

Questões do ENADE – Prova 1

- Assuntos: fundamentos de SO, processos e *threads*, comunicação interprocessos
- Questões:
 - 2005: 11, 53
 - 2008: 19
 - 2011: 18, 29
 - 2014: 20, 21
 - 2017: não caiu nenhum dos assuntos
 - 2021: 14, 17 (CC), 28 (TADS), 32 (TGTI)



1. As questões de 11 a 40, a seguir, distribuídas de acordo com o quadro abaixo, são específicas para os estudantes de cursos com perfil de Bacharelado em Sistemas de Informação. **Os demais estudantes deverão passar para a questão de número 41.**

PERFIL DO CURSO	NÚMERO DAS QUESTÕES	
	OBJETIVAS	DISCURSIVAS
Bacharelado em Sistemas de Informação	11 a 38	39 e 40

2. Você deve responder apenas às questões referentes ao perfil profissional do curso em que você está inscrito, de acordo com o estabelecido no cartão de informação do estudante.
3. Favor responder também ao questionário de percepção sobre a prova localizado no final deste caderno.

As questões de 11 a 40, a seguir, são específicas para os estudantes de cursos com perfil profissional de

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

QUESTÃO 11

Apesar de todo o desenvolvimento, a construção de computadores e processadores continua, basicamente, seguindo a arquitetura clássica de von Neumann. As exceções a essa regra encontram-se em computadores de propósitos específicos e nos desenvolvidos em centros de pesquisa. Assinale a opção em que estão corretamente apresentadas características da operação básica de um processador clássico.

- A** Instruções e dados estão em uma memória física única; um programa é constituído de uma sequência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa sequência e, quando é executada, passa-se, então, para a próxima instrução na sequência.
- B** Instruções e dados estão em memórias físicas distintas; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando o seu operando-destino necessita ser recalculado; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para o próximo operando a ser recalculado.
- C** Instruções e dados estão em uma memória física única; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando todos os seus operandos-fonte estiverem prontos e disponíveis; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para a instrução seguinte que tiver todos seus operandos disponíveis.
- D** Instruções e dados estão em memórias físicas distintas; um programa é constituído de um conjunto de instruções de máquina; uma instrução é lida da memória quando todos os seus operandos-fonte estiverem prontos e disponíveis; essa instrução é executada e o resultado é escrito no operando de destino, passando-se, então, para a instrução seguinte que estiver com todos os seus operandos disponíveis.
- E** Instruções e dados estão em memórias físicas distintas; um programa é constituído de uma sequência de instruções de máquina; uma instrução é lida da memória de acordo com a ordem dessa sequência e, quando é executada, passa-se, então, para a próxima instrução na sequência.

QUESTÃO 12

Um elemento imprescindível em um computador é o sistema de memória, componente que apresenta grande variedade de tipos, tecnologias e organizações. Com relação a esse assunto, julgue os itens seguintes.

- I Para endereçar um máximo de 2^E palavras distintas, uma memória semicondutora necessita de, no mínimo, E bits de endereço.
- II Em memórias secundárias constituídas por discos magnéticos, as palavras estão organizadas em blocos, e cada bloco possui um endereço único, com base na sua localização física no disco.
- III A tecnologia de memória dinâmica indica que o conteúdo dessa memória pode ser alterado (lido e escrito), ao contrário da tecnologia de memória estática, cujo conteúdo pode apenas ser lido, mas não pode ser alterado.

Assinale a opção correta.

- A** Apenas um item está certo.
- B** Apenas os itens I e II estão certos.
- C** Apenas os itens I e III estão certos.
- D** Apenas os itens II e III estão certos.
- E** Todos os itens estão certos.

QUESTÃO 13

Julgue os itens a seguir, acerca de algoritmos para ordenação.

- I O algoritmo de ordenação por inserção tem complexidade $O(n \times \log n)$.
- II Um algoritmo de ordenação é dito estável caso ele não altere a posição relativa de elementos de mesmo valor.
- III No algoritmo *quicksort*, a escolha do elemento pivô influencia o desempenho do algoritmo.
- IV O *bubble-sort* e o algoritmo de ordenação por inserção fazem, em média, o mesmo número de comparações.

Estão certos apenas os itens

- A** I e II.
- B** I e III.
- C** II e IV.
- D** I, III e IV.
- E** II, III e IV.

