



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 - Segundo Semestre de 2017
(Segunda chamada)

Nome -

Assinatura -

Observações:

1. Prova sem consulta e sem uso de celular ou 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) A diferença entre as multiplexações dos recursos por tempo e por espaço é que, na primeira, cada recurso é usado exclusivamente por um programa por um dado intervalo de tempo e que, na última, cada programa usa uma parte do recurso.

Resp.: V (Verdadeira).

- (b) (0,5) A principal diferença entre os nomes de caminho absoluto e relativo de um diretório é que o primeiro inclui, além do caminho do diretório, todos os caminhos dos arquivos nesse diretório, enquanto que o segundo inclui somente o caminho do diretório.

Resp.: F (Falsa), porque o caminho absoluto define o caminho de um arquivo ou diretório a partir do diretório raiz, enquanto que o caminho relativo define o caminho de um arquivo ou diretório a partir do diretório de trabalho.

- (c) (0,5) O diagrama de estados de um processo mostra os possíveis estados bloqueados em que um processo pode estar ao ser suspenso devido a precisar esperar por um evento externo.

Resp.: F (Falsa), porque o diagrama de estados mostra os três possíveis estados em que um processo pode estar durante a sua execução no sistema operacional: **Executando**, quando o processo está em execução em uma das unidades de processamento do hardware; **Pronto**, quando o processo está esperando para ser escolhido pelo escalonador para depois ser executado em uma das unidades de processamento do hardware; e **Bloqueado** quando o processo foi suspenso até que um dado evento externo ocorra.

- (d) (0,5) Uma condição de corrida sempre ocorrerá quando dois ou mais processos de um conjunto tentarem acessar, ao mesmo tempo, uma mesma região de memória.

Resp.: V (Verdadeira).

- (e) (0,5) O escalonamento de dois níveis pode ser usado pelo sistema operacional quando a memória não possui espaço suficiente para armazenar todos os processos em execução no sistema, sendo que o escalonador de alto nível é definido para minimizar o tempo de comutação dos processos entre a memória e o disco.

Resp.: V (Verdadeira).

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

- (a) (0,5) Sistema definido na terceira geração de computadores, no

qual o tempo de processamento é dividido entre os usuários, conectados ao sistema a partir de terminais, e no qual cada usuário tem a ilusão de estar usando exclusivamente a máquina.

Resp.: Sistema de Compartilhamento de Tempo.

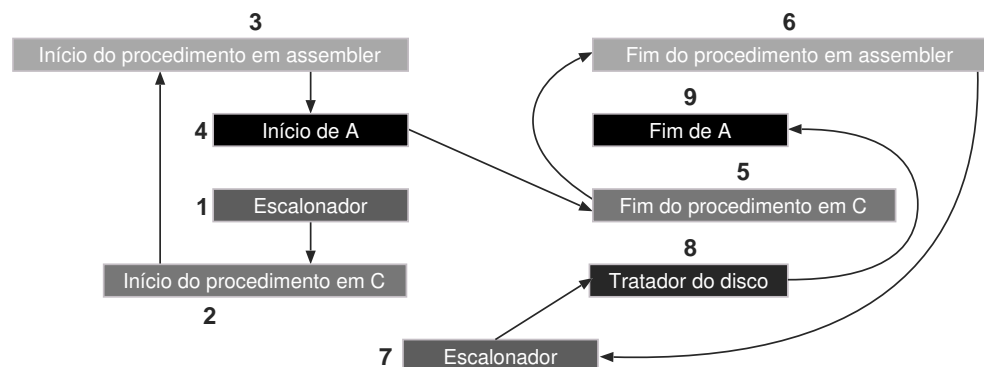
- (b) (0,5) Instrução usada para garantir a exclusão mútua que permite, em uma única operação atômica, verificar o conteúdo de uma posição de memória e, depois disso, colocar nela um valor diferente de zero.

Resp.: TSL.

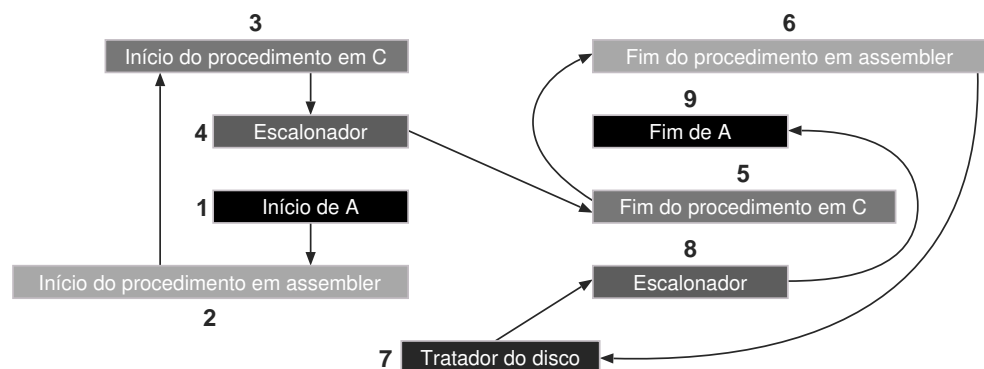
- (c) (0,5) Algoritmo de escalonamento no qual cada processo executa no processador por um dado intervalo de tempo, chamado de *quantum*, sendo que um processo somente poderá executar novamente no processador por mais um *quantum*, se necessário, depois de cada um dos outros processos ter também executado por um *quantum*.

