



Primeiros passos

A primeira coisa que precisamos conversar é sobre a plataforma para execução do curso. Se você tem um computador com **Linux**, então está tudo ok. O curso foi criado para ambiente Linux, portanto todos os exemplos e comandos de terminal são utilizando o Linux shell.

O que fazer se tiver **MS Windows** ou **MacOS**?

1. Utilizar uma máquina virtual Linux (VirtualBox ou outro software)
2. Rodar um contêiner Linux em um gerenciador como o **Docker**
3. Utilizar comandos e compiladores locais do Windows e do MacOS

Se decidir utilizar Máquina virtual ou Contêiner, então tudo certo. Lembre-se de utilizar uma imagem que tenha o GCC instalado.

Agora, se decidir utilizar a plataforma nativa do seu computador, então podemos tentar instalar a plataforma nela.

Na pasta **código** nos downloads da aula (quase toda aula tem anexos: Um PDF e uma pasta "codigo" com exemplos. Baixe o zip e descompacte).

Linux (baseado em Debian)

Como instalar as ferramentas necessárias para desenvolver em C e C++ no Linux (Debian/Ubuntu).

Instalando as ferramentas básicas

1. **Abra o terminal.**

2. Atualize a lista de pacotes:

```
sudo apt update
```

3. Instale o compilador GCC (GNU Compiler Collection), que suporta C e C++:

```
sudo apt install build-essential
```

O pacote **build-essential** inclui o **gcc**, **g++**, e outras ferramentas úteis como **make**, que facilita a compilação de programas.

Instalando editores de texto ou IDEs

Você pode usar qualquer editor de texto ou IDE que preferir. Alguns populares para desenvolvimento em C e C++ são:

- **Vim ou Nano (editores de texto simples):**

```
sudo apt install vim  
sudo apt install nano
```

- **Visual Studio Code (VS Code):**

Primeiro, adicione o repositório do VS Code:

```
wget -q0- https://packages.microsoft.com/keys/microsoft.asc | gpg --  
dearmor > packages.microsoft.gpg  
sudo install -o root -g root -m 644 packages.microsoft.gpg  
/usr/share/keyrings/  
sudo sh -c 'echo "deb [arch=amd64 signed-  
by=/usr/share/keyrings/packages.microsoft.gpg]  
https://packages.microsoft.com/repos/code stable main" >  
/etc/apt/sources.list.d/vscode.list'
```

Em seguida, instale o VS Code:

```
sudo apt update  
sudo apt install code
```

- **Code::Blocks (IDE específica para C/C++):**

```
sudo apt install codeblocks
```

Configurando o ambiente de desenvolvimento

1. Verifique se o GCC está instalado corretamente:

```
gcc --version
g++ --version
```

Você deve ver a versão do GCC e G++ instalada.

gcc e **g++** são duas maneiras de invocar o GNU C Compiler, sendo a primeira voltada para programas em C e a segunda para programas em C++.

2. Instale bibliotecas e headers de desenvolvimento (opcional, mas recomendado para desenvolvimento mais avançado):

```
sudo apt install libboost-dev libncurses5-dev libncursesw5-dev
```

Teste seu ambiente

Crie um arquivo simples para testar:

1. Baixe o anexo da aula e descompacte

2. Examine o código de exemplo:

```
#include <stdio.h> int main (int argc, char * argv[]) { printf("Meu primeiro programa em 'C'\n"); return 0; }
```

3. **Compile o programa:**

```
```bash
gcc exemplo1.c
```

#### 4. Execute o programa:

```
./a.out
```

Se tudo estiver correto, você verá a mensagem "Meu primeiro programa em 'C'" no terminal. Pronto! Seu ambiente de desenvolvimento em C e C++ está configurado. Se tiver mais dúvidas ou precisar de algo

específico, é só falar!

## Microsoft Windows

Vou explicar como instalar o MinGW e usar o GCC no Microsoft Windows para compilar e executar o seu programa em C.

### Instalando o MinGW

#### 1. Baixe o Instalador do MinGW:

- Vá até o site do [MinGW](#) e clique em "Download" para baixar o instalador `mingw-get-setup.exe`.

#### 2. Execute o Instalador:

- Abra o arquivo `mingw-get-setup.exe` e siga as instruções para instalar o MinGW. Durante a instalação, selecione a opção para instalar o `gcc`.

#### 3. Configure o Instalador:

- Quando o instalador abrir o gerenciador de pacotes, selecione as seguintes opções:
  - Em "Basic Setup", clique com o botão direito do mouse em `mingw32-gcc-g++` e escolha "Mark for Installation".
  - Em "Basic Setup", clique com o botão direito do mouse em `mingw32-gcc-objc` e escolha "Mark for Installation" (opcional, para C++ e Objective-C).
  - Em "Basic Setup", clique com o botão direito do mouse em `mingw32-gcc-objc` e escolha "Mark for Installation" (opcional, para Objective-C).
  - Clique em "Installation" e depois em "Apply Changes" para instalar os pacotes selecionados.

#### 4. Adicione o MinGW ao PATH:

- Adicione o diretório `bin` do MinGW ao PATH do sistema para poder usar os comandos `gcc` e `g++` no terminal. O caminho geralmente é `C:\MinGW\bin`.
- Para adicionar ao PATH:
  1. Clique com o botão direito em "Este PC" ou "Meu Computador" e selecione "Propriedades".
  2. Clique em "Configurações avançadas do sistema".
  3. Na aba "Avançado", clique em "Variáveis de Ambiente".
  4. Em "Variáveis de sistema", encontre e selecione a variável `Path`, depois clique em "Editar".
  5. Adicione o caminho `C:\MinGW\bin` ao final da lista.

### Compilando e Executando seu Programa em C

#### 1. Baixe o anexo da aula

- Localize o arquivo `exemplo1.c` com o seguinte código:

