



# Programação de Software Básico

MATA49

Prof. Babacar Mane

2021.1



# Programação de Software Básico

- **Apresentação da disciplina**

- Planejamento:

- Cronograma de aulas e slides

- Google Class - MATA49 - Programação de Software Básico

- Contatos através do e-mail:

- babacarm@ufba.br

- Utilizar no assunto o prefixo: [MATA49]



# Programação de Software Básico

- **Apresentação da disciplina**

- Metodologia:

- Aulas expositivas

- Exercícios práticos



# Programação de Software Básico

- **Apresentação da disciplina**

- Planejamento:

- Avaliação

- Um trabalho prático

- Seminário prático

- Média Final:  $((\text{TrPr} + \text{SePr}) / 2)$



# Programação de Software Básico

- **Apresentação da disciplina**

- **Conteúdo:**

- Arquitetura Intel
    - Representação dos dados em linguagem de máquina
    - Introdução a Linguagem de montagem
    - Condicionais e estrutura de repetição
    - Operações com Bits
    - Vetores
    - Instruções para manipulação de Vetores e Strings



# Programação de Software Básico

- **Apresentação da disciplina**

- **Conteúdo:**

- Pilhas
    - Procedimentos
    - Interrupções
    - Instruções sobre ponto flutuante
    - Assembly com C
    - Assembly 64 bits Intel
    - Introdução linguagem de montagem ARM

# Programação de Software Básico

- **Apresentação da disciplina**

- **Referencias Bibliográficas:**

- Slides do Prof. LEANDRO ANDRADE
    - Paul A. Carter, PC Assembly Language • Kip Irvine, Assembly Language for x86 Processors, 6th edition
    - Andrew S. Tanenbaum, Organização Estruturada de Computadores, 4ª edição, Prentice-Hall do Brasil, 2001
    - Manual oficial do Nasm
    - Hugo Perez Perez. Tutorial de linguagem Assembly. Information Systems General Coordination. University of Guadalajara. Traduzido para o português por Jeferson Amaral. 1995
    - IA-32 Intel Architecture Software Developer's Manual volume1: Basic architecture. Intel Corporation. 2003





# Programação de Software Básico

- **Apresentação da disciplina**
  - **Objetivo:**
    - Programação em linguagem de montagem Intel
    - Recursos de arquitetura para técnicas de programação
      - Arquitetura Intel
  - Conceitos de implementação de baixo nível





# Programação de Software Básico

## ➤ Introdução

- O que é uma **linguagem de montagem**?
- Qual a diferença entre **assembly** e **assembler**?



# Programação de Software Básico

## ➤ Introdução

### ➤ Tradutores:

- Programas que convertem um programa escrito em uma linguagem para outra
- Linguagem fonte → Linguagem Alvo
- Geração do programa objeto

### ➤ Interpretação:

- O programa fonte é executado por um interpretador.
- Em certos caso geração de um código intermediário (Ex: Java)

# Programação de Software Básico

## ➤ Introdução

- Quando uma linguagem de programação e uma representação simbólica de uma linguagem de maquina ela e chamada de linguagem de montagem ou assembly
- A instrução de maquina:
  - 10110000 01100001
- Pode ser representada por:
  - MOV AL, 61h
- Que significa mover o numero 97 para o registrador AL

# Programação de Software Básico

## ➤ Introdução

### ➤ Exemplo:

`mov dest, src`

- copia em *dest* o conteúdo de *src*.

`mov eax, 3;` *grava 3 no registrador eax*

`mov ebx, eax;` *grava o conteúdo de eax em ebx*

`add`

- adiciona inteiros.

`add eax, 4;` *eax = eax + 4*

`add ebx, eax;` *ebx = ebx + eax*

`sub` : mesmo formato de `add`

`inc` : `inc eax;` *eax++*

`dec` : `dec eax;` *eax--*



# Programação de Software Básico

## ➤ Introdução

- Assim as linguagens de montagens são uma representação direta das linguagens de máquina
- *Isto é uma instrução de máquina é representada por uma instrução na linguagem de montagem*

# Programação de Software Básico

## ➤ Introdução

### ➤ Por exemplo:

- O comando na linguagem C abaixo não representa uma instrução na linguagem de máquina

**`int x = y + 5`**

- Já a instrução `ADD AX, 5` representa um único comando de máquina para o processador



# Programação de Software Básico

## ➤ Introdução

➤ Os tradutores podem ser:

