

Escola	EACH		TURMA		Nota do aluno na PROVA
Curso	Sistemas de Informação				
Disciplina	Sistemas Operacionais - ACH2044	Data da Prova	31/10/14		
Professor	Clodoaldo Aparecido de Moraes Lima				
Aluno					
No. USP					

QUESTÃO 01	Valor da Questão:	2,0
-------------------	--------------------------	------------

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, 600 e 800, antes de os processos executarem, pede-se:

a) (0.5 ponto) Quais os **valores corretos** esperados para os saldos dos clientes A e B após o término da execução dos processos?

Cliente A ____ 300 ____ **cada item 0.25**

Cliente B ____ 1100 ____

b) (0.5 ponto) 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?

Cliente A ____ 500 ____ **cada item 0.25**

Cliente B ____ 1000 ____

c) (1.0 ponto) Utilizando dois semáforos (S1,S2), 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.

semaphore S1 = __1__;
0.1 ponto

semaphore S2 = __1__;

Processo 1 (Cliente A) /* saque em A */ 0.2 ponto down (S1) 1a. x := saldo_do_cliente_A; 1b. x := x - 200; 1c. saldo_do_cliente_A := x; up (S1) /* deposito em B */ 0.2 ponto down (S2) 1d. x := saldo_do_cliente_B; 1e. x := x + 100; 1f. saldo_do_cliente_B := x; up (S2)	Processo 2 (Cliente B) /*saque em A */ 0.25 ponto down (S1) 2a. y := saldo_do_cliente_A; 2b. y := y - 100; 2c. saldo_do_cliente_A := y; up (S1) /* deposito em B */ 0.25 ponto down (S2) 2d. y := saldo_do_cliente_B; 2e. y := y + 200; 2f. saldo_do_cliente_B := y; up (S2)
--	---

up(S2) → S2 (up) descontar 0.2, emprego de outro semáforo sem definir, descontar 0.2

QUESTÃO 02

Valor da Questão:

1,0

Considere um sistema de troca de processos entre a memória e o disco no qual a memória é constituída dos seguintes tamanhos de lacunas em ordem na memória: 11 KB, 5 KB, 22 KB, 17 KB, 7 KB, 10 KB, 8 KB, e 10 KB. Qual lacuna é tomada pelas solicitações sucessivas do segmento de 13 KB, 10 KB e 7 KB, para o first fit, best fit e worst fit?

(0.3 Ponto)

	First Fit	First Fit	First Fit
	Alocando 13 KB	Alocando 10 KB	Alocando 7 KB
11 KB	11 KB	10 KB 01 KB	10 KB 01 KB
5 KB	5 KB	5 KB	5 KB
22KB	13 KB 9 KB	13 KB 9 KB	13 KB 7 KB 2 KB
17 KB	17 KB	17 KB	17 KB
7KB	7 KB	7 KB	7 KB
10 KB	10 KB	10 KB	10 KB
8 KB	8 KB	8 KB	8 KB
10 KB	10 KB	10 KB	10 KB

(0.35 Ponto)

	Best Fit	Best Fit	Best Fit
	Alocando 13 KB	Alocando 10 KB	Alocando 7 KB
11 KB	11 KB	11 KB	11 KB
5 KB	5 KB	5 KB	5 KB
22KB	22 KB	22 KB	22 KB
17 KB	13 KB 4 KB	13 KB 4 KB	13 KB 4 KB
7KB	7 KB	7 KB	7 KB 0 KB
10 KB	10 KB	10 KB 0 KB	10 KB 0 KB
8 KB	8 KB	8 KB	8 KB
10 KB	10 KB	10 KB	10 KB

0.35 Ponto

	Worst Fit	Worst Fit	Worst Fit
	Alocando 13 KB	Alocando 10 KB	Alocando 7 KB
11 KB	11 KB	11 KB	7 KB 04 KB
5 KB	5 KB	5 KB	5 KB
22KB	13 KB 9 KB	13 KB 9 KB	13 KB 9 KB
17 KB	17 KB	10 KB 7 KB	10 KB 7 KB
7KB	7 KB	7 KB	7 KB
10 KB	10 KB	10 KB	10 KB
8 KB	8 KB	8 KB	8 KB
10 KB	10 KB	10 KB	10 KB

