

Introdução à Computação

o que é programação

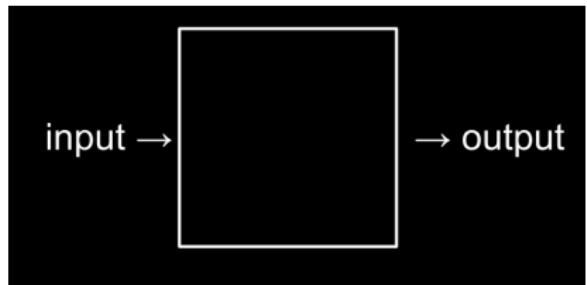
Alexandre Rademaker

o que é programação I

Programação é resolver
problemas!

Programas recebem entradas e
geram saídas!

Por que aprender a programar?
Entender o problema?



o que é programação II

Considere o problema de calcular raízes quadradas. Podemos definir a função de raiz quadrada como

$$\sqrt{x} = y \text{ such that } y > 0 \wedge y^2 = x$$

Uma função matemática perfeitamente legítima. Poderíamos usá-la para **reconhecer** se um número é a raiz quadrada de outro ou para derivar fatos sobre raízes quadradas em geral. Por outro lado, a definição não descreve um procedimento, não nos diz quase nada sobre como **encontrar** a raiz quadrada de um determinado número.

Matemática versus computação

Linguagens e Programação I

Um software é um conjunto de instruções para que o computador realize ações e tome decisões.

Existem diversas linguagens de programação, algumas entendidas diretamente pelos computadores, outras precisam ser 'traduzidas'.

As LPs são divididas em três tipos gerais:

- ▶ Linguagens de Máquina
- ▶ Linguagens Assembly
- ▶ Linguagens de Alto-nível

Linguagens e Programação II

Um computador entende diretamente sua própria **linguagem de máquina**, definida por seu projeto de hardware.

São constituídas por sequências de bits (1s e 0s), instruções associadas a operações elementares.

São dependentes de máquina: uma linguagem de máquina particular somente pode ser usada em um único tipo de computador.

São de difícil compreensão por humanos.

Linguagens e Programação III

as sequências de bits (entendidas diretamente pelo computador) são associadas à abreviações que representavam as operações elementares (mnemônicos).

Esse mnemônico formaram as bases das **linguagens assembly**.

O programa tradutor para converter os programas em linguagem assembly para programas em linguagem de máquina chama-se Assembler.

O código é claro para o ser humano, mas incompreensível para a máquina até ser traduzido.

Linguagens e Programação IV

Em assembly o programador precisa escrever muitas linhas de código para tarefas simples.

Linguagens de alto-nível aceleraram o processo de criação de softwares, onde instruções mais claras representam uma ou mais instruções para o computador.

Os compiladores traduzem programas em linguagem de alto-nível para programas em linguagem assembly e/ou diretamente para linguagem de máquina.

Compilação I

minimal.c

```
1 #include <stdio.h>
2
3 int main()
4 {
5     int x = 42;
6     return 0;
7 }
```

Compilação II

```
1 .file    "minimal.c"
2 .text
3 .p2align 4,,15
4 .globl   main
5 .type    main, @function
6 main:
7     push    ebp
8     mov     ebp, esp
9     and     esp, -16
10    sub    esp, 16
11    mov    DWORD PTR [esp+12], 42
12    mov    eax, 0
13    leave
14    ret
```

Compilação III

```
1 55          push  ebp
2 89 E5        mov   ebp,  esp
3 83 E4 F0      and   esp,  0xffffffff0
4 83 EC 10      sub   esp,  0x10
5 C7 44 24 0C 2A 00 00 00  mov  DWORD PTR [esp+0xc],0x2a
6 B8 00 00 00 00    mov  eax,0x0
7 C9          leave
8 C3          ret
```

Programas I

Programação Estruturada é uma metodologia de programação constituída por sequências, desvios e repetições de instruções de uma linguagem de programação.

Programas II

C e C++ são duas das linguagens mais [populares](#) para o desenvolvimento de software.

A linguagem C foi desenvolvida por Dennis Ritchie na Bell Laboratories. Vide [C Programming Language, Brian Kernighan and Lex Fridman](#).

O uso de C em vários tipos de computadores levou a muitas variações da linguagem.

Programas III

C99 é o último padrão ANSI para a linguagem C.

C++, uma extensão de C, foi desenvolvida por Bjarne Stroustrup no início dos anos 802, no Bell Laboratories.

C11, é o padrão atual da linguagem C++ (já existe o C14).

desenvolvimento em C I

Passos comuns utilizados na criação e execução de um programa em C (ou qualquer outra linguagem compilada).

Existem também as linguagens **interpretadas** como Python.

desenvolvimento em C II

fase 1: criando um programa

Esta fase consiste da edição de um arquivo com um programa editor de texto (notepad, vi, Emacs, etc). Importante, não processadores de texto como Word.

Você digita um programa C (programa fonte) usando o editor, faz as correções necessárias e salva o programa em um dispositivo de memória secundária, por exemplo, o HD.

Os nomes de arquivos dos programas fonte C terminam com a extenção `.c`.

desenvolvimento em C III

Uma IDE é um editor, que normalmente tem alguma interação com o compilador, que facilita o desenvolvimento de programadas.

IDE offline: [VS Code](#), [Emacs](#)

IDEs online: <http://replit.com>,
<https://github.com/features/codespaces> etc.

desenvolvimento em C IV

Fase 2: Pré-processando

Um programa pré-processador executado automaticamente antes que a fase de tradução do compilador inicie.

O pré-processador obedece a comandos chamados diretivas do pré-processador, que indicam que certas manipulações são realizadas no programa antes da compilação.

Estas manipulações usualmente juntam arquivos para serem compilados, e podem realizar substituições no texto.

Fase 3: Compilando um Programa C

O compilador traduz o código fonte (alto-nível) em um código de linguagem de máquina (código objeto).

desenvolvimento em C VI

Fase 4: Ligação (linking)

Tipicamente, um programa C contém referências para funções e dados definidos em outros arquivos, bibliotecas.

O código objeto produzido pelo compilador C contém referências a trechos definidos nas bibliotecas.

Um ligador (linker) liga o código objeto com o código das funções ausentes para produzir um programa executável.

desenvolvimento em C VII

Fase 5: Carga (loading)

Antes de um programa ser executado, ele deve ser primeiramente colocado na memória (primária).

Isto é feito pelo carregador (loader), que toma a imagem executável do disco e a transfere para a memória.

Bibliotecas podem ser também carregadas apenas neste momento.

Fase 6: Execução

Finalmente, o computador, sob o controle de sua CPU, executa o programa, uma instrução por vez.

A maioria das arquiteturas de computadores atuais podem executar várias instruções em paralelo.

desenvolvimento em C IX