➤ **Assembler (Montador):** Quando a linguagem fonte é uma representação simbólica da linguagem de máquina

❑ *Neste caso a linguagem fonte é chamada de linguagem de montagem (assembly language)*

➤ **Compilador:** Quando a linguagem fonte é uma linguagem de alto nível e o alvo é uma linguagem de máquina numérica ou uma representação simbólica





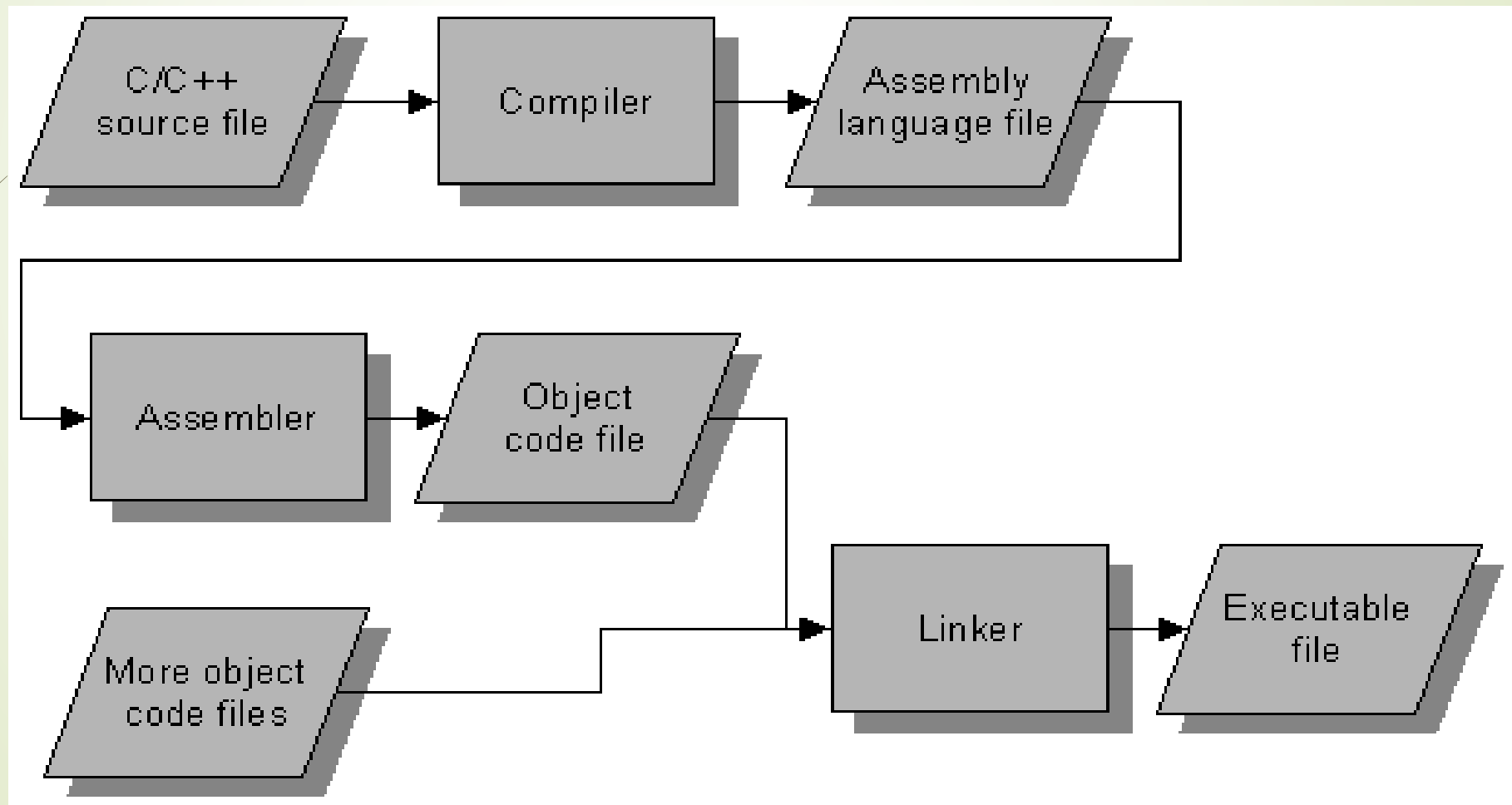
# Programação de Software Básico

## ➤ Introdução

- **Assembler** é o programa que transforma a representação simbólica da linguagem de máquina em programa objeto (executável)
- **Assembly** é a linguagem fonte de representação simbólica da linguagem de máquina

