UNIVERSIDADE DO ESTADO DE MINAS GERAIS

Docente: Rildo Afonso de Almeida

Microprocessadores & Microcontroladores

UEMG

5 - Programação C

5.5 - Variáveis

- São espaços de **memória** reservados que guardam valores durante a execução de um programa;

- **Todas** as variáveis em C devem ser **declaradas**, antes de serem **usadas**;

5 - Programação C

5.5 - Variáveis

➢Uma declaração de variável em C consiste no nome de um **tipo**, seguido do **nome** da variável, seguido de **ponto-e-vírgula**.

Ex: **tipo_da_variavel** lista_de_variaveis;

int num;

Tipo da Variável

Nome da Variável

UEMG

5 - Programação C

5.5 - Variáveis

O tipo de uma variável informa a quantidade de memória, em bytes, que a variável ocupará e a forma como um valor deverá ser armazenado.

5 - Programação C

5.5.1 – Tipos de Variáveis

Há cinco tipos básicos de dados em C:

Tipo	Bit	Bytes	Escalas
char	8	1	-128 a 127
int	16	2	-32768 a 32767
float	32	4	3.4E-38 a 3.4E+38
double	64	8	1.7E-308 a 1.7E+308
void	0	0	Nenhum valor

5 - Programação C

5.5.1 – Tipos de Variáveis

Exceto o void, os tipos de dados básicos podem ter vários modificadores precedendo-os;

signed;

unsigned;

long;

short.

5 - Programação C

5.5.1 – Tipos de Variáveis

Um modificador é usado para alterar o significado de um tipo básico para adaptá-lo mais precisamente às necessidades de diversas situações, veja:

Tipo	Bits	Início	Fim
int	16	-32.768	32.768
unsigned int	16	0	65.535
signed int	16	-32.768	32.768
short int	16	-32.768	32.768
long int	32	-2.147.483.648	2.147.483.648

5 - Programação C

5.5.2 – Nome da Variável

✓ O nome de uma variável pode ser de uma letra até palavras com no máximo 32 caracteres;

✓ Obrigatoriamente deve começar com uma letra ou undercore (“_”). O restante pode ser letras de A a Z, maiúsculas, minúsculas, números e o underscore;

5 - Programação C

5.5.2 – Nome da Variável

Ex: a; num; essa_e_uma_variavel; tambem_essa;

Cuidados:

- O nome de uma variável não pode ser igual a uma palavra reservada;
- O nome de uma variável não pode ser igual a de uma função declarada pelo programador ou pelas bibliotecas do C.

UEMG

5 - Programação C

5.5.2 – Nome da Variável

Eis algumas palavras reservadas da linguagem C:

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	Short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
asm	pascal	far	huge
interrupt	near	_cs	_cs

5 - Programação C

5.5.2 – Nome da Variável

Em C, letras maiúsculas e minúsculas são tratadas diferentemente.

```
int variavel;
int Variavel;
int VaRiAVeL;
int VARIAVEL;
```

Ou

```
int variavel, Variavel, VaRiAVeL, VARIAVEL;
```

5 - Programação C

5.5.3 – Exemplo de Variáveis

```
#include <stdio.h>
/* Exemplo da variável Char */
int main()
{
char Ch;
Ch='D';
printf("%c", Ch);
return 0;
}
```

%c indica que printf() deve colocar um caracter na tela.

stdio.h – Cabeçalho Padrão Entrada e Saída

5 - Programação C

5.5.3 – Exemplo de Variáveis

```
#include <stdio.h>
/* Exemplo da variável Inteiro */
int main()
{
int num;
num= 10;
printf("%d", num);
return 0;
}
```

%d indica que printf() deve colocar um inteiro na tela.

5 - Programação C

5.5.3 – Exemplo de Variáveis

```
#include <stdio.h>
/* Exemplo da variável String*/
int main()
{
        char nome[20];
        printf("Digite seu nome:");
        gets (nome);
        printf("\n\nSeu nome é: %s" , nome);
        return 0;
}
```

Função para leitura de String (Char).

