

Básico de GCC

Prof. André Grégio

O gcc é um **compilador** de código aberto muito popular para a linguagem C, suportando também outras linguagens, como C++. Um compilador para C é um programa que transforma código escrito na linguagem de programação C em um programa que o seu computador consegue entender e executar.

Quando você escreve um programa em C, você está essencialmente escrevendo instruções detalhadas que você quer que o computador execute. Porém, o computador não entende diretamente código escrito em C, mas um conjunto específico de instruções chamado código de máquina. Portanto, para que o computador seja capaz de executar seu programa escrito em C, o código C precisa ser traduzido para código de máquina. É aí que entra o compilador.

Um compilador como o GCC pega o seu código C, verifica se está tudo sintaticamente correto (ou seja, se segue todas as regras da linguagem C) e traduz as instruções para código de máquina. O resultado da compilação (tradução) é um arquivo binário, também chamado de executável, que pode ser executado no seu computador. Superficialmente, um compilador possui três tarefas principais feitas por analisadores:

- **Analisador Léxico:** lê o código-fonte e divide-o em partes menores chamadas *tokens*, como variáveis, funções e operadores.
- **Analisador Sintático:** verifica se os *tokens* gerados no passo anterior estão organizados de uma maneira que faça sentido de acordo com as regras da linguagem de programação (verificador gramatical).
- **Analisador Semântico:** verifica se as instruções do programa fazem sentido em um dado contexto (por exemplo, se o programador tenta somar um número inteiro com uma palavra) e atribui significado a essas instruções.

Voltando ao GCC e pressupondo que você já está com seu ambiente para programação pronto, vamos compilar e executar um programa via linha de comando em **três passos**:

1. Crie um arquivo chamado `alomundo.c` com o seguinte conteúdo:

```
#include <stdio.h>

int main() {
    printf("Alô Mundo!\n");
    return 0;
}
```

2. Em um terminal do Linux, por exemplo, compile seu código da maneira mais básica possível com o seguinte comando (o sinal de cifrão, \$, mostrado em um terminal *shell*, é também chamado de *prompt* e indica que você pode executar comandos; não se preocupe, veremos mais assuntos relacionados à *shell* no decorrer deste curso):

```
$ gcc alomundo.c -o alomundo
```

O comando acima diz ao gcc para compilar o arquivo `alomundo.c` e gerar um executável chamado `alomundo`. A opção `-o` é usada para especificar o nome do arquivo de saída (*output*). Sem esta opção, o compilador gera por padrão um executável cujo nome é `a.out`.

3. Após compilar o seu código no passo dois, você pode executá-lo com o seguinte comando, também escrito em um terminal:

```
$ ./alomundo
```

Se tudo correr bem, você verá a frase `Alô Mundo!` aparecer no seu terminal...

Algumas das flags mais usadas do gcc estão listadas abaixo, mas lembre-se: **o manual é seu amigo!** Digite `man gcc` no seu terminal para entender o ditado. ;-)

`-o` arquivo: Especifica o nome do arquivo de saída. Se você não usar esta opção, o gcc criará um arquivo chamado `a.out` por padrão.

`-Wall`: Habilita todos os alertas mais comuns do compilador (*Warning all*). Isso pode ajudar a identificar problemas potenciais no seu código.

`-Wextra`: Habilita alertas extras que não são ativados pelo `-Wall`.

`-lm`: Vincula a biblioteca de matemática (`libm`), necessária se você precisar usar funções matemáticas como `sqrt` (raiz quadrada) no seu código.

É possível combinar várias *flags* de compilação em um único comando do gcc. Por exemplo, o comando a seguir compila o arquivo `codigo.c` com todos os avisos de alerta habilitado, inclusive alertas extras, inclui a biblioteca de matemática e gera um arquivo de saída chamado `programa`:

```
gcc -Wall -Wextra codigo.c -o programa -lm
```