Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
Professores: Valmir C. Barbosa e Felipe M. G. França
Assistente: Alexandre H. L. Porto

Quarto Período
Gabarito da AP1 - Primeiro Semestre de 2014

Nome -
Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
3. Você pode usar lápis para responder as questões.
4. Ao final da prova devolva as folhas de questões e as de respostas.
5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.

---

1. (1,5) Suponha que a multiprogramação não tenha sido usada, que um programa A tenha terminado sua execução após 15s, e que ele tenha feito uma operação de E/S que foi inicializada após 1/3 desse tempo e durou 1/3 desse tempo. Após o término da execução de A, um programa B, que já podia executar quando A começou a executar, executou por 4s sem fazer operações de E/S. Se a multiprogramação passar a ser usada somente para evitar a ociosidade do processador ao executar as operações de E/S, o processador ainda ficará ocioso? E se B começar sua execução antes de A? Justifique a sua resposta.

   Resp.: Pelo enunciado, vemos que o programa A fez uma operação de E/S com duração de 5s após executar no processador também por 5s.
   -No caso de A ser o primeiro a executar, o processador ficará ocioso por 1s porque B poderá começar a sua execução após A executar por 5s e fazer a sua operação de E/S, já que o tempo de execução de B é de 4s e B não faz operações de E/S.
   -Se B for o primeiro a executar, então o processador ainda ficará ocioso por 5s, porque não existe nenhum outro programa para ser executado após A executar pelos primeiros 5s.

2. (2,5) Diga se as seguintes afirmativas são falsas ou verdadeiras. Para responder, escreva apenas F ou V para cada item em seu caderno de respostas.

   (a) (0,5) Um dos motivos de uma chamada ao sistema operacional não ser, em geral, executada diretamente pelos processos é devido à execução da chamada depender do hardware e ser de difícil implementação.

   Resp.: V (Verdadeira).

   (b) (0,5) O conceito de máquina virtual foi criado para permitir que processos de usuário executassem de modo seguro no modo supervisor.

   Resp.: F (Falsa), pois o conceito foi criado para permitir que vários sistemas operacionais pudessem executar na mesma máquina

sem interferirem uns com os outros.

(c) (0,5) Dizemos que existe um pseudoparalelismo quando dois ou mais processos executam alternadamente em duas ou mais unidades de processamento do hardware.

Resp.: F (Falsa), pois o pseudoparelismo ocorre quando não é possível executar os processos em paralelo, ou seja, quando o hardware tem um número de processadores inferior ao número de processos a serem executados.

(d) (0,5) O conceito de exclusão mútua foi definido para evitar que dois ou mais processos executassem simultaneamente no computador, caso existissem duas ou mais unidades de processamento no hardware.

Resp.: F (Falsa), pois o conceito foi criado para evitar uma condição de corrida, a qual ocorre quando dois ou mais processos compartilham um recurso não-preemptivo e tentam acessá-lo de modo concorrente.

(e) (0,5) Quando um algoritmo de escalonamento é preemptivo, o hardware do computador precisa ter um temporizador que gere periodicamente interrupções interceptáveis pelo processador.

Resp.: V (Verdadeira).

3. (1,5) Diga a quais conceitos vistos em aula se referem as seguintes definições:

(a) (0,5) Conceito definido na terceira geração e que permite a cada programa usar a CPU por um dado intervalo de tempo sempre que possível evitando deixar a CPU ociosa quando operações de E/S são feitas.

Resp.: Multiprogramação.

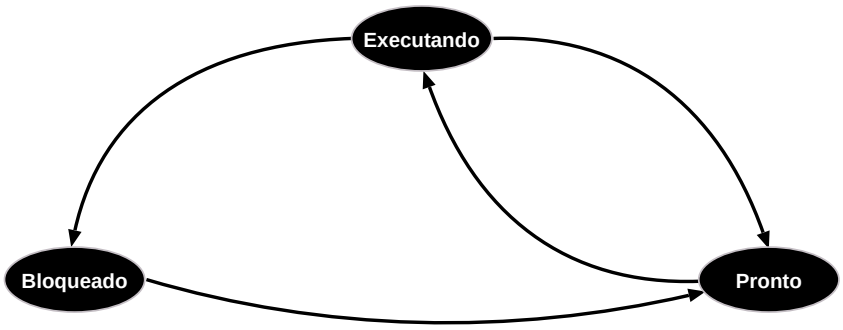
(b) (0,5) Parte do código de um processo no qual é acessado um recurso compartilhado com outros processos.

Resp.: Seção crítica ou região crítica.

(c) (0,5) Parte do sistema operacional responsável por comutar o uso do processador entre os processos do sistema.

Resp.: Escalonador ou agendador.

4. (1,5) Considere o diagrama de estados de um processo dado na aula 4 e reproduzido a seguir. Responda:



(a) (0,8) Por que, quando um processo é desbloqueado, ele passa do estado **Bloqueado** ao estado **Pronto** e não diretamente ao estado **Executando**?

