

Programmer en C sous unix

1 Instructions simples pour impatientes

À l'aide d'un éditeur de texte (par exemple `emacs`, `kwrite` ou encore `vim`), on tape le code dans un fichier dont le nom se termine par `.c`.

Le `.c` doit contenir une fonction nommée `main` avec le type suivant :

```
int main(int argc, char *argv[])
```

C'est cette fonction qui est exécutée au lancement du programme. Le programme se termine avec elle, donc elle doit se charger de lancer ce qu'il faut. C'est exactement la même chose que les fonctions `main` des classes Java.

On n'oublie pas d'enregistrer, puis, on compile. Par exemple, pour compiler le fichier `ploum.c`, on tape la commande `gcc ploum.c`. Cela crée un fichier `a.out`, qui contient le programme compilé directement exécutable.

Pour exécuter `a.out`, il faut taper la commande `./a.out` (on met un `./` pour dire au shell que le programme à exécuter se trouve dans le répertoire courant, car normalement il va chercher les programmes dans une certaine liste de répertoires). Contrairement au compilateur Java, le compilateur C produit un programme en langage machine, donc exécutable sans passer par un interpréteur.

2 Plus de détails

2.1 Choisir le nom de l'exécutable créé par le compilateur

Il suffit d'ajouter l'option `-o nomvoulu` à la ligne de commande de compilation.

Par exemple, `gcc -o plouf ploum.c` va créer un exécutable nommé `plouf` au lieu de `a.out`. On tapera donc `./plouf` au lieu de `./a.out`.

2.2 Compilation séparée

Lorsque l'on compile `ploum.c` avec la commande `gcc ploum.c`, un fichier `a.out` est créé, mais également un *fichier objet* temporaire de type `.o`. Ce fichier contient le code de `ploum.c` traduit en langage machine, mais sous une forme pas encore exécutable. Après avoir créé le fichier temporaire `.o`, le compilateur effectue une deuxième étape (appelée *linkage* ou encore *édition de liens*) pour créer le `a.out` exécutable.

Cette séparation en deux phases permet de diviser ses programmes en plusieurs fichiers, ce qui permet, entre autre, de ne recompiler que la partie qui a changé par exemple.

L'idée consiste à lancer le compilateur sur chacun des `.c`, en lui disant de s'arrêter à la première phase (appelée *compilation*). Par exemple, si l'on a au départ `ploum.c`, `plim.c` et `plam.c` (l'un seul d'entre eux contenant une fonction `main`), on va exécuter les commandes : `gcc -c ploum.c`, `gcc -c plim.c`, et `gcc -c plam.c` (l'option `-c` sert à dire au compilateur de ne faire que la génération du `.o`). On a alors obtenu `ploum.o`, `plim.o`, et `plam.o`, que l'on veut rassembler en un unique exécutable. Cela se fait avec la commande `gcc ploum.o plim.o plam.o` (on peut toujours rajouter un `-o` pour choisir le nom de l'exécutable créé).

2.3 Récupération de la ligne de commande

La fonction `main(int argc, char *argv[])` appelée au lancement du programme reçoit en argument la ligne de commande qui a servi à lancer le programme.

Le premier argument est un entier, qui indique le nombre d'éléments de cette ligne de commande. Le second est un tableau de strings contenant ces éléments. Par exemple, si l'on tape la ligne de commande `./a.out pouet eheh plouf`, `argc` vaudra 4, et `argv` est un tableau de chaînes de caractères qui vaudra `{"./a.out", "pouet", "eheh", "plouf"}`.

3 Quelques fonctionnalités utiles d'emacs

Si vous choisissez d'utiliser emacs pour taper vos `.c`, voici quelques astuces pour vous simplifier la vie.

colorisation Activer l'option « Syntax highlighting » du menu « Options »

indentation automatique La touche « Tab » dit à emacs d'indenter automatiquement la ligne

compilation On peut lancer les compilations à l'intérieur d'emacs. L'avantage est qu'emacs peut lire les messages d'erreur et sauter directement aux lignes concernées. Il faut choisir la commande « Compile » du menu « Tool ». Emacs demande alors d'entrer la ligne de commande de compilation (dans la ligne tout en bas de la fenêtre). La commande de compilation proposée par défaut sera a priori `make -k`. Ce n'est pas ce qu'on veut (pour l'instant), il faut l'effacer, puis taper la bonne, et appuyer sur entrée. La fenêtre d'emacs se scinde alors en deux, et les messages du compilateur apparaissent en bas. On peut passer en revue toutes les erreurs en tapant « contrôle x ´ » (un accent grave, pas une apostrophe).