

Module Algorithmie et programmation

La fonction main et les arguments

Pierre Künzli

Inspiré des cours de Paul Albuquerque, Guido Bologna et Orestis Malaspinas

Généralités

- Point d'entrée du programme.
- Retourne le code d'erreur du programme :
 - 0 : tout s'est bien passé.
 - Pas zéro : problème.
- La valeur de retour peut être lue par le shell qui a exécuté le programme.
- EXIT_SUCCESS et EXIT_FAILURE (de `stdlib.h`) sont des valeurs de retour **portables** de programmes C.

Exemple

```
int main() {  
    // ...  
    if (error)  
        return EXIT_FAILURE;  
    else  
        return EXIT_SUCCESS;  
}
```

- Le code d'erreur est lu dans le shell avec \$?

```
$ ./prog  
$ echo $?  
0 # tout s'est bien passé par exemple  
$ if [ $? -eq 0 ]; then echo "OK" ; else echo "ERROR"; fi  
ERROR # si tout s'est mal passé
```

La fonction main

- Elle peut être déclarée de 4 façon différentes :
 1. `void main()`.
 2. `int main()`.
 3. `void main(int argc, char **argv)`.
 4. `int main(int argc, char **argv)`.
- `argc` est le nombre d'arguments passés à la ligne de commande : **le premier est celui du programme lui-même.**
- `argv` est un tableau de chaînes de caractères passés sur la ligne de commande.

Lea arguments de la fonction main

Pour la fonction dans le programme prog

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

Pour l'exécution suivante on a

```
$ ./prog -b 50 file.txt
```

```
argc == 4  
argv[0] == "prog"  
argv[1] == "-b"  
argv[2] == "50"  
argv[3] == "file.txt"
```

Les arguments

Conversion des arguments

- Les arguments sont toujours stockés comme des **chaînes de caractère**.
- Peu pratique si on veut manipuler des valeurs numériques.
- Fonctions pour faire des conversions :

```
int atoi(const char *nptr);  
long atol(const char *nptr);  
long long atoll(const char *nptr);  
double atof(const char *nptr);
```

Exemple d'utilisation

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

int main(int argc, char **argv) {
    if (argc != 3) {
        char *progrname = argv[0];
        printf("usage: %s name age\n", progrname);
        return EXIT_FAILURE;
    }

    char *name = argv[1];
    int age = atoi(argv[2]);

    printf("Hello %s, you are %d years old.\n", name, age);
    return EXIT_SUCCESS;
}
```

```
$ ./prog Paul 29
Hello Paul, you are 29 years old.
```