

Arquitetura e Organização de Computadores II

Lista 1

1. Como montadores e ligadores trabalham juntos?
2. Como o estudo da linguagem de montagem pode aumentar a sua compreensão dos sistemas operacionais?
3. Qual o significado de um relacionamento um-para-muitos ao comparar uma linguagem de alto nível para linguagem de máquina?
4. Explique o conceito de portabilidade, que se aplica as linguagens de programação.
5. A linguagem de montagem para processadores x86 é a mesma para os processadores VAX ou Motorola 68x00?
6. Dê um exemplo de uma aplicação de sistemas embarcados.
7. O que é um driver de dispositivo?
8. Você acha que a verificação de tipo em variáveis de ponteiro é mais forte (mais rigorosa) em linguagem de montagem ou em C e C++?
9. Nomeie dois tipos de aplicativos que seriam mais adequados à linguagem de montagem do que a uma linguagem de alto nível.
10. Por que uma linguagem de alto nível não é uma ferramenta ideal para escrever um programa que acessa diretamente uma determinada marca de impressora?
11. Por que uma linguagem de montagem geralmente não é usada para criar programas aplicativos de grande porte?
12. Traduza a seguinte expressão em C++ para a linguagem de montagem:
$$X = (Y * 4) + 3$$
13. Descreva o conceito de máquina virtual com suas próprias palavras.
14. Por que os programadores não escrevem programas aplicativos em linguagem de máquina?
15. (V/F): Quando um programa interpretado em linguagem L1 é executado, cada uma das suas instruções é decodificada e executada por um programa escrito em linguagem L0.

16. Explique a técnica de tradução quando se trata de linguagens em diferentes níveis de máquina virtual.
17. Que software permite que programas compilados em Java possam rodar em qualquer computador?
18. Explique o termo Bit Menos Significativo (LSB).
19. Explique o termo Bit Mais Significativo (MSB).
20. Qual é a representação decimal de cada um dos seguintes números binários inteiros sem sinal?
 - a. 11111000
 - b. 11001010
 - c. 11110000
 - d. 00110101
 - e. 10010110
 - f. 11001100
21. Qual é a representação decimal destes mesmos números considerando que eles são binários inteiros com sinal em complemento de 2?
22. Qual é a soma de cada par de números inteiros binários?
 - a. 00001111 + 00000010
 - b. 11010101 + 01101011
 - c. 00001111 + 00001111
 - d. 10101111 + 11011011
 - e. 10010111 + 11111111
 - f. 01110101 + 10101100
23. Quantos bytes estão contidos em cada um dos seguintes tipos de dados?
 - a. word
 - b. doubleword
 - c. quadword
24. Quantos bits estão contidos em cada um dos seguintes tipos de dados?
 - a. word
 - b. doubleword
 - c. quadword

25. Qual é o número mínimo de bits necessários para representar cada um dos seguintes números decimais inteiros sem sinal?
- 65
 - 256
 - 32768
 - 4095
 - 65534
 - 2134657
26. Qual é a representação hexadecimal de cada um dos seguintes números binários?
- 1100 1111 0101 0111
 - 0101 1100 1010 1101
 - 1001 0011 1110 1011
 - 0011 0101 1101 1010
 - 1100 1110 1010 0011
 - 1111 1110 1101 1011
27. Qual é a representação binária dos números hexadecimais abaixo?
- E5B6AED7
 - B697C7A1
 - 234B6D92
 - 0126F9D4
 - 6ACDFA95
 - F69BDC2A
28. Qual é a representação decimal sem sinal de cada inteiro hexadecimal?
- 3A
 - 1BF
 - 4096
 - 62
 - 1C9
 - 6A5B
29. Qual é a representação hexadecimal de 16 bits de cada número inteiro decimal com sinal?
- 26
 - 452
 - 32
 - 62
30. Os 16 bits representam números hexadecimais inteiros assinados. Converta para decimal.
- 7CAB
 - C123
 - 7F9B
 - 8230

31. Qual é a representação decimal dos seguintes números binários com sinal?
- 10110101
 - 00101010
 - 11110000
 - 10000000
 - 11001100
 - 10110111
32. Qual representação binária de 8 bits (em complemento de dois) de cada um dos seguintes decimais inteiros?
- 5
 - 36
 - 16
 - 72
 - 98
 - 26
33. Por que foi inventado Unicode?
34. Qual é o maior valor que se pode representar usando um número inteiro de 256 bits sem sinal?
35. Qual é o maior valor positivo que se pode representar usando um número inteiro de 256 bits com sinal em complemento de 2?

Os exercícios abaixo são relativos à linguagem Assembly da arquitetura IA-32

36. Mostrar o conteúdo da memória na área de dados definida pelas sentenças:

.data

valorbyte	BYTE	88h
valorword	WORD	4433h
valordword	DWORD	11226677h
valorfinal	?	

valorbyte: _____
 valorbyte+1: _____

Valorbyte+7: _____

37. Escrever o código que armazena os valores (tamanho1 e tamanho2) em tam1 e tam2.

```
.data
valorbyte    BYTE    88h
valoresvetor  BYTE    20 DUP    (0)
tamanho1 = $ - valorbyte
tamanho2 = $ - valoresvetor
tam1 DWORD    ?
tam2 DWORD    ?
.code
main PROC

.....

.....

.....

.....

exit
main ENDP
END main
```

38. Porque as instruções abaixo são inválidas?

```
.data
bVal BYTE 100
bVal2 BYTE ?
wVal WORD 2
dVal DWORD 5
```

```
.code                                RESP:
```

mov ds, 44	
mov eip,dVal	
mov bVal2,bVal	
mov esi,wVal	
mov 25,bVal	

39. Escrever o código que rearranja os valores de 3 doublewords no seguinte vetor:
arrayD DWORD 33, 11, 22, usando instruções MOV e XCHG.

```
.data
arrayD DWORD 11,22,33
.code
```

--

40. Escrever um código que soma os 3 bytes definidos pela sentença :
myBytes BYTE 80h, 66h, 0A5h, usando deslocamento direto, ou seja. myBytes,
myBytes +1, myBytes +2. Terminar o programa com o resultado em al.
41. Escrever um código que soma os 3 bytes definidos no exercício anterior, porém,
com o resultado final em ax, usando adição de 16 bits, instrução movzx para
carregar o valor da posição myBytes em ax, e instrução movzx para carregar o valor
de myBytes +1 e myBytes + 2 em bx. O resultado final é igual ao obtido no
exercício 5?