

UNIVERSIDADE DE SÃO PAULO
SEGUNDO SEMESTRE LETIVO DE 2015
PROVA P1 OFICIAL

Escola	EACH		TURMA		Nota do aluno na PROVA
Curso	Sistemas de Informação				
Disciplina	Sistemas Operacionais - ACH2044	Data da Prova	01/10/15		
Professor	Clodoaldo Aparecido de Moraes Lima				
Aluno					
No. USP					

QUESTÃO 01	Valor da Questão:	1,0
-------------------	--------------------------	-----

Relacione a segunda coluna de acordo com a primeira.

Atenção: **Um item relacionado incorretamente anula um item relacionado corretamente.**

- | | |
|---|--|
| <ul style="list-style-type: none"> (a) Inibição de Interrupção (b) Variáveis de Travamento (c) Instrução TSL (d) Sleep/Wake up (e) Semáforos (f) Monitores (g) Rendez-vous Estendido | <ul style="list-style-type: none"> (c) Solução de hardware em ambiente de múltiplos processadores (a) Processos maliciosos podem apoderar-se da CPU não permitindo que o Sistema Operacional retome o controle da CPU. (f) É uma unidade básica de sincronização de alto nível (b) Utiliza uma variável compartilhada, mas pode conduzir à condição de disputa. (d) Pode conduzir à existência de um estado onde os processos encontram-se bloqueados, situação denominada de deadlock (perda de sinal). (e) É um mecanismo de sincronização inter-processos composto das operações DOWN e UP. (g) Permite que se execute uma ação após ter recebido um canal de comunicação, mas antes de deixar o outro processo continuar. |
|---|--|

Cada item incorreto – 0.15

QUESTÃO 02	Valor da Questão:	1,0
-------------------	--------------------------	-----

Ordene (2 a 11) as atividades que (o nível mais baixo do) SO faz quando ocorre uma interrupção:

- | | |
|--------|---|
| (7) | O ponteiro da pilha é alterado, para que aponte para uma pilha temporária, usada pelo tratador do processo (assembly) |
| (4) | O novo PC é carregado do vetor de interrupções |
| (9) | Terminado o procedimento, o escalonador é chamado para decidir qual o próximo processo a executar |
| (1) | O controle (hardware) termina a execução da instrução atual |
| (8) | O procedimento para tratar desse tipo de interrupção é chamado (linguagem de alto nível) |
| (3) | Há o desvio para o endereço especificado no vetor de interrupção apropriado |
| (6) | Remove a informação da pilha do controle (colocada pela interrupção) |
| (10) | O controle volta ao código assembly (restaura registradores e o mapa da memória do processo) |
| (2) | O program counter, program status word e outros registradores são empilhados (no controle) |
| (11) | Roda o novo processo |
| (5) | Procedimento em assembly salva os registradores (no BCP) |

Cada item correto – 0.1

QUESTÃO 03	Valor da Questão:	2,5
-------------------	--------------------------	-----

Cinco processos A, B, C, D, E chegam em um centro de computação ao mesmo tempo. Eles têm tempos de execução estimados de 10, 8, 6, 12 e 4. Suas prioridades, definidas externamente, são 2, 4, 5, 3 e 1, com 5 sendo a mais alta.

a) (2,0) Assuma que somente o processo D tenha um surto de CPU a cada 2 e que a E/S tenha duração de 4. Apresente o escalonamento destes processos considerando os seguintes algoritmos

(0,5) Round Robin

(0,5) Prioridade (preemptivo)

(0,5) First-come, First-served (na ordem 6, 10, 8, 12, 4)

(0,5) Shortest Remaining Time Next

Round Robin

E												X	X	X							
D										X	X										
C							X	X	X												X
B				X	X	X												X	X	X	
A	X	X	X													X	X	X			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Pronto	[B,C, D,E]	[B,C, D,E]	[B,C, D,E]	[C,D, E,A]	[C,D, E,A]	[C,D, E,A]	[D,E, A,B]	[D,E, A,B]	[D,E, A,B]	[E,A, B,C]	[E,A, B,C]	[A, B,C]	[A, B,C]	[A, B,C]	[B, C,E]	[B,C, E, D]	[B,C, E, D]	[C,E, D,A]	[C,E, D,A]	[C,E, D,A]	[E,D, A, B]
Bloq	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[D]	[D]	[D]	[D]	[]	[]	[]	[]	[]	[]

