

Linguagem C

Linguagem C

- Surgiu nos anos 70 criada por Dennis Ritchie.
- É uma linguagem estruturada e padronizada (ANSI C).
- Facilita a portabilidade.
- Oferece ao programador **poucas restrições**, facilidade de **acesso ao hardware**.
- Geralmente o código gerado pelo compilador é tão **rápido** e **eficiente** em consumo de memória quanto programação direta em *assembly*.

Declaração de variáveis

- Variáveis devem ser declaradas **antes** de serem usadas.
- Assim permitindo que o compilador saiba de antemão informações como **tipo** e **espaço gasto em memória** podendo fazer verificação durante o processo de compilação.

Funções

- Conjunto de comandos que executa uma determinada operação no programa.
- Normalmente é útil quando se necessita **repetir um mesmo trecho de código** várias vezes em um programa.

Cuidado na escrita

Case Sensitive: Significa que na linguagem C as letras **maiúsculas** e **minúsculas** são tratadas como caracteres diferenciados.

- Ex.: Podemos declarar uma variável **For**, apesar de haver uma palavra reservada **for**, mas isto não é uma coisa recomendável de se fazer pois pode gerar confusão!!

Comentários

- ***/* comentário de uma ou várias linhas.*** O padrão não permite comentários aninhados (um dentro do outro), mas alguns compiladores os aceitam.

****/***

- ***// comentário de uma linha***

Palavras reservadas

<i>auto</i>	<i>else</i>	<i>register</i>	<i>union</i>
<i>break</i>	<i>enum</i>	<i>return</i>	<i>unsigned</i>
<i>case</i>	<i>extern</i>	<i>short</i>	<i>void</i>
<i>char</i>	<i>float</i>	<i>signed</i>	<i>volatile</i>
<i>const</i>	<i>for</i>	<i>sizeof</i>	<i>while</i>
<i>continue</i>	<i>goto</i>	<i>static</i>	
<i>default</i>	<i>if</i>	<i>struct</i>	
<i>do</i>	<i>int</i>	<i>switch</i>	
<i>double</i>	<i>long</i>	<i>typedef</i>	

ESTRUTURA BÁSICA DE UM PROGRAMA

- 1) INCLUSÃO DE ARQUIVOS EXTERNOS (.h)***
- 2) DECLARAÇÃO DE VARIÁVEIS GLOBAIS***
- 3) DECLARAÇÃO DE PROTÓTIPOS DE FUNÇÕES***
- 4) FUNÇÕES***
- 5) FUNÇÃO PRINCIPAL (main)***

Protótipos de Funções

- Protótipos são **declarações de funções**. Isto é, você declara uma função que irá usar.
- O compilador toma então conhecimento do **formato** daquela função antes de compilá-la.
- O código correto será então gerado. Um protótipo tem o seguinte formato:
tipo_de_retorno nome_da_função (declaração_de_parâmetros);

Protótipos de Funções

- *tipo_de_retorno nome_da_função (declaração_de_parâmetros);*
- onde o tipo_de_retorno, o nome_da_função e a declaração_de_parâmetros são os mesmos que você pretende usar quando realmente escrever a função. Repare que os protótipos têm uma nítida semelhança com as declarações de variáveis.

Exemplo de programa em C

```
#include <stdio.h>
int Square (int a);
int main ()
{
    int num;
    num=Square(num);
    return 0;
}
int Square (int a)
{
    return (a*a);
}
```

Tipos de Dados

- **char** (sinalizados –positivo ou negativo)
ou não sinalizado – positivo)
- **int** (sinalizados – positivo ou negativo)
ou não sinalizado – positivo)
- **float/double** (valores inteiros ou fracionários)
- **void** (tipo vazio, ou um "tipo sem tipo". Não retorna e/ou recebe valor – ex. Usado em uma função sem parâmetros)

Modificadores de tipos de dados

- **short** (reduz a faixa de representação do tipo)
- **long** (amplia a faixa de representação do tipo)
- **signed** (para especificar a representação de valores sinalizados – compl. de dois)
- **unsigned** (para especificar a representação de valores não sinalizados)

Modificadores de acesso

- **static** (indica que a variável deve ter endereço fixo na memória). É similar a uma variável global mas que pode ter seu acesso restrito.
- **volatile** (indica que a variável pode ter seu valor modificado repentinamente).

Exercício

- Completar a tabela dos tipos de dados do Microcontrolador AVR Atmega2560.
- Utilize o Atmel Studio.

TIPOS DE DADOS

ATmega2560

TIPO	FAIXA DE VALORES	TAMANHO EM BYTES
signed char		
char		
int		
unsigned int		
unsigned long int		
long int		
float		
double		
long double		

Exercício

- Repita o exercício anterior para o microcontrolador AVR ATmega2560, com os tipos listados abaixo e que estão definidos no arquivo “stdint.h” (inclua-o no projeto):
 - uint8_t, int8_t, uint16_t, int16_t, uint32_t e int32_t

Exercício

- Completar a tabela dos tipos de dados do Microcontrolador HCS08QE128.
- Utilize o CodeWarrior.

TIPOS DE DADOS

HCS08QE128

TIPO	FAIXA DE VALORES	TAMANHO EM BYTES
signed char		
char		
int		
unsigned int		
unsigned long int		
long int		
float		
double		
long double		