

# Exercices de TP sur le shell et les fichiers

Fabien Rico

## 1 Le manuel

`man` (pour *manual*) est une commande disponible sur les systèmes d'exploitation de type Unix. Elle permet d'obtenir de l'aide sur la plupart des programmes disponibles sur le système, mais aussi sur les fonctions du langage C, et de certaines bibliothèques. Pour l'invoquer, la commande est :

```
man nom_de_la_commande
```

Par exemple,

```
man du
```

permet d'obtenir le manuel de la commande `du`.

Une fois que la page manuel est affichée, les touches de déplacement permettent de naviguer dedans. Pour quitter (revenir à la ligne de commande), il suffit de taper `Q`. Il y a beaucoup de subtilités que l'on passe ici sous silence, mais vous pouvez maintenant faire `man man` pour en apprendre plus sur `man`!

### 1.1 Man

Voici ci-dessous un extrait de la page de manuel de la commande `ls`.

#### NAME

`ls` - list directory contents

#### SYNOPSIS

```
ls [OPTION]... [FILE]...
```

#### DESCRIPTION

List information about the FILES (the current directory by default). Sort entries alphabetically if none of `-cftuvSUX` nor `--sort` is specified.

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

`-c` with `-lt`: sort by, and show, ctime (time of last modification of file status information); with `-l`: show ctime and sort by name; otherwise: sort by ctime, newest first

`-h`, `--human-readable`  
with `-l` and/or `-s`, print human readable sizes (e.g., 1K 234M 2G)

`--si` likewise, but use powers of 1000 not 1024

`-l` use a long listing format

`-r`, `--reverse`  
reverse order while sorting

`-R`, `--recursive`  
list subdirectories recursively

`-s`, `--size`  
print the allocated size of each file, in blocks

`-S` sort by file size, largest first

`-t` sort by modification time, newest first

`-u` with `-lt`: sort by, and show, access time; with `-l`: show access time and sort by name; otherwise: sort by access time, newest first

- 1) Quelles options de `ls` utiliser pour lister le contenu du répertoire courant dans l'ordre inverse de l'ordre alphabétique?

- 2) Quelles options utiliser pour lister le contenu du répertoire par ordre de date de dernière modification, avec les fichiers modifiés le plus récemment à la fin ?
- 3) Quelles options pour ranger les fichiers par ordre de taille décroissante ?

## 2 Arborecence de fichiers : chemins

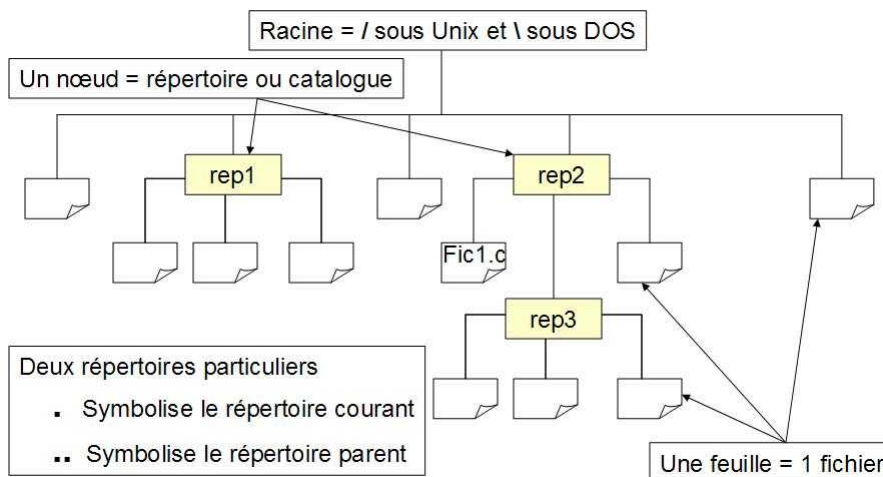
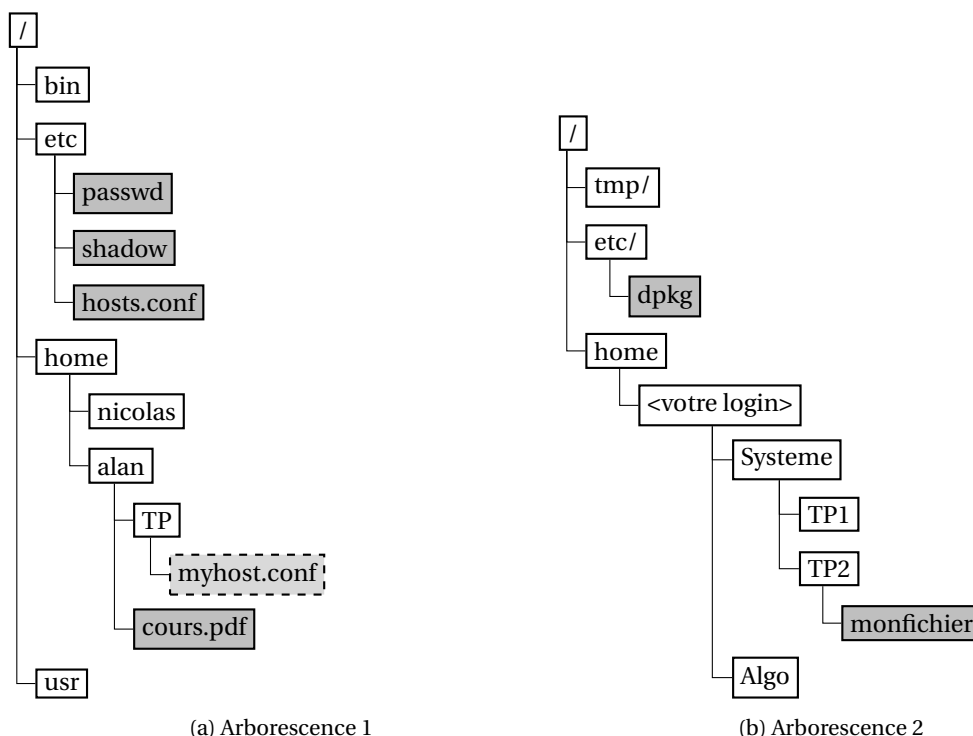


