



Universidade Federal de Viçosa – Departamento de Informática
INF451 - Sistemas Operacionais

Prova 1 - 100 Pts – 2017/II – 25/09/2017 (Início: 10h00)

Nome: GRABANIO

Matrícula: _____

OBSERVAÇÕES:

- As questões podem ser resolvidas a lápis.
- A duração da prova é de 2 (duas) horas
- Questões como múltipla escolha, relacionar colunas, marcar verdadeiro ou falso e afins, podem ser feitas nos próprios enunciados das questões
- Deve-se utilizar o verso das folhas de questões para a resolução da prova.
- Permitida consulta: **não**.

Questão 1.

(15 pontos)

Considere as frases a seguir e diga se as afirmativas apresentada é Verdadeira ou Falsa. No caso da afirmativa ser falsa, mostre a razão.

- (**F**) Um sistema operacional pode ser dividido em duas partes básicas: a interface gráfica e o gerenciador de hardware.
- (**F**) Um programa de sistema pode ser visto como um módulo do sistema operacional que faz a invocação de chamadas de sistema para acesso aos periféricos.
- (**F**) A sincronização baseada em semáforos é mais eficiente e mais genérica do que a sincronização baseada em troca de mensagens.
- (**F**) As threads de usuário executam subordinadas aos processos do mesmo, mas podem ser escalonadas pelo próprio SO.
- (**V**) O chaveamento de processos pelo escalonador é um processo usualmente relativamente custoso e pode influenciar na calibração dos parâmetros de uma política de escalonamento.

Escrever justificativas aqui e/ou no verso

Questão 2.

(15 pontos)

Relacione as colunas abaixo. Observação: nem todas as sentenças da coluna à direita combinam com opções da coluna à esquerda. Mas todas opções da primeira coluna correspondem a uma das sentenças da segunda.

Coluna 1	Coluna 2
A. <i>Starvation</i>	(-) Tempo médio decorrido do momento em que uma tarefa é submetida até o momento em que ela é terminada
B. <i>Lightweight process</i>	(P) Maneira de assegurar que outros processos utilizem um espaço de memória ou arquivo que estiver em uso por um processo.
C. Chaveamento do escalonador	(G) Chamada de sistema para criar um processo filho.
D. Exclusão mútua	(I) Porção do espaço de memória de um processo dedicado ao código do programa associado àquele processo.
E. Problema do Jantar dos Filósofos	(B) Sinônimo para thread
F. Semáforo	(S) Sistema de comunicação entre processos bastante utilizado em sistemas paralelos
G. Instrução <i>fork</i>	(L) Número de tarefas por unidade de tempo que um sistema foi capaz de concluir
H. <i>Deadlock</i>	(N) Tipo especial de variável que sinaliza um dos estados impedido ou desimpedido. Utilizada no controle de exclusão mútua e sincronização em problemas de comunicação entre processos
I. Segmento de texto	(E) Situação hipotética que tem como objetivo ilustra uma variante de problema de comunicação entre processos.
J. Sistema de troca de mensagens	(H) Situação de congestionamento onde processos se colocam mutuamente à espera uns dos outros sem que ninguém libere um recurso vital.
K. Algoritmo não preemptivo	(M) Diz respeito ao algoritmo de escalonamento propriamente dito, sem levar em conta a calibração de seus parâmetros
L. Vazão do escalonador	(K) Política de escalonamento que não interrompe que um processo em execução na CPU, a menos que este bloqueie ou a libere voluntariamente
M. Mecanismo de escalonamento	(O) Método de sincronização de processos ou threads quando se tem necessidade de reiniciar a sincronia entre os mesmos antes que uma nova fase de execução seja atingida.
N. <i>Mutex</i>	(A) Fenômeno onde um processo nunca consegue acesso a tempo de CPU
O. Barreira	(F) Tipo especial de variável contadora para auxiliar no controle de exclusão mútua e sincronização em problemas de comunicação entre processos
	(C) Operação que envolve salvar e carregar registradores, mapas de memória e outros itens ligados ao contexto do processo para troca de estado do mesmo.
	(-) Sistema de comunicação entre processos exclusivo para modelo cliente-servidor

Questão 3.

(20 pontos)

- Qual a diferença entre um processo *CPU-bound* e um processo *I/O-bound*?
- Qual a função/papel da tabela de processo? Que tipo de informação pode ser encontrada na mesma?
- Que tipo de fenômeno o problema do Jantar dos Filósofos visa ilustrar? Explique.
- Quais os possíveis tipos de término de um processo? Classifique cada um deles como "voluntário" ou "involuntário".

Respostas aqui e/ou no verso

a) CPU - BOUND: CONCENTRA ~~AS~~ INSTAÇÕES DE CPU;
I/O - BOUND: CONCENTRA INSTAÇÕES DE I/O

b) PODEM RECUPERAR INFORMAÇÕES DE CONTEXTO DO PROCESSO.

c) DEADLOCK

d) VOLUNTÁRIO: FIM ^{NORMAL} DE EXECUÇÃO; ERRO INTERNO
INVOLUNTÁRIO: ERRO FATAL/ABORTO PELO S.O.;
INTERRUPÇÃO PELO USUÁRIO

→ RESPOSTAS BEM RESUMIDAS, APENAS PARA REFERÊNCIA. CORREÇÃO DISCRICIONÁRIA DEPENDE DO QUE O ALUNO ESCREVEU...

