Travaux Dirigés d'algorithmique n°1

Cours d'Informatique de Première Année

—DEUG Sciences S2-SM/STPI—

Initiation à Unix

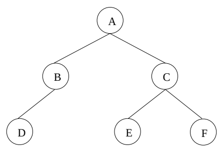
Lors de cette séance, nous introduisons les commandes de base du système UNIX et nous présentons le langage C.

▸ Exercice 1. *(Commandes de base)*

- *Commentez la séquence d'instructions suivantes :*

```
ls
mkdir TOTO
cd toto
cd TOTO
ls -a
pwd
cd ..
touch fic
ls -la
rmdir TOTO
mv fic tata.c
touch tutu.c
ls
rm *.c
```

- *À partir de votre répertoire "maison", créer l'arborescence des répertoires ci-dessous, et revenir dans le répertoire initial.*



- *Créer un fichier* `myfic` *dans le répertoire* `E`*. Vérifier qu'il existe.*

- *Taper la commande* `cd; rm -rf A`*.*

▸ Exercice 2. *(Droits d'accès)*

- *La commande* `chmod` *permet de fixer les droits d'accès aux fichiers et aux répertoires. Ces droits apparaissent en début de ligne quand on effectue la commande* `ls -l`*. Taper la commande* `man chmod`*.*

- *Taper la séquence d'instructions suivantes :*

```
touch fichier
chmod 644 fichier
ls -l fichier
chmod 777 fichier
ls -l fichier
```

- *Créer trois fichiers* `fic1`*,* `fic2` *et* `fic3`*. Donner tous les droits d'accès à tous pour* `fic1` *et uniquement aux utilisateurs de votre groupe pour* `fic2`*. Enfin, donner les droits de lecture seule à tous pour le fichier* `fic3`*.*

- *Que se passe-t-il si on enlève les droits en exécution sur un répertoire ?*

▸ **Exercice 3.** ***(Mon premier programme** C**)***
Créer un répertoire `TP1` *et vous y placer. À l'aide de l'éditeur de texte emacs, taper le texte ci-dessous en respectant scrupuleusement la syntaxe.*

```
#include <stdio.h>

int main(void)
{
  printf("Voici mon programme en C.\n");
  return 0;
}
```

Sauvegarder votre travail dans un fichier de nom `premier.c`*. Compiler ce code source à l'aide de la commande* `gcc -Wall premier.c`*. Vous obtenez alors un fichier exécutable nommé* `a.out`*. Exécuter le programme en tapant* `a.out`*. Expliquer en détail le source C.*

▸ **Exercice 4.** ***(Somme de deux entiers)*** *Tout programme fait appel à des données. Celles-ci peuvent prendre des valeurs évoluant pendant toute l'exécution du programme. On parle alors de variables. Le type d'une variable détermine les valeurs qu'elle peut prendre ainsi que les opérations pouvant s'y appliquer. En C, les principaux types standards de variables sont :* `char` *(caractère),* `int` *(entier) et* `float` *(réel). Exécuter et commenter le programme suivant :*

```
#include <stdio.h>

int main(void)
{
  int i,j;

  i=5;  j=3;
  printf("i=%d, j=%d, somme=%d\n",i,j,i+j);
  return 0;
}
```

Modifier le programme pour afficher la différence, le produit puis le quotient de deux variables. Dans ce dernier cas, que se passe-t-il si le diviseur est nul ?

▸ **Exercice 5.** ***(Conversion Francs-Euros)*** *Écrire un programme qui convertit une valeur (supposée en Francs) en Euros. (Rappel : 1 Euro = 6.55957 F).*