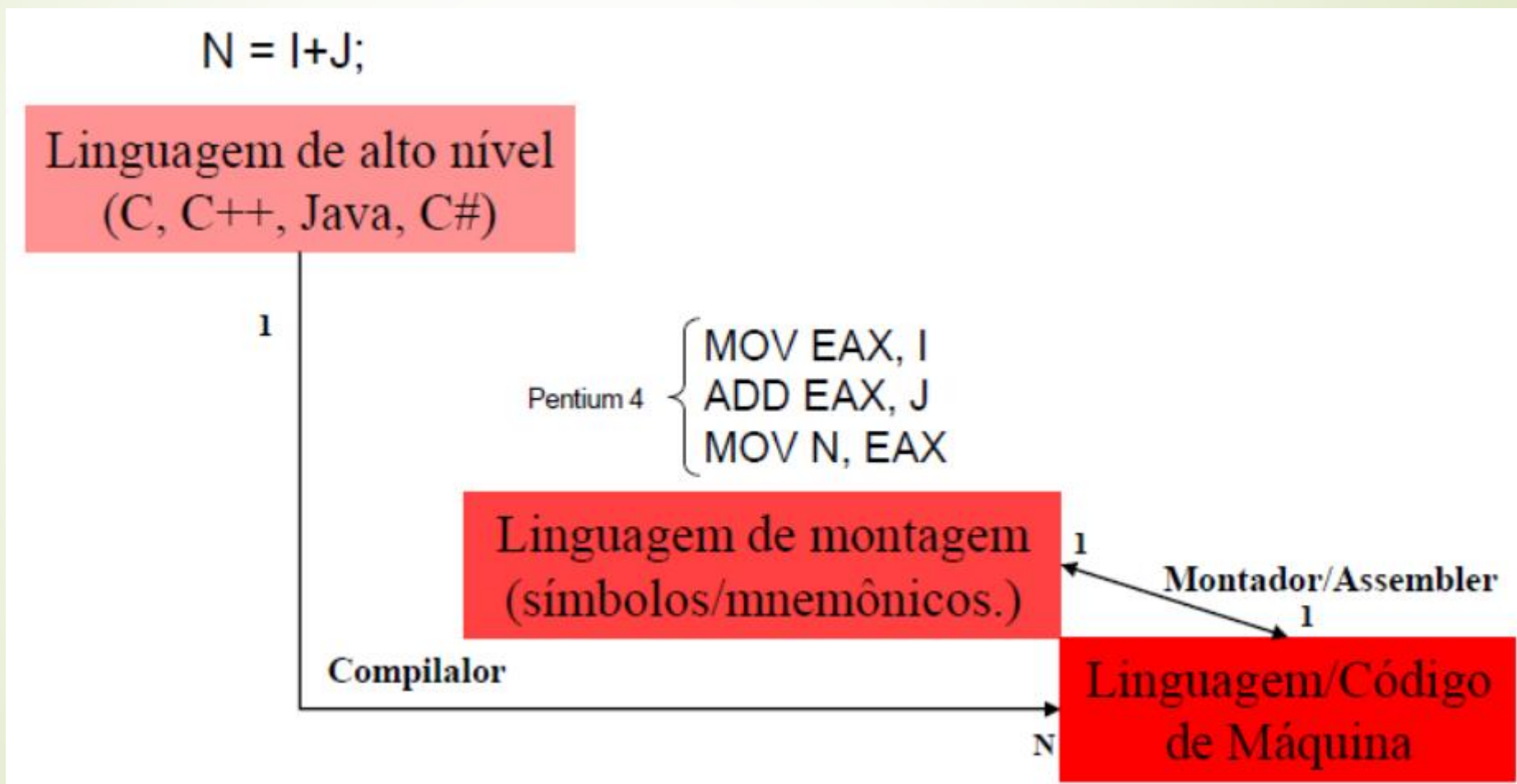
# Programação de Software Básico

## ➤ Introdução



# Programação de Software Básico

## ➤ Introdução





# Programação de Software Básico

- **Introdução**
- **Cada declaração produz uma instrução de máquina**
- **Facilita o processo de programação**
  - *Acredite! Pior seria usar somente binários e hexadecimais*
- **Diretamente ligado a arquitetura do processador**
  - *Dificulta o reuso para outras arquiteturas*