Questão 4.

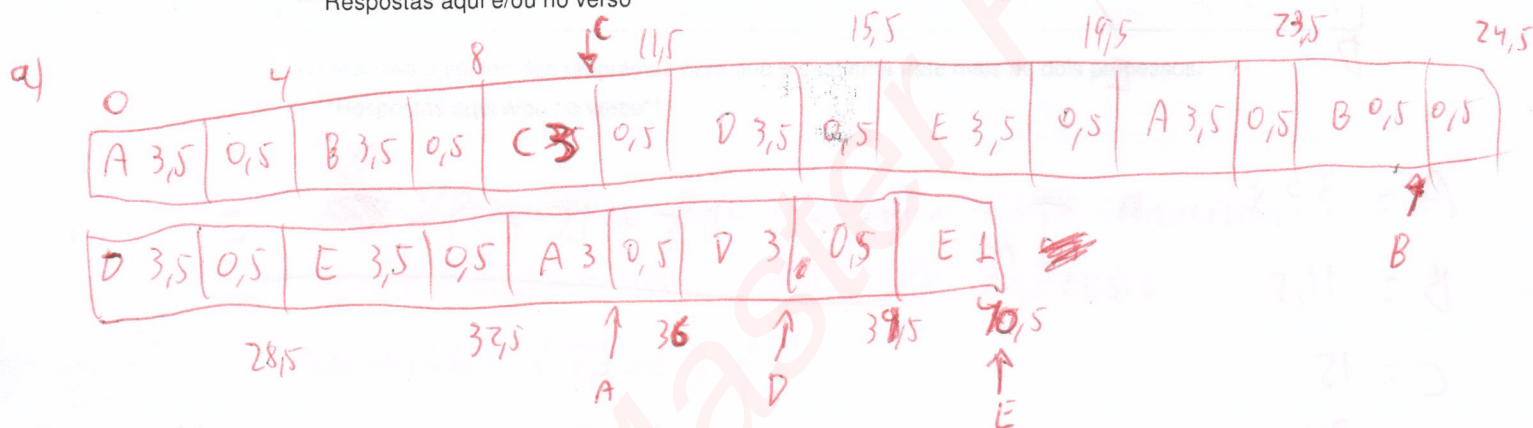
(15 pontos)

Em um sistema de computação em lote, 5 tarefas chegam simultaneamente. O sistema é capaz de estimar tempos de execução para as tarefas. Os dados de tais tarefas estão na tabela abaixo, onde a maior prioridade é aquela com o valor mais alto. Determine o tempo médio de execução dos processos neste sistema, adotando-se um tempo de chaveamento de 0,5 unidades de tempo e algoritmo de escalonamento Round Robin, com:

Tarefa	A	B	C	D	E
Tempo de execução estimado	10	4	3	10	8

- a) Quantum igual à metade do tempo médio de execução estimado para todos processos.
 b) quantum igual ao tempo médio de execução estimado para todos processos.

Respostas aqui e/ou no verso



A 35,5

B 24,0

C 11,0

D 39,0

E 40,5

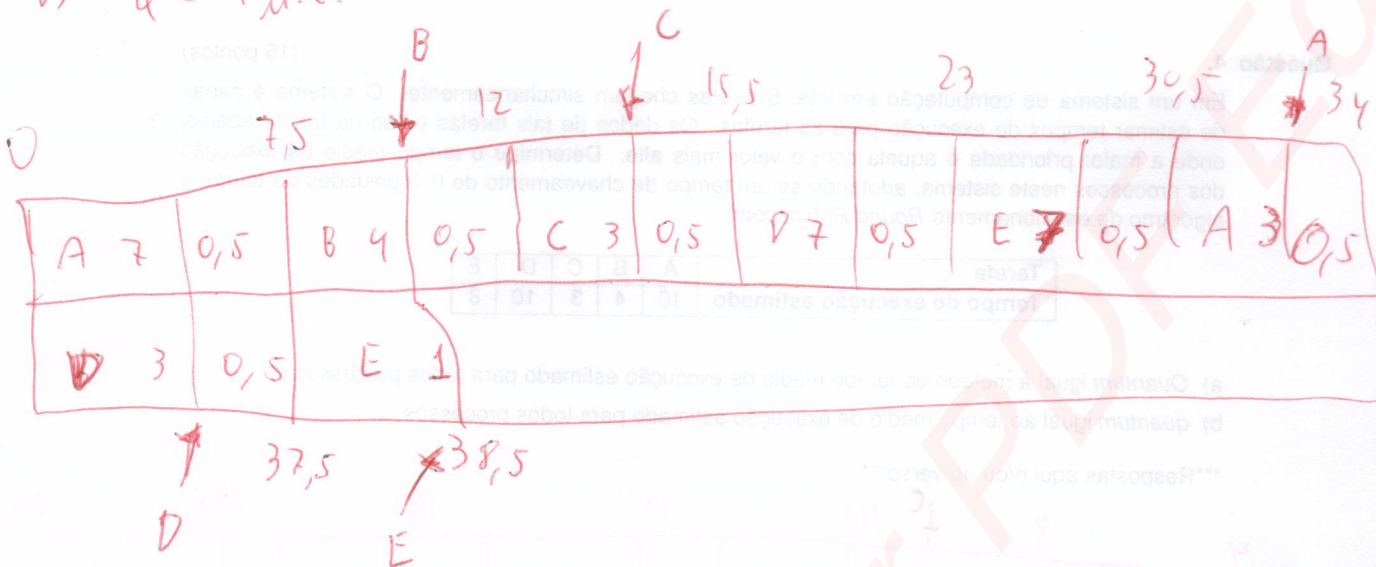
$$T_M = \frac{35,5 + 24 + 11 + 39 + 40,5}{5}$$

