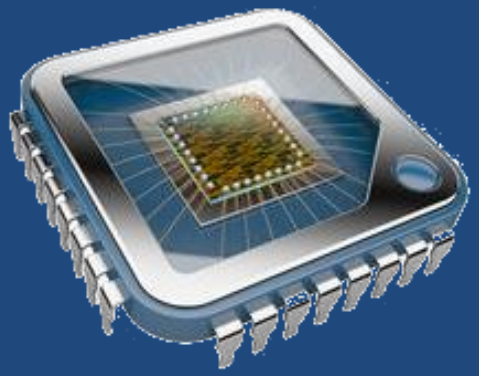
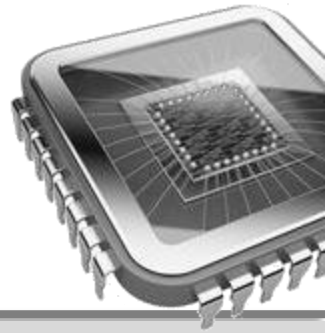


Introdução à Programação
Aula 16

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação

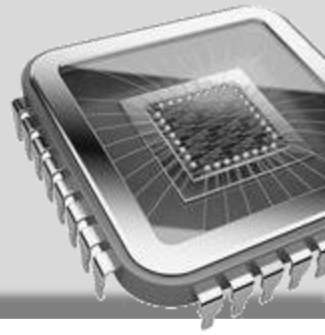
GESE
Grupo de Engenharia de Sistemas Embarcados

CEComp
Engenharia da Computação
UNIVASF



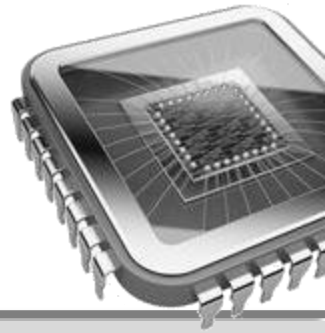
COMO UM PROGRAMA C É COMPILADO?

Compilação de um Programa



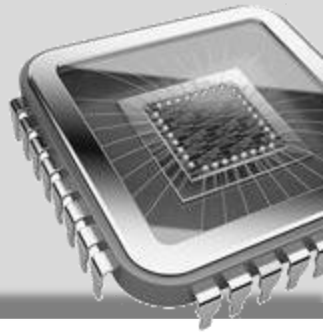
- A compilação de um programa em C é feito resumidamente em duas etapas:
  - **Compilação**: Traduzir cada .c em um objeto .o
  - **Linking**: Unir todo os objetos em um programa executável.

Primeiro a compilação ocorre em partes, um arquivo por vez, evitando saturar a memória e o tempo de otimização. Depois só precisa recompilar os *sources* que você alterar. Dessa forma ele ganha tempo.



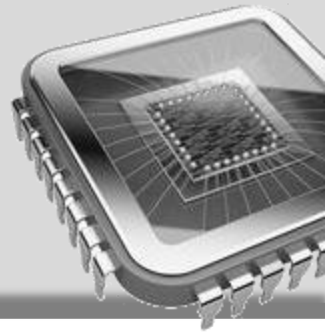
QUAL A IMPORTÂNCIA DE USAR BIBLIOTECAS?

Bibliotecas em C



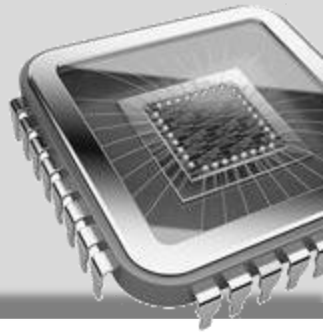
- Já percebeu que todos os programas em linguagem C usam as bibliotecas básicas da linguagem.
  - stdio.h, stdlib.h, math.h, string.h etc.
  - Cada biblioteca tem um conjunto de funções que são descritas em um arquivo-cabeçalho (*header-file*) .h, que tem o mesmo nome da biblioteca.
  - Essas bibliotecas também podem ser chamadas de API (*application programming interface*).