# Programação de Software Básico

## ➤ Introdução

Linguagem de máquina	Linguagem de montagem
BA0B01	mov dx,msg
B409	mov ah,9
B44C	mov ah,4Ch
CD21	int 21h
48656C6C6F2C20576F	msg db 'Hello, World!',0Dh,0Ah,'\$'

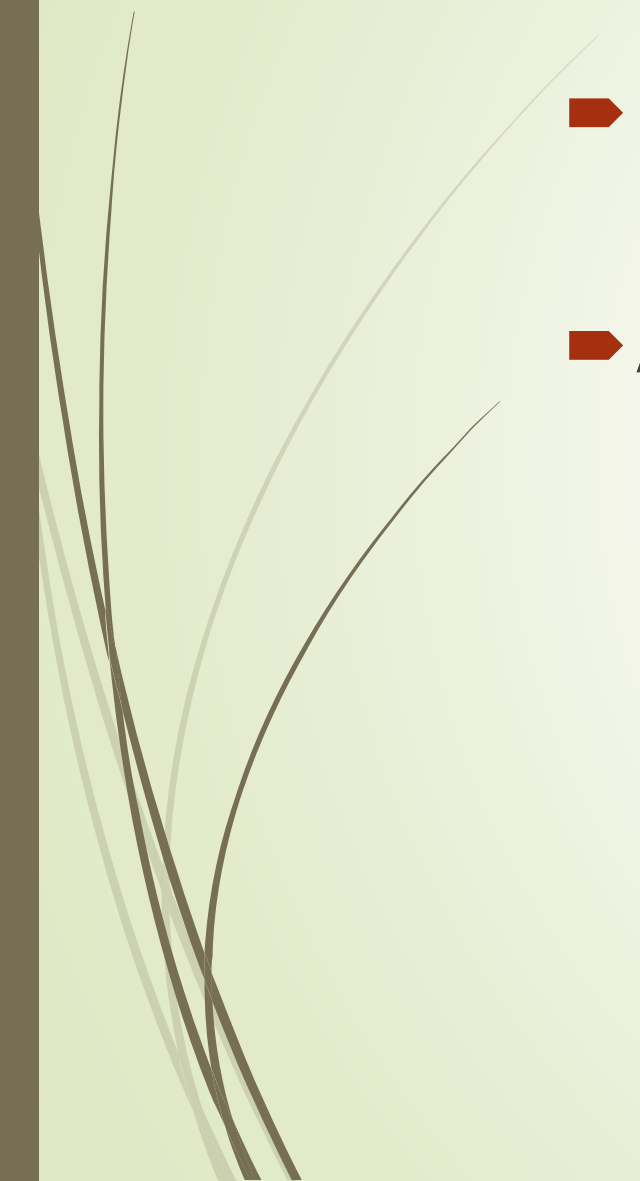


# Programação de Software Básico

- **Introdução**
- **Existência de linguagens híbridas**
  - São constituídas por instruções de alto nível
  - Porém permitem a execução de instruções de máquina
- **Exemplo: Linguagem C**
  - Por isso é considerada por muitos autores como uma linguagem de “médio” nível