E			X																			
D				X	X						X	X					X	X				
C	X	X																				
B									X	X												
A						X	X	X			X											
0	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43
Pronto	[E, D, A, B]	[E, D, A, B]	[D, A, B]	[A, B]	[A, B]	[B]	[B]	[B]	[A]	[A, D]	[A]	[]	[]									
Bloq	[]	[]	[]	[]	[]	[D]	[D]	[D]	[D]	[]				[D]	[D]	[D]	[D]			[D]	[D]	[D]

E																					
D		X	X					X	X												
C																					
B																					
A																					
0	44	45	46	47	48	49	50	51	52												
Pronto																					
Bloq			[D]	[D]	[D]	[D]															

$$T_m = (32+31+23+52+24)/5 = 32,4$$

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2015

PROVA P1 OFICIAL

Prioridade (preemptivo) 2, 4, 5, 3 e 1 → B, C, D, A, E

E																				
D															X	X				X
C	X	X	X	X	X	X														
B							X	X	X	X	X	X	X	X						
A																	X	X	X	X
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Pronto																				
Bloq																				

E										X	X			X	X					
D	X						X	X				X	X				X	X		
C																				
B																				
A		X	X	X	X			X	X											
0	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
Pronto																				
Bloq																				

E																				
D			X	X																
C																				
B																				
A																				
0	43	44	45	46																
Pronto																				
Bloq	[D]	[D]																		

$$T_m = (30+14+6+46+36)/5 = 26,4$$

First-come, First-served (na ordem 6, 10, 8, 12, 4) → C, A, B, D, E

E																				
D																				
C	X	X	X	X	X	X														
B																	X	X	X	X
A							X	X	X	X	X	X	X	X	X	X				
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Pronto	[A,B,D,E]	[A,B,D,E]	[A,B,D,E]	[A,B,D,E]	[A,B,D,E]	[A,B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[B,D,E]	[D,E]	[D,E]	[D,E]	[D,E]
Bloq	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

E						X	X	X	X											
D				X	X					X	X				X	X				
C																				
B	X	X	X																	
A																				
0	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41
Pronto	[D,E]	[D,E]	[D,E]	[E]	[E]															
Bloq	[]	[]	[]	[]	[]	[D]	[D]	[D]	[D]	[]	[]	[D]	[D]	[D]	[D]	[]	[]	[D]	[D]	[D]

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2015

PROVA P1 OFICIAL

E																				
D	X	X					X	X					X	X						
C																				
B																				
A																				
0	43	44	45	46	47	48	49	50	51	52	53	54	55	56						
Pronto																				
Bloq	[]	[]																		

$$T_m = (16+24+6+56+30)/5 = 26,4$$

Shortest Remaining Time Next 10, 8, 6, 12 e 4 → E, C, B, A, D

E	X	X	X	X																	
D																					
C					X	X	X	X	X	X											
B											X	X	X	X	X	X	X	X			
A																			X	X	X
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
Pronto	[C,B, A, D]	[C,B, A, D]	[C,B, A, D]	[C,B, A, D]	[B, A, D]	[B, A, D]	[B, A, D]	[B, A, D]	[B, A,D]	[B, A,D]	[A,D]	[A,D]	[A,D]	[A,D]	[A,D]	[A,D]	[A,D]	[A,D]	[D]	[D]	[D]
Bloq	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]	[]

E																					
D								X	X					X	X				X	X	
C																					
B																					
A	X	X	X	X	X	X	X														
0	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Pronto	[D]	[D]	[D]	[D]	[D]	[D]	[D]														
Bloq	[]	[]	[]	[]	[]	[]	[]	[]	[]	[D]	[D]	[D]	[D]	[]	[]	[D]	[D]	[D]	[D]	[]	[]

E																				
D					X	X					X	X				X	X			
C																				
B																				
A																				
0	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60		
Pronto																				
Bloq	[]	[]																		

$$T_m = (28+18+10+60+4)/5 = 24$$

b) (0,5) Ignore o tempo gasto com a troca de processos. Determine o tempo médio de execução completa (turnaround time) desses processos para cada um dos algoritmos acima.

