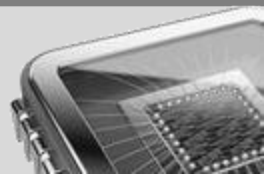
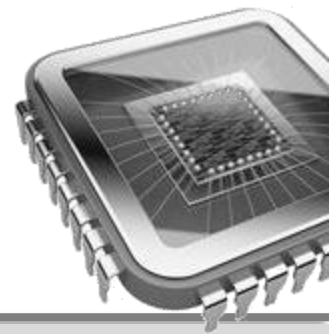


Introdução à Programação

Aula 04

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

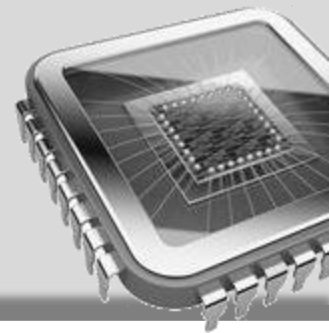




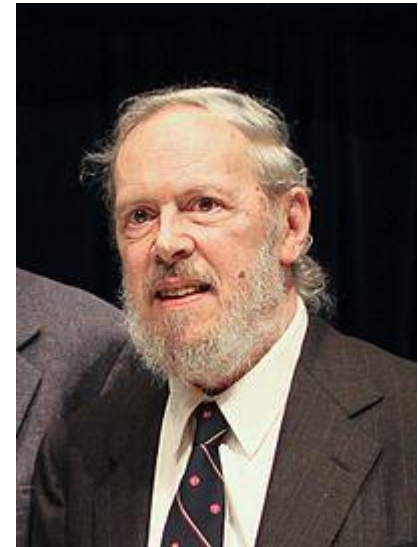
QUAL LINGUAGEM IREMOS ESTUDAR?

Introdução à Programação

Linguagens de Programação C



- Linguagem de programação C
 - Criada em 1972 por Dennis Ritchie
 - AT&T Bell Labs
 - Para desenvolver o S.O. Unix
 - Originalmente foi escrita em Assembly
 - A C é uma linguagem de proposito geral



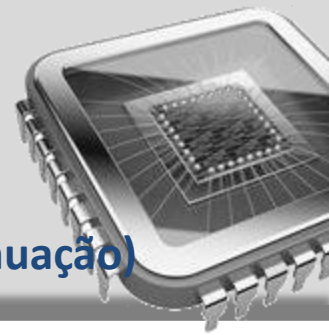
Dennis Ritchie



A linguagem foi chamada de C, porque suas características foram obtidas a partir de uma linguagem anteriormente chamada de B.

Introdução à Programação

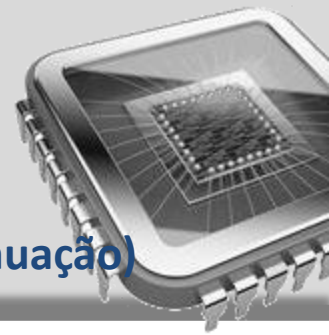
Linguagens de Programação C (continuação)



- A linguagem C é uma das linguagens de programação mais populares
- Existem poucas arquiteturas para as quais não existem compiladores para C
- C tem influenciado muitas outras linguagens de programação
- O Instituto Norte-americano de Padrões (ANSI) estabeleceu uma especificação para linguagem C
 - ANSI X3.159-1989: Programming Language C (C ANSI)

Introdução à Programação

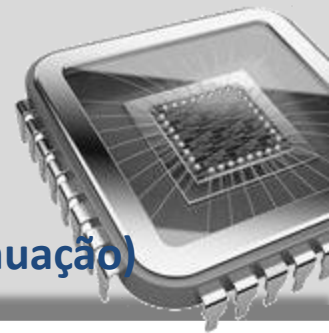
Linguagens de Programação C (continuação)



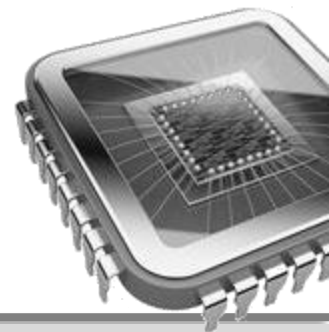
- Características básicas:
 - Case Sensitive
 - Diferencia maiúsculo e minúsculo
 - Suporta tipos de dados primitivos
 - Caractere
 - Inteiro
 - Ponto flutuante
 - Possui estruturas de controle para a programação estruturada

Introdução à Programação

Linguagens de Programação C (continuação)



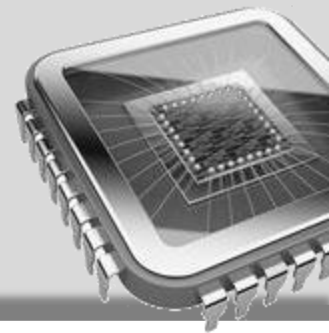
- Características básicas (continuação):
 - Possui operadores
 - Aritméticos
 - Lógicos,
 - Relacionais
 - Condicionais
 - Bit a bit
 - Apresenta funções de entrada e saída formatadas
 - Todo o programa em C tem uma função principal
 - Todas as linhas (instruções) do programa deve terminar com “;”



COMO PROGRAMAR EM C?