Descontar de cada item 0.05 (total 0.15), quando não apresentar a lacuna restante

QUESTÃO 03	Valor da Questão:	1,0
-------------------	--------------------------	-----

Um processo rodando com um escalonador de múltiplas filas (CTSS) necessita de 67 quanta para ser finalizado. Quantas vezes ele será colocado para rodar, incluindo a primeira vez (antes que tenha começado a rodar)?

Quando for posto para rodar, entra na primeira fila (quantum = 1). Rodado seu quantum, vai para a segunda fila (rodando 2 quanta). Findado seu tempo, vai para a terceira fila e, quando posto novamente para rodar, rodará 4 quanta, e assim por diante. Então, ele irá rodar $1 + 2 + 4 + 8 + 16 + 32 + 4$ (dos 7 possíveis), sendo escalonado 7 vezes. Note que matematicamente, como a cada escalonamento ele ganha o dobro do tempo, ele será escalonado $\log_2 67 = 6; 06$ vezes para rodar seus 67 quanta.

QUESTÃO 04	Valor da Questão:	1,0
-------------------	--------------------------	-----

Considere 4 processos, P_1 a P_4 , com prioridades 1, 3, 2 e 5, respectivamente, em que 5 é a mais alta. Seguindo o algoritmo de loteria, mostre a ordem de escalonamento dos processos acima, sabendo que a cada um é **atribuído um número de bilhetes igual à sua prioridade**, na ordem $P_1 \dots P_4$, e que os bilhetes sorteados são os de número 11, 5, 10, 3, 9, 4, 2, 6, 8, 1, e 7, nessa ordem.

Supondo a distribuição de bilhetes na forma consecutiva, teremos que P_1 recebe os bilhetes de 1, P_2 de 2-4, P_3 5-6 e P_4 de 7-11. Com a sequência de sorteios dada, os processos escalonados são $P_4, P_3, P_4, P_2, P_4, P_2, P_2, P_3, P_4, P_1$ e P_4 .

0.15 atribuição de bilhete aos processos

0.85 escalonamento –

1 erro 0.05

2 erros 0.15

3 erros 0.25

QUESTÃO 05	Valor da Questão:	2
-------------------	--------------------------	---