```
#include <stdio.h>

int main (int argc, char * argv[]) {
 printf("Meu primeiro programa em 'C'\n");
 return 0;
}
```

## 2. Abra o Prompt de Comando:

- Pressione **Win + R**, digite **cmd** e pressione Enter.

## 3. Navegue até o Diretório do Arquivo:

- Use o comando **cd** para mudar para o diretório onde você salvou o arquivo **exemplo1.c**. Por exemplo, se você salvou em **C:\Users\SeuUsuario\Documents**, digite:

```
cd C:\Users\SeuUsuario\Documents
```

## 4. Compile o Programa:

- Use o comando **gcc** para compilar o arquivo **exemplo1.c**:

```
gcc exemplo1.c -o exemplo1.exe
```

- Isso criará um executável chamado **exemplo1.exe**.

## 5. Execute o Programa:

- Digite o nome do executável para executar o programa:

```
exemplo1.exe
```

- Você verá a saída:

```
Meu primeiro programa em 'C'
```

# MacOS

Agora veremos como instalar o GCC no macOS usando o **Homebrew** e como compilar e executar o seu programa em C.

Lembrando que você pode utilizar as ferramentas de desenvolvimento do **Xcode** em vez do GCC.

## Instalando o Homebrew

Homebrew é um gerenciador de pacotes para macOS que facilita a instalação de software. Vamos usá-lo para instalar o GCC.

### 1. Instale o Homebrew:

- Abra o Terminal (você pode encontrar o Terminal na pasta Aplicativos > Utilitários).
- Execute o seguinte comando para instalar o Homebrew:

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.s
h)"
```

- Siga as instruções na tela para completar a instalação.

### 2. Adicione o Homebrew ao PATH:

- Depois da instalação, o script mostrará um comando que você deve adicionar ao seu arquivo de configuração do shell (por exemplo, `.zshrc` ou `.bash_profile`). Ele se parecerá com algo assim:

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >>
/Users/yourusername/.zprofile
eval "$(/opt/homebrew/bin/brew shellenv)"
```

- Execute esse comando no terminal.

## Instalando o GCC

### 1. Atualize o Homebrew:

```
brew update
```

### 2. Instale o GCC:

```
brew install gcc
```

- Homebrew irá instalar a versão mais recente do GCC. No entanto, no macOS, o comando `gcc` será prefixado com a versão, por exemplo, `gcc-11` para a versão 11 do GCC.

## Compilando e Executando seu Programa em C

### 1. Baixe o anexo da aula:

- Localize o arquivo `exemplo1.c` com o seguinte conteúdo:

```
#include <stdio.h>

int main (int argc, char * argv[]) {
 printf("Meu primeiro programa em 'C'\n");
 return 0;
}
```

## 2. Abra o Terminal:

- Use o Spotlight (Cmd + Espaço) e digite "Terminal" para abrir.

## 3. Navegue até o Diretório do Arquivo:

- Use o comando `cd` para mudar para o diretório onde você salvou o arquivo `exemplo1.c`. Por exemplo, se você salvou em `Documents`, digite:

```
cd ~/Documents
```

## 4. Compile o Programa:

- Use o comando `gcc` seguido do número da versão instalada (por exemplo, `gcc-11`) para compilar o arquivo `exemplo1.c`:

```
gcc-11 exemplo1.c -o exemplo1
```

- Isso criará um executável chamado `exemplo1`.

## 5. Execute o Programa:

- Digite o seguinte comando para executar o programa:

```
./exemplo1
```

- Você verá a saída:

```
Meu primeiro programa em 'C'
```