Resp.: Porque o processo em execução ou algum dos outros processos no estado **Pronto** pode ser mais prioritário do que o processo que foi desbloqueado.

(b) (0,7) Por que o estado **Bloqueado** é necessário, ou seja, por que um processo bloqueado não pode ser colocado diretamente no estado **Pronto**?

Resp.: Porque os processos que estão no estado **Pronto** podem executar imediatamente, caso uma unidade de processamento fique livre ou um dos processos em execução deixe de ser o mais prioritário. Já um processo bloqueado somente pode executar depois que o evento que ele está esperando ocorre.

5. (1,5) Um aluno de sistemas operacionais deu a solução a seguir para resolver um problema no qual dois processos, A e B, acessam uma unidade de fita (recurso não-preemptivo) protegida pelo semáforo de contagem *acesso*. A solução do aluno está correta, ou seja, ela impede a ocorrência de impasses e garante o acesso exclusivo de um processo à unidade de fita? Se você acha que o aluno está correto basta responder que sim mas, se você acha que ele está errado, indique quais erros existem na solução do aluno.

   Semáforo *acesso*: inicializado com o valor 2.

```
void ProcessoA(void)
{
  while (1)
  {
    P(acesso);
    lerfita();
  }
}

void ProcessoB(void)
{
  while (1)
  {
    P(acesso);
    P(acesso);
    escreverfita();
    V(acesso);
    V(acesso);
  }
}
```

Resp.: Existem três erros na solução proposta. Como o semáforo *acesso* objetiva dar acesso exclusivo à fita, ele nunca pode assumir valor maior que 1. É devido a isso que, em geral, os semáforos binários são usados para garantir o acesso exclusivo, mas um semáforo de contagem também pode ser usado se for inicializado em 1 e se sempre executarmos a operação **P** sobre o semáforo imediatamente antes de acessar a fita e a operação **V** sobre o mesmo semáforo imediatamente após acessar a fita. Logo, o primeiro erro está na inicialização do semáforo *acesso*, pois ele deveria ter sido inicializado com o valor 1. O segundo erro está no fato de o algoritmo para o processo A não executar a operação **V** sobre o semáforo *acesso* após ler a fita. Finalmente, o último erro está no algoritmo do processo B. Como descrevemos anteriormente, deve ser executada somente uma operação **P** antes de o processo escrever na fita e somente uma operação **V** após o processo escrever na fita.

6. (1,5) Suponha que o sistema operacional use o algoritmo *round robin*, com um quantum de 2 unidades de tempo, ao escalonar os processos, e que a ordem de execução dos três únicos processos, A, B e C, até terminarem tenha sido ABCABCABABAAAA. Suponha agora que o sistema operacional tenha passado a usar o algoritmo de prioridades, sendo cada prioridade reduzida em 3 unidades a cada 2 unidades de tempo, e sendo que cada processo executa até a sua prioridade deixar de ser maior que todas as outras. Quais serão os novos tempos de término dos processos se a prioridade inicial de cada processo for igual à metade do número de unidades de tempo que ele precisa executar até terminar? Justifique a sua resposta.

Resp.: Pelo enunciado, os tempos de execução dos processos A, B e C são, respectivamente, 16, 8 e 4 unidades de tempo. Logo, as prioridades iniciais dos processos A, B e C são, respectivamente, 8, 4 e 2. Como veremos a seguir existem quatro possíveis respostas, devido a um empate nas prioridades dos processos A e C quando elas são iguais a -1 e -4. Então, nas tabelas a seguir mostramos as quatro possíveis ordens de execução dos processos quando o sistema operacional usa o algoritmo por prioridades. Na primeira linha das tabelas mostramos, da esquerda para a direita, os tempos de execução conforme os processos vão sendo escolhidos pelo algoritmo de escalonamento, antes de

executarem no processador. Na segunda linha mostramos os processos executados e, na última linha, mostramos suas prioridades logo após a execução. Como podemos ver, os tempos de término dos processos são de, respectivamente, 28, 22 e 14 unidades de tempo para a primeira e a terceira tabelas e de, respectivamente, 28, 22, e 16 unidades de tempo para a segunda e a quarta tabelas.

0	2	4	6	8	10	12	14	16	18	20	22	24	26
A	A	B	C	A	B	C	A	B	A	B	A	A	A
5	2	1	-1	-1	-2	-4	-4	-5	-7	-8	-11	-14	-17

0	2	4	6	8	10	12	14	16	18	20	22	24	26
A	A	B	C	A	B	A	C	B	A	B	A	A	A
5	2	1	-1	-1	-2	-4	-4	-5	-7	-8	-11	-14	-17

0	2	4	6	8	10	12	14	16	18	20	22	24	26
A	A	B	A	C	B	C	A	B	A	B	A	A	A
5	2	1	-1	-1	-2	-4	-4	-5	-7	-8	-11	-14	-17

0	2	4	6	8	10	12	14	16	18	20	22	24	26
A	A	B	A	C	B	A	C	B	A	B	A	A	A
5	2	1	-1	-1	-2	-4	-4	-5	-7	-8	-11	-14	-17