
Séances 1 et 2 – Algo Prog en C

1. Créer et compiler un programme

1.1 quelques commandes système à connaître

Tapez votre login (nom d'utilisateur), puis votre mot de passe.

Faire apparaître un terminal.

Commandes système à connaître :

- cd (permet de changer de répertoire courant)
- ls
- pwd
- rm
- mkdir
- cp

Que signifient ces commandes ? Pour le savoir tapez :

```
man nomCommande
```

Exemple : `man rm`

Ensuite (toujours dans le terminal) :

- lister le contenu du répertoire.
- Afficher le nom du répertoire courant

- Créer un répertoire nommé `algoProg`, et un sous-répertoire nommé `tp0`
- Se placer dans ce sous-répertoire
- On va créer quelques fichiers (vides) ; pour cela, on peut utiliser la commande `touch` (exemples `touch toto` créé le fichier vide nommé « toto »)
- créer les fichiers vides suivants : `toto.un`, `toto1.a`, `toto1.b`, `toto2.o`, `toto3.o`
- vérifier qu'ils sont bien créés
- effacer le fichier « `toto.un` » (vérifier qu'il n'existe plus)
- effacer en une seule commande tous les fichiers « `toto1` » quel que soit leur extension (vérifier)
- enfin, effacer tous les fichiers ayant comme extension « `o` »

Pour écrire un programme C, il faut taper le programme dans un éditeur de texte (`kate`, `xemacs`, `nedit`, `edit`...).

Dans l'éditeur, tapez le programme suivant :

```
int main(void)
{
printf("Un premier programme.\n");
}
```

Sauvegarder sous le nom `minimal.c`

2 Compiler et exécuter un programme

Compiler (appel au compilateur " gcc ") ce fichier en tapant : \$ cc minimal.c

Quel fichier obtient-on ? Exécutez-le en tapant son nom

Ensuite supprimez ce fichier

Puis créez un programme objet à partir du programme source `minimal.c` en tapant :

`$ gcc -c minimal.c`

Tapez : `$ ls minimal.`*

Créez un programme exécutable à partir du programme objet en tapant :

`$ gcc minimal.o -o minimal`

Observez les fichiers créés à chaque étape par le compilateur. Utiliser la commande `nm` pour lister les symboles contenus dans les fichiers objets et exécutables.

Utilisez la commande : ” `$ cc -E minimal.c` ” afin de n'effectuer que la phase de précompilation.

Modifiez le programme source en y incluant en première instruction : ” `#include <stdio.h>` ” . Reprendre les fichiers construits à chaque étape et regardez les différences avec ce que vous aviez obtenu la première fois (sans cette instruction).

Enfin, effacez tous les fichiers, SAUF le fichier `minimal.c`, puis tapez `gcc minimal.c -o minimal`

Que constatez-vous ? Exécutez le programme.

3. premiers exercices de programmation

Ecrire un programme nommé exo3 qui affiche bonjour.

Ecrire un programme avec les lignes suivantes :

```
...  
int naiss ;  
int annee ;  
annee = 2009 ;  
printf("on est en %d : \n", annee);
```

Exécuter ce nouveau programme, puis ajouter les 2 lignes suivantes :

```
....  
printf("quelle est votre année de naissance ?\n ");  
scanf("%d", &naiss);
```

Enfin compléter votre nouveau code de telle sorte qu'il affiche l'âge de l'utilisateur.