

Iniciado em	quinta-feira, 27 jun. 2024, 15:56
Estado	Finalizada
Concluída em	quinta-feira, 27 jun. 2024, 16:12
Tempo empregado	15 minutos 26 segundos
Avaliar	Ainda não avaliado

Questão 1

Correto

Atingiu 1,00 de 1,00

Em um sistema operacional, qual é a principal função de um lock?

- ☒ Garantir que apenas um processo possa executar uma seção crítica de cada vez. ✓
- ☐ Permitir que vários processos acessem uma seção crítica simultaneamente.
- ☐ Nenhuma das opções acima.
- ☐ Bloquear todos os processos que tentam acessar uma seção crítica.

Sua resposta está correta.

A resposta correta é: Garantir que apenas um processo possa executar uma seção crítica de cada vez.

Questão 2

Correto

Atingiu 1,00 de 1,00

Qual das seguintes opções melhor descreve uma condição de corrida em sistemas operacionais?

- ☐ É uma situação em que um processo termina sua execução antes de outros processos.
- ☒ É uma situação em que vários processos acessam e manipulam uma variável compartilhada simultaneamente. ✓
- ☐ É uma situação em que um processo continua a executar em um loop infinito.
- ☐ É uma situação em que um processo nunca acessa o recurso compartilhado está sendo usado por outro processo.

Sua resposta está correta.

A resposta correta é: É uma situação em que vários processos acessam e manipulam uma variável compartilhada simultaneamente.

Questão 3

Completo

Vale 1,00 ponto(s).

O que é um Impasse (Deadlock) em sincronização de processos concorrentes?

Explique com as tuas palavras! Não copie trechos de livros, web ou app de IA!

É quando dois ou mais processos ficam simultaneamente dependendo de um outro para que ele então execute, de modo que fecha-se um ciclo no qual todos ficam eternamente aguardando e nenhum é concluído.

Um deadlock ocorre quando dois ou mais processos ficam impedidos de serem executados devido a uma situação de espera circular.

Por exemplo, se P1 está esperando P2, que está esperando P3, e P3 está esperando P1, temos um deadlock.

Primitivas como semáforos e mutex (locks) podem ser usadas para evitar deadlocks e garantir a exclusão mútua.

Questão 4

Correto

Atingiu 1,00 de 1,00

Considere dois processos, P1 e P2, que compartilham um semáforo, S.

P1 executa a operação wait(S) e P2 executa a operação signal(S).

Qual das seguintes afirmações é verdadeira?

- ☒ Se P1 executa wait(S) primeiro, P1 será imediatamente desbloqueado após P2 executar signal(S). ✓
- ☐ Se P1 executa wait(S) primeiro, P2 será bloqueado até que P1 libere o semáforo.
- ☐ Nenhuma das opções acima.
- ☐ Se P1 executa wait(S) primeiro, P2 não poderá executar signal(S) até que P1 tenha liberado o semáforo.

Sua resposta está correta.

A resposta correta é: Se P1 executa wait(S) primeiro, P1 será imediatamente desbloqueado após P2 executar signal(S).

Questão 5

Correto

Atingiu 1,00 de 1,00

No contexto de sistemas operacionais, o que é abandono (starvation)?

- ☐ É uma situação em que um processo é incapaz de avançar porque os recursos necessários ainda não foram criados.
- ☐ É uma situação em que um processo é incapaz de avançar porque está esperando indefinidamente para ser escalonado pelo SO.
- ☐ É uma situação em que um processo termina abruptamente.
- ☒ É uma situação em que um processo é incapaz de avançar porque está esperando indefinidamente por um recurso ✓
que está continuamente sendo concedido a outros processos.

Sua resposta está correta.

A resposta correta é: É uma situação em que um processo é incapaz de avançar porque está esperando indefinidamente por um recurso que está continuamente sendo concedido a outros processos.

Questão 6

Incorreto

Atingiu 0,00 de 1,00

No contexto de exclusão mútua, um processo precisa acessar uma região crítica sem ser interrompido por outros processos. Qual afirmação a seguir está incorreta?

- ☐ Quando um processo precisa entrar em uma região crítica, ele permanece bloqueado se outro processo já estiver nessa região.
- ☐ Dois processos nunca podem estar na mesma região crítica.
- ☒ Um processo fora da região crítica não pode impedir outro processo de acessá-la. ✗
- ☐ Após um processo deixar a região crítica, outro processo deve aguardar um determinado tempo para acessá-la.

Sua resposta está incorreta.

A resposta correta é: Após um processo deixar a região crítica, outro processo deve aguardar um determinado tempo para acessá-la.

Questão 7

Correto

Atingiu 1,00 de 1,00

Existem maneiras de controlar a sincronização de execução de processos, como por exemplo, quando um recurso único do sistema operacional necessita ser acessado por vários processos. Uma maneira clássica se dá pelo uso de semáforos, ou quando não é necessário efetuar contagem, uma versão simplificada, chamada de:

- ☐ Block
- ☒ Lock ✓
- ☐ Fork
- ☐ Sync

Sua resposta está correta.

A resposta correta é: Lock

Questão 8

Completo

Vale 1,00 ponto(s).

Qual o problema com a solução que desabilita as interrupções para implementar a exclusão mútua?

Explique com as tuas palavras! Não copie trechos de livros, web ou app de IA!

O problema é que daí podem surgir erros inesperados, deadlocks ou mesmo deixar que processos não importantes monopolizem o processamento do sistema.

Questão 9

Incorreto

Atingiu 0,00 de 1,00

Qual é a principal diferença entre um semáforo e um lock (mutex)?

- ☒ Um semáforo pode ser usado para controlar o acesso a múltiplos recursos idênticos, enquanto um lock é usado para controlar o acesso a um único recurso. ✗
- ☐ Um semáforo é uma variável inteira, enquanto um lock é uma variável booleana.
- ☐ Um semáforo pode ser usado para resolver problemas de sincronização de processos, enquanto um lock não pode.
- ☐ As opções B e C.

Sua resposta está incorreta.

A resposta correta é: Um semáforo é uma variável inteira, enquanto um lock é uma variável booleana.

Questão 10

Completo

Vale 1,00 ponto(s).

No contexto de sincronização de processos, a implementação de alguns mecanismos como a instrução TSL (Test and Set Lock) são baseados em "espera ocupada".

O que é espera ocupada e qual o seu problema?

Explique com as tuas palavras! Não copie trechos de livros, web ou app de IA!

É quando um processo fica constantemente verificando se um recurso está disponível antes de acessá-lo, o que consome desnecessariamente processamento, não é eficiente.