

Programação I

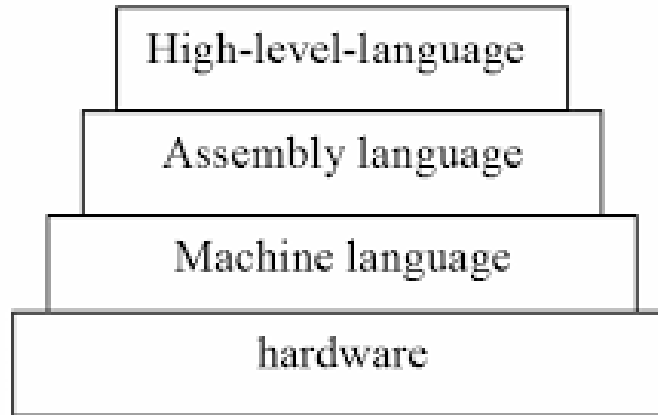
PRG29002

Engenharia de Telecomunicações 2ª Fase

Professor: Cleber Jorge Amaral

2016-1

Níveis de linguagens



+
Similaridade
com a linguagem
humana

-



Introdução ao C

- ▶ Um programa em C é composto por um conjunto de Funções.
- ▶ A função pela qual o programa começa a ser executado chama-se “main()”.
- ▶ Após cada comando em C deve-se colocar um ; (ponto-e-vírgula).
- ▶ É uma linguagem “tipada”, ou seja, os dados precisam ter tipos definidos
- ▶ Possui estruturas diversas de fluxo e controle como “if... else”, “Switch case”, etc.

Identificadores

- ▶ São os nomes que o programador dá a suas variáveis, constantes e funções
- ▶ Deve sempre iniciar com uma letra ou “_” (underscore)
- ▶ A partir do segundo caracter pode também conter números
- ▶ A linguagem não suporta caracteres especiais como letras acentuadas
- ▶ Identificadores não podem ser escritos com espaço, exemplo “buscarCodigo()”, não pode ser escrito como “buscar codigo()”
- ▶ A linguagem C é case-sensitive. Por exemplo, as variáveis “numero”, “Numero” e “NUMERO” são endereços diferentes
- ▶ Deve ter no máximo 31 caracteres (compatível com TurboC)

Boas práticas

- ▶ A indentação adequada facilita a compreensão do código
- ▶ O uso de nomes auto-explicativos facilita a compreensão e manutenção futura
- ▶ É comum variaar maiúsculas e minúsculas para facilitar a leitura como “QtMedidas”, “ValorMedio”

Preparando o ambiente

- ▶ Como precisaremos digitar alguns comandos, vamos utilizar o terminal do linux, abra portanto o terminal
- ▶ Por padrão o linux inicia na pasta do usuário, ficará algo assim: `aluno@sj-redes1-d1:~$`, daqui em diante utilizaremos o `~$` para simbolizar a pasta do usuário
- ▶ Execute os seguintes comandos para criar e entrar na pasta do usuário:
 - `~$ mkdir ExerciciosC`
 - `~$ cd ExerciciosC`
- ▶ Observe que agora o terminal exhibe algo como `aluno@sj-redes1-d1:~/ExerciciosC$`

Editando um arquivo

- ▶ O arquivo “.c” é o código-fonte de nosso projeto, é onde digitaremos o código na linguagem C
- ▶ Depois precisaremos compilar este código para transformá-lo em executável e finalmente poder rodá-lo, para compilar utilizaremos o compilador gcc do linux
- ▶ Para editar o arquivo “.c” podemos utilizar qualquer editor como o “gedit” do linux que é bem parecido com o “bloco de notas” do windows.
- ▶ Porém como teremos que entrar com linhas de comando para compilar, vamos utilizar um editor chamado pico que roda no terminal do linux

Passo-a-passo criando o OlaMundo.c

- ▶ 1) Crie o arquivo através do pico “nome do arquivo.c”
~/ExerciciosC\$ pico OlaMundo.c
- ▶ 2) Digite dentro do arquivo em branco criado o seguinte código:

```
#include <stdio.h>
main()
{
    printf("Olá Mundo!\n");
}
```

- ▶ 3) Compile o código (nenhuma linha impressa significa que não deu erro de compilação)
~/ExerciciosC\$ gcc OlaMundo.c -o OlaMundo

criando o OlaMundo.c (cont.)

- ▶ 4) Observe que na sua pasta agora tem dois arquivos um é o código fonte (arquivo .c) e o outro é o executável
~/ExerciciosC\$ ls -l
- ▶ 5) Execute o programa criado
~/ExerciciosC\$./OlaMundo
- ▶ 6) Como resultado você deve ver a mensagem na tela:
Olá Mundo!
- ▶ 7) O sistema volta para a linha de comando
~/ExerciciosC\$

Exercícios

- ▶ Crie novos arquivos e experimente escrever diversas mensagens de diferentes textos.
- ▶ Teste também o uso do “\n”

Variáveis

- ▶ Por uma questão de eficiência de uso de memória e processamento o C possui diversos tipos de variáveis, vamos agora trabalhar com alguns deles que servirão para praticamente todas as nossas necessidades
 - char: ocupa 1 byte na memória e varia de -127 a +127
 - int: ocupa 4 bytes e varia de -2.147.483.648 a +2.147.483.647
 - double: ocupa 8 bytes e possui dez dígitos de precisão
 - char[]: esta é o mesmo char descrito acima mas aqui simbolizando uma cadeia/vetor de caracteres (string)
- ▶ Apenas para conhecimento neste momento, há outros tipos como short, float e long double e os tipos que não são de precisão podem ainda ser signed ou unsigned

Exemplo de uso de variável

```
#include <stdio.h>
main()
{
    /* aqui começam as declarações de variáveis */
    int x; /* declaração de uma variável inteira */
    double y; /* declaração de uma variável real */

    /* aqui começam as instruções do programa principal */
    x=5; /* atribuindo o valor 5 (constante) a variável x */
    y=6.5;
}
```

Exercícios

- ▶ Imprima na tela os valores de sua variavel utilizando o comando printf

```
printf ("O valor de x é %d\n",x);
```

Obrigado pela atenção e participação!

Cleber Jorge Amaral (cleber.amaral@ifsc.edu.br)

Horários de atendimento (2016-1):
Quintas-feiras as 17:30 no laboratório de Programação

Sextas-feiras as 17:30 no Laboratório de Meios de
Transmissão