ESTRUTURA DE DADOS I

Diretivas de compilação

Profª Amanda Tameirão

Objetivos da aula de hoje

- Compreender o que são diretivas;
- Aplicar diretivas.


endif
ifdef
Diretivas
ifndef
Compilação
include
define
undef
else
elif

Conceito

Instruções para o pré-processador.

Pré-processador é um programa que examina e controla o arquivo fonte antes da compilação, baseando-se nas diretivas.

Iniciadas com #, terminam na mesma linha, a não ser que o último caractere seja \, neste caso ela terminará na linha seguinte.

#include

include

Inclui um arquivo em um determinado ponto

Normalmente utilizado com arquivos .h (cabeçalho)

Geralmente contém:

- Definições de tipos
- Declarações globais
- Definições de constantes e macros
- Protótipos de funções

include

Possui duas formas de utilização:

#include <arquivo> ou #include “arquivo”

A diferença é:

#include <arquivo> procura o arquivo nos diretórios do sistema. Exemplo: #include <stdio.h>

#include “arquivo” procura o arquivo nos diretórios do usuário, a partir do atual até 10 níveis anteriores. Exemplo: #include “pessoal.h”

Bibliotecas

Bibliotecas

Bibliotecas são criadas para que as funções sejam separadas, e, sempre que precisar, serem disponibilizadas em qualquer código.

Geralmente concentradas por tipos:

math.h
string.h
stdio.h
etc

Bibliotecas

Principal **.c ou .cpp**	**Cabeçalho** **.h**	**Funções** **.c ou .cpp**
Função principal	Possui os protótipos das funções Declarações globais (structs, typedef, etc.) Macros Constantes	Funções que deseja disponibilizar

Como criar os arquivos

No Eclipse	No Dev
Após criar um projeto Clique em NEW – Header File Coloque o nome**.h** (não esqueça da extensão) Clique em NEW – Source File Coloque o nome**.c** (não esqueça da extensão)	Crie um projeto Clique em New File Renomeie o arquivo e altere a extensão para **.h** Clique em New File Renomeie o arquivo e altere a extensão para **.c**

#define

#undef

define

Define uma constante ou macro (que atua como uma *string* de substituição).

Toda vez que o compilador encontrar o nome da constante/macro deverá substituir pelo valor ou sequencia fornecida.

Por convenção as nomes das constantes/macros são escritos com letras maiúsculas.

Aplicação do define

```
#include <stdio.h>

#define PI 3.1416
#define VERSAO "2.02"

int main() {
    printf("\nPrograma versão %s", VERSAO);
    printf("\nO número PI vale %.5f", PI);
    return 0;
}
```

#define – Uma outra opção é definir as macros com argumentos.

```
#include <stdio.h>
#define MAX(a,b) ((a > b) ? (a) : (b))
#define MIN(a,b) ((a <= b) ? (a) : (b))
int main() {
    int i = 7, j = 12;
     printf("\nMaior valor %d", MAX(i,j));
    printf("\nMenor valor %d", MIN(i,j));
    return 0;
}
```

#define

Qual o motivo do excesso de parênteses?

```
#include <stdio.h>
#define SQR(x) x*x
```

Ao executar SQR(a+b) o pré-processador o converterá em SQR(a+b*a+b) e consequentemente o cálculo estará incorreto pois ele multiplicará b*a e depois somará a e b ao resultado da multiplicação.

Criação ideal #define SQR(x) (x)*(x)

#undef – Desativa uma macro utilizada, ou seja, ela deixará de existir.

```
#include <stdio.h>
#define SQR(x) (x)*(x)
int main() {
   int i = 7, j = 12;
   printf("\nO valor é %d", SQR(i+j));
   #undef SQR
   //A partir deste ponto SQR não poderá ser utilizada
   return 0;
}
```

EDI

Aula de hoje

Declaração de diretivas de compilação

EDI

Próxima aula

Exercício aplicando diretivas de compilação