QUESTÃO 53

O problema do *buffer* limitado de tamanho N é um problema clássico de sincronização de processos: um grupo de processos utiliza um *buffer* de tamanho N para armazenar temporariamente itens produzidos; processos produtores produzem os itens, um a um, e os armazenam no *buffer*; processos consumidores retiram os itens do *buffer*, um a um, para processamento. O problema do *buffer* limitado de tamanho N pode ser resolvido com a utilização de semáforos, que são mecanismos de *software* para controle de concorrência entre processos. Duas operações são definidas para um semáforo s : $\text{wait}(s)$ e $\text{signal}(s)$.

Considere o problema do *buffer* limitado de tamanho N cujos pseudocódigos dos processos produtor e consumidor estão mostrados na tabela abaixo. Pode-se resolver esse problema com a utilização dos semáforos *mutex*, *cheio* e *vazio*, inicializados, respectivamente, com 1, 0 e N .

processo produtor	processo consumidor
produz item	comando_e comando_f
comando_a comando_b	retira do buffer
coloca no buffer	comando_g comando_h
comando_c comando_d	consome o item

A partir dessas informações, para que o problema do *buffer* limitado de tamanho N cujos pseudocódigos foram apresentados possa ser resolvido a partir do uso dos semáforos *mutex*, *cheio* e *vazio*, é necessário que comando_a, comando_b, comando_c, comando_d, comando_e, comando_f, comando_g e comando_h correspondam, respectivamente, às operações

- A $\text{wait}(\text{vazio})$, $\text{wait}(\text{mutex})$, $\text{signal}(\text{mutex})$, $\text{signal}(\text{cheio})$, $\text{wait}(\text{cheio})$, $\text{wait}(\text{mutex})$, $\text{signal}(\text{mutex})$ e $\text{signal}(\text{vazio})$.
- B $\text{wait}(\text{cheio})$, $\text{wait}(\text{mutex})$, $\text{signal}(\text{mutex})$, $\text{signal}(\text{vazio})$, $\text{wait}(\text{vazio})$, $\text{signal}(\text{mutex})$, $\text{signal}(\text{mutex})$ e $\text{wait}(\text{cheio})$.
- C $\text{wait}(\text{mutex})$, $\text{wait}(\text{vazio})$, $\text{signal}(\text{cheio})$, $\text{signal}(\text{mutex})$, $\text{wait}(\text{mutex})$, $\text{wait}(\text{vazio})$, $\text{signal}(\text{cheio})$ e $\text{signal}(\text{mutex})$.
- D $\text{wait}(\text{mutex})$, $\text{wait}(\text{vazio})$, $\text{signal}(\text{cheio})$, $\text{signal}(\text{mutex})$, $\text{wait}(\text{mutex})$, $\text{wait}(\text{cheio})$, $\text{signal}(\text{vazio})$ e $\text{signal}(\text{mutex})$.
- E $\text{wait}(\text{vazio})$, $\text{signal}(\text{mutex})$, $\text{signal}(\text{cheio})$, $\text{wait}(\text{mutex})$, $\text{wait}(\text{cheio})$, $\text{signal}(\text{mutex})$, $\text{signal}(\text{vazio})$ e $\text{signal}(\text{mutex})$.

QUESTÃO 54

Considere que, durante a análise de um problema de programação, tenha sido obtida a seguinte fórmula recursiva que descreve a solução para o problema.

$$C(i, j) = \begin{cases} 0 & \text{se } i = 0 \text{ ou } j = 0 \\ 1 + C(i-1, j-1) & \text{se } 0 < i \leq M, 0 < j \leq N \text{ e } i = j \\ \max\{C(i, j-1), C(i-1, j)\} & \text{se } 0 < i \leq M, 0 < j \leq N \text{ e } i \neq j \end{cases}$$

Qual a complexidade da solução encontrada?

- A $O(n \times \log n)$
- B $O(n^2)$
- C $O(n^2 \times \log n)$
- D $O(2^n)$
- E $O(n^3)$

RASCUNHO

QUESTÃO 19

Uma alternativa para o aumento de desempenho de sistemas computacionais é o uso de processadores com múltiplos núcleos, chamados *multicores*. Nesses sistemas, cada núcleo, normalmente, tem as funcionalidades completas de um processador, já sendo comuns, atualmente, configurações com 2, 4 ou mais núcleos. Com relação ao uso de processadores *multicores*, e sabendo que *threads* são estruturas de execução associadas a um processo, que compartilham suas áreas de código e dados, mas mantêm contextos independentes, analise as seguintes asserções.