Resp.: *Round robin*.

- 4. (1,5) Um aluno de sistemas operacionais disse que o diagrama a seguir representa, durante o tratamento de uma interrupção ocorrida durante a execução de um processo A, a correta ordem das ações executadas. O diagrama do aluno está correto? Se você achar que sim, basta dizer isso mas, se você achar que não, diga quais foram os erros do aluno.



Resp.: O diagrama do aluno está errado porque ele cometeu três erros devidos a permutações nos passos. Nos passos de 1 até 4 existem dois erros, pois o aluno permutou os passos 1 com o 4 e o 2 com o 3. Note que o processo A precisa estar executando quando a interrupção do disco ocorre, porque os procedimentos em assembler e em C e o escalonador são executados exatamente para suspender A quando a interrupção ocorrer para, depois disso, executar o tratador do disco. Finalmente, ele também permutou os passos 7 e 8. Neste caso, o escalonador deve ser chamado depois do tratador do disco porque essa segunda execução do escalonador vai reiniciar A e não o tratador do disco, que foi iniciado pela primeira execução do escalonador. O diagrama correto é o dado a seguir:



5. (1,5) Suponha que dois processos, A e B, compartilhem um conjunto, inicialmente vazio, que pode armazenar um número ilimitado de ele-

mentos. O processo A continuamente coloca três elementos no conjunto, e o processo B continuamente remove dois elementos do conjunto, calcula o produto desses elementos, e coloca o resultado novamente no conjunto. Como dois semáforos, um binário e um de contagem, podem ser usados para garantir a correta execução dos processos? Justifique a sua resposta.

Resp.: O semáforo binário *acesso* é usado para garantir que um dos processos possa acessar exclusivamente o conjunto. Já o semáforo de contagem *cheio* conta o número de elementos no conjunto, e é usado para bloquear o processo B se o conjunto estiver vazio. Como inicialmente o conjunto está vazio, então o semáforo *cheio* é inicializado com 0, e como nenhum processo está inicialmente acessando o conjunto, então o semáforo *acesso* é inicializado com 1. A seguir mostramos os códigos dos processos A e B:

```
void ProcessoA(void)
{
    while (1);
    {
        // Garante o acesso exclusivo ao conjunto.
        P(acesso);
        // Código para gerar os três elementos e inseri-los no conjunto.
        // Libera o acesso exclusivo ao conjunto.
        V(acesso);
        // Usa três operações V sobre cheio para registrar que três
        // elementos foram inseridos no conjunto.
        V(cheio);
        V(cheio);
        V(cheio);
    }
}
```

```

void ProcessoB(void)
{
    while (1);
    {
        // Usa duas operações P sobre cheio para garantir que
        // existam pelo menos dois elementos no conjunto.
        P(cheio);
        P(cheio);
        // Garante o acesso exclusivo ao conjunto.
        P(acesso);
        // Código para obter dois elementos do conjunto, multiplicá-los, e
        // colocar o resultado no conjunto.
        // Libera o acesso exclusivo ao conjunto.
        V(acesso);
        V(cheio);
    }
}

```

6. (1,5) Suponha que o algoritmo *round robin* tenha sido usado pelo sistema operacional, com um quantum de 2 unidades de tempo, e que três processos, A, B e C, tenham sido executados por, respectivamente, 6, 4 e 5 quanta, sendo que A usou integralmente o seu último quantum, assim como B, enquanto que C usou somente metade do seu. Suponha agora que o sistema operacional passe a usar o algoritmo por prioridades, no qual o processo em execução executa até existir um outro processo com prioridade menor, e no qual cada prioridade é incrementada por 2 unidades a cada 3 unidades de tempo. Quais serão os tempos de término de A, B e C, se as suas prioridades iniciais forem de, respectivamente, 7, 1 e 4? Justifique a sua resposta.

Resp.: Pelo enunciado, vemos que os tempos totais de execução no processador para os processos A, B e C são, respectivamente, 12, 8 e 9 unidades de tempo. Usando estes tempos e as prioridades dadas no enunciado para A, B e C, teremos a execução dada na tabela a seguir. Nessa tabela mostramos, na primeira linha, o tempo antes de o processo da coluna ter sido executado. Na segunda linha mostramos, da esquerda para a direita, a ordem de execução dos processos no processador. Finalmente, na última linha mostramos, para cada coluna, a

prioridade do processo antes de ele executar no processador no tempo dado na mesma coluna. Pela tabela, vemos que os tempos de término de A, B e C serão, respectivamente, de 29, 11 e 20 unidades de tempo.

0	3	6	9	11	14	17	20	23	26
B	B	C	B	C	A	C	A	A	A
1	3	4	5	6	7	8	9	11	13