Round Robin -> $T_m = (32+31+23+52+24)/5 = 32,4$ (0,15 ponto)

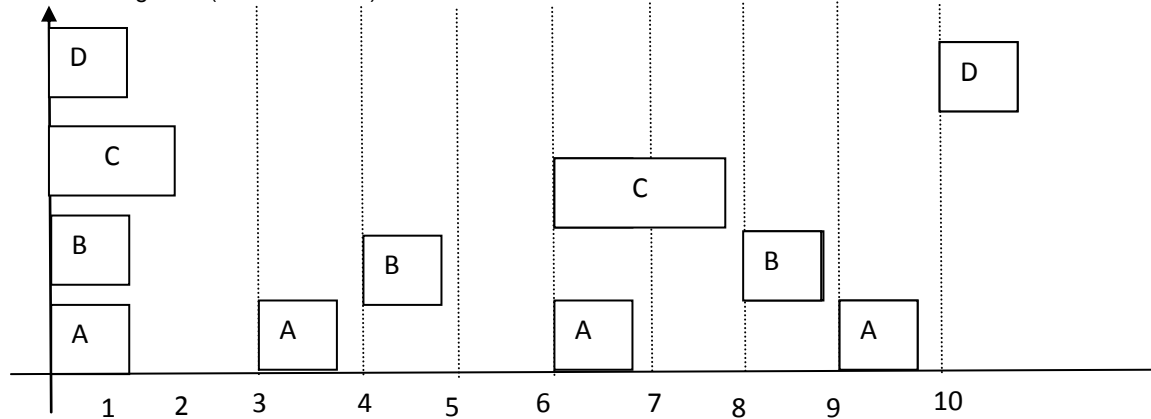
Prioridade-> $T_m = (30+14+6+46+36)/5 = 26,4$ (0,15 ponto)

FIFO -> $T_m = (16+24+6+56+30)/5 = 26,4$ (0,10 ponto)

Shortest Remaining Time Next -> $T_m = (28+18+10+60+4)/5 = 24$ (0,10 ponto)

QUESTÃO 04 **Valor da Questão:** 2,0

Um sistema de tempo real tem quatro eventos periódicos com períodos de 3, 4, 6 e 10 ms cada. Suponha que os quatro eventos requeiram 1, 1, 2 e 1 ms de tempo de CPU, respectivamente. Ilustre o escalonamento dos processos segundo (durante 20 ms)



a) Rate Monotonic Scheduling

b) Earliest Deadline First

D											D										D
C							C							C						C	
B					B				B					B				B			B
A				A			A			A				A			A			A	
	A	B	C	A	B	C	A	C	B	A	C	D	A	B	C	A	B	C	A	C	B
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Cada item errado -0.05

Não parou em 10 - 0,5

Parou não explicou – 0.2

D											D										D
C							C							C						C	
B					B				B					B				B			B
A				A			A			A				A			A			A	
	A	B	C	A	C	B	A	D	B	A	C	C	A	B	C	A	C	B	D	A	
	A	B	C	C	A	B	A	D	B	A	C	C	A	B	C	C	A	B	D	A	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Cada item 0.0375

QUESTÃO 05	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:

<p>Processo 1 (Cliente A)</p> <pre>/* 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;</pre>	<p>Processo 2 (Cliente B)</p> <pre>/* 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;</pre>
---	---

Supondo que os valores dos saldos de A e B sejam, respectivamente, 800 e 600, 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 500 0.25

