

Unix (1)

L'ensemble du TD nécessite d'utiliser une fenêtre de type "TERMINAL", aussi appelé "CONSOLE".

1 Description et utilisation de commandes Unix

1.1 EXERCICE Ouvrez un "TERMINAL" et testez la commande `exit`, puis réouvrez votre "TERMINAL".

1.2 EXERCICE Essayez les commandes suivantes `who`, `date`, `cal`, `echo`. Commentez leur fonctionnement.

1.3 EXERCICE Essayez `man cal`. Comment obtenir de l'aide en ligne sur le manuel ?

1.4 EXERCICE Essayez `du -sh ~`. Trouvez dans le manuel la signification des paramètres et options de la commande `du`.

1.5 EXERCICE Il est préférable de consulter le manuel en anglais. En effet, il est plus complet et plus précis (pas d'erreurs de traduction possible). Trouvez dans le manuel comment l'afficher en anglais.

2 L'interprète de commandes

On peut éditer la ligne de commande :

→ déplacement dans la liste des commandes (**historique**) :

C-p ou Up	(<i>previous</i>) : commande précédente
C-n ou Down	(<i>next</i>) : commande suivante
C-r	(<i>reverse search</i>) : recherche dans l'historique

→ déplacement du curseur dans la ligne

C-a ou Home	(<i>abcdef...</i>) : début de ligne
C-e ou End	(<i>end</i>) : fin de ligne
C-b ou Left	(<i>backward</i>) : caractère précédent
C-f ou Right	(<i>forward</i>) : caractère suivant

→ suppressions :

C-d ou Suppr	(<i>delete</i>) : caractère sous le curseur
DEL	caractère précédent le curseur
C-k	(<i>kill</i>) : jusqu'à la fin de ligne

→ validation/annulation

RETURN ou C-j	validation de la ligne
C-c	(<i>cancel</i>) : annulation de la ligne en cours

2.1 EXERCICE Essayez les commandes d'édition de ligne :

- faites afficher le calendrier de l'année courante ;
- utilisez l'historique et l'édition de ligne pour afficher successivement les calendriers des mois de janvier, de décembre, et de février de l'année courante. Essayez la recherche avec `C-r`.

3 Visite de l'arborescence utilisateur

3.1 EXERCICE Essayez

```
$ ls
$ pwd
$ cd_..
    (pensez à séparer la commande de ses paramètres par un espace représenté ici par le caractère _)
$ pwd
$ ls
```

3.2 EXERCICE Essayez les séquences suivantes

```
$ ls e TAB
$ cd e TAB
```

3.3 EXERCICE Sélectionnez un texte (bouton gauche de la souris) dans une fenêtre `TERMINAL` pour ensuite le coller (bouton du milieu de la souris) dans une autre fenêtre `TERMINAL`. Vérifiez la possibilité de faire la même opération de copier-coller entre du texte contenu dans votre navigateur web et une fenêtre `TERMINAL`.

3.4 EXERCICE Exécutez la suite de commandes :

```
$ cd
$ cd_Bureau (n'oubliez pas le mécanisme de complétion)
$ pwd
```

Maintenant pour aller dans votre répertoire d'accueil plusieurs possibilités :

```
$ cd_.. ou
$ cd_~ ou encore
$ cd
```

3.5 EXERCICE Dessinez votre arborescence.

3.6 EXERCICE Essayez différentes options de la commande `ls` dans votre répertoire d'accueil. Commentez. Y a-t-il une notion de fichier caché ?

3.7 EXERCICE Exécutez, en utilisant les mécanismes d'édition de `bash`, les commandes

```
$ man ls
$ ls Bureau
$ ls -F Bureau
$ ls -a Bureau
$ ls -aF Bureau
$ ls -l Bureau
```

4 Création et suppression de répertoires

4.1 EXERCICE Créez un répertoire `tmp` et un répertoire `Projets` dans votre répertoire d'accueil. Plusieurs possibilités, par exemple :

```
$ mkdir ~/tmp
$ cd
$ mkdir Projets
```

Si le chemin est celui d'un fichier (ou répertoire) déjà présent, cela provoque une erreur : le nom de base doit être nouveau dans le répertoire de référence.

```
$ mkdir eclair
$ ls -F
$ cd eclair
$ rmdir ../eclair
$ cd ..
$ rmdir eclair
$ ls -F
```

4.2 EXERCICE Essayez `rmdir ~` et commentez le résultat obtenu.

5 Manipulation de fichiers

5.1 Copies

5.1 EXERCICE Que font les commandes :

```
$ mkdir test
$ cd test
$ touch doc1.html
$ touch doc2.html
$ ls
$ cp doc1.html doc2.html ~/tmp
$ ls ~/tmp
```

5.2 Suppressions

5.2 EXERCICE Essayez la suite de commandes :

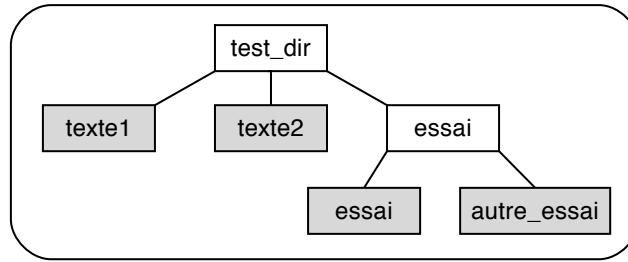
```
$ rm doc2.html
$ ls
$ rm ~/tmp/doc1.html
$ ls
$ cd ~/tmp
$ ls
```

Pour la suppression interactive essayer : `rm -i <fichier> ...` (répondre `y` pour confirmer la suppression et `n` pour infirmer).

