

TP1_presentation_Hategekimana_Fabrice

2.1 Commande ‘man’

a) Utilisez la commande shell ‘man’ pour afficher le manuel de la commande.

```
$man man
```

Cette commande me donne la présentation de man. La description dit: man is the system’s manual pager. Each page argument given to man is normally the name of a program, utility or function....

```
$ man echo
```

man me donne la présentation de la commande echo. La description dit: Echo the STRING(s) to standard output.

```
$ man head
```

man me donne la présentation de la commande head. La description dit: Print the first 10 lines of each FILE to standard output....

b) Essayez de rechercher dans un moteur de recherche une commande qui fait ce que vous avez en tête: par exemple, essayez la recherche suivante: “bash how to merge two files”

J’ai essayé la recherche et voila ce que je trouve dans le fameux site stack overflow: <https://stackoverflow.com/questions/3806874/how-to-merge-two-files-line-by-line-in-bash>

Il suffit de faire:

```
$ paste file1.txt file2.txt > fileresults.txt
```

2.2 Commande ‘echo’

La commande echo permet d’afficher les chaînes ou variables qu’on lui présente. Les paramètre/option de cette commande sont:

- e enable interpretation of backslash escapes
- E disable interpretation of backslash escapes (default)

-help display this help and exit

-version output version information and exit)

2.3 Créer un fichier

Pour créer un fichier vide à partir de la ligne de commande, il suffit d'utiliser:

```
$ touch [nom du fichier]
```

2.4 Nautilus

Cette commande ouvre le gestionnaire de fichier par défaut de GNOME. Si on donne un nom de répertoire en paramètre, il ouvre le gestionnaire de fichier sur ce répertoire.

2.5 Fichiers

La commande:

```
wc -w < foo.txt
```

Va mettre le contenu de foo.txt dans l'entrée standard de wc.

La commande:

```
$ cat foo.txt | wc -w
```

Va passer le contenu du fichier foo.txt par pipe (par un FIFO) que wc viendra chercher.

La commande:

```
wc -w foo.txt
```

Va mettre le fichier foo.txt directement dans les paramètres de la commande wc

Exercices

La commande:

```
$ head foo.txt -n 6
```

Va afficher les 6 premières lignes du fichier foo.txt

La commande:

```
$ tail foo.txt -n 6
```

Va afficher les 6 dernières lignes du fichier foo.txt

La commande:

```
$ sort foo.txt>out1.txt 2>out2.txt
```

Va mettre le contenu de la sortie standard dans le fichier out1.txt alors que le fichier out2.txt va prendre l'erreur de la commande. Donc si foot.txt, out2.txt va avoir dans son contenu une erreur disant que le fichier foo.txt n'existe pas.

La commande:

```
$ evince foo.pdf
```

Va lancer le programme directement dans le processus du terminal (le terminal se bloc)

La commande:

```
$ evince foo.pdf &
```

Va lancer le programme dans un nouveau processus (le terminal ne va pas se bloquer)

la commande:

```
$ echo $SHELL
```

Affiche le shell par défaut du système

la commande:

```
$ echo $0
```

Affiche le shell actuellement utilisé.