

Iniciado em Tuesday, 9 Nov 2021, 14:47

Estado Finalizada

Concluída em Tuesday, 9 Nov 2021, 15:36

Tempo empregado 49 minutos 29 segundos

Notas 16,00/17,00

Avaliar 9,41 de um máximo de 10,00(94%)

Questão 1

Correto

Atingiu 1,00 de 1,00

Sejam as duas threads abaixo executadas concorrentemente e compartilhando as variáveis A e B.

Thread 0	Thread 1
<code>A = 0;</code>	<code>B = 0;</code>
<code>...</code>	<code>...</code>
<code>A = 1;</code>	<code>B = 1;</code>
<code>...</code>	<code>...</code>
<code>if (B == 0)</code>	<code>if (A == 0)</code>
<code> P1 ();</code>	<code> P2 ();</code>

Com relação à situação acima, julgue os itens seguintes.

- I. As funções P1() e P2() nunca serão executadas simultaneamente.
- II. As funções P1() e P2() poderão não ser executadas.
- III. Uma das funções (P1() ou P2()) necessariamente será executada.
- IV. Apenas P1() ou apenas P2() será executada.

Estão corretas as assertivas:

Escolha uma opção:

- ☒ a. I e II. ✓
- ☐ b. I, II, III e IV.
- ☐ c. I e III.
- ☐ d. II e IV.
- ☐ e. III e IV.

Sua resposta está correta.

A resposta correta é: I e II.

Questão 2

Correto

Atingiu 1,00 de
1,00

Com relação a processos e threads, analise as seguintes proposições e, a seguir, marque a alternativa CORRETA.

I. Um processo é constituído de código executável, dados referentes ao código, pilha de execução, valor do contador de programa, valor do apontador de pilha, valores dos registradores do hardware e um conjunto de outras informações necessárias à execução do programa.

II. O bloco de controle de processo (descriptor de processo) mantém todas as informações que o sistema operacional necessita para gerenciar o processo, tais como identificador do processo, estado do processo, prioridade de escalonamento, ponteiro para o processo pai, entre outras.

III. Processo é uma unidade básica de utilização de CPU, que possui como benefício o compartilhamento da seção de código, seção de dados, arquivos abertos, entre outras informações, com outros processos.

IV. Threads, na API POSIX, podem ser criadas através da função `pthread_create()`.

Escolha uma opção:

- ☒ a. Somente as proposições I, II e IV são corretas. ✓
- ☐ b. Somente as proposições I e IV são corretas.
- ☐ c. Somente as proposições I, II e III são corretas.
- ☐ d. Somente as proposições III e IV são corretas.

Sua resposta está correta.

A resposta correta é: Somente as proposições I, II e IV são corretas.

Questão 3

Correto

Atingiu 1,00 de 1,00

Sobre estados de processos em Sistemas Operacionais, analise as assertivas e assinale a alternativa que aponta a(s) correta(s).

I. Um processo, em um sistema multiprogramável (multitarefa), não é executado todo o tempo pelo processador. Durante sua existência, ele passa por uma série de estados.

II. Execução (running): um processo é classificado como *running* quando aguarda uma oportunidade para executar, ou seja, esperando que o sistema operacional aloque a CPU para sua execução.

III. Pronto (ready): um processo é dito neste estado quando está sendo processado pela CPU. Em sistemas com apenas um processador, somente um processo pode estar pronto em um dado instante de tempo.

IV. Bloqueado (waiting): neste estado um processo aguarda algum evento externo ou por algum recurso para poder prosseguir seu processamento. Como exemplo, podemos citar o término de uma operação de entrada/saída ou a espera de uma determinada data e/ou hora para poder continuar sua execução.

Escolha uma opção:

- ☐ a. Apenas II, III e IV.
- ☐ b. Apenas I.
- ☐ c. I, II, III e IV.
- ☒ d. Apenas I e IV. ✓
- ☐ e. Apenas I e III.

Sua resposta está correta.

A resposta correta é: Apenas I e IV.

Questão 4

Correto

Atingiu 1,00 de 1,00

O gerenciamento de processos em sistemas modernos é feito, quase sempre, com o uso de preempção de processos através de técnicas de compartilhamento de tempo. O que a introdução de processadores com vários núcleos (cores) altera nesse gerenciamento?

Escolha uma opção:

- ☐ a. Torna possível o uso do algoritmo de escalonamento shortest job first (SJF).
- ☐ b. Torna possível o uso de threads para a execução de processos concorrentes.
- ☐ c. Torna possível a concorrência efetiva de processos paralelos.
- ☐ d. Torna possível separar os demais mecanismos de gerenciamento do sistema operacional do gerenciamento de processos.
- ☒ e. Torna possível a paralelização efetiva de processos concorrentes. ✓

Sua resposta está correta.

