ⓘ As respostas corretas estão ocultas.

Pontuação deste teste: **27** de 30

Enviado 8 abr em 11:25

Esta tentativa levou 40 minutos.

Pergunta 1

2 / 2 pts

Em um sistema operacional, um processo pode, em um dado instante de tempo, estar em um de três estados: em execução, pronto ou bloqueado. Considere as afirmativas abaixo sobre as possíveis transições entre estes estados que um processo pode realizar.

I. Do estado em execução para o estado bloqueado

II. Do estado em execução para o estado pronto

III. Do estado pronto para o estado em execução

IV.Do estado pronto para o estado bloqueado

V. Do estado bloqueado para o estado em execução

VI. Do estado bloqueado para o estado pronto

Quais são as afirmativas verdadeiras?

- Somente as afirmativas I, II e são verdadeiras.
- **Somente as afirmativas I, II, III e VI são verdadeiras.**
- Todas as afirmativas são verdadeiras.
- Somente as afirmativas I, III, IV e VI são verdadeiras.
- Somente as afirmativas I, III, IV e V são verdadeiras.

Pergunta 2

2 / 2 pts

Os sistemas operacionais modernos utilizam o conceito de fila circular no escalonamento de processos. O processo que está no início da fila de processos prontos é selecionado, executado por algum tempo e, ao término da fatia de tempo, retorna para o final da fila. O mecanismo apresentado permite que as aplicações sejam:

- executadas apenas uma vez, pois o esquema de filas não permite que processos já selecionados possam retornar para a mesma fila.
- selecionadas no meio da fila, por terem mais prioridade que os demais processos.
- selecionadas conforme a sua prioridade dentro do sistema.
- executadas conforme são criadas ou esgotem as suas fatias de tempo.
- executadas de forma aleatória dentro da fila de processos prontos.

Pergunta 3

3 / 3 pts

Considere um sistema operacional com escalonamento por prioridades, no qual a avaliação do escalonamento é realizada em um intervalo mínimo de 5 ut. Neste sistema, os processos A e B competem por uma única UCP. Desprezando os tempos de processamento relativo às funções do sistema operacional, a tabela a seguir fornece os estados dos processos A e B ao longo do tempo, medido em intervalos de 5 ut (E = execução, P = pronto e W = espera). O processo A tem menor prioridade

que o processo B. Em que tempos A sofre preempção?

	00-04	05-09	10-14	15-19	20-24	25-29	30-34	35-39	40-44	45-49
Processo A	P	P	E	E	E	P	P	P	E	W
Processo B	E	E	W	W	P	E	E	E	W	W

	50-54	55-59	60-64	65-69	70-74	75-79	80-84	85-89	90-94	95-99	100-105
Processo A	P	E	P	P	E	E	W	W	P	E	E
Processo B	W	P	E	E	W	W	P	E	E	–	–

- (●) Entre os instantes 24-25 e 59-60.
- ( ) Entre os instantes 9-10, 39-40, 69-70.
- ( ) Entre os instantes 19-20, 54-55, 79-80.
- ( ) Entre os instantes 79-80.
- ( ) Nunca, pois o processo B tem maior prioridade do que o processo A.

Pergunta 4

2 / 2 pts

Starvation ocorre quando:

- (●) Pelo menos um processo é continuamente postergado e não executa.

A prioridade de um processo é ajustada de acordo com o tempo total de execução do mesmo.

O processo tenta mas não consegue acessar uma variável compartilhada

Dois ou mais processos são forçados a acessar dados críticos alternando estritamente entre eles.

Pelo menos um evento espera por outro evento que não vai ocorrer.

Pergunta 5

3 / 3 pts

Considere o seguinte programa com dois processos concorrentes. O escalonador poderá alternar entre um e outro, isto é, eles poderão ser intercalados durante sua execução. As variáveis x e y são compartilhadas pelos dois processos e inicializadas antes de sua execução.

```
programa P
    int x = 0;
    int y = 0;

processo A {
    while (x == 0)  ;
    print(“a”);
    y = 1;
    y = 0;
    print(“d”);
    y = 1;
```

```
processo B {
    print(“b”);
    x = 1;
    while (y == 0)
;
    print(“c”);
}
```

}

Qual é a possível saída:

- adbc ou bcad
- dbca ou dcab
- Nenhuma das opções
- abdc ou abcd
- badc ou bacd

Pergunta 6

3 / 3 pts

Considere a situação em que 4 processos A, B, C, D concorrem por recursos da máquina. Há 2 unidades de disco, 3 unidades de fita e 1 unidade de impressão. Os processos se encontram na seguinte situação:

1. O processo A está de posse de 2 unidades de disco;
2. O processo B está de posse de 2 fitas e requisita 1 unidade de disco;
3. O processo C está de posse de 1 unidade de impressão e requisita uma unidade de disco;
4. O processo D está de posse da outra fita;