Ao dividirem suas atividades em múltiplas *threads* que podem ser executadas paralelamente, aplicações podem se beneficiar mais efetivamente dos diversos núcleos dos processadores *multicores*

porque

o sistema operacional nos processadores *multicores* pode alocar os núcleos existentes para executar simultaneamente diversas seqüências de código, sobrepondo suas execuções e, normalmente, reduzindo o tempo de resposta das aplicações às quais estão associadas.

Acerca dessas asserções, assinale a opção correta.

- A As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- B As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira.
- C A primeira asserção é uma proposição verdadeira, e a segunda, uma proposição falsa.
- D A primeira asserção é uma proposição falsa, e a segunda, uma proposição verdadeira.
- E Tanto a primeira quanto a segunda asserções são proposições falsas.

QUESTÃO 20 – DISCURSIVA

Tabelas de dispersão (tabelas *hash*) armazenam elementos com base no valor absoluto de suas chaves e em técnicas de tratamento de colisões. As funções de dispersão transformam chaves em endereços-base da tabela, ao passo que o tratamento de colisões resolve conflitos em casos em que mais de uma chave é mapeada para um mesmo endereço-base da tabela.

Suponha que uma aplicação utilize uma tabela de dispersão com 23 endereços-base (índices de 0 a 22) e empregue $h(x) = x \bmod 23$ como função de dispersão, em que x representa a chave do elemento cujo endereço-base deseja-se computar. Inicialmente, essa tabela de dispersão encontra-se vazia. Em seguida, a aplicação solicita uma seqüência de inserções de elementos cujas chaves aparecem na seguinte ordem: 44, 46, 49, 70, 27, 71, 90, 97, 95.

Com relação à aplicação descrita, faça o que se pede a seguir.

- A Escreva, no espaço reservado, o conjunto das chaves envolvidas em colisões.

(valor: 4,0 pontos)

RASCUNHO – QUESTÃO 20 - A

- B Assuma que a tabela de dispersão trate colisões por meio de encadeamento exterior. Esboce a tabela de dispersão para mostrar seu conteúdo após a seqüência de inserções referida.

(valor: 6,0 pontos)

RASCUNHO – QUESTÃO 20 - B

QUESTÃO 18

Um vendedor de artigos de pesca obteve com um amigo o código executável (já compilado) de um programa que gerencia vendas e faz o controle de estoque, com o intuito de usá-lo em sua loja. Segundo o seu amigo, o referido programa foi compilado em seu sistema computacional pessoal (sistema A) e funciona corretamente. O vendedor constatou que o programa executável também funciona corretamente no sistema computacional de sua loja (sistema B). Considerando a situação relatada, analise as afirmações a seguir.

- I. Os computadores poderiam ter quantidades diferentes de núcleos (*cores*).
- II. As chamadas ao sistema (*system call*) do sistema operacional no sistema A devem ser compatíveis com as do sistema B.
- III. O conjunto de instruções do sistema A poderia ser diferente do conjunto de instruções do sistema B.
- IV. Se os registradores do sistema A forem de 64 *bits*, os registradores do sistema B poderiam ser de 32 *bits*.

É correto o que se afirma em

- A III, apenas.
- B I e II, apenas.
- C III e IV, apenas.
- D I, II e IV, apenas.
- E I, II, III e IV.

QUESTÃO 19

Uma equipe está realizando testes com base nos códigos-fonte de um sistema. Os testes envolvem a verificação de diversos componentes individualmente, bem como das interfaces entre os componentes.

No contexto apresentado, essa equipe está realizando testes em nível de

- A unidade.
- B aceitação.
- C sistema e aceitação.
- D integração e sistema.
- E unidade e integração.

QUESTÃO 20

Considere que G é um grafo qualquer e que V e E são os conjuntos de vértices e de arestas de G , respectivamente. Considere também que grau(v) é o grau de um vértice v pertencente ao conjunto V . Nesse contexto, analise as seguintes asserções.

Em G , a quantidade de vértices com grau ímpar é ímpar.

PORQUE

Para G , vale a identidade dada pela expressão

$$\sum_{v \in V} \text{grau}(v) = 2|E|$$

Acerca dessas asserções, assinale a opção correta.

