Laboratório de Algoritmos I

Aula 1 – Introdução

Thiago Silva Vilela

O que é um programa?

- Nós já sabemos o que é um **programa**, uma vez que usamos programas o tempo todo no nosso dia a dia.
  - o navegador é um programa;
  - o editor de texto que usamos é um programa;
  - os aplicativos de celular que usamos são programas.

- Um **programa** é montado utilizando **algoritmos** e uma **linguagem de programação**.

Algoritmos

- **Algoritmo**: conjunto finito de instruções
  - Usualmente, começa com a primeira instrução
  - Execução sequencial, uma instrução de cada vez, com possibilidade de saltos para outras instruções
  - Instruções individuais suficientemente elementares, ou primitivas
  - Sempre deve alcançar uma instrução PARE, para terminar a execução do algoritmo.
- Utiliza dados (**entrada**) e gera um resultado (**saída**).

Algoritmos

- Exemplo: Verificar se a soma de dois números é maior que 20.

1) Leia dois números;
2) Some o segundo número com o primeiro;
3) Se o resultado for maior que 20, escreva que a soma foi maior que 20 e vá para o passo 5.
4) Escreva que a soma não foi maior que 20.
5) PARE.

Algoritmos

- Idealmente, cada instrução em um algoritmo precisa ser:
  - Não ambígua;
  - Uma única operação bem definida;
  - Simples.

- Um algoritmo **não é** um programa de computador!
  - Algoritmo são somente os passos necessários para realizar uma tarefa.
  - Eles **são utilizados** para escrever programas de computador.

Linguagens de Programação

- Uma **linguagem de programação** é utilizada para escrever programas de computador.
- É uma forma de nos comunicarmos com o computador.
- Possuem uma sintaxe bem definida;
- Possuem vocabulário restrito;

Linguagens de Programação

- Todo computador tem uma linguagem de programação nativa, que é a linguagem que ele compreende.
  - Conhecida como código de máquina

- O código de máquina, no entanto, apresenta uma série de problemas. Um deles é que ele é expresso em código binário.

Linguagens de Programação

- Soma dois números:

0010 0001 0000 0100

0001 0001 0000 0101

0011 0001 0000 0110

0111 0000 0000 0001

0000 0000 0101 0011

1111 1111 1111 1110

0000 0000 0000 0000

Linguagens de Programação

- Felizmente, existem linguagens de programação de **alto nível**. Nessas linguagens escrevemos em uma linguagem mais próxima de humana e mais distante da linguagem de máquina.

- Nosso programa é então traduzido para a linguagem de máquina, para que possa ser entendido pelo computador.

- Esse processo de tradução é chamado de **compilação**.

Linguagens de Programação

- Soma dois números:

```
int main() {
   int a = 2;
   int b = 3;
   int c = a + b;
   printf("Soma = %d\n", c);
}
```

Linguagens de Programação

- Nesse curso usaremos a linguagem de programação C.
  - Possui tanto características de baixo nível quanto de alto nível;
  - Linguagem de propósito geral;
  - Largamente utilizada;

Primeiro programa em C

- Vejamos o seguinte programa:

```
#include <stdio.h>
/* Um Primeiro Programa */
int main (int argc, char *argv[]) {
    printf ("Ola! Eu estou vivo!\n");
    return 0;
}
```

Primeiro programa em C

- Vamos analisar passo a passo nosso primeiro programa.

```
#include <stdio.h>
```

- Essa linha diz ao compilador que ele deve incluir o arquivo-cabeçalho **stdio.h**.
  - Neste arquivo existem declarações de funções úteis para entrada e saída de dados (printf).
  - Toda vez que você quiser usar uma destas funções deve-se incluir este comando.
  - O C possui diversos Arquivos-cabeçalho.

Primeiro programa em C

```
/* Um Primeiro Programa */
```

- Quando fazemos um programa, uma boa idéia é usar comentários que ajudem a elucidar seu funcionamento.
- No caso acima temos um comentário.
- O compilador C desconsidera qualquer coisa que esteja começando com /* e terminando com */.
- Um comentário pode, inclusive, ter mais de uma linha.

Primeiro programa em C

```
int main (int argc, char *argv[])
```

- Essa linha indica que estamos definindo uma função de nome main;
- Todos os programas em C têm que ter uma função main, pois é esta função que será chamada quando o programa for executado;
- O conteúdo da função é delimitado por chaves { };
- O código que estiver dentro das chaves será executado seqüencialmente quando a função for chamada;
- A palavra **int** indica que esta função retorna um inteiro. O que significa este retorno será visto posteriormente;
- A última linha do programa, **return 0;** indica o número inteiro que está sendo retornado pela função, no caso o número 0.
- **int argc** e **char *argv[]** são os parâmetros da função. Falaremos sobre isso mais tarde.

Primeiro programa em C

```
printf ("Ola! Eu estou vivo!\n");
```

- A única coisa que o programa realmente faz é chamar a função **printf()**, passando a string (uma string é uma seqüência de caracteres, como veremos brevemente) "Ola! Eu estou vivo!\n" como argumento.
- É por causa do uso da função **printf()** pelo programa que devemos incluir o arquivo-cabeçalho **stdio.h**.
- A função **printf()** neste caso irá apenas colocar a string na tela do computador. O **\n** é uma constante chamada de constante barra invertida. No caso, o **\n** é a constante barra invertida de "new line" e ele é interpretado como um comando de mudança de linha.

Curiosidade – Constantes de Barra Invertida

- Segue uma lista com algumas constantes de barra invertida interessantes:

\n	Nova linha
\t	Tabulação horizontal
\”	Aspas
\'	Apóstrofo
\\	Barra invertida
\v	Tabulação vertical
\a	Sinal sonoro

Boas Práticas

- Um programa deve ser sempre o mais legível possível!
- Para isso existem algumas boas práticas de programação:
  - **Nunca** escreva um código sem indentação.
  - Use espaços/linhas em branco a seu favor.
  - Cuidado com mais de uma instrução por linha.