Bibliotecas em C

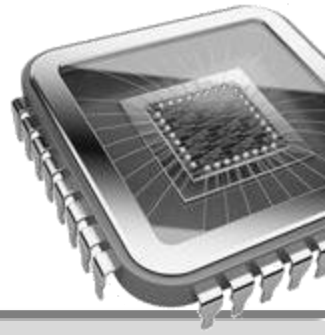


- Para ter acesso a uma biblioteca, o programa deve incluir uma cópia do arquivo *header* (.h), com a diretiva #include, para que o pré-processador inclua uma cópia no programas.
  - Se o arquivo *header* está localizado na pasta include do compilador basta usar os símbolos <>.
  - Quando o arquivo *header* está fora da pasta include do compilador, mas no mesmo diretório do código fonte deve usar aspas duplas no include.
  - Caso o arquivo esteja em outra pasta, é preciso informar o caminho correto entre aspas duplas.

Arquivos header
Benefícios

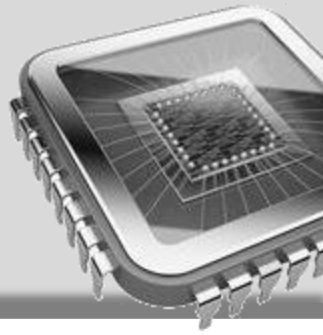


- **Reutilização de código**: Fica mais fácil depois, em um projeto, reutilizar funções pré-definidas e compiladas, não é preciso reimplementá-las.
- **Organização**: Omitindo a implementação, apenas a lógica utilizada no arquivo principal é vista pelo programador.
- **Omissão da implementação**: Com isso, o usuário limita-se à interface da biblioteca.



COMO CRIAR UM ARQUIVO DE HEADER?

Arquivo header (protótipo) calculadora.h



```
#ifndef CALCULADORA_H
#define CALCULADORA_H
```

Guarda de cabeçalho: evita inclusão cíclica

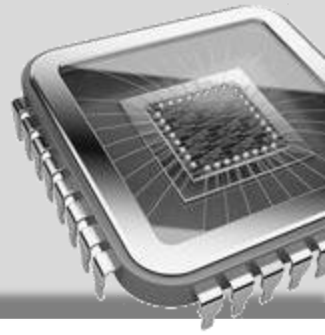
Inclusão de interfaces, declaração de variáveis, definição de macros...

```
float soma(float x, float y);
float subtracao(float x, float y);
float multiplicacao(float x, float y);
float divisao(float x, float y);

#endif
```

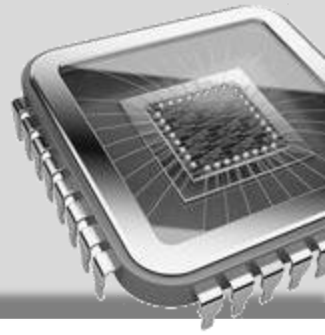
Declaração dos protótipos das funções.

Arquivo header (implementação) calculadora.c



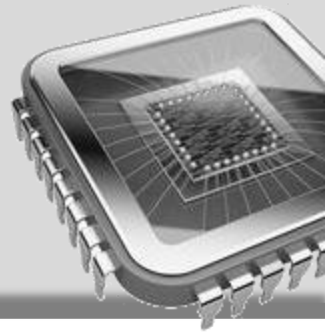
```
#include “calculadora.h”
/* Retorna o valor da soma de dois números */
float soma(float x, float y) {
  return x + y;
}
/* Retorna o valo da diferença de dois números*/
float subtracao(float x, float y) {
  return x – y;
}
```

Arquivo header (implementação) calculadora.c (continuação)