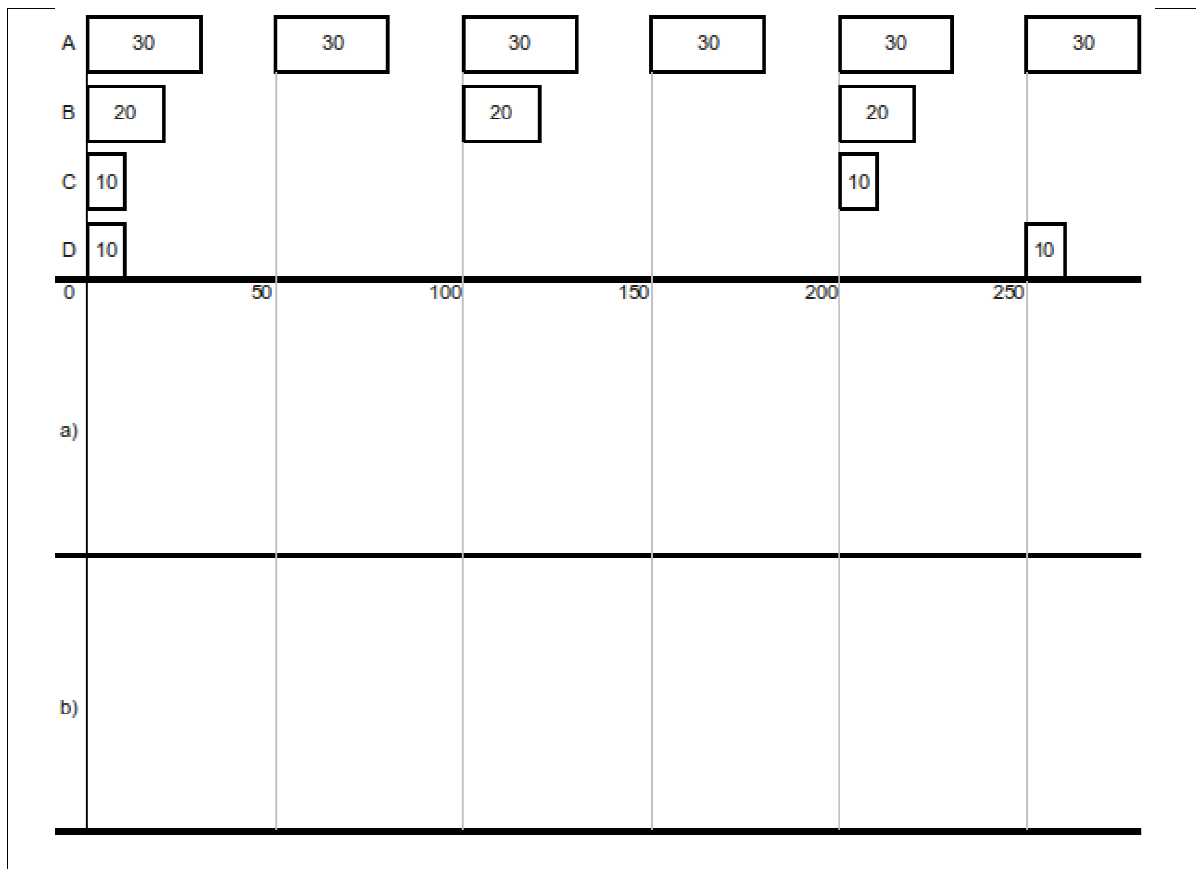
Um sistema de tempo real tem quatro eventos periódicos com períodos de 50, 100, 200 e 250 ms cada. Suponha que os quatro eventos requeiram 30, 20, 10 e 10 ms de tempo de CPU, respectivamente. Ilustre o escalonamento dos processos segundo

a) (1,0) Rate Monotonic Scheduling

Cada intervalo de 50 – 0.2

b) (1,0) Earliest Deadline First

Cada intervalo de 50 – 0.2



QUESTÃO 06 **Valor da Questão:** 3

Cinco processos em batch, A a E, chegam em um centro de computação quase que ao mesmo tempo. Eles têm tempos de execução estimados de 12, 10, 4, 8 e 6. Suas prioridades, definidas externamente, são 2, 3, 1, 5 e 4, com 5 sendo a mais alta. Para cada um dos seguintes algoritmos, determine o tempo médio de execução completa (mean turnaround time) desses processos. Ignore o tempo gasto com a troca de processos.

- Round Robin
- Prioridade
- First-come, First-served (na ordem 6, 12, 8, 4, 10)
- Shortest Remaining Time Next

Pede-se

- (0.5) Para i), assuma que o sistema aceita multiprogramação, e que cada processo recebe uma fatia de 3 da CPU. Para (ii) a (iv) assuma que somente um processo pode rodar por vez, rodando até o fim. Todos os processos são CPU bound (sem E/S)..
Cada item 0.125 – cada processo 0.025
- (2.5) Para i), assuma que o sistema aceita multiprogramação, e que cada processo recebe uma fatia de 3 da CPU. Para (ii) a (iv) assuma que somente um processo pode rodar por vez, rodando até o fim. Para todos os itens assuma que somente o processo B realiza E/S. Assuma que somente o processo B tenha um surto de CPU de 2 e que a E/S tenha duração de 4.

Round Robin – 0.7 (cada processo errado 0.14)

Outros método – 0.6 (cada processo errado 0.12)



UNIVERSIDADE DE SÃO PAULO
SEGUNDO SEMESTRE LETIVO DE 2014
PRIMEIRA PROVA OFICIAL