A resposta correta é: Torna possível a paralelização efetiva de processos concorrentes.


Questão 5

Correto

Atingiu 1,00 de
1,00

No navegador Firefox, cada nova aba representa uma nova *thread*; no Chrome, cada nova aba representa um novo processo. Os projetistas do Firefox e do Chrome adotaram diferentes decisões porque:

Escolha uma opção:

- ☐ a. Os projetistas do Firefox conceberam o navegador para executar, prioritariamente, em ambiente Linux, que suporta apenas *multithreading*; os projetistas do Chrome conceberam esses navegadores para executar, prioritariamente, em ambiente Windows, que suporta multiprocessamento e *multithreading*.
- ☒ b. Os projetistas do Firefox consideraram o menor consumo de recursos da máquina mais importante; os projetistas do Chrome priorizaram a robustez do navegador: se uma aba “pendurar”, não é necessário “matar” todo o navegador. 
- ☐ c. Os projetistas do Firefox consideraram a robustez mais importante do que a economia de recursos da máquina; os projetistas do Chrome consideraram mais importante a implementação de um navegador que usasse poucos recursos (memória, CPU) da máquina.
- ☐ d. Os projetistas do Firefox consideraram importante que cada aba tivesse a sua área de dados privada, compartilhando o mesmo código; os projetistas do Chrome preferiram que as abas compartilhassem a sua área de memória de uma maneira simples.

Sua resposta está correta.

A resposta correta é: Os projetistas do Firefox consideraram o menor consumo de recursos da máquina mais importante; os projetistas do Chrome priorizaram a robustez do navegador: se uma aba “pendurar”, não é necessário “matar” todo o navegador.

Questão 6

Correto

Atingiu 1,00 de 1,00

Sobre exclusão mútua é correto afirmar que:

Escolha uma opção:

- ☐ a. É uma técnica usada em programação concorrente para evitar que dois ou mais processos tenham acesso simultaneamente a múltiplos recursos privados.
- ☐ b. É uma técnica usada em programação concorrente para resolver *deadlock*.
- ☐ c. É uma técnica usada em programação concorrente para evitar *deadlock*.
- ☒ d. É uma técnica usada em programação concorrente para evitar que duas ou mais threads tenham acesso simultaneamente a um recurso compartilhado (seção crítica). ✓
- ☐ e. É uma técnica usada em programação não concorrente para evitar que duas ou mais threads tenham acesso simultaneamente a um recurso compartilhado (seção crítica).

Sua resposta está correta.

A resposta correta é: É uma técnica usada em programação concorrente para evitar que duas ou mais threads tenham acesso simultaneamente a um recurso compartilhado (seção crítica).

Questão 7

Correto

Atingiu 1,00 de 1,00

A situação na qual vários processos manipulam um mesmo conjunto de dados concorrentemente e o resultado depende da ordem na qual os acessos são feitos é denominada:

Escolha uma opção:

- ☐ a. espera ocupada (*busy waiting*).
- ☐ b. exclusão mútua.
- ☒ c. condição de corrida (*race condition*). ✓
- ☐ d. *deadlock*.
- ☐ e. região crítica.

Sua resposta está correta.

A resposta correta é: condição de corrida (*race condition*).

Questão 8

Incorreto

Atingiu 0,00 de
1,00

Um programa Java chamado CorridaDeSapos que cria várias *threads* foi disparado a partir do bash no Linux. Durante sua execução, a chamada do comando `ps -eLf` forneceu a seguinte saída:

UID	PID	PPID	LWP	C	NLWP	STIME	TTY	TIME	CMD
marcio	8094	2454	8094	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8095	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8096	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8097	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8098	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8099	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8100	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8101	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8102	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8103	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8104	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8105	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8106	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8107	0	19	15:20	pts/2	00:00:00	java
									CorridaDeSapos
marcio	8094	2454	8108	75	19	15:20	pts/2	00:05:06	java
									CorridaDeSapos
marcio	8094	2454	8109	76	19	15:20	pts/2	00:05:09	java
									CorridaDeSapos
marcio	8094	2454	8110	75	19	15:20	pts/2	00:05:07	java
									CorridaDeSapos
marcio	8094	2454	8111	76	19	15:20	pts/2	00:05:08	java
									CorridaDeSapos
marcio	8094	2454	8112	76	19	15:20	pts/2	00:05:08	java
									CorridaDeSapos
marcio	8142	2510	8142	0	1	15:27	pts/9	00:00:00	ps -eLf

A coluna "LWP" fornece o identificador da *thread* e a coluna "NLWP" fornece o número total de threads de um processo. Isso mostra que:

Escolha uma opção:

- ☐ a. Threads Java são mapeadas para processos Linux mas têm a memória compartilhada entre elas.