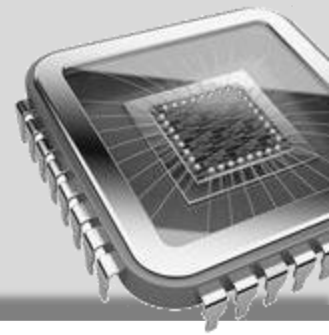
Linguagens de Programação C

Ambiente de Programação

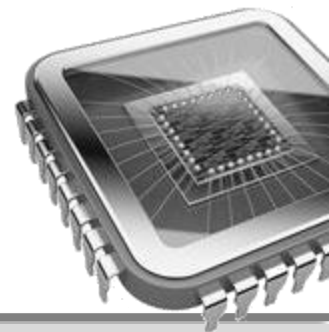


- Um ambiente de programação é o conjunto de ferramentas usadas no desenvolvimento de programas
- Essas ferramentas podem ser:
 - Sistema de arquivos
 - Editor de texto
 - Linkeditor
 - compilador

Ambiente de Programação Interface de Desenvolvimento



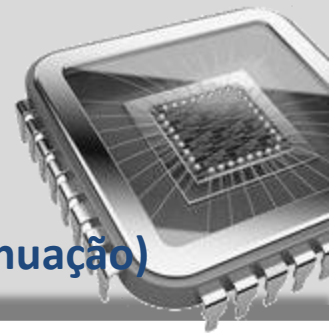
- Ambiente Integrado de Desenvolvimento (IDE)
 - Inclui uma grande coleção de ferramentas integradas
 - Pode fornecer ainda programas auxiliares para o desenvolvimento
 - Gerador de códigos: ferramenta que possui a capacidade de gerar códigos a partir de um determinado modelo (diagrama de blocos)



QUAL IDE SERÁ UTILIZADA?

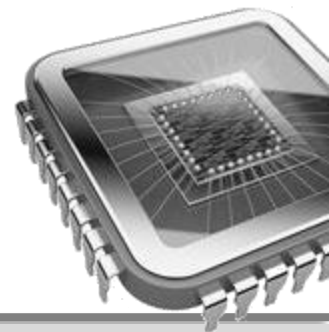
Ambiente de Programação Interface de Desenvolvimento

(continuação)



NÃO USAREMOS IDE!

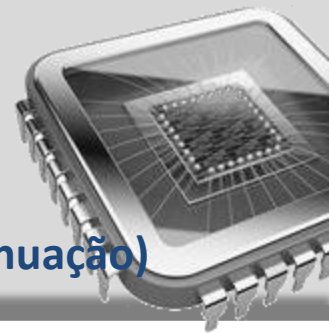
- IDE cria *makefiles* e regras próprias de compilação que acabam dificultando o aprendizado
- Estudantes de computação precisam saber como as coisas acontecem...
 - Entender os comandos (linha a linha)
 - Entender os parâmetros de compilação (como se compila)
 - O porque do fechamento daquele par de “{ }”
 - Aprender as palavras chaves/reservadas
 - Aprender a sintaxe da linguagem



E O QUE USAREMOS?

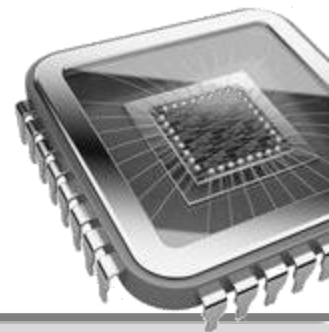
Ambiente de Programação Interface de Desenvolvimento

(continuação)



USAREMOS O COMPILADOR + EDITOR DE TEXTO

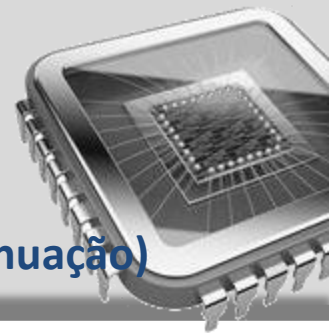
- Sem geração de código automático
- Sem fechamento automático de par de “{ }”
- Sem endentação automática
- Sem sintaxe padrão da linguagem
- Sofrimento (“programação na mão”)



QUANDO TEREI O PRAZER EM USAR UMA IDE?