- A As duas asserções são proposições verdadeiras, e a segunda é uma justificativa correta da primeira.
- B As duas asserções são proposições verdadeiras, mas a segunda não é uma justificativa correta da primeira.
- C A primeira asserção é uma proposição verdadeira, e a segunda uma proposição falsa.
- D A primeira asserção é uma proposição falsa, e a segunda uma proposição verdadeira.
- E Tanto a primeira quanto a segunda asserções são proposições falsas.

QUESTÃO 21

No desenvolvimento de um *software* que analisa bases de DNA, representadas pelas letras A, C, G, T, utilizou-se as estruturas de dados: pilha e fila. Considere que, se uma sequência representa uma pilha, o topo é o elemento mais à esquerda; e se uma sequência representa uma fila, a sua frente é o elemento mais à esquerda.

Analise o seguinte cenário: “a sequência inicial ficou armazenada na primeira estrutura de dados na seguinte ordem: (A,G,T,C,A,G,T,T). Cada elemento foi retirado da primeira estrutura de dados e inserido na segunda estrutura de dados, e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A). Finalmente, cada elemento foi retirado da segunda estrutura de dados e inserido na terceira estrutura de dados e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A)”.

Qual a única sequência de estruturas de dados apresentadas a seguir pode ter sido usada no cenário descrito acima?

- A Fila - Pilha - Fila.
- B Fila - Fila - Pilha.
- C Fila - Pilha - Pilha.
- D Pilha - Fila - Pilha.
- E Pilha - Pilha - Pilha.



QUESTÃO 29

Uma antiga empresa de desenvolvimento de *software* resolveu atualizar toda sua infraestrutura computacional adquirindo um sistema operacional multitarefa, processadores *multi-core* (múltiplos núcleos) e o uso de uma linguagem de programação com suporte a *threads*.

O sistema operacional multitarefa de um computador é capaz de executar vários processos (programas) em paralelo. Considerando esses processos implementados com mais de uma *thread* (*multi-threads*), analise as afirmações abaixo.

- I. Os ciclos de vida de processos e *threads* são idênticos.
- II. *Threads* de diferentes processos compartilham memória.
- III. Somente processadores *multi-core* são capazes de executar programas *multi-threads*.
- IV. Em sistemas operacionais multitarefa, *threads* podem migrar de um processo para outro.

É correto apenas o que se afirma em

- A** I.
- B** II.
- C** I e III.
- D** I e IV.
- E** II e IV.

ÁREA LIVRE

QUESTÃO 30

Suponha que se queira pesquisar a chave 287 em uma árvore binária de pesquisa com chaves entre 1 e 1 000. Durante uma pesquisa como essa, uma sequência de chaves é examinada. Cada sequência abaixo é uma suposta sequência de chaves examinadas em uma busca da chave 287.

- I. 7, 342, 199, 201, 310, 258, 287
- II. 110, 132, 133, 156, 289, 288, 287
- III. 252, 266, 271, 294, 295, 289, 287
- IV. 715, 112, 530, 249, 406, 234, 287

É válido apenas o que se apresenta em

- A** I.
- B** III.
- C** I e II.
- D** II e IV.
- E** III e IV.

ÁREA LIVRE

QUESTÃO 19

O algoritmo de traçado de raios (*ray-tracing*) é considerado um marco no desenvolvimento de técnicas de computação gráfica para a geração de imagens realistas.



Imagem 1

Disponível em: <<http://www.wikipedia.org>>. Acesso em: 18 jun. 2014.



Imagem 2

Disponível em: <<http://www.colegioweb.com.br>>. Acesso em: 18 jun. 2014.



Imagem 3

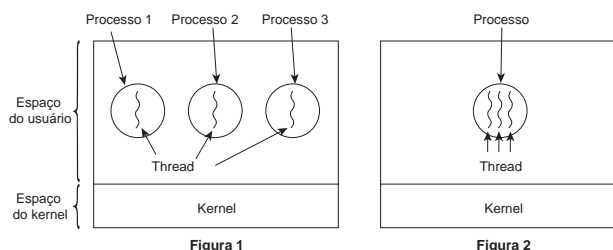
Disponível em: <<http://www.cgw.com>>. Acesso em: 18 jun. 2014.

