

**AVALIAÇÃO 01 – PROGRAMAÇÃO DE SISTEMAS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**  
**UNIVERSIDADE FRANCISCANA – UFN. 2025-02. Peso:5,0.**

**PROFESSOR:** André F. dos Santos.

**Nome do aluno:** Pedro Henrique Camabarre . **Data:** 28/08/25.

OBS: Marque apenas uma alternativa nas questões de múltipla escolha. Desligar o celular durante a prova.

**1) No fluxo clássico de construção de um programa, você escreve em linguagem de alto nível, o compilador traduz para Assembly e, em seguida, outro componente transforma esse Assembly em código de máquina; por fim, um ligador produz o executável.**

Assinale a alternativa que descreve exatamente o papel do montador nesse fluxo:

- a) Converter código de máquina em linguagem de alto nível.
- b) Converter Assembly em código de máquina (objeto/binário).
- c) Reunir vários objetos em um executável (linkagem). ✓
- d) Interpretar Assembly em tempo de execução, sem gerar binário.
- e) Converter linguagem de alto nível diretamente em binário.

**2) Em um código com salto para rótulo ainda não declarado (referência futura), há duas estratégias de montagem: 1-pass (uma passagem) e 2-pass (duas passagens). No 2-pass, a 1<sup>a</sup> varredura cria a SYMTAB (rótulo → endereço) e a 2<sup>a</sup> gera os bytes finais; no 1-pass, frequentemente se usa fixup/backpatch para preencher endereços depois.**

Assinale a alternativa correta sobre as diferenças:

- a) 1-pass é mais rápido, mas tem limitações com referências futuras; 2-pass resolve rótulos desconhecidos criando SYMTAB e gravando depois. ✓
- b) 1-pass sempre gera código mais otimizado.
- c) 2-pass não utiliza tabela de símbolos. ✓
- d) 1-pass e 2-pass são idênticos na prática.
- e) 1-pass é “inviável” e não se usa em casos reais.

**3) A SYMTAB (tabela de símbolos) é central no montador. Ela mapeia rótulos e variáveis para endereços, permitindo que instruções como JMP LOOP ou operandos rotulados sejam resolvidos.**

Selecione a alternativa que melhor descreve a função da SYMTAB:

- a) Guardar apenas os opcodes das instruções (MOT).
- b) Armazenar rótulos/variáveis e seus endereços para resolver referências. ✓
- c) Estimar tempo de execução de trechos do programa.
- d) Converter C/C++ diretamente em Assembly.
- e) Otimizar automaticamente instruções aritméticas.

**4) Considere o trecho:**

**Código:**

```
MOV EAX, 5
CMP EAX, 5
JE equal
JMP end
equal:
; ...
end:
```

**O que significa a instrução JE nesse contexto?**

- a) Jump if Error
- b) Jump if Equal ✓
- c) Jump if End
- d) Jump if Even
- e) Jump if Enter

**5) Considere a mini-ISA didática abaixo:**

- MOV R0, #imm
- ADD R0, #imm
- JMP label
- HLT

Assinale a alternativa que corresponde aos tamanhos (bytes) das instruções em assembly acima;

- a) MOV=1, ADD=1, JMP=2, HLT=1  
 b) MOV=2, ADD=1, JMP=2, HLT=2  
 c) MOV=1, ADD=2, JMP=1, HLT=1  
~~d) MOV=2, ADD=2, JMP=2, HLT=1~~  
 e) MOV=2, ADD=2, JMP=3, HLT=1

✓

**6) Analise o programa mini-ISA:**

; Cada MOV/ADD/JMP = 2 bytes; HLT = 1 byte

①	MOV R0, #5	→ 2 bytes	0, 1
②	ADD R0, #3	→ 2 bytes	2, 3
③	JMP END	→ 2 bytes	:
④	LOOP: ADD R0, #1	→ 2 bytes	:
⑤	JMP LOOP	→ 2 bytes	:
	END: HLT	→ 1 byte	10

✗ 0, 0

- a) Numere os endereços por linha, começando em 0, considerando o tamanho de cada instrução.

~~0, 1, 2, 3, 4, 5~~, cada uma corresponde a 2 bytes de memória  
~~5 corresponde a 1 byte de memória~~

**7) No ambiente DOS (Emu8086), qual serviço lê 1 caractere com eco, retornando o código em AL?**

- ~~a)~~ INT 21h com AH=01h  
 b) INT 21h com AH=08h  
 c) INT 21h com AH=0Ah  
 d) INT 10h com AH=0Eh  
 e) INT 21h com AH=09h

✓

**8) O que acontece no seguinte código assembly e qual será a saída exibida no resultado?**

Código:

```
.data
num1 db 2
num2 db 7
num3 db 4
resultado db 0
msg db 'Resultado: $'

.code
start:
mov ax, @data
mov ds, ax

        mov al, num1
        add al, num2
        sub al, num3

        mov resultado, al
        mov ah, 09h
        lea dx, msg
        int 21h

        mov ah, 02h
        mov dl, resultado
        add dl, 30h
        int 21h

        mov ah, 4Ch
        int 21h
end start
```

Responda: Qual é a saída apresentada na tela ao final da execução?

Resultado: \$5

✓

9) Identifique e Diferencie 'operando imediato' de 'endereço de label' na nossa mini-ISA. Dê exemplos válidos para cada um.

Programa 'exemplo.asm'

; exemplo.asm — mini-ISA (MOV/ADD/JMP/HLT)

; Cada MOV/ADD/JMP ocupa 2 bytes; HLT ocupa 1 byte.

```

MOV R0, #5 ; carrega 5 em R0      } operando imediato
ADD R0, #3 ; R0 = 5 + 3 = 8
JMP END    ; salto para rótulo ainda não definido (referência futura)
LOOP: ADD R0, #1 ; incrementa R0 (exemplo de referência para trás)
      JMP LOOP ; volta para LOOP
END:   HLT     ; fim
    
```

} endereço de label

O operando imediato como o nome diz, recebe o valor na mesma linha do registrador, como no exemplo acima.

Já o endereço de label direciona os códigos para executar as operações imediatas.

✓

10) Explique de forma objetiva o que é um montador (assembler), destacando sua função no processo de tradução de programas e como ele lida com rótulos e símbolos durante a montagem.

O montador converte o assembly em código de máquina e utiliza a SYMTAB para padronizar e otimizar as linhas de código do programa.

✓