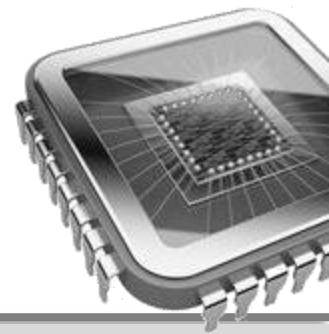
Ambiente de Programação

Interface de Desenvolvimento (continuação)



QUANDO JÁ ESTIVER DOMINANDO A LINGUAGEM

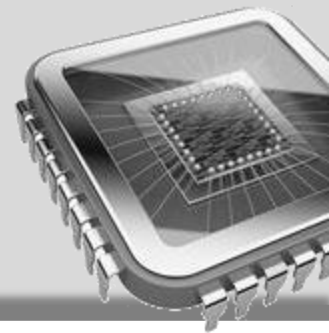
- Entendendo toda as etapas de compilação de um programa
- Apresentar poucos erros no console/prompt...
 - Pode usar um editor de texto que melhore a legibilidade (NOTEPAD++)
- Quando você estiver pensando em programação durante o café-da-manhã. Lanche, almoço e jantar...
 - Pode ir para uma IDE (Eclipse ou netbeans)



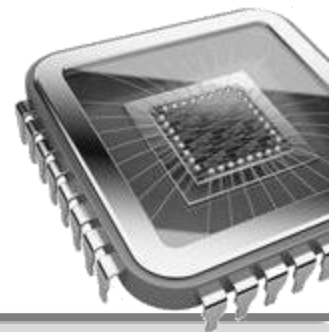
COMPILADOR

Ambiente de Programação

Compilador



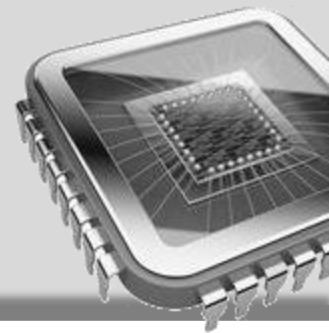
- Vamos nos concentrar no compilador **GNU Compiler Collection (GCC)**
 - Mas qualquer outro compilador aderente ao padrão da linguagem C pode ser utilizado
- No Linux o compilador gcc já vem instalado na maioria das distribuições Linux
- No Windows existem alguns pacotes que implementam um subconjunto do gcc
 - Minimalist GNU for Windows (MinGW)
 - GNU + Cygnus + Windows (CygWin)



EDITOR DE TEXTO

Ambiente de Programação

Editor de Texto

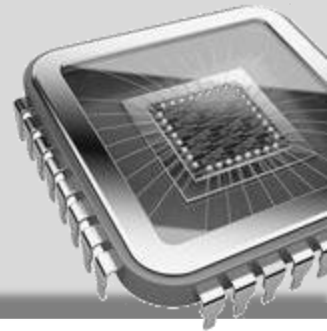


EDITOR DE TEXTO LINUX

- No linux há excelentes opções de editores de texto
 - Gedit (Editor de texto padrão GUI)
 - Kate (pode acoplar um terminal virtual e funcionar como uma IDE)
 - Geany (Tem características básicas de uma IDE)
 - Vi (O editor de texto mais utilizado do mundo)
 - Emacs (Um ambiente de trabalho: documentos, email, web, música, IRC, agenda...)

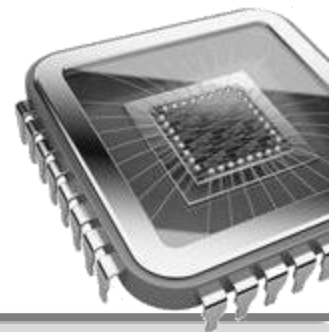
Ambiente de Programação

Editor de Texto (continuação)



EDITOR DE TEXTO WINDOWS

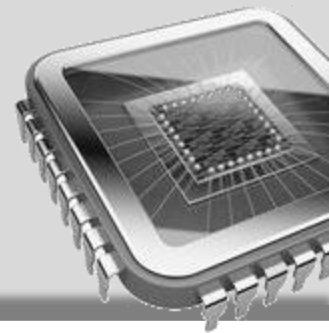
- Para Windows não há muitas opções:
 - Bloco de notas
 - Notepad++ (Editor que oferece uma interface agradável e melhor que o bloco de notas)



