

Disciplina: Programação de Sistemas
Professor: André Flores dos Santos
Nome: Pedro Henrique de Brito Canabarro
Data: 07/08/2025

1. Resumo Geral sobre Assembly:

A linguagem Assembly é uma linguagem de programação de baixo nível que serve como uma representação simbólica da linguagem de máquina, a linguagem compreendida diretamente pela CPU. Enquanto a linguagem de máquina é composta apenas por números binários, o Assembly utiliza meios mnemônicos que facilitam a leitura e escrita por seres humanos, como MOV, ADD, JMP, entre outros.

Assembly é considerada de baixo nível porque mantém uma correspondência direta com as instruções do hardware. Cada instrução Assembly equivale, geralmente, a uma única instrução de máquina. Isso permite controle total sobre o hardware, sendo útil em aplicações como sistemas embarcados, drivers de dispositivo e sistemas operacionais.

2. Variáveis em Assembly

Em Assembly, as variáveis são representadas por rótulos associados a posições de memória e são declaradas usando diretivas como DB, DW e DD. Essas diretivas informam ao assembler quanto espaço alocar para cada variável e qual valor inicial atribuir.

- DB (Define Byte): aloca 1 byte.
Ex: var1 DB 10

- DW (Define Word): aloca 2 bytes.
Ex: var2 DW 1000

- DD (Define Double Word): aloca 4 bytes.
Ex: var3 DD 12345678 h

Exemplo:

```
section .data
valor1 DB 5
valor2 DW 100
valor3 DD 12345
```

3. Registradores e sua Importância

Registradores são pequenas áreas de armazenamento extremamente rápidas dentro do processador, utilizadas para armazenar dados temporários durante a execução das instruções. Principais registradores na arquitetura x86:

- EAX: acumulador, usado em operações aritméticas.
- EBX: registrador base.
- ECX: contador, muito usado em loops.
- EDX: dados, usado em multiplicações e divisões.

Exemplo de uso: MOV EAX, 5 MOV EBX, 10 ADD EAX, EBX ; EAX agora contém 15

4. Instruções Aritméticas e Lógicas

As instruções aritméticas e lógicas manipulam dados diretamente nos registradores.

- ADD, SUB: soma e subtração.
- MUL, DIV: multiplicação e divisão.
- AND, OR, XOR: operações lógicas bit a bit.

Exemplo:

MOV EAX, 5

ADD EAX, 3 ; EAX = 8

XOR EAX, EAX ; EAX = 0 (zera o registrador)

5. Controle de Fluxo em Assembly

O controle de fluxo é feito com instruções de salto, que redirecionam a execução do programa.

- JMP: salto incondicional.
- JE, JG, JL: saltos condicionais (igual, maior, menor).

Exemplo de loop:

MOV ECX, 5

loop_inicio:

; alguma operação

DEC ECX

JNZ loop_inicio ; repete enquanto ECX ≠ 0

6. Modos de Endereçamento

- Imediato: valor constante. Ex: MOV AX, 10
- Direto: endereço direto de memória. Ex: MOV AX, [1000h]
- Indireto: endereço via registrador. Ex: MOV AX, [BX]

Exemplo:

MOV AL, 25 ; imediato

MOV AX, [var1] ; direto

MOV SI, var1

MOV AX, [SI] ; indireto

7. Ferramentas para Programar em Assembly

- NASM (Netwide Assembler): assembler moderno e popular para x86.
- MASM (Microsoft Assembler): muito usado em sistemas Windows.
- TASM (Turbo Assembler): da Borland, usado em sistemas antigos.
- EMU8086: simulador para aprendizado da arquitetura 8086.
- MARS 4.5: simulador para Assembly MIPS, arquitetura RISC.

Essas ferramentas permitem escrever, montar e depurar código Assembly com suporte a diferentes arquiteturas.