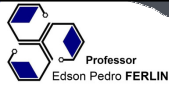


Tipos de Dados Avançados

Prof. Edson Pedro Ferlin

Modificadores de Tipo

- Os modificadores de tipo fazem com que tenhamos outros tipos de dados além dos cinco básicos;
- São divididos em quatro categorias:
 - de Acesso;
 - de Armazenamento;
 - de Tipo de Função.
- Um modificador de tipo precede o tipo base que ele modifica no comando de declaração.
- Forma Geral: **Modificador Tipo_base Lista_de_variáveis;**

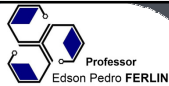


Modificadores de Acesso

Const

- Faz com que durante a execução, o programa não possa alterar uma variável declarada com este modificador, exceto pelo fato de que podemos dar à variável um valor inicial;
- Exemplo:

```
const float versao=3.20;
```
- Evitam que os argumentos de uma função sejam modificados por aquela função.



Modificadores de Acesso

Volatile

- Indica que o valor de uma variável pode ser alterado sem uma especificação explícita do programa;
- Dessa forma, será examinado o valor da variável cada vez que for feita uma referência a ela;
- Exemplo:

```
int timer;  
volatile int clock;  
  
timer = clock;  
printf("Tempo decorrido é %d", clock-timer);
```

Modificadores Armazenamento Auto

- Usada para declarar variáveis locais;
- Por *default* as variáveis locais são Auto;

Modificadores Armazenamento Extern

- Diz ao compilador que os tipos e nomes das variáveis que se seguem já foram declarados em um outro lugar;
- Deixa o compilador saber quais os tipos e nomes dessas variáveis globais sem realmente criar armazenamento para elas novamente;

```
int x, y;
char ch;
main()
{
    ...
}
func1()
{
    x = 123;
}
```

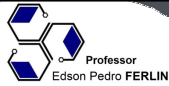
```
extern int x, y;
extern char ch;
func2()
{
    x = y/10;
}
func3()
{
    y = 10;
}
```

Modificadores Armazenamento Static

- São variáveis permanentes dentro de suas próprias funções ou arquivos;
- São diferentes das variáveis globais, porque elas não são conhecidas fora de suas funções ou arquivos, mas mantêm seus valores entre as chamadas;
 - Locais: É uma variável que retém seu valor entre as chamadas das funções;
 - Globais: Possibilidade de criar funções gerais de modo a incluir em bibliotecas para uso posterior.

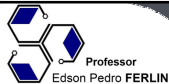
Modificadores Armazenamento Register

- Aplica-se apenas a variáveis do tipo int e char;
- Exige que o compilador mantenha o valor das variáveis declaradas com esse modificador no registrador da CPU, e não na memória, onde as variáveis normais são armazenadas;
- Mais rapidez no processamento;
- Ideal no controle de laços.



Modificadores Tipo de Função Pascal

- Informa ao compilador para usar uma convenção de passagem de parâmetros do tipo Pascal para os argumentos da função em lugar do método normal do C;
- Escrever funções em C que serão usadas por outros compiladores;
- Usar rotinas de biblioteca de um compilador Pascal declarando-as no início de seu programa em C como sendo do tipo Pascal.



Modificadores Tipo de Função Cdecl

- Informa ao compilador que ele deve compilar uma função de forma que seus parâmetros sejam passados de uma maneira compatível com outras funções de C;
- Usa-se quando o compilador está ajustado para compilar usando a convenção de chamada Pascal.

Modificadores Tipo de Função

Interrupt

- Informa ao compilador que a função que ele modifica será usado como um controlador de interrupção;
- Preserve todos os registradores da CPU;
- Saia da função mediante um IRET (Retorno de Interrupção).

Conversão de Tipo

Atribuição

- Ocorre quando se mistura tipos de dados diferentes;
- Regra: converte o valor do lado direito (expressão) do comando para o tipo do lado esquerdo (objeto);

```
int x
char ch
float f;
func()
{
    ch = x;    /* 8 bits inferiores */
    x = f;     /* parte não fracionária */
    f = ch;    /* para ponto flutuante */
    f = x;     /* para ponto flutuante */
}
```

Quando se converte de um tipo maior para outro menor, perde-se os bits superiores.

Contato



eferlin@live.com



(BLOG) professorferlin.blogspot.com

(SITE) professorferlin.webnode.com.br

(YOUTUBE) ProfEdsonPedroFerlin