O processo D requisita a unidade de impressão e, logo após, o processo A faz a mesma requisição. A situação acima:

- não configura deadlock mesmo que A e C participem de um ciclo

- não configura deadlock pois B não participa de um ciclo
- configura deadlock pois A, B, C, D participam de um ciclo
- configura deadlock mesmo que B e D não participem de um ciclo

Pergunta 7

3 / 3 pts

A concorrência entre processos ocorrem em diversas situações no âmbito dos sistemas operacionais e outros tipos de processos. Alguns destes problemas foram estudados e se tornaram clássicos pela sua ocorrência e aplicabilidade. Um destes é o problema do produtor consumidor, onde temos um ou mais processos “produzindo” uma informação e um ou mais processos “consumindo” a informação. Neste problema o mais importante é controlar o acesso dos processos envolvidos à porção de memória compartilhada de forma a não permitir que dois processos não “consumam” a mesma informação ou uma informação não possa ser consumida enquanto ela ainda está sendo produzida. A implementação mais segura para este cenário é com o uso de dois semáforos contadores, geralmente chamados de: Cheio e Vazio. O semáforo Cheio controla o quão cheio está o buffer de memória e o Vazio controla o tanto que o mesmo buffer está vazio e admite mais itens serem produzidos.

Considere o código abaixo que implementa a solução para o problema do produtor/consumidor:

```
void produtor ()
{
        int item;
        while (TRUE) {
                item = produzir();
1. _____
```

```
                    2. _____
                        insere(item);
                    3. _____
                    4. _____
            }
 }

void consumidor()
{
            int item;
            while(TRUE) {
                    5._____
                    6._____
                        item = remover();
                    7._____
                    8._____
                        consumir(item);
            }
}
```

Assinale a alternativa que preenche corretamente as lacunas.

- 1. wait(&cheio); 2. wait(&mutex); 3. signal(&mutex); 4. signal(&vazio); 5. wait(&vazio); 6. wait(&mutex); 7. signal(&mutex); 8. signal(&cheio);
- 1. wait(&mutex); 2. wait(&vazio); 3. signal(&cheio); 4. signal(&mutex); 5. wait(&mutex); 6. wait(&cheio); 7. signal(&vazio); 8. signal(&mutex);

1. wait(&vazio); 2. wait(&mutex); 3. signal(&mutex); 4. signal(&cheio); 5. wait(&cheio); 6. wait(&mutex); 7. signal(&mutex); 8. signal(&vazio);

1. wait(&mutex); 2. wait(&cheio); 3. signal(&vazio); 4. signal(&mutex); 5. wait(&mutex); 6. wait(&vazio); 7. signal(&cheio); 8. signal(&mutex);

Pergunta 8

3 / 3 pts

O tema “comunicação entre processos” (IPC) pode ser resumido em três partes: 1) como um processo pode passar informações para outro processo; 2) certificar que dois ou mais processos não “se atrapalhem” ao tentar gravar dados em regiões de memória compartilhada; 3) implementar a condição de espera ociosa, quando um processo depende do encerramento de outro processo.

Em relação ao tema comunicação entre processos, analise as afirmações a seguir:

I. O termo “condições de corrida” é utilizado para descrever a tentativa de dois ou mais processos, escreverem em uma área de memória compartilhada ao mesmo tempo.

II. Para evitar que dois ou mais processos escrevam em uma área de memória compartilhada ao mesmo tempo, o processo deve entrar na chamada “região crítica”, que consiste na parte do código do programa responsável por fazer acesso a área de memória compartilhada.

III. Semáforos binários inicializados com o valor 1 podem ser utilizados por dois ou mais processos para assegurar que apenas um deles consiga entrar em sua região crítica, em um determinado momento.

Está correto somente o que se afirma em:

- [ ] I
- [ ] II
- [ ] II e III
- [ ] I e III
- [x] I, II, III

Pergunta 9

3 / 3 pts

Considerando um escalonamento SJF preemptivo:

Processo	Tempo chegada	Tempo de burst
P1	0,0	7
P2	2,0	4
P3	4,0	1
P4	5,0	4

Qual o tempo médio de espera dados os processos acima?

- [ ] 2
- [ ] 4
- [x] 3
- [ ] 5
- [ ] 6

Pergunta 10

3 / 3 pts

Considerando um escalonamento Round-Robin com quantum = 20 e overhead = 5:

Processo	Tempo de burst
P_1	53
P_2	17
P_3	68
P_4	24

Qual a taxa de utilização da CPU, aproximadamente, dados os processos acima?

- 82
- 78
- 80
- 84
- 76

Incorreta

Pergunta 11

0 / 3 pts

Considerando um escalonamento Round-Robin com quantum = 20 e overhead = 5:

Processo	Tempo de burst
P_1	53

P_4	24

Qual o tempo médio de retorno dados os processos acima?

- (•) 132
- ( ) 141
- ( ) 162
- ( ) 125
- ( ) 151

Pontuação do teste: **27** de 30