



Universidade Federal de Santa Catarina
Centro Tecnológico de Joinville
Departamento de Engenharias de Mobilidade
EMB5632 - Sistemas Operacionais
Lista de exercícios - pré-prova

1. Qual o problema com a solução que desabilita as interrupções para implementar a exclusão mútua?
2. O que é espera ocupada e qual o seu problema?
3. Explique o que são semáforos e dê um exemplo de sua utilização.
4. Em uma aplicação concorrente que controla saldo bancário em contas correntes, dois processos compartilham uma região de memória onde estão armazenados os saldos dos clientes A e B. Os processos executam, concorrentemente os seguintes passos:

Processo 1 (Cliente A) /* saque em A */ 1a. x := saldo_do_cliente_A; 1b. x := x - 200; 1c. saldo_do_cliente_A := x; /* deposito em B */ 1d. x := saldo_do_cliente_B; 1e. x := x + 100; 1f. saldo_do_cliente_B := x;	Processo 2 (Cliente B) /* saque em A */ 2a. y := saldo_do_cliente_A; 2b. y := y - 100; 2c. saldo_do_cliente_A := y; /* deposito em B */ 2d. y := saldo_do_cliente_B; 2e. y := y + 200; 2f. saldo_do_cliente_B := y;
--	--

Supondo que os valores dos saldos de A e B sejam, respectivamente, 500 e 900, antes de os processos executarem, pede-se:

- a) Quais os valores corretos esperados para os saldos dos clientes A e B após o término da execução dos processos?
- b) Quais os valores finais dos saldos dos clientes se a sequência temporal de execução das operações for: 1a, 2a, 1b, 2b, 1c, 2c, 1d, 2d, 1e, 2e, 1f, 2f?
- c) Utilizando semáforos, proponha uma solução que garanta a integridade dos saldos e permita o maior compartilhamento possível dos recursos entre os processos, não esquecendo a especificação da inicialização dos semáforos.

5. Considere um programa concorrente com três threads, X, Y e Z, mostradas abaixo.

```
int n = 1;      /* variável compartilhada entre X, Y e Z */

void X() {      |      void Y() {      |      void Z() {
    n = n * 16;   |      n = n / 7;      |      n = n + 40;
}                |      }                |      }
```

Mostre como as threads poderiam usar semáforos para garantir que o valor final de n seja sempre 8.

6. **[Silberschatz 1994, 5.3mod]** Considere o seguinte conjunto de processos, onde “tempo CPU” representa a duração do próximo ciclo de CPU: Os processos chegam na ordem A, B, C, D, E, todos no instante zero.

processo	tempo CPU	prioridade
A	10	3
B	1	1
C	2	3
D	1	4
E	5	2

- (a) Faça diagramas temporais ilustrando a execução desses processos usando FCFS, SJF, prioridade não preemptiva (o menor valor tem a maior prioridade) e round-robin (quantum = 2). (O valor de prioridade só deve ser considerado no algoritmo de prioridade não preemptiva, e ignorado nos demais)
- (b) Qual o tempo de retorno de cada processo em cada algoritmo?
- (c) Qual o tempo de espera de cada processo em cada algoritmo? (O tempo de espera é dado pelo tempo que o processo fica esperando em uma fila, sem efetivamente usar nenhum recurso.)
- (d) Quais das escalas resultam no menor e no maior tempo de espera médio para todos os processos? Quais são esses tempos?

7. Suponha que os processos do exercício anterior são escalonados por um algoritmo de múltiplas filas com realimentação, onde as filas são escalonadas por prioridade e os processos em cada fila por RR, de acordo com o seguinte esquema:

- fila 1, quantum = 1
- fila 2, quantum = 3
- fila 3, quantum = 5

Os processos sempre iniciam na fila 1, e mudam de fila caso não tenham encerrado seu ciclo de CPU ao término do quantum. Faça um diagrama de tempo mostrando a execução dos processos e calcule o tempo de espera de cada processo e o tempo médio de espera para o conjunto.

8. [Tanenbaum 1987, 2.20] Cinco processos estão esperando para serem executados. As durações previstas de seus ciclos de CPU são 9, 6, 3, 5 e X. Em que ordem esses processos devem ser executados para minimizar o tempo de retorno médio do conjunto? (Sua resposta dependerá de X.)

9. [Oliveira 2004, 4.7] Quatro programas devem ser executados em um computador. Todos os programas são compostos por dois ciclos de processador e dois ciclos de E/S. A entrada e saída de todos os programas é feita sobre a mesma unidade de disco. Os tempos para cada ciclo de cada programa são mostrados abaixo:

Programa	CPU	Disco	CPU	Disco
P1	3	10	3	12
P2	4	12	6	8
P3	7	8	8	10
P4	6	14	2	10

Construa um diagrama de tempo mostrando qual programa está ocupando o processador e o disco a cada momento, até que os quatro programas terminem. Suponha que o algoritmo de escalonamento utilizado seja round-robin, com quantum de 4 unidades. Qual a taxa de ocupação do processador e do disco? O que acontece se a velocidade do disco dobrar?