UEMG

5 - Programação C

5.5.4 – A função printf()

A função printf() tem a seguinte forma geral:

printf(string_de_controle, lista_de_argumentos)

Teremos, na string de controle, uma descrição de tudo que a função vai colocar na tela.

5 - Programação C

5.5.4 – A função printf()

Isto é feito usando-se os códigos de controle, veja alguns exemplos:

Código	Significado
%d	Inteiro
%f	Float
%c	Caractere
%s	String
%%	Coloca um % na tela

5 - Programação C

5.5.5 – Variáveis Locais

- São variáveis declaradas dentro de uma função;
- Só podem ser referenciadas por comandos que estão dentro do bloco no qual as variáveis foram declaradas;

```
#include <stdio.h>
int main()
{
    int x;
    int y;
    x = 10;
    y = 20;
}
```

Início do Bloco

Fim do Bloco

5 - Programação C

5.5.5 – Variáveis Locais

- As variáveis também podem ser declaradas dentro de qualquer outro bloco de código;

5 - Programação C

5.5.5 – Variáveis Locais

```
#include <stdio.h>
int main()
{
    int x;
    scanf("%d", &x);
    if(x = = 1)
    {
    char s[30];
    printf("Entre com o nome");
    gets(s);
    /* Faz alguma coisa*/
    }
}
```

Criação da Variável

Morte da Variável

UEMG

5 - Programação C

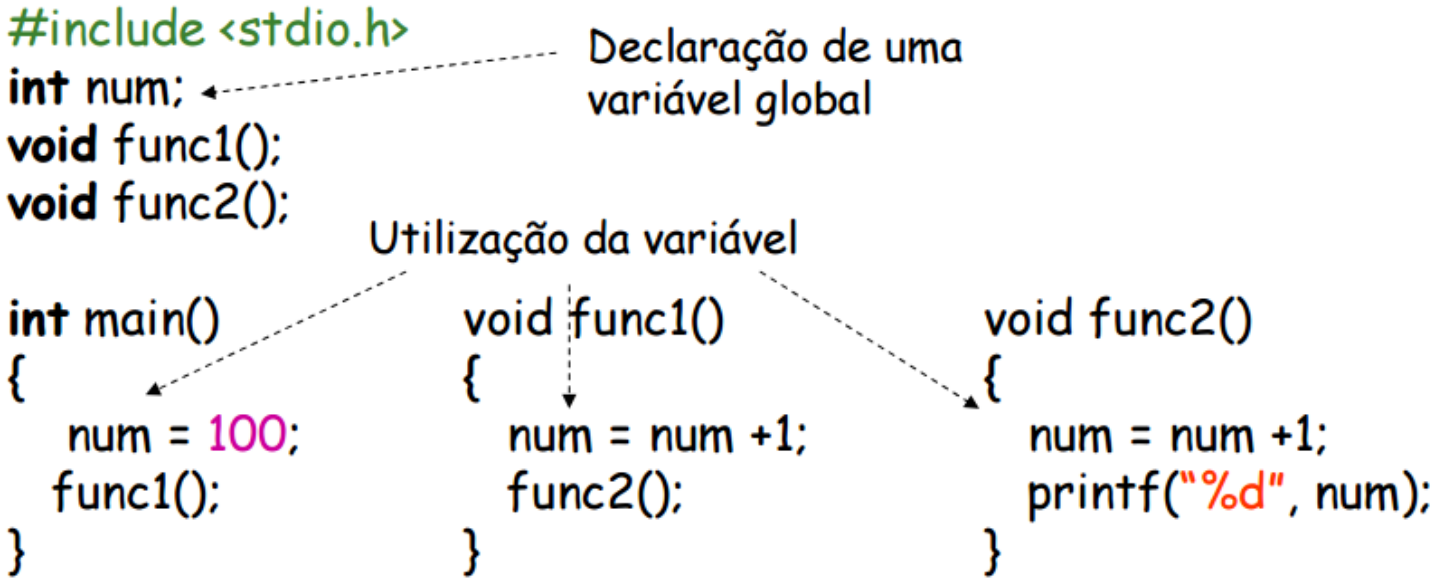
5.5.6 – Variáveis Globais

- São variáveis reconhecidas pelo programa inteiro e podem ser usadas por qualquer bloco de código;