$$T_M = \frac{150}{5} = 30 \text{ m.t.}$$

* Quantum: $T_M = \frac{10 + 4 + 3 + 10 + 8}{5} = \frac{35}{5} = 7 \text{ m.t.}$

$$Q = \frac{T_M}{2} = 3,5 \text{ m.t.}$$

b) $Q = 7 \text{ m.t.}$



$$A = 37,5$$

$$B = 11,5$$

$$C = 15$$

$$D = 37$$

$$E = 38,5$$

$$T_m = 37,5 + 11,5 + 15 + 37 + 38,5 =$$

$$= \frac{135,5}{5} = 27,1 \text{ m.t.}$$

Questão 5.

(15 pontos)

Imagine dois processos hipotéticos contendo trechos de programa dentro e trechos fora de suas respectivas regiões críticas. Nestes processos, a exclusão mútua é implementada utilizando-se uma variável inteira compartilhada *turno*, que controla a vez de quem é de entrar na região crítica. Os códigos dos processos encontram-se listados a seguir.

```
1 while (true) {
2     while (turno != 0);
3     executa_regiao_critica();
4     turno = 1;
5     executa_regiao_ao_critica();
6 }
```

```
1 while (true) {
2     while (turno != 1);
3     executa_regiao_critica();
4     turno = 0;
5     executa_regiao_ao_critica();
6 }
```

Reescreva o código dos programas para que o esquema trate mais de dois processos.

Respostas aqui e/ou no verso

// Variável que identifica este processo
 MINHA_VEZ = ID; // ID do processo
 WHILE (true) {

WHILE (turno != MINHA_VEZ);


EXECUTA_REGIAO_CRITICA();

turno = (turno + 1) % N; // N: número de processos

EXECUTA_REGIAO_NAO_CRITICA();

}

(20 pontos)

P3 



- Faça um diagrama de tempo mostrando como será a utilização do processador e do disco
- Faça um diagrama de estados para o processo P_3 , a partir do instante $t = 0$, considerando que ele executa apenas as instruções mostradas na figura acima. Para cada transição de estado, coloque o tempo em que ela ocorre no arco correspondente do diagrama.
- Calcule o tempo médio geral de retorno dos processos. (OBS: o tempo de retorno, para um processo, é o tempo decorrido entre sua submissão e o seu término)

- Faça um diagrama de tempo mostrando como será a utilização do processador e do disco
- Faça um diagrama de estados para o processo P_3 , a partir do instante $t = 0$, considerando que ele executa apenas as instruções mostradas na figura acima. Para cada transição de estado, coloque o tempo em que ela ocorre no arco correspondente do diagrama.
- Calcule o tempo médio geral de retorno dos processos. (OBS: o tempo de retorno, para um processo, é o tempo decorrido entre sua submissão e o seu término)

Boa prova.

Prof. Marcos

$P_2, P_3, P_3, P_2, P_2, P_2, P_2, P_1, P_1, P_1, P_1, P_3, -$
 $- P_2, P_2, P_3, P_3, P_3, P_3, P_3, P_3, P_2, P_2, P_2, P_2, P_2, P_2, P_2, P_2, P_2$

(- P2, P2, P3, P3, P3, P3, P3, P3, P2, P2, P2, P2, P2, P2, P2, P2)

18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36

$\{P_2, P_2, P_2, P_1, P_1, P_1, P_1, P_1, P_3\}$ - - - - - P_1, P_1

$\{P_1, P_3, P_3, P_3, P_3, P_3, P_3, P_3, P_1, P_1, P_1, P_1, P_1, P_1, P_1, P_1, P_3, P_3\}$

$P_1, P_3, P_3, P_3, P_3, P_3, P_3, P_1, P_1, P_1, P_1, P_1, P_1, P_1, P_3, P_3,$

36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
P1	P1	P1	P1	P1	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P3	P2	P2	P2
P3	P2	P2	P2	P2	P2	P2	P2	P2	P1	P1	P1	P1	-	-	-	-	-	-

$P_3, P_2, P_2, P_2, P_2, P_2, P_2, P_2, P_2, P_1, P_1, P_1, P_1, - - - - -$

↑ ↑ ↑
(1) (3) (2)

