Scéance d'utilisation de UNIX


```
whoami
pwd
ls
cd pc
pwd
cd ..
pwd
ls
cd ift159
mkdir temp
ls
cd temp
pwd
touch unFichier
ls -l
cp unFichier unAutreFichier
ls
mv unAutreFichier aeffacer
ls
rm aeffacer
ls
mv unFichier ../unAutreNom
cd ..
ls
rm unAutreNom
ls
cd temp
pwd
cd
```

Scéance d'utilisation de UNIX

pwd
cp -R ift159 repertoire
rm repertoire
mkdir rep_a_effacer
rmdir rep_a_effacer
rmdir repertoire
rm -r repertoire
ls ift159
rmdir ift159/temp
cd ift159
mkdir temp
cd temp
pwd
nedit programme.cpp &
< ENTRER LE PROGRAMME EN C++>
g++ programme.cpp
./a.out

- 8.1 Changement de mot de passe : *passwd*
- 8.2 Information sur l'espace utilisé par le compte : *quota -v*
- 8.3 renseignement divers
  - **`whoami`** permet de connaître l'utilisateur connecté
  - **`pwd`** indique le répertoire courant
  - **`ls`** indique le contenu du répertoire courant, on peut indiquer un répertoire particulier en argument (par exemple avec **`ls ift159/`**)
  - **`ls -a`** indique tous le contenu du répertoire courant y compris ce qui est caché
  - **`ls -l`** indique le contenu du répertoire courant sous la forme d'une liste
  - **`ls -al`** cumule les deux précédentes commandes
- 8.4 déplacement
  - **`cd`** permet de changer de répertoire en se déplacant dans le répertoire dont le nom est passé en argument.
- 8.5 création
  - **`mkdir arg1`** crée un répertoire dont le nom est **`arg1`**.
  - **`touch arg1`** crée un fichier dont le nom est **`arg1`**.
- 8.6 copier
  - **`cp arg1 arg2`** copie le fichier passé en premier argument (**`arg1`**). Si le deuxième argument (**`arg2`**) est un répertoire, le fichier sera copié en un fichier du même nom dans le répertoire cible. Si le deuxième argument est un fichier, écrasera ce fichier par la copie. Si le deuxième argument n'est pas un répertoire ou un fichier existant, la copie portera le nom de ce deuxième argument.
  - **`cp -R arg1 arg2`** copie le répertoire passé en premier argument (**`arg1`**). Le deuxième argument (**`arg2`**) doit être un répertoire. Le répertoire **`arg1`** sera copié en un répertoire du même nom dans le répertoire **`arg2`**.
- 8.7 deplacer (renommer) :
  - **`mv arg1 arg2`** déplace le fichier passé en premier argument (**`arg1`**). Si le deuxième argument est un fichier ou n'existe pas, renomme le fichier **`arg1`** avec le nom **`arg2`**. Si le deuxième argument est un répertoire, déplace effectivement le fichier **`arg1`** dans le répertoire **`arg2`**.
- 8.8 suppression : **`rmdir`**, **`rm`**
  - **`rmdir arg1`** supprime le répertoire **`arg1`**. Ce répertoire doit être vide.
  - **`rm arg1`** supprime le fichier **`arg1`**, en demandant confirmation à chaque étape (sur rigel).
  - **`rm -r arg1`** supprime le répertoire **`arg1`**, en demandant confirmation à chaque étape (sur rigel).

Scéance d'utilisation de UNIX

- 8.9 editeur : **`nedit &`**
  - Ouvre l'éditeur de fichier **`nedit`**. On peut préciser un fichier en particulier en argument. Attention, évitez d'oublier le symbole à la fin de la commande (le **`&`**). Celui-ci permet de garder le terminal libre.
- 8.10 compilation : g++
  - **`g++ prog.cpp`** Lance la compilation du programme **`prog.cpp`**. Le fichier exécutable résultant portera le nom de **`a.out`**.
  - **`g++ prog.cpp -o mon_prog`** Lance la compilation du programme **`prog.cpp`**. Le fichier exécutable résultant portera le nom de **`mon_prog`**.
- 8.11 exécution d'un programme
  - il suffit simplement de taper son nom en étant dans le répertoire dans lequel il se trouve. Par exemple, si le programme se nomme **`mon_prog`** il faut taper la commande **`mon_prog`**. Parfois le système peut nécessiter de taper **`./`** avant le nom du programme, ce qui donne **`./mon_prog`**. Finalement on peut toujours donner le chemin complet (absolu ou relatif) vers le programme, par exemple, **`ift159/tp1/mon_prog`** ou **`/home/gradues/fraikin/ift159/tp1/mon_prog`**.
- 8.12 afficher le contenu d'un fichier : *cat, more, less*
  - Utilisation de *less* :
  - L'utilisation d'<espace> ou de <CTRL-f> permet d'avancer d'une page.
  - L'utilisation de <CTRL-b> permet de reculer d'une page.
  - La flèche du haut ↑ et du bas ↓ permettent de se déplacer d'une ligne.
  - La barre oblique ou « *slash* » </> permet de faire des recherches ainsi que le <?>.
  - Le <p> revient au début.
  - La touche <q> permet de quitter.