

C : introduction

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



- 1 Introduction
- 2 Installations
 - Compilateur
 - IDE
- 3 Avant de commencer
 - Règles de nommage
 - Instructions
 - Bibliothèque standard du C
- 4 Premier projet C

- Langage de programmation
 - Compilé
 - Procédural
 - Fortement typé : acceptant le typage statique
 - Sensible à la casse
- Créé par le professeur américain **Dennis Ritchie** en 1972 aux laboratoires **Bell** à New Jersey aux États-Unis
- Conçu pour le développement du système d'exploitation **UNIX**
- Influence majeure sur plusieurs langages : **C++**, **Java**, **C#**, **PHP**...
- Normalisé depuis 1989 par **ANSI** puis adopté par **ISO** (International Organization for Standardization)

C : avantages

- Un des langages les plus utilisés dans le monde
- Langage de bas niveau : rapide et efficace
- Excellente documentation et vaste communauté
- Portable : **Windows, Mac OS, Linux**
- Permet la programmation procédurale

C : inconvéient

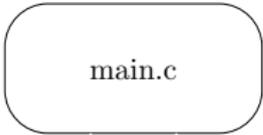
Gestion manuelle de la mémoire : allocation dynamique, libération de la mémoire, risques de fuites...

C : quelques versions (normalisées avec ISO)

- **C89/C90** : première norme **ANSI** en 1989, reprise par **ISO** en 1990
- **C99** (sortie en 1999) : types entiers fixes, nouvelles fonctionnalités du préprocesseur...
- **C11** (sortie en 2011) : support amélioré du multi-threading, nouvelles bibliothèques...
- **C18** (sortie en 2018) : corrections mineures et améliorations de la stabilité

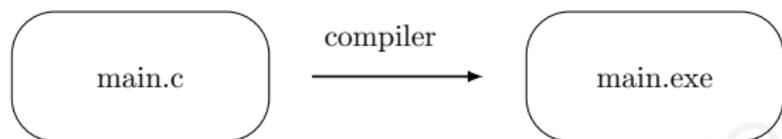
Comment ça fonctionne ?

- On écrit un programme dans un fichier `.c`
- Ensuite, le compilateur traduit le code source et génère un fichier exécutable `.exe` ou `a.out` selon le système
- On lance le fichier exécutable

