

Programmation C

Semestre 5 - 2020

[Installer le shell Bash sous Windows 10, sans Dual Boot.](#)

Table des Matières

Table des Matières	2
Introduction	3
Compilation	3
Redirections de base	4
Variables d'environnement	4
Assert	5
Const	5
Valgrind	5
Compiler plusieurs fichiers entre eux pour en créer un exécutable	6

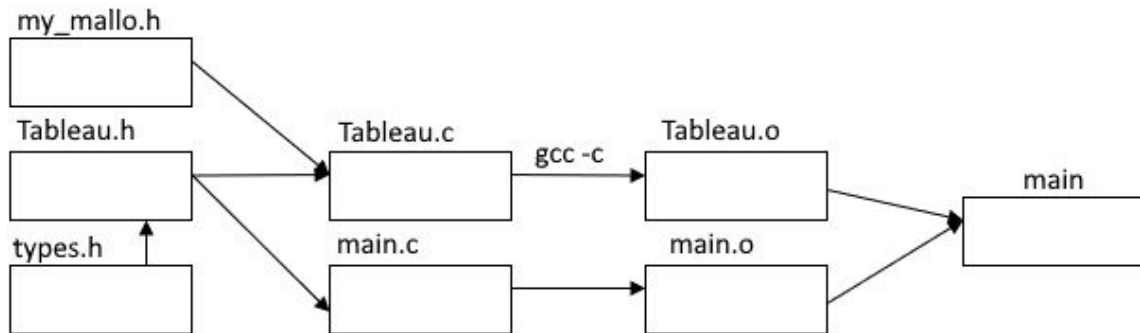
Introduction

Les erreurs et les warnings sont à éviter à tout prix.

Exemple de fonction basique :

```
int main ()
{
    return EXIT_SUCCESS; // Utiliser EXIT_SUCCESS au lieu de 0
}
```

Compilation



Imaginons main a été créé le 8, Tableau.c a été modifié pour la dernière fois le 5 et main.c le 3. Lequel on recompile ?

- Tout ? Non, trop coûteux en temps et ressources.
- Tableau.c, car il a été modifié en dernier.

Redirections de base

- “>” écrire le retour des printf au début d’un fichier (écrase le fichier actuel)
- “>>” écrire le retour des printf en fin d’un fichier
- “2>” écrire les erreurs d’un code au début d’un fichier (écrase le fichier actuel)
- “2>>” écrire les erreurs d’un code en fin d’un fichier
- “<” envoie le contenu d’un fichier dans le scanf d’une commande
- “|” appelé pipe, renvoie le résultat d’une commande en entrée de la suivante
- “&&” permet d’exécuter plusieurs commandes à la suite en une ligne

Variables d’environnement

on peut voir la liste des variables d’environnement avec : “\$ set”

- “& set | grep PATH”
- “& set | grep HOME” trouve le chemin de votre HOME

On peut créer notre propre variable d’environnement !

Exemple :

`$TOTO = “Bonjour”` => TOTO se rajoute dans la liste du set.

Dans `int main(int ac, char * av[], char * env [])`

- `int ac, char * av[]` => variables en entrée
- `char * env []` => liste des variables d’environnement (dernière case du tableau est nulle)

Si on affiche dans un code la dernière case du tableau, puis qu’on fasse `$TOTO = “Bonjour”` : en réexécutant le code TOTO ne s’affichera pas. Il est stocké dans le SHELL, mais n’est pas ajouté à la liste du main. Pour rajouter TOTO, il faudra faire :
`$export TOTO`

Note : on peut définir une variable d’environnement en même temps que l’export :
`$export TUTU = “Salut”`

Assert

`assert()` : permet de détecter une erreur. Il faut mettre la condition vrai en paramètre. (nécessite la librairie `<assert.h>`)

Une macro permet de désactiver les assertions dans le code : `#define NDEBUG`. Comme la macro est utilisé à la compilation, il doit se placer avant l'inclusion de la librairie `assert.h`

Il est possible de définir un fichier global `config.h` servant à inclure et modifier en une seule fois `NDEBUG` dans tous les sous-fichiers d'un projet.

Il est aussi possible de faire un `#define NDEBUG` à la compilation en ligne de commande en utilisant l'option `-DNDEBUG`. `-D` peut remplacer le `#define` de n'importe quelle macro.

Const

La paramètre `const` permet de transformer un objet en constante, le rendant impossible à modifier par la suite. C'est basiquement le `final` de java.

Exemples d'utilisation :

- `const int MY_CONST = 3;`
- `#define MY_CONST 3`

Valgrind

Vous permet de lancer des programmes en vérifiant qu'il n'y a pas de fuite mémoire. Pensez à l'utiliser sur tous vos codes

Compiler plusieurs fichiers entre eux pour en créer un exécutable

- `mygcc -c fichier1.c`
- `mygcc -c fichier2.c`
- `ls ->` fichier1.o et fichier2.o ont été créés
- `mygcc fichier1.o fichier.o -o nom_executable`