- ☐ b. Threads Java são mapeadas para threads Linux e participam do escalonamento executado pelo sistema operacional nativo.
- ☒ c. Threads Java são mapeadas para threads Linux e seu escalonamento é executado pela JVM. ✖

Sua resposta está incorreta.

A resposta correta é: Threads Java são mapeadas para threads Linux e participam do escalonamento executado pelo sistema operacional nativo.

Questão 9

Correto

Atingiu 1,00 de 1,00

O estado de um processo é definido, em parte, pela sua atividade presente. Quando o processo está esperando para ser atribuído a um processador, ele se encontra em um estado denominado:

Escolha uma opção:

- ☐ a. de execução.
- ☐ b. de espera.
- ☒ c. pronto. ✔
- ☐ d. novo.
- ☐ e. encerrado.

Sua resposta está correta.

A resposta correta é: pronto.

Questão 10

Correto

Atingiu 1,00 de 1,00

Nos conceitos de ciência da computação, um processo é um módulo executável que pode conter threads. Um conceito importante sobre threads que estão contidas no mesmo processo é que:

Escolha uma opção:

- ☐ a. controlam o acesso da memória dinâmica (heap) dos processos externos.
- ☐ b. acessam a memória estática ao contrário de processos convencionais.
- ☐ c. possuem proteção contra problemas comuns de processos, como vazamento e acessos inválidos da memória.
- ☒ d. podem compartilhar a memória do processo. ✔
- ☐ e. evitam que processos concorrentes acessem recursos do processo principal.

Sua resposta está correta.

A resposta correta é: podem compartilhar a memória do processo.

Questão 11

Correto

Atingiu 1,00 de 1,00

Em sistemas operacionais, uma região crítica é

Escolha uma opção:

- ☐ a. um trecho de programa no qual existe um recurso ao qual somente o sistema operacional pode ter acesso.
- ☐ b. um trecho de programa que deve ser executado em paralelo com a seção crítica de outro programa.
- ☒ c. um trecho de programa no qual existe o compartilhamento de algum recurso, mas o acesso ao recurso é mutuamente exclusivo. ✓
- ☐ d. um trecho de programa cujas instruções podem ser executadas em paralelo e em qualquer ordem.

Sua resposta está correta.

A resposta correta é: um trecho de programa no qual existe o compartilhamento de algum recurso, mas o acesso ao recurso é mutuamente exclusivo.

Questão 12

Correto

Atingiu 1,00 de 1,00

Qual o significado do conceito *multithread* num sistema operacional?

Escolha uma opção:

- ☐ a. Utilização de porções de memória para guardar dados que vêm de dispositivos com baixa velocidade para agilizar o processamento de algum dado.
- ☐ b. Condição em que processos ficam bloqueados esperando por algum evento que nunca vai acontecer.
- ☐ c. Capacidade de processar algum job assim que este chega ao sistema através da sobreposição de operações muito lentas com processamento.
- ☒ d. É uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente. ✓
- ☐ e. Guardar dados para posterior utilização, evitando a repetição de acesso a dispositivos lentos.

Sua resposta está correta.

A resposta correta é: É uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas simultaneamente.

Questão 13

Correto

Atingiu 1,00 de 1,00

Imagine que várias threads devem acessar a mesma base de dados (bd), algumas para ler e outras para escrever. Contudo, existem algumas restrições para a realização de leituras e escritas:

I. Enquanto uma thread estiver escrevendo, nenhuma outra thread pode acessar a base de dados (bd).

II. Enquanto uma thread estiver lendo, somente threads leitoras podem acessar a base de dados (bd).

Seja a solução abaixo:

```
pthread_mutex_t db;
pthread_mutex_t mutex;
int rc = 0; // Quantidade de leitores lendo ou querendo ler a base de dados (bd)

void reader() {
    while(1) { // Repete eternamente
        (1);
        rc=rc+1; // Um novo leitor
        if(rc==1)
            (2); // Caso este seja o primeiro leitor...
        (3);
        read_data_base(); // le a bd
        (4);
        rc=rc-1; // Um leitor a menos...
        if(rc==0)
            (5); // Caso este seja o ultimo leitor.
        (6);
        use_data_read(); // Utiliza as informações da bd
    } // Fim do while
}

void writer() {
    while(1) { // Repete eternamente
        think_up_data(); // pensa em informações para adicionar à bd
        (7);
        write_data_base(); // escreve novas informações na bd
        (8);
    }
}
```

Para a solução acima, que operações quais deveriam ser as operações 1, 2, 3, 4, 5, 6, 7 e 8?

