UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL

Escola	EACH	TURMA		Nota do aluno na PROVA
Curso	Sistemas de Informação			
Disciplina	Sistemas Operacionais - ACH2044	Data da Prova	18/01/12	
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.

(a)	Inibição de Interrupção	(c)	Solução de hardware em ambiente de múltiplos processadores
(b)	Variáveis de Travamento	(a)	Processos maliciosos podem apoderar-se da CPU não permitindo que o Sistema Operacional retome o controle da CPU.
(c)	Instrução TSL	(f)	É uma unidade básica de sincronização de alto nível
(d)	Sleep/Wake up	(b)	Utiliza uma variável compartilhada, mas pode conduzir à condição de disputa.
(e)	Semáforos	(d)	Pode conduzir à existência de um estado onde os processos encontram-se bloqueados, situação denominada de deadlock (perda de sinal).
(f)	Monitores	(e)	É um mecanismo de sincronização inter-processos composto das operações DOWN e UP.
(g)	Caixa Posta	(j)	Permite que se execute uma ação após ter recebido um canal de comunicação, mas antes de deixar o outro processo continuar.
(i)	Porto	(g)	São filas de mensagens não associadas, a princípio, com nenhum processo.
(j)	Rendez-vous Estendido	(i)	A comunicação entre processos locais ou remotos, em um sistema estruturado com portos, será feita pela execução de primitivas síncronas (ou assíncronas) do tipo envia e recebe. Possui dono, que será o processo que o criar

QUESTÃO 02	**Valor da Questão:**	1,0

Ordene (1 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 nivel)
(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)

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL

QUESTÃO 03	**Valor da Questão:**	2,0

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) (1,6) Assuma que somente o processo B 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

(0,4) Round Robin

(0,4) Prioridade (preemptivo)

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

(0,4) Shortest Remaining Time Next

b) (0,4) 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.

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL

QUESTÃO 04	**Valor da Questão:**	1,5

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) (0,75) Rate Monotonic Scheduling

b) (0,75) 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		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 -0.075

Não parou em 10 - 0,30

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	C	
	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

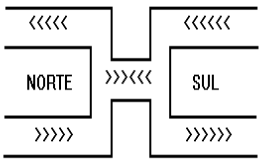
UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL

QUESTÃO 05	**Valor da Questão:**	1,0

Os veículos que chegam do norte e do sul devem atravessar uma ponte que possui somente uma pista (figura a seguir). Deste modo, num dado momento somente podem cruzar um ou mais veículos vindos do mesmo sentido. Escreva um algoritmo, utilizando o esqueleto de código abaixo, para os veículos vindos do norte e do sul, representando como eles atravessam a ponte. Utilize as primitivas SLEEP e WAKEUP. Explicite também se existe alguma possibilidade de falha decorrente da sincronização ter sido feita usando essas primitivas.




```
Programa Principal
Início {
Executa_Concorrente(T1,T2);
}

T1: Código Carros_Dir_Norte
Início {
    Viaja_Dir_Norte/Ponte( );
    Atravessa_Ponte( );
    Viaja_Ponte/Dir_Norte( );
}
```

```
T2: Código Carros_Dir_Sul
  Início {
     Viaja_Dir_Sul/Ponte( );
     Atravessa_Ponte( );
     Viaja_Ponte/Dir_Sul( )
          }
```

1.

Programa Principal
Variável Compartilhada: Passa;
Início {
Passa = 0;
Executa_Concorrente(T1,T2);
}

T1: Código Carros_do_Norte
Início {
Viaja_Dir_Norte/ponte ( ) ;
Se (Passa < 0) então
SLEEP;
Senão
Passa = Passa + 1;
Atravessa_ponte ( ) ;
Passa = Passa - 1;
Se (Passa == 0) então
WAKEUP;

T2: Código Carros_do_Sul
Início {
Viaja_Dir_Sul/Ponte ( ) ;
Se (Passa > 0) então
SLEEP ;
Senão
Passa = Passa - 1;
Atravessa_Ponte ( ) ;
Passa = Passa + 1;
Se (Passa == 0) então
WAKEUP ;

```
    Viaja_Ponte/Dir_Norte ( ) ;                    Viaja_Ponte/Dir_Sul ( ) ;
}                                                  }
```

Apesar desta solução não implicar em Espera Ocupada, ela tem o problema de que a disputa apenas se transferiu da região crítica (atravessar a ponte) para a variável Passa. Assim:

a) Se um processo for interrompido imediatamente após ler o valor da variável Passa (== 0) e antes de alterá-la, pode-se permitir que um carro do sul e outro do norte entrem juntos na ponte.
b) Se um processo for interrompido imediatamente após ler o valor da variável Passa (== 1 ou == -1) e antes de dormir, pode-se perder o WAKEUP feito pelo outro processo, deixando um carro do sul ou um carro do norte dormindo por um tempo indeterminado.

