

Laboratório de Programação - Linguagem C

Prof. Cezar Macegoza

Histórico da Linguagem

- Em 1973 Dennis Ritchie reescreveu o sistema operacional UNIX em uma linguagem de alto nível (Ling. C).
- Surgem diversas implementações utilizando C, porém os códigos gerados eram incompatíveis.
- ANSI (American National Standard Institute), em 1983 padronizou a ling. C.



Características da Linguagem

- Portabilidade entre máquinas e sistemas operacionais.
- Alia características de linguagens de alto e baixo nível (liberdade ao programador).
- Programas Estruturados.
Total interação com o Sistema Operacional.
- Código compacto e rápido, quando comparado ao código de outras linguagem de complexidade análoga.

Características da Linguagem

- C é uma linguagem compilada: lê todo o código fonte e gera o código objeto (ling. de máquina) uma única vez.
- Linguagens Interpretadas: lê o código fonte, traduz e executa cada vez que o programa for executado.

Características da Linguagem

- Sempre que o código fonte for alterado ele deve ser novamente compilado.
- C é "case sensitive"
- Deve-se listar antecipadamente todas as variáveis utilizadas no programa.

Palavras Reservadas

- São palavras que têm um significado especial para a linguagem.
- auto, break, case, if, for, while, begin, end, continue, return, const,....
- C entende tais palavras apenas em letras minúsculas

Bibliotecas Utilizadas

- Conjunto de funções para realizar tarefas específicas.
- Biblioteca padrão C - ANSI - funções básicas.
- As primeiras linhas do programa indicam as bibliotecas utilizadas

`#include "minha_biblioteca.h" ou #include <minha_biblioteca.h>`

Primeiro Programa

```
/* Primeiro Programa em C */  
#include <stdio.h>  
int main()  
{  
printf("Meu primeiro programa em C\n");  
}
```


Primeiro Programa

```
/* Primeiro Programa em C */ comentários #  
include <stdio.h> /*biblioteca de E/S */  
main() /*função principal – inicio do programa*/ { /*marca início  
da função*/  
printf("Meu primeiro programa em C\n"); /*  
função para escrever na tela*/  
} /*marca fim da função*/
```

Como compilar

Utilizar um editor qualquer: vi ou vscode

Digite o programa anterior

Salve com o nome **p1.c**

Como compilar

- Para compilar e gerar executavel:

```
>> gcc -o p1 p1.c <ENTER>
```

- Executando:

```
>> ./p1 <ENTER>
```

Variável

- Variável: “objeto” que pode assumir diversos valores;
- Espaço de memória de um certo tipo de dado associado a um nome para referenciar seu conteúdo

```
Main ( ) {  
    int idade;  
    idade = 30;  
    printf (“ A idade é : %d”, idade);  
}
```

Nomes de variáveis

- quantos caracteres quiser (32);
- comece com letras ou sublinhado: Seguidos de letras, números ou sublinhados
- C é sensível ao caso:
peso <> Peso <> pEso
- não podemos definir um identificador com o mesmo nome que uma palavra chave: auto static extern int long if while do

Declaração de variáveis

- Instrução para reservar uma quantidade de memória para um certo tipo de dado, indicando o nome pelo qual a área será referenciada

>> tipo nome-da-variável; ou >> tipo nome1,
nome2,...,nomen EX: char nome;

int idade, num;

Tipos básicos de dados

tipo	bytes	escala
char	1	-128 a 127
int	2	-32.768 a 32.767
float	4	3.4e-38 a 3.4e+38
double	8	1.7e-308 a 1.7e+308

Programa Exemplo 2

```
#include <stdio.h>
int main( )
{
    int soma=10;
    float money=2.21;
    char letra= 'A';
    double pi=2.01E6;
    printf ("valor da soma = %d\n", soma);
    printf ("Valor de Money = %f\n", money);
    printf("Valor de Letra = %c\n", letra);
    printf("Valor de Pi = %e\n", pi);
    return 0;
}
```


Instruções de E/S - Scanf()

- Leitura de dados tipados via teclado
- Scanf (“string de controle”, lista de argumentos);

Exemplo: `scanf("%d",&idade);`

OBS: Para seqüência de caracteres (%s), o caracter & não deverá ser usado.

Comandos de E/S - printf()

- Apresentação de dados no monitor
- `printf("string de controle", lista de argumentos);`

Exemplo:

```
printf ("Digite a sua idade:\n"); scanf ("%d", &idade);  
printf("Sua idade é: %d", idade);
```

String de Controle

%c → character

%d → inteiro

%e → número ou notação científica

%f → ponto flutuante

%o → octal

%x → hexadecimal

%s → string (cadeia de caracteres)

%lf → double

Programa Exemplo 3

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

```
int main ( )  
{
```

```
char a ;  
printf ( "digite um character" );  
scanf ( "%c", &a );  
printf ( " \n %c = %d em decimal", a, a);  
printf ("%o em octal, %x em hexadecimal", a, a);  
  
}
```

Operador de Endereço &

- um endereço de memória é o nome que o computador usa para identificar uma variável
- toda variável ocupa uma área de memória e seu endereço é o primeiro byte por ela ocupado

Ex :

inteiro → 2 bytes float → 4 bytes char → 1 byte

Operador de Endereço &

Quando usamos & precedendo uma variável estamos falando do endereço desta variável na memória

```
Ex: Main ( ) {  
    int num;  
    num = 2;  
    printf ("valor = %d, endereço = %p", num, &num);  
}
```

Caracteres Especiais

\n nova linha

\r enter

\t tabulação (tab) \b retrocesso

\" aspas

\\ barra

Exercicio 1

Faça um programa para ler e escrever na tela o seu nome e a sua idade.

Exercicio 2

Faça um programa para ler e escrever na tela os seguintes dados:

- Nome
- Endereço
- Telefone
- Cidade
- Cep

Operadores Aritméticos

Operador	Ação
+	Adição
*	Multiplicação
/	Divisão
%	Resto da divisão inteira
-	Subtração (unário)
--	Decremento
++	Incremento

Operadores Relacionais e Lógicos

Operador	Ação
>	Maior que
>=	Maior ou igual que
<	Menor que
<=	Menor ou igual que
==	Igual a
!=	Diferente de
&&	Condição "E"
	Condição "OU"
!	Não

Operadores - Observação

Em C o resultado da **comparação** será **ZERO** se resultar em **FALSO** e **DIFERENTE DE ZERO** no caso de obtermos **VERDADEIRO** num teste qualquer.

Operadores - Exemplo

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

```
int main ()
```

```
{
```

```
    int verdadeiro, falso;
```

```
    verdadeiro = (15 < 20);
```

```
    falso  = (15 == 20);
```

```
    printf("Verd.= %d,Falso= %d",verdadeiro, falso);
```

```
    return 0;
```

```
}
```

Pré e Pós-incremento

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

```
int main()
```

```
{
```

```
int cont = 0, x;
```

```
x=++cont;
```

```
printf("X=%d, Cont=%d\n", x, cont);
```

```
x=cont++;
```

```
printf("X=%d, Cont=%d\n", x, cont);
```

```
}
```

Operador Sizeof

- Este operador retorna o tamanho da variável ou tipo que está em seu operando.
- Por exemplo “sizeof(char)” resultaria em 1.

Operador Sizeof

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

```
int main()
```

```
{
```

```
int cont = 0;
```

```
printf("Verd.= %lu",sizeof(cont));
```

```
}
```