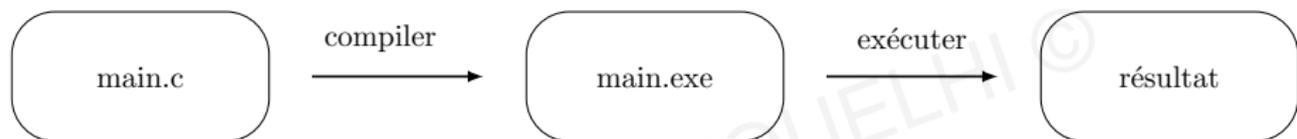


main.c

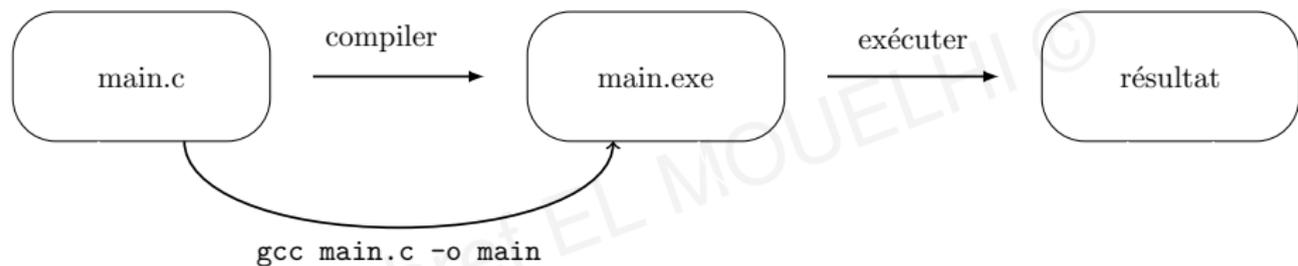
© Achref EL MOUELHI ©

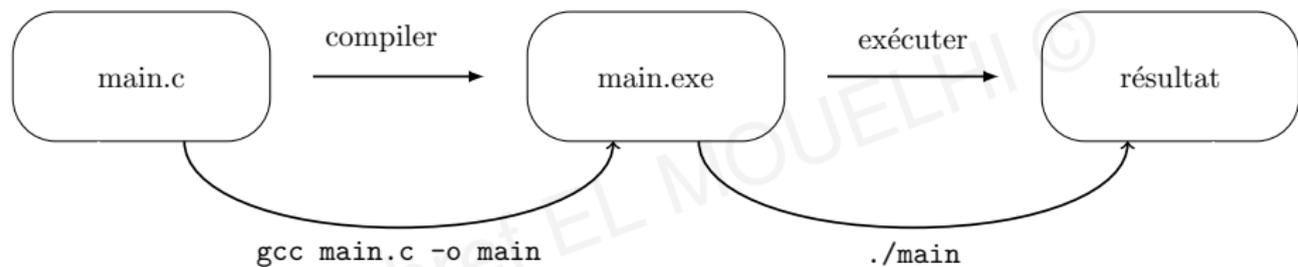


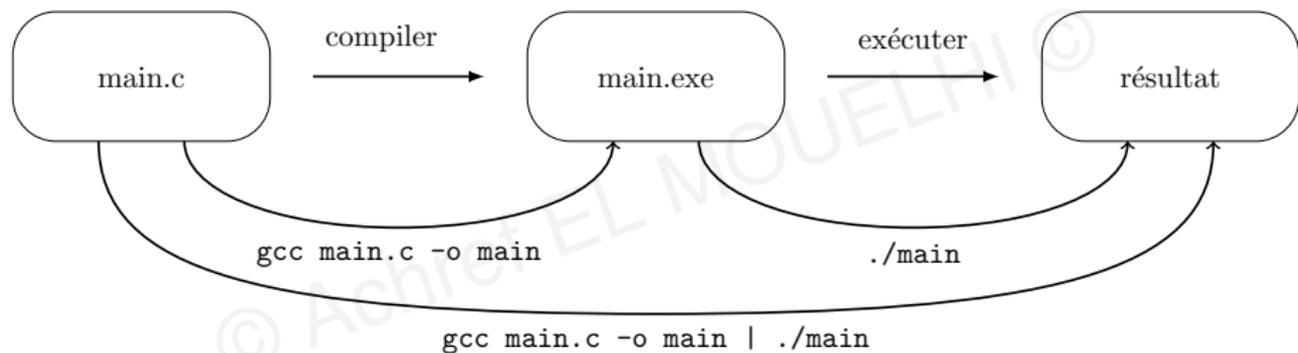
© Achref EL MOUËLHI ©



© Achref EL MOUELHI







De quoi on a besoin (le minimum) ?

- Un éditeur de texte (**Bloc-notes, Notepad++, Sublime Text, Atom...**)
- Un compilateur
 - **GCC (GNU Compiler Collection)** : une famille de compilateur pour les langages **C, C++, Java, Fortran...**, multi-plateforme (pour **Windows, Linux, Android**) et utilisé par le noyau du système **Unix**
 - `gcc` : pour **C**
 - `g++` : pour **C++**
 - `gcj` : pour **Java**
 - Clang : un compilateur pour les langages **C, C++** et **Objective-C**, multi-plateforme (pour **Windows, Mac, Linux, Android, iOS**)
 - Sun Studio : un **IDE** (comprenant un compilateur) pour les langages **C, C++** et **Fortran** développé par **Sun Microsystems**, multi-plateforme (pour **Linux, Solaris**)

Pour installer GCC (GNU Compiler Collection)

- Allez à <https://www.msys2.org/>
- Cliquez sur le lien pour télécharger (Download the installer : `msys2-x86_64-20250221.exe`)
- Lancez l'installation

© Achref EL MOUËZ

Pour installer GCC (GNU Compiler Collection)

- Allez à <https://www.msys2.org/>
- Cliquez sur le lien pour télécharger (Download the installer : `msys2-x86_64-20250221.exe`)
- Lancez l'installation

Dans la console qui vient de s'ouvrir, exécutez (à refaire si la console se referme)

```
pacman -Syu
```

Pour installer GCC (GNU Compiler Collection)

- Allez à <https://www.msys2.org/>
- Cliquez sur le lien pour télécharger (Download the installer : `msys2-x86_64-20250221.exe`)
- Lancez l'installation

Dans la console qui vient de s'ouvrir, exécutez (à refaire si la console se referme)

```
pacman -Syu
```

Exécutez ensuite

```
pacman -S --needed base-devel mingw-w64-x86_64-toolchain
```

Introduction

Remarque

N'oubliez pas d'ajouter une variable de chemin pour le `bin` de **Mingw-w64** (défini généralement dans `C:\msys64`).

© Achref EL M...

Introduction

Remarque

N'oubliez pas d'ajouter une variable de chemin pour le `bin` de **Mingw-w64** (défini généralement dans `C:\msys64`).