5 - Programação C

5.5.6 – Variáveis Globais

- As variáveis globais são criadas declarando-as fora de qualquer função. Veja:



5 - Programação C

5.5.6 – Variáveis Globais

- As variáveis globais encontram-se armazenadas em uma região fixa da memória, separada para esse propósito pelo compilador C;

- Variáveis globais são úteis quando o mesmo dado é usado em muitas funções em seu programa;

5 - Programação C

5.5.6 – Variáveis Globais

- Alerta: Variáveis globais ocupam memória durante todo o tempo em que seu programa estiver executando, portanto, evite usar variáveis globais desnecessárias.

5 - Programação C

5.5.7 – Constantes

- Variáveis com o modificador const não podem ser modificadas por seu programa;

```
#include <stdio.h>
int  main()
{
const  int  num = 100;
}
```

Cria uma variável inteira chamada **num**, com um valor inicial 100, que seu programa não pode modificar.

5 - Programação C

5.5.8 – Constantes Hexadecimais e Octais

- Muitas vezes precisamos inserir constantes hexadecimais (base dezesseis) ou octais (base oito) no nosso programa.

5 - Programação C

5.5.8 – Constantes Hexadecimais e Octais

- Em C as constantes hexadecimais começam com **0x** e as constantes octais começam em **0**.

```
int varHex = 0x80;
int varOct = 012;
```

128 em decimal

10 em decimal

UEMG

Exercícios

1) Quais são os tipos básicos de dados na linguagem C padrão?

2) Qual é a diferença entre uma variável **signed** e outra **unsigned**?

3) Qual a Magnitude de armazenamento do tipo **long int** ?

UEMG

Exercícios

4) No programa seguinte, qual o escopo (global/local) de cada uma das variáveis?

```
int teste;
int funcao1 (int a, int b)
{
return a+b
}
main() {
int valor;
valor = 5;
teste = funcao1 (valor, 10);
}
```

UEMG

Exercícios

5) Qual o resultado armazenado na variável “y” no programa seguinte?

```
int x, y;
x = 5;
y = (x==5) * 2;
```

6) Indique o resultado armazenado na variável “x” do programa seguinte:

```
int x, y;
long int z;
z = 0x1234;
y = z+1;
x = y + 1;
```

UEMG

Exercícios

7) No programa seguinte, qual o valor armazenado em “z”?

```
int x;
long y;
float z;
x = 153;
y = 420;
z = y / x;
```

UEMG

Bibliografia Básica

MIYADAIRA, A. N. *Microcontroladores PIC18: aprenda e programe em Linguagem C* Ed. Érica, 1ª Ed., 2009, São Paulo.

LUZ, C. E. S. *Programando Microcontroladores PIC em Linguagem C com base no PIC4520.* Ed. Ensino Profissional, 2011, São Paulo.

SILVA, R. A. *Programando Microcontroladores PIC.* Ed. Ensino Profissional, 2011, São Paulo.

PEREIRA, F. *Microcontroladores PIC – Programação em C.* Érica: São Paulo, 2003.

UEMG

Bibliografia Complementar

TAUB, H.. Circuitos Digitais e Microprocessadores. McGraw Hill do Brasil, 1984.

ZILLER, Roberto M. *Microprocessadores: Conceitos Importantes*. Edição do Autor, 2000.

DALTRINI, Beatriz M., JINO, M., MAGALHÃES, L. P.. *Introdução à Computação Digital*. Makron Books, 1999.