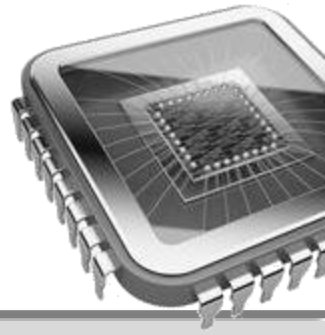
```
...
/* Retorna a multiplicação de dois números */
float multiplicacao(float x, float y) {
  return x * y;
}
/* Retorna a divisão de x por y, exceto quando y = 0 */
float divisao(float x, float y) {
  if (y == 0)
    return -1;
  else
    return x/y;
}
```

Arquivo header (chamada) main.c



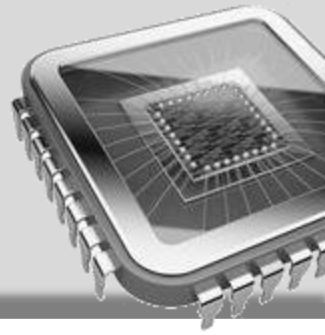
```
#include <stdio.h>
#include <stdlib.h>
#include “calculadora.h”

int main() {
  printf(“2 + 3 = %f\n”, soma(2,3));
  printf(“7.5 – 2.5 = %f\n”, subtracao(7.5,2.5));
  printf(“8.5 * 2 = %f\n”, multiplicacao(8.5,2));
  return 0;
}
```



COMO COMPILAR E EXECUTAR O PROGRAMA?

COMPILAÇÃO

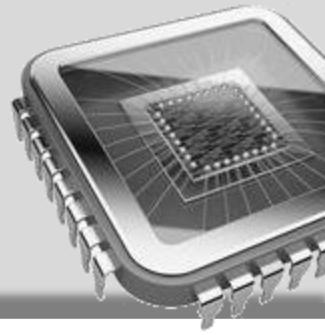


- Para compilar os arquivos do programa e transformá-los em um arquivo executável, basta:

```
gcc calculadora.c main.c –o calculadora
```

A primeira fase da compilação é um pré-processamento, executado por um pré-processador, que cuida de todas as linhas de código que começam com #. A segunda fase, compila os arquivos pré-processados.

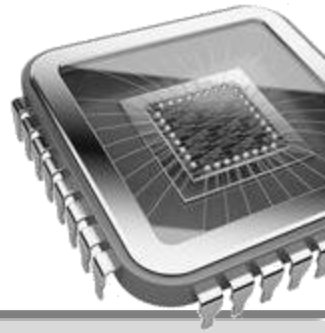
Execução



- O programa pode ser executado:

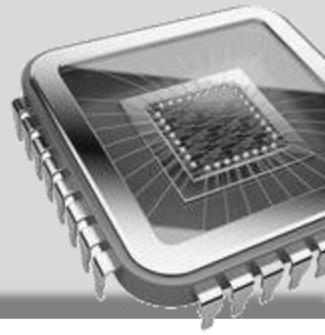
./calculadora

Se a função main do seu programa tiver parâmetros, digite os correspondentes argumentos na linha de comando. Por exemplo, se o seu programa tiver dois argumentos, digite:
./calculadora 2 3



MAKE E MAKEFILE

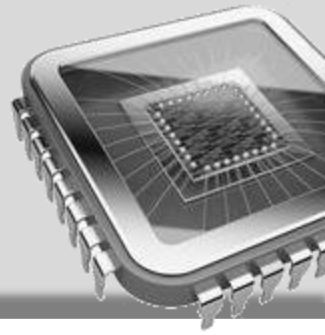
Make e Makefile



- Para automatizar a compilação do seu programa, reduzindo assim a digitação de longas linhas de comando, basta utilizar o **make**.

Para usar o make é necessário criar o arquivo makefile.

Makefile



all: calculadora

calculadora: calculadora.c main.c gcc calculadora.c main.c -o calculadora

clean: rm -rf *.o calculadora

Se o comando não for precedido por um Tab. Você obterá o erro: “*** missing separator. Stop”;
O makefile pode ser nomeado como: makefile ou Makefile.