QUESTÃO 07	**Valor da Questão:**	1.0

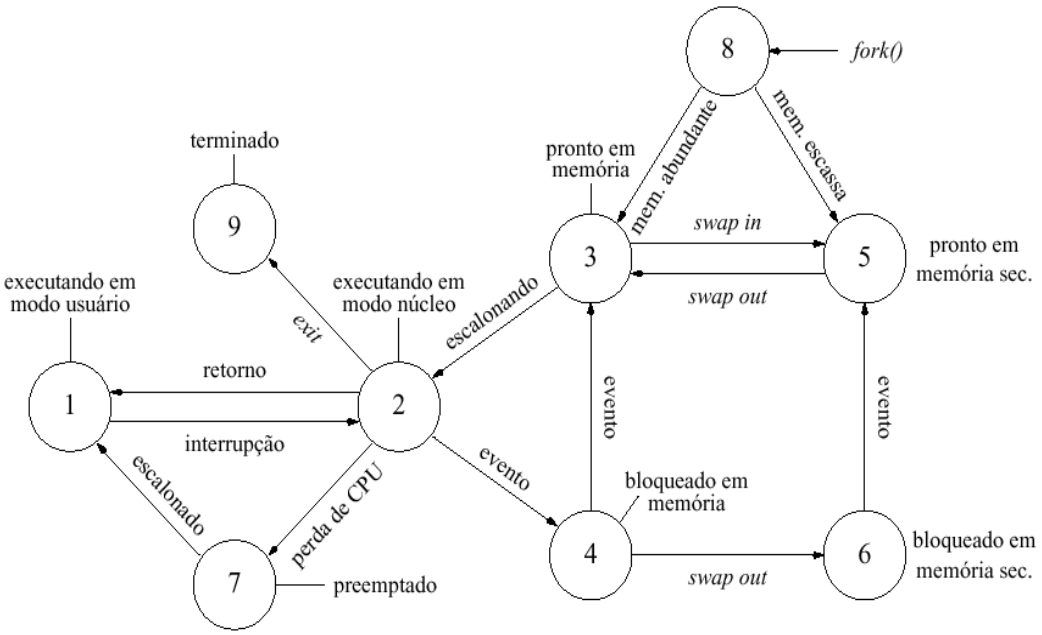
A partir do diagrama completo de transição de estados para processos em UNIX (ver figura abaixo), apresente uma possível sequencia de estados referente ao seguinte histórico de um processo: o processo foi criado e iniciou a execução de instruções comuns (toda instrução que não corresponde a uma chamada 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 um 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 seqüência de estados a cada evento listado no histórico.

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL



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

Cada item 0,125
7-1 = -0.1
Cada sem justificativa = 0.05

QUESTÃO 08	**Valor da Questão:**	1.5

Considere o problema dos leitores/escritores, onde existem processos que lêem o valor das variáveis compartilhadas, chamados leitores e processos que escrevem na região compartilhada, chamados escritores. Os leitores podem ler de modo concorrente, enquanto os escritores só podem executar em exclusão mútua. O seguinte código resolve o problema dos leitores/escritores, desde

que sejam colocados adequadamente os semáforos. Nesta implementação, foi feita a suposição de que, enquanto houver um leitor acessando a base de dados, o escritor é suspenso. Havendo pelo menos um leitor ativo, leitores subseqüentes serão admitidos e o escritor permanecerá suspenso até que nenhum leitor esteja presente.

```
typedef int semaphore;
semaphore mutex = 1;   /* Controla o acesso a rc
semaphore db = 1;      /* Controla o acesso a base de dados
int rc = 0;            /*Número de processos lendo ou querendo ler

void leitor (void)
{
        while (TRUE) {
                ___a) down(&mutex)_____;
                rc++;   /* Um leitor a mais agora
                if (rc ==1) __c) down(&db)_____;
                _b) up(&mutex)_____;
                read_data_base();    /* Acessa os dados
                a) down(&mutex)_____;
                rc--; /* Um leitor a menos agora
                if (rc = = 0) _d) up(&db)_____;
                _b) up(&mutex)_____;
                use_data(); /*Região não critica
        }
}
void escritor (void)
{
        while (TRUE) {
                prepare_data(); /* Região não critica
                _c) down(&db)_____;
                write_data_base(); /* Atualiza os dados
                _d) up(&db)_____:
        }
}
```

Considerando a) down(&mutex), b) up(&mutex), c) down(&db), d) up(&db) preencha adequadamente os 8 espaços no programa acima com a opção adequada.

QUESTÃO 09	Valor da Questão:	1,5

Como serão alocados os processos de 89K, 407K, 126K e 455K (nesta ordem) para os algoritmos (a) First-fit, (b) Best-fit e (c) Worst-fit (d) Next Fit? Considere as partições de memória de 100K, 500K, 200K, 300K e 600K (nesta ordem). Explique qual algoritmo utiliza mais eficientemente a memória.

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2012

PROVA P1 OFICIAL

	First	Best	Worst	Next
100 K	89 K 11 K	89 K	100 K	89 K
		11 K		11 K
500 K		407 K	126K	219 K
	407 K	93 K	374 K	407 K
	93 K			93 K
200 K	126 K 74 K	126 K		126 K
		74 K		74 K
300 K	300 K	300 K	300 K	300 K
600 K	455 K	455 K	89 K	455 K
	145 K	145 K	407 K	145 K
			104 K	

0,25 cada
Item 0,05
Worst - Falha na alocação de 455 K

b) First, Best, Next - mesmo desempenho
Next - mais rápido para este problemas
0,2