COMPILAÇÃO

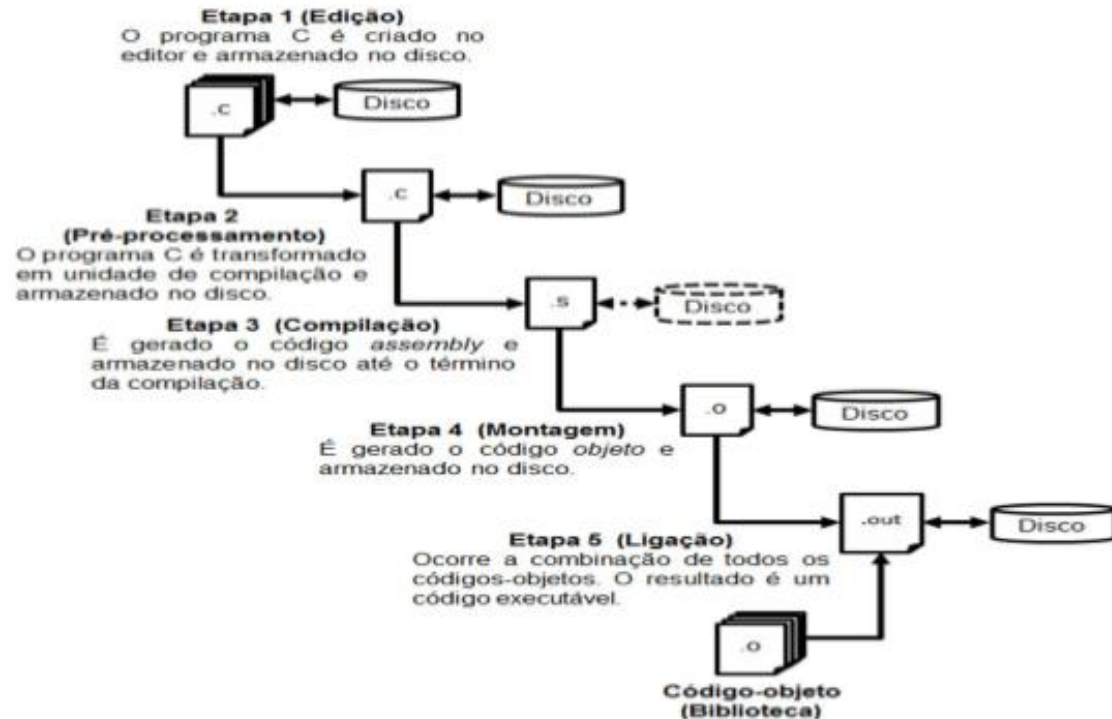
Ambiente de Programação

Compilação



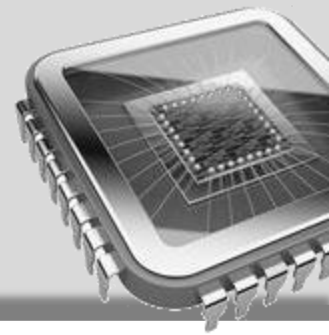
- O compilador C realiza o processo de compilação em 5 etapas:

- Edição
- Pré-processamento
- Compilação
- Montagem
- Ligação



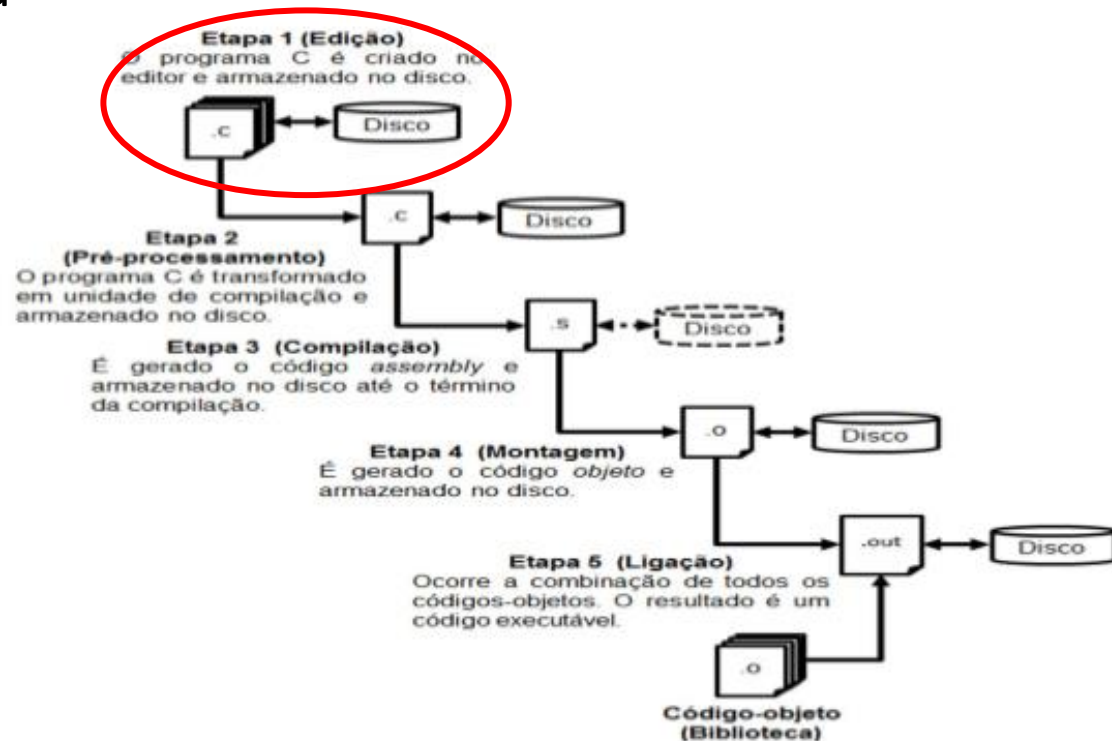
Ambiente de Programação

Compilação (continuação)



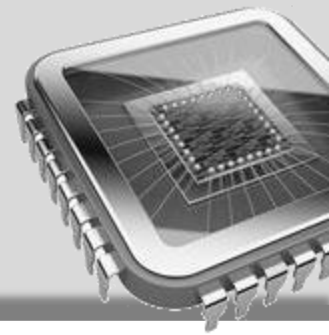
Edição

- Consiste de um programa de edição
- Os arquivos do programa devem ser então armazenados em um dispositivo de armazenamento secundária
- Os arquivos devem ter a extensão .c



Ambiente de Programação

Compilação (continuação)

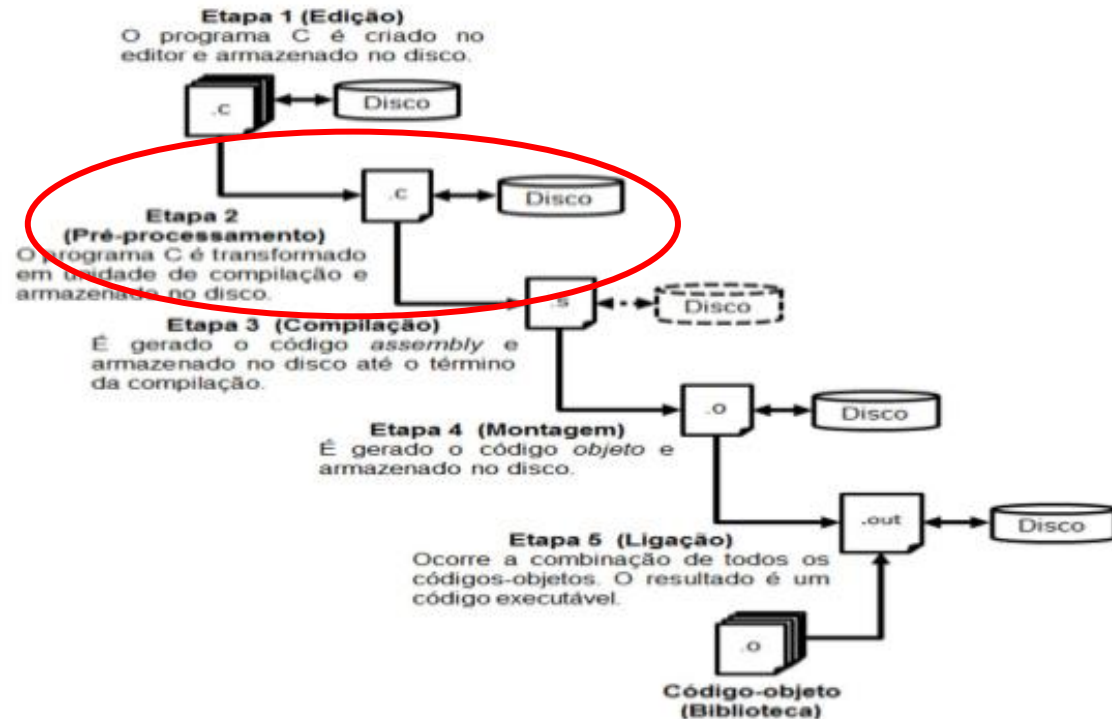


Pré-processamento

- Responsável por modificar o código-fonte
- Ocorre nessa etapas:
 - a suspensão de espaços
 - substituição de macros
 - Inclusão de códigos
- É gerado o código de unidade de compilação