A partir da análise das imagens apresentadas, conclui-se que a técnica de traçado de raios foi utilizada para a geração

- A** apenas da imagem 1.
- B** apenas da imagem 2.
- C** apenas das imagens 1 e 3.
- D** apenas das imagens 2 e 3.
- E** das imagens 1, 2 e 3.

QUESTÃO 20

Um processo tem um ou mais fluxos de execução, normalmente denominados apenas por *threads*.



TANENBAUM, A. D. **Sistemas operacionais modernos**. 3. ed. São Paulo: Pearson Prentice Hall, 2010 (adaptado).

A partir das figuras 1 e 2 apresentadas, avalie as afirmações a seguir.

- I. Tanto na figura 1 quanto na figura 2, existem três *threads* que utilizam o mesmo espaço de endereçamento.
- II. Tanto na figura 1 quanto na figura 2, existem três *threads* que utilizam três espaços de endereçamento distintos.
- III. Na figura 2, existe um processo com um único espaço de endereçamento e três *threads* de controle.
- IV. Na figura 1, existem três processos tradicionais, cada qual tem seu espaço de endereçamento e uma única *thread* de controle.
- V. As *threads* permitem que várias execuções ocorram no mesmo ambiente de processo de forma independente uma das outras.

É correto apenas o que se afirma em

- A** I, II e III.
- B** I, II e IV.
- C** I, III e V.
- D** II, IV e V.
- E** III, IV e V.

ÁREA LIVRE



QUESTÃO 21

O fragmento de código a seguir, escrito em Java, descreve duas implementações diferentes para um *lock*. Ambas possuem um método denominado *acquire* e um método denominado *release*.

<pre>class LockA { private int turn = 0 public void acquire(int tid) { while (turn == (1 - tid)); } public void release(int tid) { turn = (1 - tid); } }</pre>	<pre>class LockB { public void acquire() { disableInterrupts(); } public void release() { enableInterrupts(); } }</pre>
--	---

Considera-se que:

- as duas implementações de *lock* são utilizadas por aplicações com, no máximo, duas *threads*;
- uma aplicação que utilizar qualquer uma destas implementações invocará o método *acquire* antes de entrar em sua seção crítica e o método *release* após deixar a seção crítica;
- tanto o método *acquire* quanto o método *release* são operações atômicas nas duas implementações de *lock*;
- para a implementação que requer um *tid* (*thread id*), assume-se que ele sempre será 0 ou 1;
- os métodos *disableInterrupts* e *enableInterrupts* são utilizados para desabilitar e habilitar respectivamente as interrupções do processador onde o código for executado. O código desses dois métodos foi desenvolvido para ser utilizado em uma máquina com um ou dois processadores.

A partir das informações apresentadas, avalie as afirmações a seguir.

- I. A implementação de *LockA* garante progresso.
- II. A implementação de *LockB* garante progresso.
- III. A implementação de *LockA* garante exclusão mútua.
- IV. A implementação de *LockB* garante exclusão mútua.

É correto apenas o que se afirma em

- A** I e II.
- B** II e III.
- C** III e IV.
- D** I, II e IV.
- E** I, III e IV.



QUESTÃO 14

O primeiro computador criado foi o ENIAC (*Electronic Numerical Integrator And Computer*), desenvolvido por Eckert e Mauchly na Universidade da Pennsylvania, Estados Unidos. O projeto iniciou-se em 1943, financiado pelo governo americano. O período era da Segunda Guerra Mundial e o objetivo era poder calcular de forma mais ágil as melhores trajetórias para transporte de armas e mantimentos em meio aos exércitos inimigos. Esse é o tipo de cálculo que pequenos aparelhos celulares fazem hoje para encontrar rotas nas cidades por meio de GPS (*Global Positioning System*) e análise de mapa. O projeto só foi concluído em 1946, tarde demais para ser utilizado para a Segunda Guerra, mas foi bastante utilizado até 1955.

Muitos projetos surgiram depois do ENIAC, mas eles eram barrados por algumas dificuldades e limitações, como por exemplo, o fato de não serem programados e trabalharem com números decimais. O problema de trabalhar com decimais é que cada algarismo armazenado possui 10 estados possíveis, representando os números de 0 a 9. Dentro de um sistema eletrônico, isso é complicado por que a carga de cada dispositivo, seja transistor, seja válvula, deveria ser medida para se verificar que número ela estava representando. Os erros eram muito frequentes. Bastava que uma válvula estivesse fora da temperatura ideal para que os resultados das operações comesçassem a sair errado. Von Neumann recomendou, então, que, em sua arquitetura, os dados e instruções passassem a ser armazenados em código binário, facilitando a análise dos mesmos e reduzindo a quantidade de erros.