Cliente B 900 0.25

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 700 0.25

Cliente B 800 0.25

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; 0,1 ponto

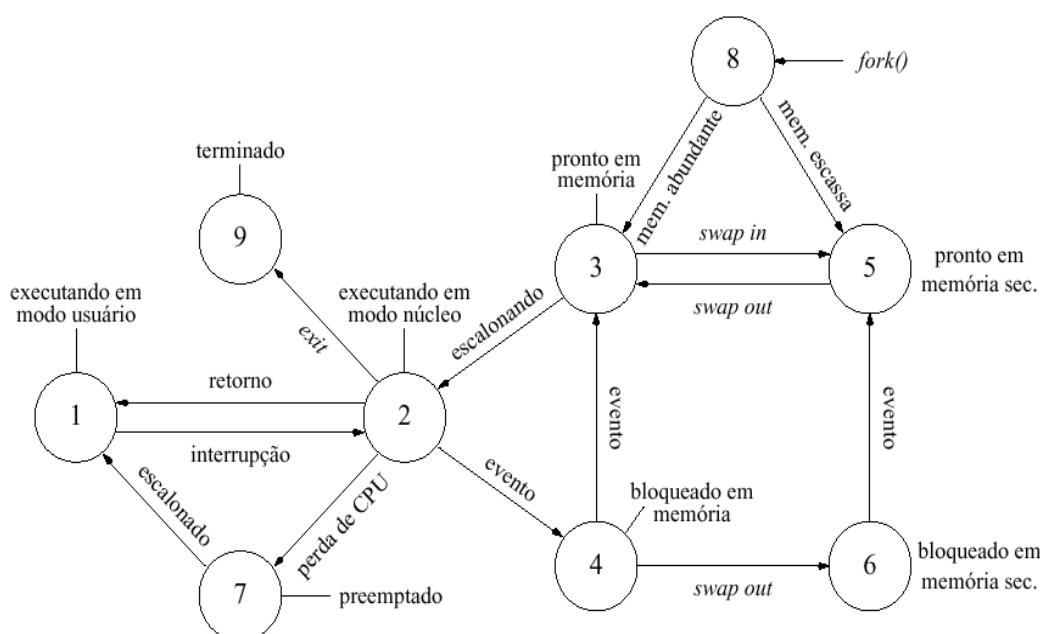
<p>Processo 1 (Cliente A)</p> <pre>/* saque em A */ 0,4 Ponto down (&S1) 1a. x := saldo_do_cliente_A; 1b. x := x - 200; 1c. saldo_do_cliente_A := x; up (&S1) /* deposito em B */ down (&S2) 1d. x := saldo_do_cliente_B; 1e. x := x + 100; 1f. saldo_do_cliente_B := x; up (&S2)</pre>	<p>Processo 2 (Cliente B)</p> <pre>/* saque em A */ 0.4 ponto down (&S1) 2a. y := saldo_do_cliente_A; 2b. y := y - 100; 2c. saldo_do_cliente_A := y; up (&S1) /* deposito em B */ down (&S2) 2d. y := saldo_do_cliente_B; 2e. y := y + 200; 2f. saldo_do_cliente_B := y; up (&S2)</pre>
--	--

Alterou o código -0.2 para cada trecho

QUESTÃO 06	Valor da Questão:	1.5
-------------------	--------------------------	-----

A partir do diagrama completo de transição de estados para processos em UNIX (ver figura abaixo), apresente uma possível sequência de estados referente ao seguinte histórico de um processo: o processo foi criado e iniciou a execução de instruções comuns (toda instrução que não corresponde a uma chamada de sistema), sem deixar a CPU até a ocorrência de uma requisição de E/S, a qual demanda um tempo 'longo' para ser atendida. Uma vez atendida esta requisição, o processo voltou imediatamente a executar instruções comuns, até liberar a CPU para outro processo. Ao ganhar a CPU novamente, o processo prosseguiu executando instruções comuns até o seu término.

Nota: é necessário associar as transições presentes na sequência de estados a cada evento listado no histórico.



O processo foi criado	8 → 3
Iniciou a execução	3 → 2 → 1
Requisitou E/S	1 → 2 → 4
Atendida a requisição	4 → 3
Voltou a executar	3 → 2 → 1
Liberou a CPU	1 → 2 → 7
Ganhou a CPU	7 → 1
Executou até o fim	1 → 2 → 9

Cada item 0,1875

7-1 = -0.1

Cada sem justificativa = 0.0875