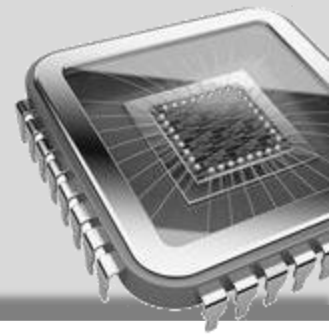
```
gcc -E <prog>.c
```

```
gcc -E <prog.c> -o code.e
```



Ambiente de Programação

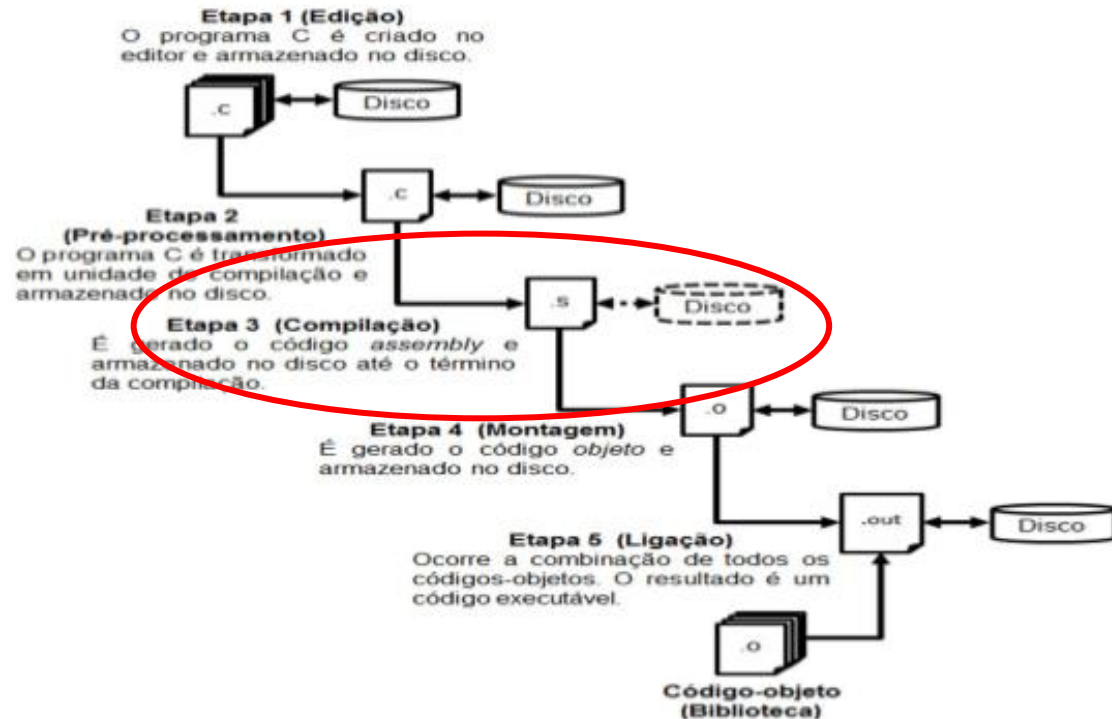
Compilação (continuação)



Compilação

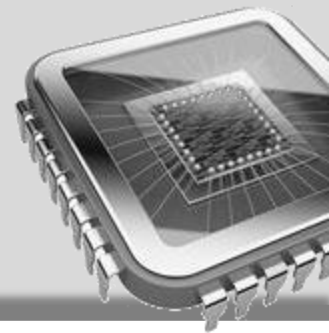
- Responsável pela análise sintática e semântica da unidade de compilação
- Gera o código assembly correspondente

```
gcc -S <prog>.c
```



Ambiente de Programação

Compilação (continuação)



