indique que ce mode d'accès n'est pas autorisé ;
  "`rw- r-- r--`" indique donc que le fichier `essai` est accessible en lecture et écriture par le propriétaire du fichier, en lecture seulement par les membres du groupe propriétaire et les autres utilisateurs ;
— "1" est le nombre de références (*hard link*) vers ce fichier ;
— "`taktaks`" est le nom de l'utilisateur propriétaire du fichier ;
— "`ensinf`" est le nom du groupe propriétaire du fichier ;
— "`34`" est la taille du fichier ;
— "`5 mars 16:40`" indique la date de dernière modification du fichier ;
— "`essai`" est le nom du fichier.

Les droits d'accès à un fichier peuvent être modifiés par le propriétaire du fichier grâce à la commande "`chmod`" :

```
$ chmod g+w essai
```

donne le droit en écriture aux membres du groupe.

La syntaxe de `chmod` est suivante :

```
chmod  [qui]op[permission] fichier
```

où :

— `qui` : est une combinaison des lettres `u` (*user*=utilisateur), `g` (groupe), `o` (*other*=autre) ou `a` (*all*=tous) équivalent à `ugo` ;

— `op` : `+` permet d'ajouter un droit d'accès, `-` de retirer un droit d'accès, et `=` de mettre un droit d'accès de manière absolue ;

— `permission` : est une combinaison des lettres `r` (*read*=lecture), `w` (*write*=écriture), `x` (exécution)

1. Changez les droits du répertoire `rep1` pour donner l'accès en écriture à tous les membres du groupe ;
2. Déplacez-vous dans le répertoire `rep2` et retirez les droits en exécution pour tout le monde au répertoire `rep3`. Déplacez-vous dans le répertoire `rep3`. Que se passe-t-il ? Corrigez le problème ;
3. Modifiez les droits du répertoire `rep3` pour que seul le propriétaire ait les droits d'accès en lecture, écriture et exécution, pour que le groupe ait les droits d'accès en lecture et exécution, et pour que les autres n'aient aucun droit sur ce répertoire.