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 2019

Nome -
Assinatura -

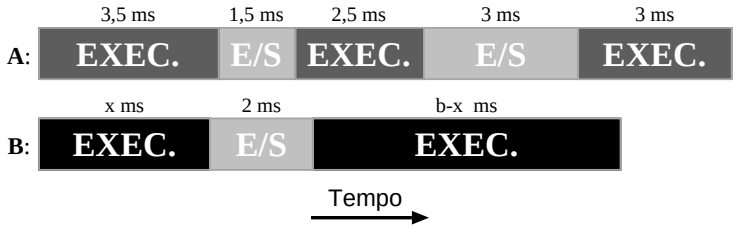
Observações:

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.

---

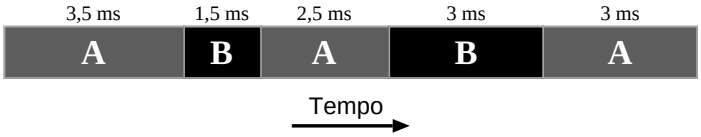
1. (1,5) Suponha que dois programas, A e B, sejam os únicos que precisam executar no sistema operacional. O programa A precisa executar por 9 ms e faz duas operações de E/S, com durações de, respectivamente, 1,5 ms e 3 ms, após executar por, respectivamente, 3,5 ms e 2,5 ms no processador. Já o programa B precisa executar por b ms e faz uma única operação de E/S com duração de 2 ms. Se a multiprogramação for usada somente para evitar a ociosidade do processador quando operações de E/S são feitas, qual será o menor valor possível de b? Justifique a sua resposta.

   Resp.: Pelo enunciado, vemos que o programa A executará por 3,5 ms antes de fazer a primera operação de E/S, com duração de 1,5 ms, por mais 2,5 ms entre a primeira e a segunda operações de E/S, e finalmente por mais 3 ms após o término da segunda operação, com duração de 3 ms. Vamos supor que o programa B executará por x ms antes de fazer a sua operação de E/S, com duração de 2 ms, e por $b - x$ ms após fazer essa operação. A figura a seguir ilustra esse comportamento de A e de B.



   Como A faz duas operações de E/S e B faz somente uma operação de E/S, e como A e B são os únicos programas que precisam executar no sistema operacional, então B não poderá executar antes de A, porque não seria possível evitar a ociosidade do processador durante a segunda operação de E/S de A. Agora, quando A executar antes de B, o tempo de x ms em que B executa antes de fazer a sua operação de E/S deverá ser pelo menos igual ao tempo da primeira operação de E/S feita por A, que é de 1,5 ms. Além disso, o tempo $b - x$ em que B executa após fazer a sua operação de E/S deverá ser pelo menos igual ao tempo da segunda operação de E/S feita por A, que é de 3 ms. Podemos concluir

que $x \geq 1,5$ e $b - x \geq 3$, ou seja, $1,5 \leq x \leq b - 3$. Como desejamos o menor valor possível para b, obtemos $x = 1,5$ e $b = 4,5$. Isso é ilustrado na figura a seguir:



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) O sistema operacional pode executar no modo usuário porque, neste modo, é possível acessar o hardware do computador.

   Resp.: F (Falsa), porque, na verdade, o sistema operacional executa no modo supervisor, devido a somentre ser possível acessar o hardware nesse modo.

   (b) (0,5) Uma máquina virtual define uma cópia exata do sistema operacional em execução no computador, de tal modo que um processo executando em uma máquina virtual não pode interferir com os resultados dos processos executando em outras máquinas virtuais.

   Resp.: F (Falsa), pois uma máquina virtual, além de ser uma cópia exata do hardware, também garante que o sistema operacional não interfira com os demais.

   (c) (0,5) Em um dado momento, um processo pode estar em um dos seguintes estados: **Executando**, se estiver em execução em uma unidade de processamento; **Pronto**, se puder executar mas não existirem unidades de processamento disponíveis para executá-lo; e **Bloqueado**, se estava executando e foi suspenso porque precisa esperar pela ocorrência de um evento externo.

Resp.: V (Verdadeira).

(d) (0,5) Nunca ocorrerão condições de corrida se um recurso compartilhado por um conjunto de processos for acessado, por cada processo, fora da seção crítica associada ao recurso.

Resp.: F (Falsa), porque a condição de corrida sempre será evitada se cada processo acessar o recurso compartilhado somente quando estiver dentro da seção crítica associada ao recurso.