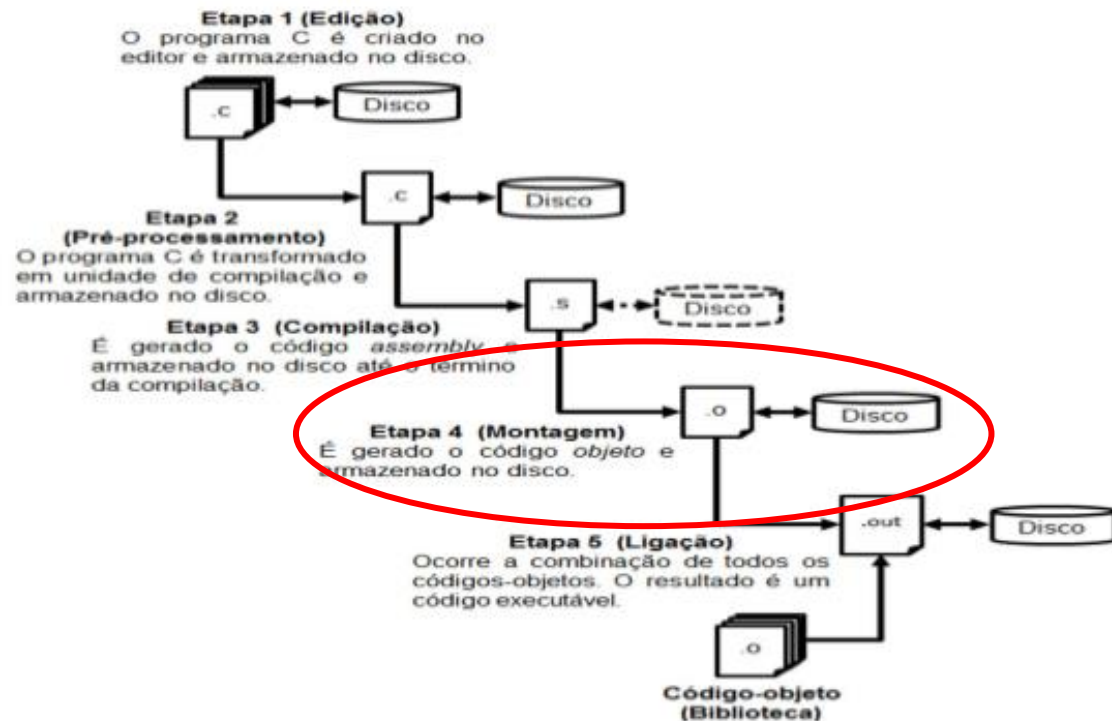
Montagem

- Responsável por gerar o código-objeto

- Os comandos assembly são transformado em linguagem de máquina

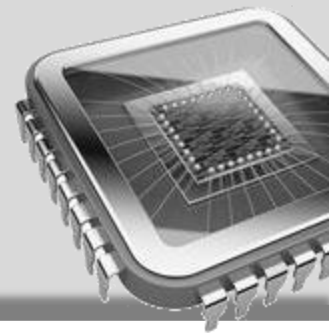
```
gcc -c <prog>.c
```

```
gcc -c <prog.c> -o code.s
```



