

### QUESTÕES OBJETIVAS

#### Questão 1.1

Os modelos de computadores são históricos e representam a abstração da arquitetura de computadores projetados em diferentes épocas da história e que deixaram ideias e conceitos explorados nas atuais arquiteturas dos computadores modernos. O modelo de von-Neumann determina que os programas junto com os dados devem ser armazenados \_\_\_\_\_.

Assinale a alternativa que completa corretamente a lacuna:

- a) no dispositivo de entrada
- b) no dispositivo de saída
- c) na unidade de lógica e aritmética (ULA)
- d) na memória
- e) na unidade de controle

#### RESOLUÇÃO

A resposta correta é: na memória.

#### Justificativa

O modelo de Von Neumann determina que o programa deve ser armazenado na memória (retomando a ideia de Turing). Isso é totalmente diferente da arquitetura dos primeiros computadores mecânicos, nos quais somente os dados eram armazenados na memória.

A memória é a parte do computador que armazena os dados e os programas durante o processamento. A unidade de lógica e aritmética (ULA) é o elemento do computador que realiza operações lógicas e matemáticas dos dados. Os dispositivos de entrada ou de saída recebem a entrada de dados e dos programas e/ou fornecem os resultados dos processamentos. Finalmente, a unidade de controle controla as operações de memória, a unidade de lógica e aritmética e os equipamentos de entrada e de saída.

#### Questão 1.2

Localizado na CPU, um registrador é uma pequena memória de alta velocidade. Em relação ao conceito de registradores, pode-se afirmar que:

- I. Registradores têm função específica.
- II. Registradores armazenam resultados de modo permanente e controlam informações
- III. O registrador do tipo contador de programa indica a próxima instrução a ser buscada para a execução.
- IV. O registrador de instrução mantém a instrução que está sendo executada.

- V. O registrador de dados mantém os dados de entrada, os resultados intermediários e os resultados finais.

Assinale a alternativa correta:

- a) V-V-V-V-F.
- b) F-V-V-V-F.
- c) V-F-V-V-F.
- d) V-F-V-V-V.
- e) F-F-F-V-F.

### RESOLUÇÃO

A resposta correta é: V-F-V-V-V.

### Justificativa

A alternativa II está incorreta, uma vez que registradores são memória de tamanho pequeno, rápido e volátil, armazenam resultados temporários e algum tipo de informação.

### Questão 1.3

Quando um processo é criado, o sistema operacional cria as páginas virtuais que o processo necessita. Sobre esse assunto, complete as lacunas:

\_\_\_\_\_ é uma técnica que usa a \_\_\_\_\_ como uma cache para armazenamento e resulta em tempo de acesso a essa memória de disco estatisticamente não muito pior que a \_\_\_\_\_.

\_\_\_\_\_ é um recurso de memória virtual que implementa a quebra de \_\_\_\_\_ em páginas com endereçamento virtual dividido em páginas virtuais.

Assinale a alternativa que completa corretamente as lacunas:

- a) memória secundária, memória virtual, memória principal, paginação, processo
- b) memória secundária, memória virtual, memória principal, processo, paginação
- c) memória principal, memória secundária, memória virtual, paginação, processo
- d) memória virtual, memória principal, memória secundária, processo, paginação
- e) memória virtual, memória secundária, memória principal, paginação, processo

### RESOLUÇÃO

A resposta correta é: memória virtual, memória secundária, memória principal, paginação, processo

### Justificativa

Memória principal e memória secundária são dispositivos que são parte do hardware do computador e, portanto, não podem ser técnicas. A memória principal é rápida e, portanto, não teria sentido usá-la como cache para tentar alcançar a velocidade da memória secundária, que é mais lenta que a cache a memória principal. Portanto, **memória virtual** é uma técnica que usa a **memória secundária** como uma cache para armazenamento e resulta em tempo de acesso a essa memória de disco estatisticamente não muito pior que a **memória principal**. **Paginação** é um recurso de memória

virtual que implementa a quebra de **processo** em páginas com endereçamento virtual dividido em páginas virtuais.

#### Questão 1.4

Em multiprogramação, a CPU compartilha o seu tempo entre os vários programas e os diferentes dispositivos periféricos que necessitam da sua atenção. O mecanismo de interrupção constitui a base para a implementação desse esquema de compartilhamento, sobre esse assunto escolha a alternativa correta:

- a) Para controlar dispositivos de entrada e saída de dados, a CPU continuamente monitora o status desses dispositivos como discos ou teclados.
- b) O mecanismo de interrupções permite a CPU execute sem sofrer interrupções quando há algo a ser feito.
- c) Interrupções são mecanismos usados para interromper o processamento normal da CPU.
- d) O manipulador de interrupções é um dispositivo físico da CPU que faz o tratamento das interrupções.
- e) O tratador de interrupção trabalha exclusivamente de maneira síncrona.