FIGURE 1 – Arborecence et chemin sous Linux

La figure 1 résume les concepts à retenir.

- *Chemin absolu* d'un fichier : c'est le nom "complet" du fichier, donné par son chemin d'accès depuis la racine. Exemple, sous Unix : `/rep/fic1.c`.
- *Répertoire de travail* (répertoire courant) : répertoire sur lequel un utilisateur est positionné à un moment donné. Ce répertoire est nommé "."
- *Chemin relatif* d'un fichier : référence d'un fichier à partir du répertoire de travail.
- *Répertoire personnel* (sous Unix) : répertoire sur lequel est positionné un utilisateur suite à son authentification. Ce répertoire est nommé "~" (tilde). On parle aussi de "home utilisateur" ou de "racine personnelle". Sous Unix c'est souvent un sous-répertoire de `/home`.



### 3 Chemins (TD)

La figure 2a représente l'arborescence présente dans un ordinateur sous Unix : (Les fichiers sont en grisé, les autres noeuds sont des répertoires).

Supposons que vous êtes l'utilisateur `al.an`. Donner des commandes successives pour :

- Vous placer dans votre répertoire d'accueil
- Y créer un répertoire nommé TP.
- Faire une copie du fichier `hosts.conf` dans le répertoire TP. Puis la renommer `myhosts.conf`.

#### 3.1 Navigation relative et absolue - TD ou TP

Donner les commandes ou réaliser les manipulations suivantes :

- 1) Réaliser la hiérarchie de répertoires et de fichiers décrite dans la figure 2b  
Si ce n'est pas le cas, effectuer les modifications nécessaires (à l'aide des commandes `touch`, `cp` et `mkdir`?)
- 2) Se placer dans le répertoire TP2. Pour copier un fichier vers le répertoire TP1, vous pouvez au choix (réaliser chacune de ces manipulations) :
  - utiliser le chemin relatif (du répertoire TP1 par rapport à où je suis, ie TP2) : `cp nomfichier ../TP1/`
  - utiliser le chemin absolu du répertoire TP1 : `cp nomfichier /home/moi/Systeme/TP1/`
  - utiliser le raccourci de votre *home* : `cp nomfichier ~/Systeme/TP1/`
- 3) Se placer à la racine de votre répertoire personnel. Puis, *sans se déplacer de nouveau*, et en s'inspirant des manipulations précédentes :
  - copier un des fichiers du répertoire `/etc/dpkg` dans le répertoire TP2.
  - lister le contenu du répertoire TP1.
  - déplacer les fichiers qui n'ont rien à faire de TP1 vers `/tmp`.

### 4 Autopsie d'un mail

Le fichier `mail.zip` est une archive contenant un fichier de mail.

Ouvrez le avec un éditeur de texte.

- 1) Quel partie forme l'entête? Les données?
- 2) Retrouvez le sujet du mail, le destinataire, l'expéditeur? Quel chemin a suivi le mail?
- 3) Dans la partie donnée, que signifie le premier `Content type`?
- 4) Combien y a-t-il de parties? Quel est le moyen de les retrouver?
- 5) A quoi correspondent chaque partie?
- 6) Extrayez l'image du mail.
- 7) Les données de l'image sont encodées en base64, pourquoi?

### 5 Contenu d'un epub

Je souhaite faire une étude statistique sur les mots utilisés dans une oeuvre de Molière. Mais je ne dispose que d'un format epub disponible ici. Vous devez donc extraire le texte de la pièce.

- 1) Affichez le début du fichier avec la commande `head` que semble-t-il contenir?
- 2) Que dit la commande `find` de ce fichier?
- 3) Extrayez les données du fichier
- 4) Retirer les balises, les noms des personnages (voir la commande `sed` et les expressions régulières)
- 5) Fournissez un fichier texte contenant la pièce avec le moins de sautes possible.