BRITO, A. V. *Introdução a Arquitetura de Computadores*. UFPB Virtual, 2020. Disponível em: <http://producao.virtual.ufpb.br/>. Acesso em: 05 maio 2020 (adaptado).

Acerca da arquitetura de Von Neumann, avalie as asserções a seguir e a relação proposta entre elas.

- I. Embora as arquiteturas de computadores tenham evoluído muito do ENIAC aos modernos notebooks de hoje, a arquitetura de Von Neumann, conceito da década de 1950, tem se mantido até os dias atuais.

PORQUE

- II. A arquitetura de Von Neumann permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada; essa técnica é utilizada para acelerar a velocidade de operação da CPU, uma vez que a próxima instrução a ser executada está normalmente armazenada nos registradores da CPU e não precisa ser buscada da memória principal, que é muito mais lenta.

A respeito dessas asserções, assinale a opção correta.

- A** As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
B As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
C A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
D A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
E As asserções I e II são proposições falsas.

Área livre

QUESTÃO 17

Durante parte do tempo, um processo está ocupado realizando computações internas e outras coisas que não levam a condições de corrida. No entanto, às vezes, um processo tem de acessar uma memória compartilhada ou arquivos, ou realizar outras tarefas críticas que podem levar a corridas. Essa parte do programa onde a memória compartilhada é acessada é chamada de **região crítica** ou **seção crítica**. Se conseguíssemos arranjar as coisas de maneira que dois processos jamais estivessem em suas regiões críticas ao mesmo tempo, poderíamos evitar as corridas. Embora essa exigência evite as condições de corrida, ela não é suficiente para garantir que processos em paralelo cooperem de modo correto e eficiente usando dados compartilhados. Precisamos que quatro condições se mantenham para chegar a uma boa solução.

1. Dois processos jamais podem simultaneamente estar dentro de suas regiões críticas.
2. Nenhuma suposição pode ser feita a respeito de velocidades ou de número de CPUs.
3. Nenhum processo executando fora de sua região crítica pode bloquear qualquer processo.
4. Nenhum processo deve ser obrigado a esperar eternamente para entrar em sua região crítica.

Em um sentido abstrato, o comportamento que queremos é mostrado na figura a seguir.