5.3 Déplacements et renommages

```
$ cd ~/test
$ mv doc1.html doc.html
$ ls -F
$ mv doc.html ..
$ ls -F
$ ls -F ..
$ mv ../doc.html .
$ ls -F ..
$ ls -F
```

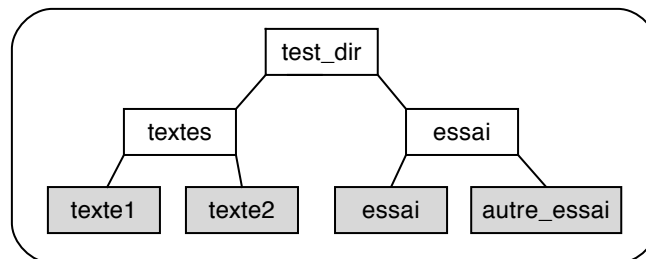
5.3 EXERCICE Créez à partir de votre répertoire d'accueil l'arborescence



Les cases blanches sont des répertoires, tandis que les cases grisées sont des fichiers texte à créer. La marche à suivre pour cet exercice est la suivante :

- créez les répertoires ;
- éditez chaque fichier avec l'éditeur de texte emacs, ajoutez du texte et sauvegardez.

5.4 EXERCICE Placez-vous dans le répertoire `test_dir`, et ramenez-vous en deux commandes à :



5.5 EXERCICE Il est possible de spécifier des listes de chemins au moyen de motifs. Le motif `*` représente tous les fichiers du répertoire courant dont le nom ne commence pas par un point. Le motif `t*` représente tous les fichiers du répertoire courant dont le nom commence par un `t`.

Placez vous dans le répertoire `textes`. Essayez les commandes :

```

$ mv t* ..
$ ls
$ ls ..

```

Déplacez les fichiers `texte1` et `texte2` dans le répertoire `essai`. Déplacez les fichiers `essai` et `autre_essai` dans le répertoire `textes`.

5.6 EXERCICE Depuis `test_dir` essayez `cp essai textes` ; commentez le résultat. Essayez avec l'option `-R`.

5.4 Liens

5.7 EXERCICE Dans un répertoire vide, essayez les commandes suivantes :

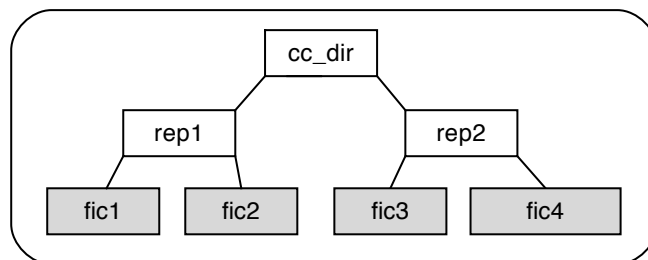
```

$ touch fichier
$ ln fichier hardlink
$ ln -s fichier symlink
$ ls -F
$ rm fichier
$ ls -l
$ more hardlink
$ more symlink

```

5.8 EXERCICE (POUR CEUX QUI SONT EN AVANCE)

- Créez l'arborescence suivante où `fic4` est un lien symbolique vers le fichier `fic1`



- Une fois l'arborescence créée, placez-vous dans `cc_dir` et exécutez la commande `tree`
- Supprimez `cc_dir` et son contenu

6 Examen de fichiers réguliers

6.1 EXERCICE Essayez `cat <fichier>`.

6.2 EXERCICE Essayez `more <fichier>`. De même avec la commande `less`. Vaut-il mieux utiliser la commande `more` ou `less` et pourquoi ?

6.3 EXERCICE Essayez

```

$ cd Bureau
$ file terminal.desktop
$ file *
$ file .*
  
```

6.4 EXERCICE Essayez

```

$ wc terminal.desktop
$ wc *
$ wc .*
  
```

\$ `wc` : par défaut, lecture au clavier ; entrez des lignes et terminer en tapant `C-d` en début de ligne

7 Comparaison de fichiers - Pour ceux qui sont en avance

La commande `diff` sert à afficher les différences entre deux fichiers.

7.1 EXERCICE Copiez un fichier, modifiez-le, et comparez les deux fichiers au moyen de `diff`.

8 Que faut-il faire si un programme « plante » ?

Pour avoir la liste de tous vos processus : `ps ux`.

Notez les identifiants (PID) des processus. Pour tuer un processus essayez :

- la manière diplomatique : `kill -s SIGTERM <pid>`
- la manière forte : `kill -s SIGKILL <pid>`

8.1 EXERCICE Lancez une application graphique, par exemple `xclock &` trouvez son (PID) et tuez-la avec `kill`.

8.2 EXERCICE Trouvez un exemple de processus dont le parent n'est pas le processus `init`. Pour afficher le processus parent, faites afficher le `ppid`. Tuez le père du processus et observez le comportement du fils. Si le processus fils est toujours vivant, recherchez son nouveau père.

8.3 EXERCICE À l'aide de la commande `ps` affichez l'arbre des processus. Y a-t-il des processus qui ont le processus `init` comme parent ?

8.4 EXERCICE Relancez une application graphique (comme pour l'exercice 8.1). Connectez vous en mode `console`¹ et tuez l'application graphique à l'aide de la commande `kill`. Revenez en mode `graphique`².

À la fin du TD, pensez à vous déconnecter

1. Pour basculer en mode `console` utiliser la séquence de touches `Ctrl Alt F1`
2. Pour basculer en mode `graphique` utiliser la séquence de touches `Ctrl Alt F7`