Pour vérifier la version de `gcc` installé, exécutez

```
gcc --version
```

On peut aussi utiliser un IDE (Environnement de développement intégré)

- pour éviter d'utiliser la console et les commandes
- pour avoir
 - l'auto-compilation
 - la coloration syntaxique
 - l'auto-complétion
 - l'indentation automatique...
 - ...
- pour une meilleure structuration du projet

Exemple d'IDE pour C++

- **Visual Studio Code** : open-source, multi-plateforme (**Windows, Mac, Linux**)
- CodeBlocks : open-source, multi-plateforme (**Windows, Mac, Linux**) et écrit en **C++** utilisant par défaut GNU Gcc Compiler et supportant plusieurs autres
- Visual C++ : disponible en version gratuite Visual C++ Express pour **Windows**
- Eclipse avec le plugin **CDT6** (open-source)
- Netbeans (open-source)
- XCode : pour **Mac**
- DevC++ : pour **Windows**
- ...

Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit.
- Extensible selon le langage de programmation.

© Achref EL M...

C

Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit.
- Extensible selon le langage de programmation.

VSC : téléchargement

`code.visualstudio.com/download`

Quelques raccourcis pour VSC

- Pour activer la sauvegarde automatique : aller dans `File > AutoSave`
- Pour indenter son code : `Alt` `Shift` `f`
- Pour commenter/décommenter : `ctrl` `:`
- Pour faire une sélection multiple : `Ctrl` `f2`
- Pour dupliquer une sélection : `Ctrl` `d`
- Pour consulter la liste des raccourcis : `Ctrl` `k` + `Ctrl` `s`

En cas de problème d'affichage de lettres accentués dans le terminal, exécutez

```
chcp 65001
```

Quelques extensions **VSC** pour **C**

- **C/C++** : pour l'intelliSense
- **Code Runner** : pour l'ajout d'un bouton de compilation

Règles de nommage en C

- Pour les variables et les fonctions : **snake_case**
- Pour les constantes : **ALL_CAPS_SNAKE_CASE** (majuscule avec underscores)
- Pour les types définis par l'utilisateur (`struct`, `enum`, `typedef`) : **PascalCase** ou **snake_case**

© Achre

C

Règles de nommage en C

- Pour les variables et les fonctions : **snake_case**
- Pour les constantes : **ALL_CAPS_SNAKE_CASE** (majuscule avec underscores)
- Pour les types définis par l'utilisateur (`struct`, `enum`, `typedef`) : **PascalCase** ou **snake_case**

Pour plus de détails

<https://wprock.fr/blog/conventions-nommage-programmation/>

Instructions

- Chaque instruction se termine par ;
- Il est possible d'écrire plusieurs instructions sur une même ligne (**mais** ce n'est pas une bonne pratique)

Bibliothèque standard du C

- Fournit un ensemble de fonctions pour la gestion des entrées/sorties, des chaînes de caractères, des allocations mémoire, etc.
- Contient plusieurs composants importants :
 - **stdio.h** : gestion des entrées/sorties (`printf`, `scanf`, `fopen`, `fclose`...)
 - **stdlib.h** : allocation mémoire dynamique (`malloc`, `free`), conversion de types...
 - **string.h** : manipulation des chaînes de caractères (`strlen`, `strcpy`, `strcat`...)
 - **math.h** : opérations mathématiques (`sin`, `cos`, `pow`, `sqrt`...)

Démarche

- Créez un répertoire `cours-c` dans votre espace de travail
- Lancez **VSC** et allez dans `File > Open Folder...` et choisissez `cours-c`
- Dans `cours-c`, créez un fichier `main.c`

Ajoutons le code suivant dans `main.c`

```
#include <stdio.h>

int main() {
    printf("Hello, World!\n");
    return 0;
}
```

Explication

- `#include <stdio>` : la librairie permettant de gérer les flux entrant/sortant (input/output).
- `int main()` : le point d'entrée vers notre projet (programme).
- `printf` : permet d'afficher le message situé entre "...".
- `return 0` :
 - `main` doit retourner 0 pour une sortie sans erreur, autre valeur sinon.
 - Si aucun `return` n'est présent, le compilateur considère que la valeur de retour est par défaut 0.

Comment exécuter le programme ? (voir le résultat)

- Soit en faisant clic droit et ensuite choisir `Run Code`
- Soit en cliquant sur `▷` en haut à droite
- Soit avec le raccourci clavier `Ctrl` + `Alt` + `N`

© Achref EL

Comment exécuter le programme ? (voir le résultat)

- Soit en faisant clic droit et ensuite choisir `Run Code`
- Soit en cliquant sur `▷` en haut à droite
- Soit avec le raccourci clavier `Ctrl` + `Alt` + `N`

Constats

- Un fichier `main.exe` généré.
- **Hello, World!** s'affiche dans la console de **VSC**.