#### RESOLUÇÃO

A resposta correta é: Interrupções são mecanismos usados para interromper o processamento normal da CPU.

#### Justificativa

- (a) Os mecanismos de interrupção foram criados para evitar que a CPU fique continuamente monitorando o status dos periféricos.
- (b) O mecanismo de interrupções permite que o hardware "chame a atenção" da CPU quando há algo a ser feito.
- (d) O manipulador ou *handler* de interrupções é um programa que determina a natureza da interrupção e que realiza o tratamento adequado.
- (e) O tratador de interrupção é NA MAIORIA das vezes assíncrono (SEM comunicação entre o programa interrompido e o tratador), MAS existe interrupção síncrona, como o caso do *Trap*.

#### QUESTÕES DISSERTATIVAS

#### Questão 2

Um código em assembly é um código em linguagem de montagem. Descreva a função de cada linha do trecho de código a seguir:

9: ....  
10: MOVE                      RA,        [3]  
11: MOVE                      RB,        [4]  
12: MOVE                      RC 0  
13: ....

## RESOLUÇÃO

Na 10a linha, tem-se o opcode MOVE e os operandos RA e [3], que são respectivamente o registrador e o endereço de memória. Move-se o conteúdo do endereço de memória [3] para o registrador RA.

Na 11a linha, tem-se novamente o opcode MOVE e os operandos RB e [4], que são respectivamente, outro registrador e outro endereço de memória. Move-se o conteúdo do endereço de memória [4] para o registrador RB.

Finalmente, na 12a linha, move-se o valor zero para o registrador RC.

Portanto, observa-se que se pode mover um valor constante (zero) ou um conteúdo da memória num endereço (por exemplo, [3]).

### Rubricas | critérios de correção

Para as linhas 10 e 11, atribua 0,6 para cada linha e distribua da seguinte forma: 0,2 para opcode MOVE, 0,2 para descrição do operando registrado e 0,2 para identificação dos endereços de memória.

Para a linha 12, atribua 0,8 da seguinte forma 0,2 para opcode MOVE. 0,2 para o operando registrador e 0,4 para a constante literal zero.

## Questão 3

*typedef* e *structs* permitem compor, respectivamente, (i) novos tipos de dados a partir de tipos pré-existentes e (ii) estruturas com conjuntos de variáveis. O trecho inicial de um código em linguagem C apresenta algumas definições usando *typedef* e *struct*. Explique as definições das *structs* com *typedef* e de seus elementos desse trecho de código.

....

```
1: typedef struct moldura{
2:     int bv;
3:     char contentRAM;
4: } frame;
```

```
5: typedef struct pagina{
6:     int bv;
7:     char contentDISCO;
8:     frame *pv;
9: }page;
```

```
10: frame RAM[2];
11: page DISCO[20];
```

```
12: int main() {
```

```
.....
```

```
}
```

## RESOLUÇÃO

No trecho inicial do código em linguagem C, foram definidas duas structs usando typedef, de modo que elas passam a ser, respectivamente, os seguintes tipos de dados: frame e page.

Entre as linhas 1 e 4, a estrutura **moldura** foi definida com um campo inteiro identificado como **bv** e um character como **contentRAM**. Essa estrutura foi definida como o tipo **frame**.

Entre as linhas 5 e 9, a estrutura **página** foi definida com três elementos: um inteiro chamado **bv**, um character chamado de contentDISCO e um ponteiro pv para o tipo **frame**. Essa estrutura foi definida como o tipo **page**.

Finalmente, nas linhas 10 e 11 foram definidos dois vetores, sendo um de tamanho 2 composto por elementos do tipo frame e o outro composto por 20 elementos do tipo page.

### Rubricas | critérios de correção

Nessa questão, basta o aluno explicar das linhas 1 a 9, pois essas linhas definem as structs. Para a explicação de struct e typedef atribua 0,2 em cada estrutura somando 0,4, no caso de acerto. Para cada tipo inteiro ou character atribua 0,2 para cada campo somando 0,8, no caso de acerto. Para o ponteiro pv atribua 0,4. Finalmente, para cada tipo frame e page, atribua 0,2, somando 0,4 no caso de acerto.