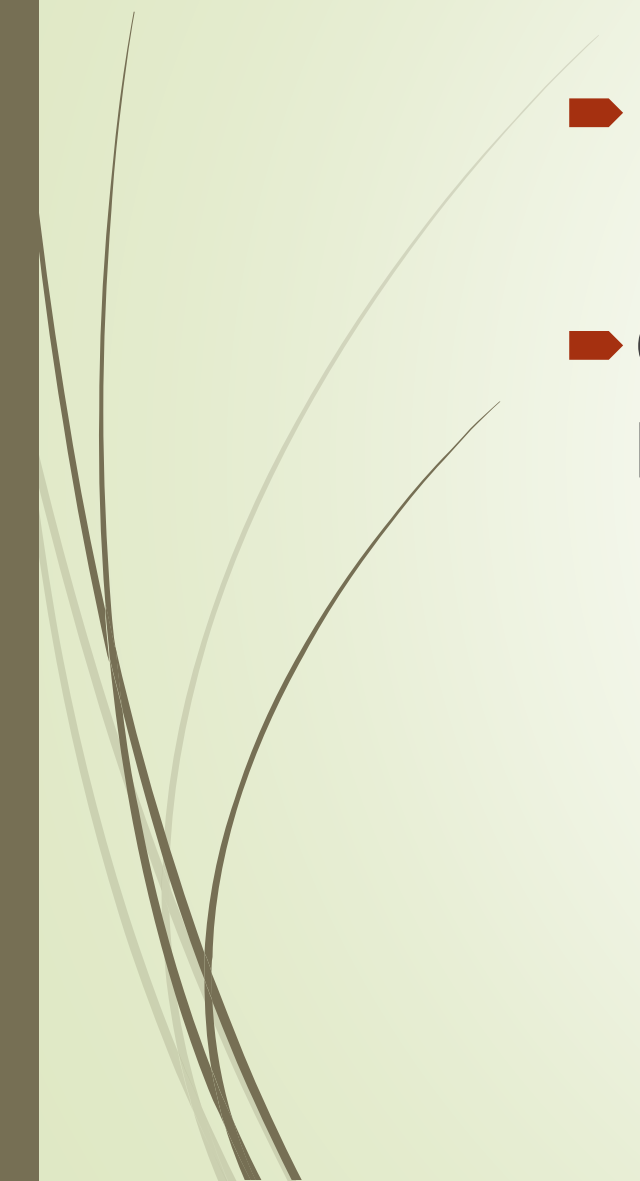
# Programação de Software Básico

- **Introdução**
  - **As linguagens de montagens estão diretamente ligadas a arquitetura do processador**
  - *É possível haver mais de uma linguagem de montagem para uma única arquitetura*
- 






# Programação de Software Básico

- **Introdução**
  - **Como é o caso da arquitetura intel x86, que possui por exemplo:**
    - ● **MASM (Microsoft Macro Assembler)**
    - ● **TASM (Turbo Assembler)**
    - ● **NASM (Netwide Assembler)**
    - ● **GAS (GNUassembler)**
- 



# Programação de Software Básico

- **Introdução**
  - **Escrever um programa em linguagem de montagem demora muito mais do que escrever o mesmo em uma linguagem de alto nível**
    - – *Isso inclui: Depuração, manutenção*
- 



# Programação de Software Básico

## ➤ Introdução

- *Se arquitetura muda, logo o código muda também!*
- *Em outras palavras, programação em linguagem de montagem não é trivial!*



# Programação de Software Básico

## ➤ Introdução

➤ *As linguagens de alto nível possuem uma relação 1-para-muitos com as linguagem de montagem*

➤ *– Por exemplo:*

➤ *Programa em C: O mesmo programa em assembly:*

# Programação de Software Básico

- Introdução
- Linguagem C


```
int Y;  
int X = (Y + 4) * 3;
```

## Assembly

```
mov     eax, Y  
add     eax, 4  
mov     ebx, 3  
imul    ebx  
mov     X, eax
```



# Programação de Software Básico

- **Introdução**
  - **Por que ainda usar/aprender linguagem de montagem?**
- 



# Programação de Software Básico

- **Introdução**
- **Busca por melhor desempenho:**
  - *O código desenvolvido em linguagem de máquina pode ser muito menor (ou mais eficiente) do que um código compilado de uma linguagem de alto nível*
- **Uso de assembly em pontos críticos de desempenho**
  - *E comum 10% do código de um programa ser responsável por 90% do tempo de execução*





# Programação de Software Básico

- **Introdução**

- *Acesso direto ao hardware*

- *Certos componentes só podem ser acessados por linguagem de montagem*

- ● *Desenvolvimento de drivers*

- ● *Ex: Controladores de dispositivos embutidos de tempo real*



# Programação de Software Básico

- **Introdução**
- **Desenvolvimento de sistemas operacionais**
  - – *Na construção de S.O é imprescindível o uso de programas ou trechos de código em assembly*
- **Manipulação de vírus e técnicas de engenharia reversa**
  - – *A maior parte dos vírus trabalha com códigos em assembly*
  - – *A engenharia reversa é o processo de tornar o um programa objeto em código assembly para ter acesso ao código-fonte do programa*



# Programação de Software Básico

## ➤ Introdução

Por que usá-la?

## ➤ Escassez de memória

## ➤ Desenvolvimento de compiladores

## ➤ Compreender o real funcionamento do computador



# Programação de Software Básico

## ➤ Introdução

Por que usá-la?

## ➤ Processo de emulação

- *Seja na emulação de vídeo games ou em sistemas de computadores e fundamental o uso de assembly para fazer uma conexão entre as linguagens de máquina dos sistemas*

# Programação de Software Básico

- Introdução

- Na disciplina utilizaremos a linguagem: (*Nasm*)

- – *Para Linux: ( Gcc + nasm)*

- – *Para Windows: (Cygwin inclui módulos do gcc e do nasm) ou SASM*