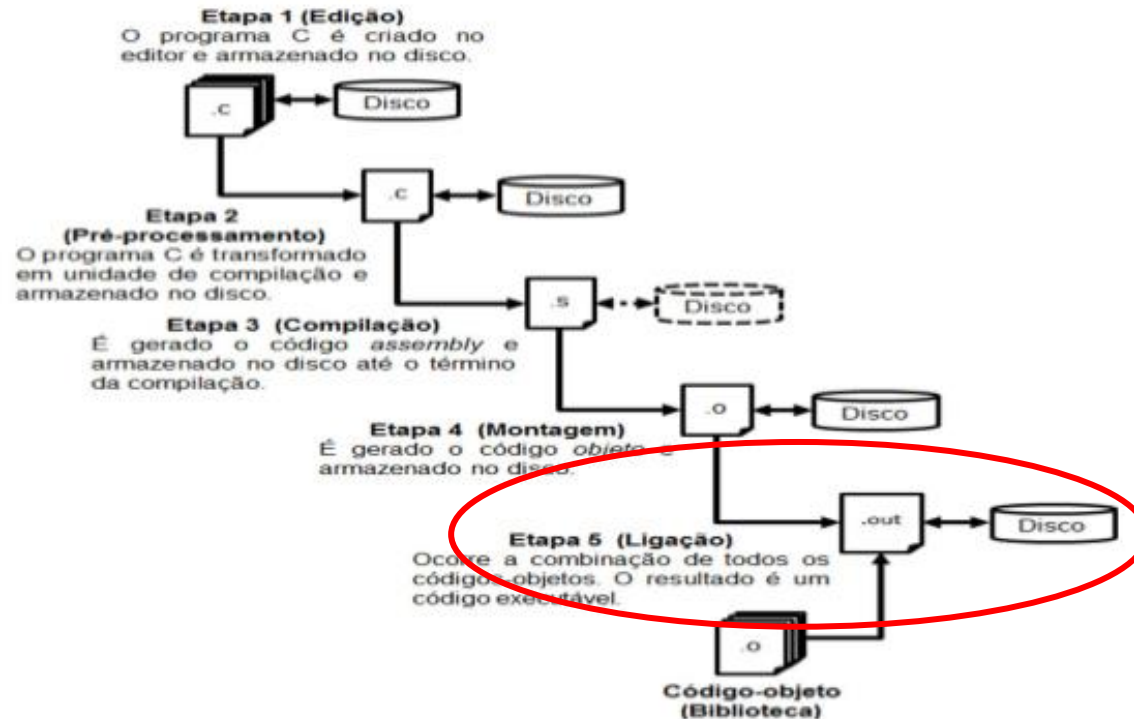
Ambiente de Programação

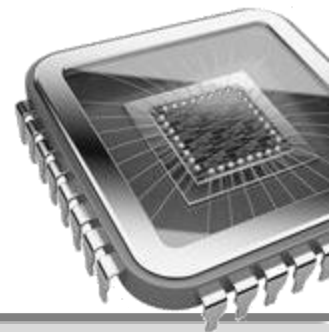
Compilação (continuação)



Ligação

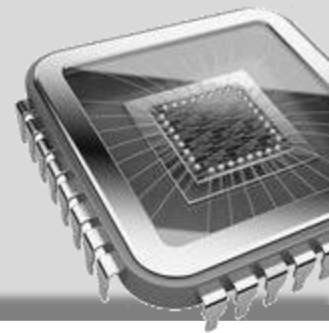
- Etapa final...
 - Onde ocorre a combinação de todos os códigos-objetos que compõem um programa
 - O resultado é um código executável





ESTRUTURA DE UM PROGRAMA C

Estrutura de um Programa C

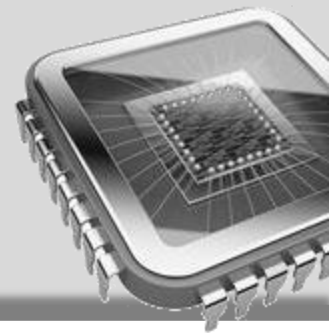


- Um programa C consiste em uma função ou várias funções.

```
<tipo_retorno> <nome_função>(<lista_parâmetro>){  
    <instrução_1>;  
    <instrução_2>;  
    ...  
    <instrução_n>;  
    return <valor_retorno>;  
}
```

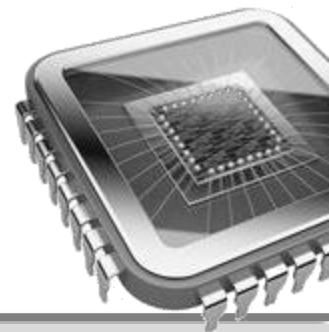
Estrutura de um Programa C

(continuação)



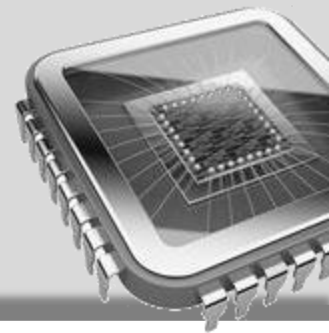
- Um programa C precisa ter uma função principal, que deve ser chamada de **main**.
 - Essa função marca o ponto inicial do programa e a execução do programa só termina quando a função **main** encerrar.
 - A função **main**, retorna um valor inteiro e pode ou não possui parâmetros.

```
int main(){  
    int a, b, c;  
    c = a+b;  
    return 0;  
}
```



COMENTÁRIOS

Comentários

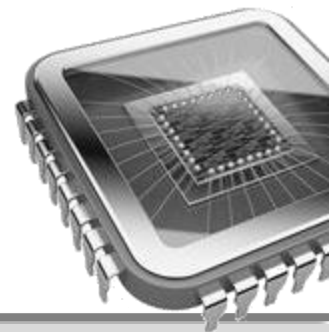


Comentário de linha

- Inicia com `//` e termina no final da linha;
- Pode iniciar no meio de uma linha e continuar até o final da linha

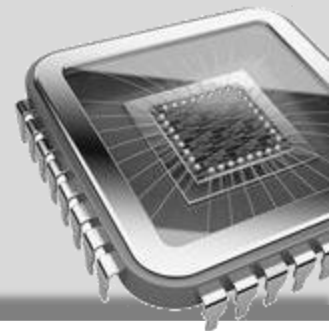
Comentário múltiplas linha

- É iniciado pelo delimitador `/*` e termina com o delimitador `*/`
- Todo o texto entre os delimitadores do comentário é ignorado pelo compilador.



PALAVRAS RESERVADAS

Palavras Reservadas



auto	break	case	char
const	continue	default	do
double	else	enum	extern
float	for	goto	if
inline	int	long	register
restrict	return	short	signed
sizeof	static	struct	switch
typedef	union	unsigned	void
volatile	while	_Alignas	_Alignof
_Atomic	_Bool	_complex	_Generic
_Imaginary	_Noreturn	_static_assert	_Thread_local