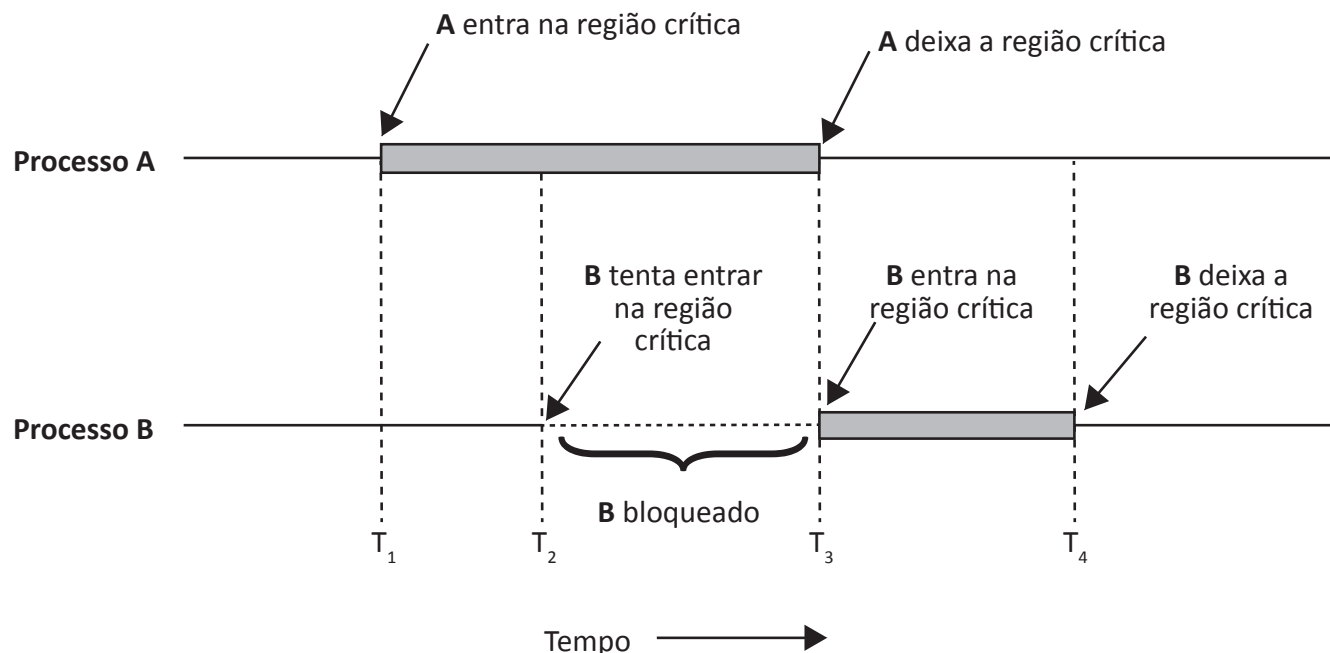


Figura - Exclusão mútua usando regiões críticas

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 4. ed. Versão para Biblioteca Virtual Pearson. São Paulo: Pearson Education do Brasil, p. 83, 2016 (adaptado).

Considerando o texto e a figura apresentados, avalie as asserções a seguir e a relação proposta entre elas.

- I. Em algumas situações, a exclusão mútua pode ser obtida por meio da desabilitação da interrupção controlada pelo Sistema Operacional, não sendo permitido que o seu controle seja feito pelo usuário.

PORQUE

- II. A desabilitação da interrupção é uma técnica que pode impedir que o processador que está executando um processo em sua região crítica seja interrompido para executar outro código, sendo mais eficiente em sistemas de multiprocessadores devido a quantidade de processos concorrentes.

A respeito dessas asserções, assinale a opção correta.

- A** As asserções I e II são proposições verdadeiras, e a II é uma justificativa correta da I.
- B** As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa correta da I.
- C** A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- D** A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- E** As asserções I e II são proposições falsas.

QUESTÃO 18

As técnicas de aprendizado de máquinas empregam um princípio de inferência denominado indução, no qual é possível obter conclusões genéricas a partir de um conjunto particular de exemplos. Estas técnicas de aprendizados indutivos podem ser divididas em dois principais tipos: os supervisionados e os não supervisionados. No aprendizado supervisionado é fornecida uma referência do objetivo a ser alcançado, isto é, um treinamento com o conhecimento do ambiente. Diferentemente do aprendizado supervisionado, o não supervisionado não utiliza referências, ou seja, não ocorre um treinamento com o conhecimento do ambiente.

PELLUCCI P. R. S. *et al.* Utilização de técnicas de aprendizado de máquina no reconhecimento de entidades nomeadas no português. Belo Horizonte. **E-xacta**, v. 4, n. 1, p. 73-81, 2011 (adaptado).

Considerando as informações do texto, avalie as afirmações a seguir.

- I. A regressão linear é um exemplo de modelo baseado no aprendizado supervisionado.
- II. A diferença entre a saída desejada e a saída gerada é o valor do erro de um aprendizado não supervisionado.
- III. O aprendizado não supervisionado é mais utilizado quando o entendimento dos dados é feito por meio de reconhecimento de padrões.
- IV. O aprendizado supervisionado é capaz de tomar decisões precisas ao receber novos dados a partir de um treinamento com dados conhecidos.

É correto apenas o que se afirma em

- A** I e III.
- B** II e III.
- C** II e IV.
- D** I, II e IV.
- E** I, III e IV.



* R 2 8 2 0 2 1 3 6 *

QUESTÃO 28

O modo clássico de encarar um sistema operacional é como um gerenciador de recursos. Desse ponto de vista, o sistema operacional é responsável pelo hardware do sistema. Nesse papel, ele recebe solicitações de acesso a recursos por parte das aplicações e concede ou nega tais acessos. Ao conceder solicitações de alocação, ele deve dispor com cuidado os recursos, de modo que os programas não interfiram uns nos outros. Por exemplo, é uma péssima ideia permitir que os programas tenham acesso sem restrição à memória uns dos outros. Se um programa com defeito (ou malicioso) escreve no espaço de memória do outro programa, o segundo programa travará, na melhor das hipóteses, ou produzirá resultados incorretos, na pior das hipóteses. Ou ainda, se o programa ofensivo modificar a memória do sistema operacional poderá afetar o comportamento de todo o sistema.

