

INE 5416/5636 - Paradigmas de programação

Turmas 04208/08238

Prof. Dr. João Dovicchi – dovicchi@inf.ufsc.br

<http://www.inf.ufsc.br/~dovicchi>

Módulos

Em C os módulos contêm declarações de funções, constantes, tipos etc. que podem ser utilizadas por outros programas ou funções.

Módulos

Em C os módulos contêm declarações de funções, constantes, tipos etc. que podem ser utilizadas por outros programas ou funções.

Vantagens:

- △ Facilitam a manutenção do código
- △ Permitem a reutilização de código

Conteúdo dos Módulos:

- declarações de tipos `struct` e `union`
- protótipos de funções
- declarações de constantes
- seleções de dependências
- definições e algoritmos de funções

Módulos .c e .h

Em C, os módulos são cabeçalhos (headers) com extensão “.h” e módulos de programas (.c) onde se define algoritmos e procedimentos de funções.

Nos arquivos de cabeçalho são declarados tipos, prototipadas as funções e programadas as seleções de dependências ou de portabilidade.

Nos arquivos de programas são definidas as variáveis, algoritmos e procedimentos de funções.

Exemplo de cabeçalho (.h)

```
/* local.h */
#include <stdio.h>
#include <math.h>

#ifndef PI
#define PI 3.14159;
#endif

#define REAL double;

int stdopt(char c, char *h);
int getparam(char *s, REAL *v, int msk);

etc...
```

Exemplo de função (.c)

```
/* soma.c */  
#include <stdio.h>  
#include "local.h"  
  
int soma(int a, int b){  
    int s;  
    s = a + b;  
    return s;  
}
```

Estrutura do programa

local.h

outros.h

func1.c \longleftrightarrow main.c

...

funcN.c

Programa principal: main.c

```
/* main.c
 * Progr. fatorial
 */
#include <stdlib.h>
#include <stdio.h>
#include "local.h" /* defs locais */

int main(){
    int i;
    double p;

    i = fat(4); /* chama a funcao fat */
    p = PI;

    printf("Fatorial de 4 = %d\n", i);
    printf("PI = %lf\n", p);
    return 0;
}
```

Definições locais: local.h

```
/* local.h
 * Define funcoes matematicas
 */

#ifndef __LOCAL_H
#define __LOCAL_H
#endif

#ifndef PI
const double PI = 3.14159265358979323844;
#endif

int fat (int); // prototipo da funcao 'fat'
```

Função fatorial: fat.c

```
/* fat.c
 * Define a funcao fatorial
 */

int fat (int n) {
    if (n==0) {
        return 1;
    } else {
        return (n*fat (n-1));
    }
}
```

Compilação

O Makefile:

```
CC = gcc
CFLAGS = -O2 -ansi -Wall
LIBDIR = -L/usr/local/lib
LIBS = -lm

INC =-I./ -I/usr/local/include

PROGS= fatorial

all: $(PROGS)

fatorial: fat.c main.c
$(CC) $(CFLAGS) $(INC) $(LIBDIR) -o $@ $? $(LIBS)

clean:
rm -f *~ *core *.o *.dat $(PROGS)
```

Roteiros: Prática 07