Escolha uma opção:

- ☐ a.
 - 1. pthread_mutex_lock(&mutex)
 - 2. pthread_mutex_lock(&db)
 - 3. pthread_mutex_unlock(&mutex)
 - 4. pthread_mutex_lock(&mutex)
 - 5. pthread_mutex_unlock(&db)
 - 6. pthread_mutex_unlock(&mutex)
 - 7. pthread_mutex_lock(&mutex)
 - 8. pthread_mutex_unlock(&mutex)
- ☐ b.
 - 1. pthread_mutex_lock(&db)
 - 2. pthread_mutex_lock(&mutex)
 - 3. pthread_mutex_unlock(&mutex)
 - 4. pthread_mutex_lock(&mutex)

```
5. pthread_mutex_unlock(&mutex)
6. pthread_mutex_unlock(&db)
7. pthread_mutex_lock(&db)
8. pthread_mutex_unlock(&db)
```



c.

```
1. pthread_mutex_unlock(&mutex)
2. pthread_mutex_unlock(&db)
3. pthread_mutex_lock(&mutex)
4. pthread_mutex_unlock(&mutex)
5. pthread_mutex_lock(&db)
6. pthread_mutex_lock(&mutex)
7. pthread_mutex_lock(&db)
8. pthread_mutex_unlock(&db)
```



d.

```
1. pthread_mutex_lock(&db)
2. pthread_mutex_lock(&db)
3. pthread_mutex_unlock(&mutex)
4. pthread_mutex_lock(&mutex)
5. pthread_mutex_unlock(&mutex)
6. pthread_mutex_unlock(&mutex)
7. pthread_mutex_lock(&db)
8. pthread_mutex_unlock(&db)
```



e.

```
1. pthread_mutex_lock(&mutex)
2. pthread_mutex_lock(&db)
3. pthread_mutex_unlock(&mutex)
4. pthread_mutex_lock(&mutex)
5. pthread_mutex_unlock(&db)
6. pthread_mutex_unlock(&mutex)
7. pthread_mutex_lock(&db)
8. pthread_mutex_unlock(&db)
```



Sua resposta está correta.

A resposta correta é:

```
1. pthread_mutex_lock(&mutex)
2. pthread_mutex_lock(&db)
3. pthread_mutex_unlock(&mutex)
4. pthread_mutex_lock(&mutex)
5. pthread_mutex_unlock(&db)
6. pthread_mutex_unlock(&mutex)
7. pthread_mutex_lock(&db)
8. pthread_mutex_unlock(&db)
```

Questão 14

Correto

Atingiu 1,00 de 1,00

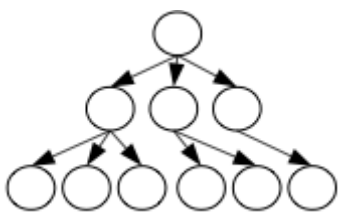
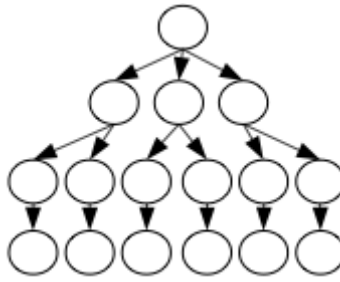
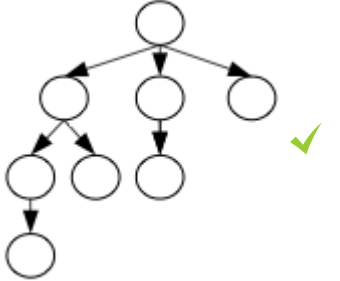
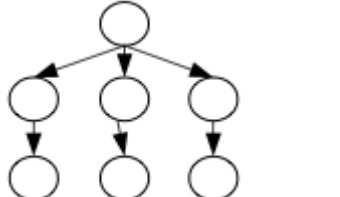
Para o trecho de código a seguir, qual o grafo que representa hierarquia de criação de processos?