STUART, B. L. **Princípios de Sistemas Operacionais**: projetos e aplicações. São Paulo: Cengage Learning, 2011 (adaptado).

Considerando que o texto alerta para a possibilidade de um programa interferir no outro, a atividade que um sistema operacional garante essa proteção é

- A** o programa antivírus.
- B** o gerenciamento de memória.
- C** o gerenciamento de arquivos.
- D** o gerenciamento de processos.
- E** o gerenciamento de entrada e saída.

QUESTÃO 29

Um algoritmo é qualquer procedimento computacional bem definido que toma algum valor ou conjunto de valores como entrada e produz algum valor ou conjunto de valores como saída.

CORMEN, T. H. *et al.* **Algoritmos Teoria e Prática**. Rio de Janeiro: Elsevier, 2002 (adaptado).

Considere o algoritmo a seguir.

Algoritmo Calcular

```
var
  mat [1..3][1..5] de inteiro =
    {{1, 2, -1, 2, 3}, {1, -3, 4, 2, 0}, {-3, 5, 2, 3, 4}}
  sl[1..3] de inteiro = {0, 0, 0}
  x, i, j : inteiro
  x <- 0
início
  para i <- 1 até 3 faça
    para j <- 1 até 5 faça
      sl[i] <- sl[i] + mat[i][j]
    fimpara
    x <- x + sl[i]
  fimpara
  imprima x
fim
```

No fim da execução do código apresentado, será exibido o valor

- A** 4.
- B** 7.
- C** 11.
- D** 22.
- E** 30.



* R 2 9 2 0 2 1 3 6 *

QUESTÃO 32

Um empreendedor, com o auxílio de um tecnólogo em gestão da tecnologia da informação, precisa definir um sistema operacional que ofereça suporte adequado ao aplicativo inovador a ser desenvolvido, que alavancará os negócios da *startup*. O objetivo é atingir o máximo de usuários possíveis e conseguir alto desempenho de processamento em dispositivos móveis. Ao pesquisar na internet, encontrou a seguinte notícia de um fabricante:

“Entre as novidades, o sistema operacional ABC para dispositivos móveis terá, entre outros recursos, nova página inicial e nova tela de bloqueio, que é ligada assim que o aparelho é suspenso.

Houve mudança, também, no *app* de fotos, que ficou mais inteligente, sendo capaz de reconhecer rostos e de mostrar imagens em um mapa. No *app* de mensagens passou a ser possível enviar textos escritos à mão e desenhos animados.

A empresa abriu o seu assistente de linguagem natural para desenvolvedores externos. Assim, será possível solicitar carro nos aplicativos mais conhecidos, publicar foto nos *apps* mais populares, fazer chamada no *Skype* por comando de voz e enviar mensagens usando *apps* como WeChat ou WhatsApp. Também será possível fazer buscas por fotos, documentos, músicas e outros arquivos por comando de voz.”

Disponível em: <http://veja.abril.com.br/tecnologia>. Acesso em: 3 jul. 2017 (adaptado).

Como indica a notícia, o ABC possibilitará novas oportunidades de desenvolvimento de aplicativos com interfaces mais naturais e imediatas, mostrando-se ideal para os negócios da *startup*. Analisando as características técnicas, o tecnólogo afirmou, corretamente, que o sistema operacional ABC

- Ⓐ teve suas funções básicas ampliadas em seu *kernel*, possibilitando uma maior capacidade de processamento dos aplicativos independente da arquitetura do dispositivo.
- Ⓑ teve sua capacidade natural ampliada em função das mudanças no *app* de fotos, com isto o seu *kernel* passou a executar menos processos tornando o dispositivo mais rápido.
- Ⓒ sofreu alteração na camada de apresentação, sua nova página inicial é tratada por um ou mais processos e seu *kernel* continua sendo responsável por escalonar os processos dos aplicativos.
- Ⓓ sofreu mudanças significativas em seu *kernel*, que passou a residir em um dispositivo fora da memória, com isso, pode haver mais processos em execução, deixando o dispositivo mais rápido.
- Ⓔ teve as funções do *kernel* reduzidas, assim os processos relativos à gerência de memória não são mais tratados pelo módulo de escalonamento, o que deixa o desempenho do dispositivo muito melhor.

Área livre