

Bases de C

TP 0 - Préparatifs

Ce TP a pour objectif de vous faire découvrir le langage C et les outils nécessaires pour le faire fonctionner. Vous démarrerez depuis l'écriture de quelques lignes de code que vous compilerez, pour finalement réaliser un petit programme de statistiques. Durant tout l'exercice, vous utiliserez au fur et à mesure quelques notions essentielles du langage C.

1 Prise en main de C

Avant de pouvoir utiliser C, vous aurez besoin d'installer l'environnement de développement : le compilateur C, et plusieurs outils pour débbugger et automatiser une partie de la chaîne de compilation. De plus, vous devez choisir un éditeur texte qui vous permettra de développer facilement (grâce à la coloration syntaxique, notamment).

Selon votre système d'exploitation, vous pouvez trouver quelques éditeurs texte comme : *notepad++*, *sublimtext*, *vim*, *emacs*, ...

Pour pouvoir *compiler* du code (c'est-à-dire transformer un code écrit dans un langage haut niveau en un code bas niveau/compréhensible par votre processeur), il vous faut disposer d'un programme faisant ce travail : le compilateur.

Plusieurs compilateurs pour le C existent : *visual studio*, *icc*, *gcc*, *clang*, ...

Nous utiliserons dans ce cours *gcc* et *clang* indifféremment.

Si vous êtes sur linux, sur des distributions dérivées de Debian (comme Ubuntu), vous devrez installer tous ces programmes :

- gcc
- g++
- make
- gdb
- valgrind
- m4
- bash
- tar
- openssl

Certains sont déjà inclus dans un pack de programmes extrêmement utile : *build-essential*

Pour installer ces programmes sur un linux dérivé de Debian, vous devez entrer dans le terminal :

```
apt-get update
```

```
apt-get install build-essential
```

```
apt-get install gcc g++ make gdb valgrind m4 bash
```

En cas d'échec pour manque de permissions, vous devrez précéder chacune de vos commandes par un **sudo**, ce qui donnera par exemple :

```
sudo apt-get update
```

Si vous avez installé un autre linux, ou un autre dérivé d'UNIX, vous devrez trouver vous-même comment installer ces programmes (et éventuellement leurs dépendances).

Si vous êtes sur macOS, vous devrez probablement installer une VM (comme pour Windows), ou alors utiliser un gestionnaire de paquets externe (comme *MacPorts*, ou *HomeBrew*) et *Xcode* que vous trouverez dans l'app store officiel d'Apple.

Une fois installés, vous pouvez fermer votre terminal, et en rouvrir un nouveau dans lequel vous pourrez taper la commande suivante :

gcc

Si tout est correctement installé, vous devriez obtenir comme réponse dans le terminal :

```
gcc: fatal error: no input files
compilation terminated.
```

Si vous utilisez **clang** malgré vous (par exemple sur macOS), vous devriez obtenir ce message :

```
clang: error: no input files
```

Pour accéder à la documentation des fonctions et appels systèmes, vous pouvez la commande **man**. La section numéro 2 sert aux appels système (**open**, **read**, **write**, **close**, ...), et la section numéro 3 sert aux fonctions des bibliothèques (**printf**, **malloc**, **memset**, **fopen**, **fread**, ...).

Pour lire la documentation de la fonction **malloc(3)**, il suffit donc de taper dans le shell :

```
man 3 malloc
```

Pour avancer dans le manuel, vous pouvez utiliser les flèches et la touche espace. Pour quitter le manuel, il suffit d'appuyer sur la touche **q** (**Q** minuscule).