



**REVISÃO PARA A PROVA 01 – PROGRAMAÇÃO DE SISTEMAS**  
**CURSO DE CIÊNCIA DA COMPUTAÇÃO**  
**UNIVERSIDADE FRANCISCANA – UFN. 2025-02. Peso:2,0.**

**PROFESSOR:** André F. dos Santos.

**Nome do aluno:** Pedro Henrique de Brito Canabarro.

**Data:** 27/08/2025.

**Destaque em amarelo a alternativa correta nas questões de múltipla escolha. Preencha seu nome e data.**  
Questões de múltipla escolha (marque apenas uma alternativa).

**1) Qual é a função principal de um montador (assembler) no fluxo de construção de um programa?**

- a) Traduzir mnemônicos de Assembly para código-objeto e resolver rótulos/símbolos.
- b) Ligar múltiplos arquivos-objeto e bibliotecas para gerar o executável final.
- c) Otimizar código de alto nível (C/Java) e gerar Assembly automaticamente.
- d) Executar o binário e gerenciar a alocação de memória em tempo de execução.
- e) Interpretar os mnemônicos diretamente em tempo real, sem gerar arquivo.

**2) Sobre montadores de 1-pass (uma passagem) e 2-pass (duas passagens), assinale a correta.**

- a) No 1-pass, o montador usa placeholders/fixups (backpatch) para rótulos futuros.
- b) No 2-pass, o montador não constrói tabela de símbolos (SYMTAB).
- c) No 1-pass, rótulos à frente são sempre proibidos e nunca podem ser usados.
- d) No 2-pass, a geração de código ocorre antes do cálculo de endereços.
- e) No 1-pass e 2-pass o custo de tempo é idêntico por definição.

**3) O que é mantido em uma Tabela de Símbolos (SYMTAB) durante a montagem?**

- a) Relação de rótulos e seus endereços no código.
- b) Relação de mnemônicos, opcodes e tamanhos das instruções.
- c) Relação de variáveis e seus valores em tempo de execução.
- d) Lista de serviços de BIOS/DOS por número de interrupção.
- e) Lista de instruções inválidas detectadas pelo processador.

**4) Na mini-ISA (Instruction Set Architecture) didática, qual instrução abaixo usa um operando imediato (valor literal)?**

- a) JMP END
- b) MOV R0, #5
- c) JMP 10
- d) ADD R0, END
- e) HLT

**5) 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

Questões dissertativas (responda de forma objetiva).

**6) Explique, em 6–8 linhas, a diferença entre montadores de 1-pass e 2-pass aplicando ao caso do 'JMP END' quando 'END' aparece depois no código.**

1 pass: monta lendo uma vez, usa placeholders/backpatches para rótulos futuros.

2 pass: na 1ª passagem constrói a tabela de símbolos, na 2ª gera o código final.

No 1 pass, gera código com endereço temporário e corrige depois.

No 2 pass, só gera o código após já saber onde o END está.

**7) Considere o programa Assembly abaixo (Emu8086):**

```
.data
num1 db 9
num2 db 4
resultado db 0
msg db 'Resultado: $'
```

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

```
mov al, num1
sub al, num2
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
```

**Pergunta:** Qual será a saída exibida pelo programa na tela? **O resultado será 5.**

**8) 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

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

**Resposta:**

Operadores imediatos: Valor literal escrito no código: MOV R0, #5 e ADD R0, #3

Endereço de Label: refere-se a um rótulo de memória: JMP END ou ADD R0, LOOP

**9) Descreva o pipeline: código fonte → objeto → executável. Qual a diferença entre arquivo-objeto e executável?**

Pipeline de compilação: código fonte (.asm) → montador/compilador → objeto (.obj); Ligador → executável (.exe/.bin)

Diferença entre o objeto e o executável é que o objeto é binário intermediário, ainda precisa de ligação, e o executável é um binário pronto para rodar no SO.

**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.**

Ele resolve rótulos e símbolos (constrói a SYMTAB), gera código objeto relocável, permite que o linker crie o executável.