

[LC081]

## La fonction `main()`

La fonction `main()` est une fonction qui n'a pas besoin d'être *déclarée*.

La fonction `main()` est de type `int`<sup>1</sup>. C'est-à-dire qu'elle est censée retourner une valeur *entière* au programme qui l'a appelée. En général celui-ci est le *système* d'exploitation.

La fonction `main()` peut retourner cette valeur :

- soit en la plaçant à la suite de l'instruction comme pour toute autre fonction,
- soit en utilisant la fonction `()` avec la valeur de retour comme argument.

La valeur de retour de la fonction `main()` est, en général et par convention : `=0` si tout est OK *{return(...)}*  
`≠0` en cas d'erreur *{exit(...)}*

Les arguments de la fonction `main()` sont au nombre de trois, sont facultatifs (sauf que si l'un est présent, les précédents doivent l'être aussi) et sont appelés en général : `argc`, `argv` et `env`.

Les syntaxes correctes de l'entête de la fonction `main()` sont alors les suivantes<sup>2</sup> :

- `int main()` ou `int main(void)`
- `int main(int argc, char *argv[])`
- `int main(int argc, char *argv[], char *env[])`

Plus précisément les trois arguments de `main()` sont dans l'ordre :

- `int argc` → un entier correspondant au nombre de paramètres de la ligne de commande (lorsque le programme est lancé par une commande interprétée par le système d'exploitation (shell sous UNIX))
- `char * argv[]` → un tableau de pointeurs vers (chaîne de) caractères. Chaque pointeur pointe vers l'un des paramètres de la ligne de commande. Un pointeur de plus est NULL et peut servir pour un test d'arrêt.
- `char * env[]` → un tableau de pointeurs vers (chaîne de) caractères. Chaque pointeur pointe vers une variable d'environnement. Il est à noter que cet argument est peu utilisé, et en fait non-standard, bien que disponible sur la plupart des systèmes. On utilisera en général de préférence la fonction `getenv()`.

Exemple : `com fich1 fich2.* 723`

(MSDOS ou Windows)

```
argc = 4
argv[0] → "com"
argv[1] → "fich1"
argv[2] → "fich2.*"
argv[3] → "723"
argv[4] = NULL
```

(UNIX)

```
argc = 5 (par exemple)
argv[0] → "com"
argv[1] → "fich1"
argv[2] → "fich2.h"
argv[3] → "fich2.c"
argv[4] → "723"
argv[5] = NULL
```

---

```
env[0] → "HOME = /u/mat20"
env[1] → ...
env[1] → ...
...
env[n] = NULL
```

<sup>1</sup> Il est à noter qu'une définition de `main()` commençant par le mot `void` est illégale (suivant les normes usuellement utilisées pour le langage C) et peut parfois conduire à des comportements erronés, voire des plantages (consulter à ce sujet : FAQ list of newsgroup comp.lang.c from <ftp://ftp.eskimo.com/u/s/scs/C-faq/>, questions 11.12 et suivantes ; ainsi que ISO Sec. 5.1.2.2.1, Sec. G.5.1; H&S Sec. 20.1 p.416; CT&P Sec. 3.10 pp. 50-51....)

<sup>2</sup> Le type `int` de ces entêtes peut être en général omis, bien que la norme recommande de le mettre