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/wait.h>
int main() {
    int i;

    for (i=0; i<3; i++)
        fork();

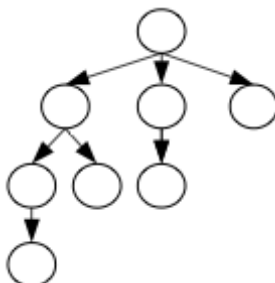
    while (wait(NULL) > 0);
}
```

Escolha uma opção:

- ☐ a. 
- ☐ b. 
- ☒ c.  ✓
- ☐ d. 

Sua resposta está correta.

A resposta correta é:



Questão 15

Correto

Atingiu 1,00 de 1,00

Considere um sistema operacional típico utilizado atualmente em computadores, que possua suporte nativo a threads. Ao efetuar o gerenciamento de processos, o sistema operacional associa alguns dos elementos enumerados abaixo a cada thread individualmente, enquanto outros são compartilhados entre threads de um mesmo processo.

1. Valores dos registradores
2. Espaço de endereçamento
3. Pilha de execução
4. Portas de comunicação e arquivos em uso

Quais dos elementos enumerados acima são associados individualmente a cada thread?

Escolha uma opção:

- ☐ a. 3 e 4.
- ☒ b. 1 e 3. ✓
- ☐ c. 1, 2 e 4.
- ☐ d. 1, 2 e 3.
- ☐ e. 2, 3 e 4.

Sua resposta está correta.

A resposta correta é: 1 e 3.

Questão 16

Correto

Atingiu 1,00 de 1,00

O problema do produtor/consumidor é um problema clássico de sincronização de threads: um grupo de threads utiliza um buffer de tamanho N para armazenar temporariamente itens produzidos; threads produtoras produzem os itens, um a um, e os armazenam no buffer; threads consumidoras retiram os itens do buffer, um a um, para processamento. O problema pode ser resolvido com a utilização de semáforos, que são mecanismos de software para controle de concorrência entre threads. Duas operações são definidas para um semáforo s : `wait(s)` e `post(s)`. Os pseudocódigos das threads produtoras e consumidoras 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_a	comando_f
comando_b	retira do buffer
coloca no buffer	comando_g
comando_c	comando_h
comando_d	consome o item

A partir dessas informações, para que o problema 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

Escolha uma opção:

- ☐ a. `wait(vazio)`, `post(mutex)`, `post(cheio)`, `wait(mutex)`, `wait(cheio)`, `post(mutex)`, `post(vazio)` e `post(mutex)`.
- ☐ b. `wait(cheio)`, `wait(mutex)`, `post(mutex)`, `post(vazio)`, `wait(vazio)`, `post(mutex)`, `post(mutex)` e `wait(cheio)`.
- ☐ c. `wait(mutex)`, `wait(vazio)`, `post(cheio)`, `post(mutex)`, `wait(mutex)`, `wait(cheio)`, `post(vazio)` e `post(mutex)`.
- ☐ d. `wait(mutex)`, `wait(vazio)`, `post(cheio)`, `post(mutex)`, `wait(mutex)`, `wait(vazio)`, `post(cheio)` e `post(mutex)`.
- ☒ e. `wait(vazio)`, `wait(mutex)`, `post(mutex)`, `post(cheio)`, `wait(cheio)`, `wait(mutex)`, `post(mutex)` e `post(vazio)`. ✓

Sua resposta está correta.

A resposta correta é: `wait(vazio)`, `wait(mutex)`, `post(mutex)`, `post(cheio)`, `wait(cheio)`, `wait(mutex)`, `post(mutex)` e `post(vazio)`.

Questão 17

Correto

Atingiu 1,00 de
1,00

Em um sistema operacional multi-tarefa, diversos processos compartilham uma ou mais CPUs do sistema. Um processo pode estar rodando, bloqueado ou pronto. É CORRETO afirmar que:

Escolha uma opção:

- ☐ a. Os processos bloqueados estão aguardando a alocação da CPU pelo sistema operacional.
- ☐ b. Um processo é bloqueado quando o sistema operacional detectar que ele não está mais respondendo.
- ☐ c. Quando o processo está bloqueado, ele irá aguardar que o administrador do sistema autorize seu funcionamento.
- ☒ d. Um processo é bloqueado quando ele requisitar uma operação de E/S. ✓
- ☐ e. Um processo é bloqueado quando o sistema operacional detectar que ele tentou fazer uma operação ilegal.

Sua resposta está correta.

A resposta correta é: Um processo é bloqueado quando ele requisitar uma operação de E/S.

◀ Escalonamento: sistemas de tempo real

Seguir para...



AF 2.2 - Escalonamento (questionário) ►