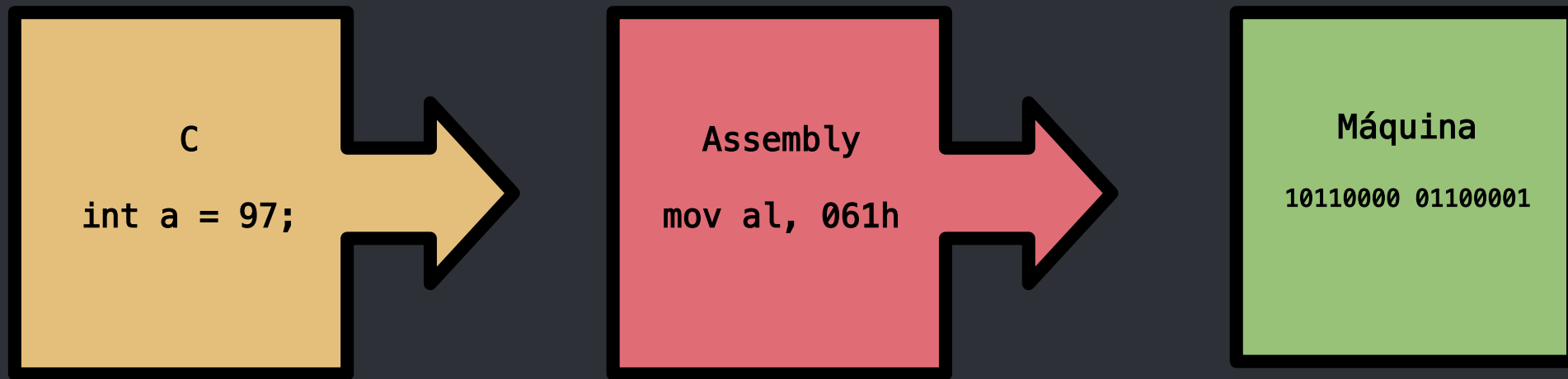


Como o C funciona?

Como o C funciona?

A **CPU** de um computador é incapaz de entender C. O conjunto limitado de instruções que uma CPU pode entender diretamente é chamado de código da máquina (ou idioma da máquina ou conjunto de instruções). Aqui está uma instrução de linguagem de máquina de amostra: `10110000 01100001`.

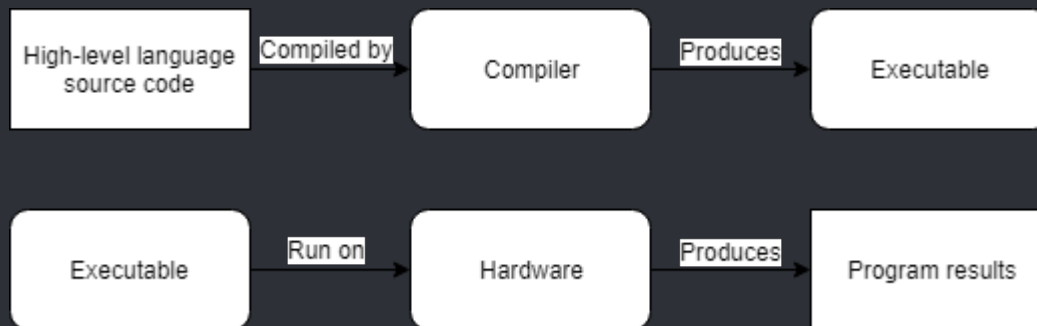
Em palavras finais podemos dizer que: o código que se escreve em C é transformado em *Assembly* e posteriormente transformado em linguagem de máquina:



Por isso o **C** é considerada uma linguagem de alto nível (longe do hardware em relação ao *Assembly*), já o *Assembly* de baixo nível porque ele transforma diretamente em linguagem de máquina.

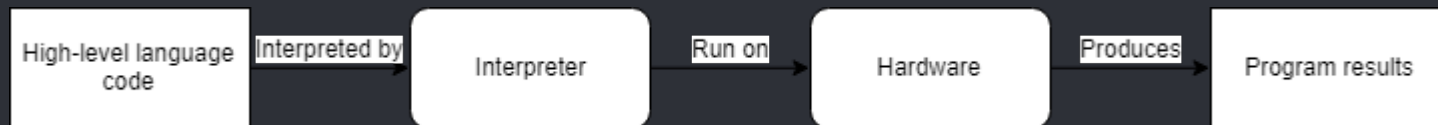
Compilação de Código

Esse processo é conhecido como: Compilação de Código . Aqui está uma representação simplificada do processo de *compilação*:



Algumas pessoas confundem **compiladores** com **interpretadores**!

Os interpretadores tendem a ser mais flexíveis que os compiladores, mas são menos eficientes ao executar programas, porque o processo de interpretação precisa ser realizado toda vez que o programa é executado. Isso significa que o intérprete é necessário toda vez que o programa é executado.



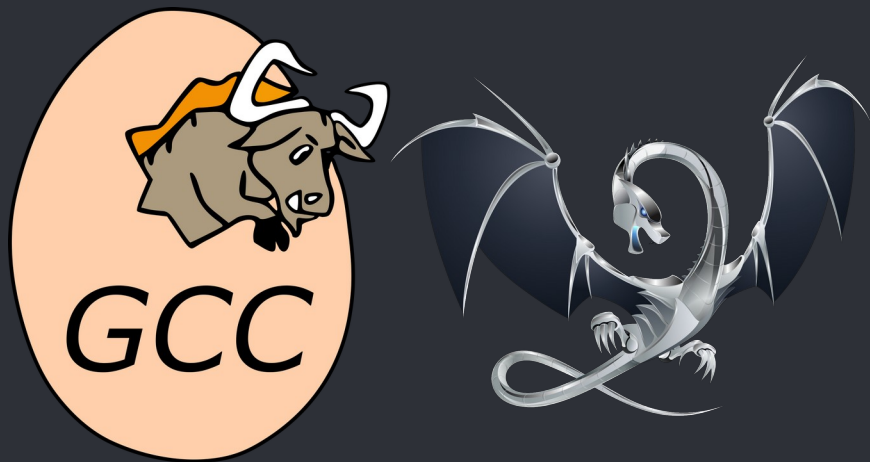
Interpretadores vs. Compiladores

- Exemplos de linguagens que usam compiladores: C, C++, C#, COBOL, Common Lisp, Delphi, Fortran, Java, Objective-C, ...
- Exemplos de linguagens que usam interpretadores: Bash, Perl, PHP, Python, JavaScript, Lua, Ruby, Haskell, ...

Se esse tipo de assunto lhe interessa, sugiro você clicar nesse link para saber mais sobre a comparação das vantagens de compiladores versus interpretadores pode ser encontrada em: <https://terminalroot.com.br/2021/10/diferencas-entre-compiladores-e-interpretadores.html> .

Compiladores

Existem diversos compiladores. Dentre eles os mais conhecidos são:



Para instalá-los utilize o gerenciador de pacotes da sua distribuição, exemplos:

```
sudo apt install gcc  
sudo apt install clang
```