(e) (0,5) Um processo poderá sempre ser bloqueado se um escalonador, preemptivo ou não-preemtivo, for usado pelo sistema operacional, mas somente poderá ser suspenso, após executar por algum tempo no processador, se o escalonador for preemptivo, pois somente este escalonador usa o temporizador do hardware.

Resp.: V (Verdadeira).

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

(a) (0,5) Nome dado ao pseudoarquivo que permite a troca de informações entre dois processos, sem que eles saibam da existência do pseudoarquivo, ao conectar a entrada de um desses processos à saída do outro processo.

Resp.: Pipe.

(b) (0,5) Nome dado à forma de paralelismo entre processos em um sistema operacional quando existe somente uma unidade de processamento.

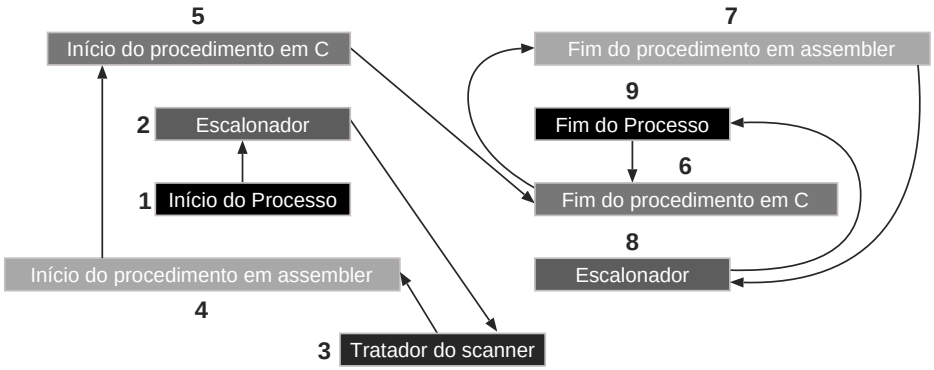
Resp.: Pseudoparalelismo.

(c) (0,5) Nome dado ao escalonador não-preemptivo que escolhe os processos de acordo com a ordem crescente dos tempos de execução

desses processos, sendo que cada processo executa até terminar ou ser bloqueado.

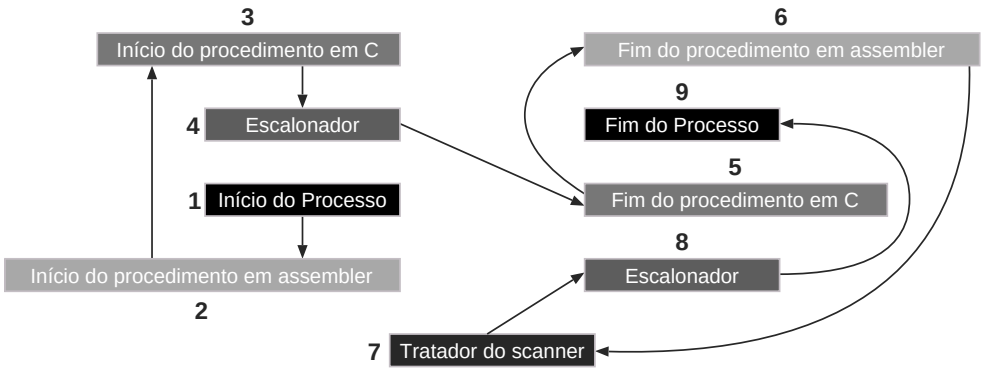
Resp.: Trabalho mais curto primeiro.

4. (1,5) Suponha que tenha ocorrido uma interrupção do *scanner* enquanto um processo estava em execução. Um aluno de sistemas operacionais disse que a figura a seguir mostra a ordem correta das ações realizadas quando essa interrupção foi tratada pelo sistema operacional. A figura do aluno está correta? Se você acha que sim, basta dizer isso mas, se você acha que não, indique os erros existentes nessa ordem.



Resp.: A figura do aluno está errada, porque somente as transições da ação "Início do procedimento em assembler" para a ação "Início do procedimento em C", da ação "Fim do procedimento em C" para a ação "Fim do procedimento em assembler", e da ação "Escalonador" da parte direita da figura para a ação "Fim do Processo"estão corretas. Quanto às demais, o correto é executar, após o processo ser suspenso, primeiramente o procedimento em assembler. Antes de este procedimento terminar, o procedimento em C é executado e, dentro dele, o escalonador é chamado para colocar o tratador do *scanner* para ser executado, sendo que esse tratador começa a executar somente depois do término dos procedimentos em C e assembler. Depois de o tratador terminar a sua execução porque já fez todas as tarefas necessárias ao correto tratamento da interrupção, o escalonador (da parte direita) é

novamente chamado para escolher o processo suspenso, que executará até terminar. A seguir está a figura com as ações executadas na ordem correta:



5. (1,5) Suponha que uma fila, que pode armazenar até x números, e que inicialmente possui y números, seja compartilhada por um conjunto de processos. A fila possui duas funções que os processos podem usar, $InsereNumeros(n_1, \ldots, n_a)$ para inserir $a > 0$ números n_1, ..., n_a no final da fila, e $RemoveNumeros(m_1, \ldots, m_b)$ para remover $b > 0$ números do início da fila e armazená-los em m_1, ..., m_b. Como dois semáforos de contagem e um semáforo binário devem ser usados, ao implementar as funções, para garantir que os processos executem sem gerar condições de corrida ou impasses? Justifique a sua resposta.

Resp.: Um semáforo binário, chamado *AcessoFila*, é usado para garantir o acesso exclusivo à fila. Em relação aos dois semáforos de contagem, o primeiro deles, chamado *LivresFila*, conta a quantidade de números que ainda podem ser inseridos na fila e é usado para bloquear o processo que chama a função $InsereNumeros$ quando a fila não possui espaço para a números adicionais. Finalmente, o segundo semáforo, chamado de *UsadosFila*, conta a quantidade de números na fila e é usado para bloquear o processo que chama a função $RemoveNumeros$ quando a fila não possui pelo menos b números. Como inicialmente a fila tem y números e não está sendo acessada através de uma das funções, então os semáforos *AcessoFila*, *LivresFila* e *UsadosFila* são inicializados, res-

pectivamente, com 1, $x - y$ e y. A seguir mostramos os códigos para as funções de acesso à fila.

void $InsereNumeros(n_1,\ldots,n_a)$
{
  // Espera a fila ter pelo menos a posições disponíveis para depois podermos
  // inserir os a números nela.
  for$(i = 1; i \leq a; i{+}{+})$
  {
    P(*LivresFila*);
  }
  // Garante o acesso exclusivo à fila.
  P(*AcessoFila*);
  // Código para inserir n_1, ..., n_a no final da fila.
  // Libera o acesso exclusivo à fila.
  V(*AcessoFila*);
  // Usa a vezes a operação **V** sobre *UsadosFila* para indicar que n_1, ..., n_a
  // foram inseridos na fila.
  for$(i = 1; i \leq a; i{+}{+})$
  {
    V(*UsadosFila*);
  }
}

void $RemoveNumeros(m_1,\ldots,m_b)$
{
  // Espera a fila ter pelo menos b números para depois podermos remover esses
  // números da fila.
  for$(i = 1; i \leq b; i{+}{+})$
  {
    P(*UsadosFila*);
  }
  // Garante o acesso exclusivo à fila.
  P(*AcessoFila*);
  // Código para remover b números do início da fila, e depois copiá-los para
  // m_1, ..., m_b.
  // Libera o acesso exclusivo à fila.
  V(*AcessoFila*);
  // Usa b vezes a operação **V** sobre *LivresFila* para indicar que m_1, ..., m_b
  // foram removidos da fila.
  for$(i = 1; i \leq b; i{+}{+})$
  {
    V(*LivresFila*);
  }
}

6. (1,5) Suponha que três processos, A, B e C, que não fazem operações de E/S, precisem executar no computador por, respectivamente, 15 ms, 11 ms e 13 ms, e que o sistema operacional use o algoritmo por *round robin* com um **quantum** de 3 ms. Suponha ainda que C é o primeiro a executar, que A comece a executar somente após C ter executado por 2 **quanta**, e que B somente comece a executar após A ter executado por 3 **quanta**. Qual será a ordem de execução dos processos? E quais serão os tempos decorridos entre o início e o término desses processos? Justifique a sua resposta.

Resp.: Quando os processos A, B e C forem executados como descrito no enunciado, a ordem de execução será CCACACABCABABB, como observado na tabela a seguir. Nesta tabela, mostramos como os processos são escolhidos pelo algoritmo, sendo que cada coluna refere-se à execução de um processo, dando o tempo de início de cada **quantum** e o processo correspondente. Pela tabela, vemos que os tempos de decorridos entre o início e o término de A, B e C serão de, respectivamente, 28 ms, 18 ms e 25 ms.

0	3	6	9	12	15	18	21	24	25	28	31	34	37	39
C	C	A	C	A	C	A	B	C	A	B	A	B	B	-