

GCC

Le compilateur **gcc(1)** est un compilateur largement utilisé pour les programmes écrits en C. Il permet de produire un fichier *exécutable*, qui pourra être exécuté pour effectuer ses opérations, à partir des fichiers *source*, qui contiennent le programme écrit en C. Les fichiers *source* ont en général l'extension `.c`, alors que le fichier *exécutable* n'a généralement pas d'extension (du moins dans le cadre de ce cours).

Soit un fichier *source* `helloworld.c` contenant le code C suivant:

```
#include <stdlib.h>
#include <stdio.h>

// Main function, prints "Hello world !"
int main(int argc, char const *argv[]) {
    printf("Hello world !\n");

    return EXIT_SUCCESS;
}
```

Pour compiler ce programme et produire l'exécutable `prog`, il suffit d'utiliser **gcc(1)**:

```
$ gcc helloworld.c -o prog
```

Discutons de chacune des parties de cette commande:

- `gcc` indique l'outil shell utilisé, ici **gcc(1)**
- `helloworld.c` indique le fichier source, en C
- l'option `-o` permet de spécifier le nom du fichier produit, qui dans ce cas est le fichier exécutable
- `prog` est le nom du fichier exécutable

On peut maintenant exécuter le programme `prog`:

```
$ ./prog
Hello world !
```

Cette description basique de **gcc(1)** est suffisante pour le début de ce cours. Pour savoir comment compiler des programmes constitués de plusieurs fichiers sources, veuillez vous référer à la partie **De grands programmes en C** de la partie Théorie du syllabus.

Compléments

De manière technique, le travail du compilateur peut être découpé selon 4 étapes distinctes:

- Appel du préprocesseur `cpp`: Supprime les commentaires, inclus les `#include` et évalue les macros
- Appel du compilateur `cc1`: Génère un fichier assembleur `.as`
- Appel de l'assembleur `as`: Génère le fichier objet `.o`
- Appel du de l'éditeur de liens `ld`: Génère l'exécutable

Différentes options peuvent être utilisées avec `gcc`:

- `-E` : Appelle uniquement le préprocesseur
- `-S` : Appelle uniquement le préprocesseur et le compilateur
- `-C` : Appelle le préprocesseur, le compilateur et l'assembleur
- `-o nom` : Détermine le nom du fichier de sortie
- `-g` : Option nécessaire pour générer les informations symboliques de débogage avec `gdb`
- `-On` : Indique le niveau d'optimisation où `n` est compris entre 0 et 3
- `-Wall` : Active tout les warnings **tous**
- `-Werror` : Considère tout les warnings comme des erreurs **tous**
- `--help` : Messages d'aide

Notons que les trois premières options ne présentent pas d'intérêt pour ce cours.