



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  
AP3 - Primeiro Semestre de 2012

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. (2,0) Diga a quais conceitos vistos em aula se referem as seguintes definições:

- (a) (0,6) Permite a cada programa em execução usar a CPU por um dado intervalo de tempo, e permite tentar evitar a ociosidade do processador durante a execução das operações de E/S;

**Resp.:** Multiprogramação.

- (b) (0,6) São as responsáveis pela definição da interface entre o sistema operacional e os processos em execução;

**Resp.:** Chamadas ao sistema operacional.

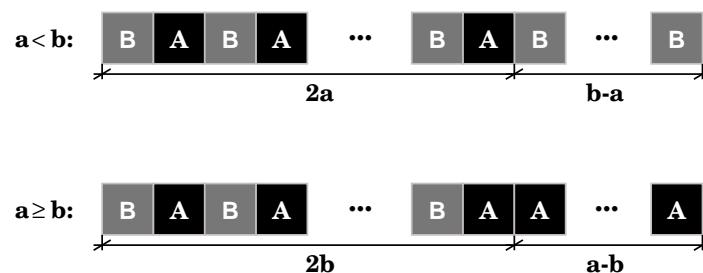
- (c) (0,8) Estruturação do sistema operacional caracterizada pela não existência de uma organização dentro do núcleo do sistema.

**Resp.:** Monolítica.

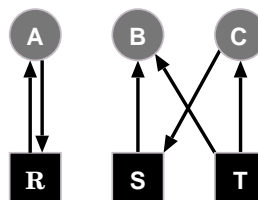
2. (2,0) Suponha que o sistema operacional use o algoritmo de escalonamento por *round robin* com um quantum de  $q$  unidades de tempo. Suponha ainda que um processo A, o único em execução no sistema, tenha executado em  $a$  quanta. Se A executar novamente agora junto com um processo B, e se B (cujo tempo necessário de execução no processador é de  $b$  quanta) começar a executar antes de A, qual será o tempo decorrido entre o início e o término da execução de A? Justifique a sua resposta.

**Resp.:** Como vimos na Aula 6, no algoritmo de escalonamento por *round robin*, cada processo executa no processador por um quantum e, depois disso, somente poderá executar novamente por mais um quantum, se necessário, após todos os outros processos no estado pronto terem também executado por um quantum. No caso da questão, A e B alternarão as suas execuções no processador até que um deles termine. Depois disso, o processo ainda em execução executará exclusivamente no processador também até terminar. Como o término da execução de

A depende dos valores de  $a$  e  $b$ , precisamos avaliar os dois possíveis casos dados a seguir, não esquecendo que B começa a executar antes de A no processador. Se  $a < b$ , A termina a sua execução antes de B, e a sequência de execução no processador é composta por  $a$  execuções alternadas de B e A seguida por  $b - a$  execuções consecutivas de B. Lembrando que cada execução de um processo no processador demora 1 quantum, ou  $q$  unidades de tempo, e que B começa executar antes de A, então o tempo decorrido entre o início e o término de A é de  $(2a - 1)q$  unidades de tempo. Agora, se  $a \geq b$ , então B termina a sua execução antes de A, e a sequência de execução é composta por  $b$  execuções alternadas dos processos B e A seguida, se  $a > b$ , por  $a - b$  execuções consecutivas de A. Novamente devido ao fato de cada quantum durar  $q$  unidades de tempo e de B começar antes de A, o tempo decorrido entre o início e o término de A é de  $2bq + (a - b)q - q$  ou  $(a + b - 1)q$  unidades de tempo. Esses dois casos encontram-se ilustrados na figura a seguir.



3. (2,0) Um aluno de sistemas operacionais disse que o grafo de recursos dado a seguir representa uma possível e correta alocação dos recursos não-preemptivos R, S e T aos processos A, B e C. Se você acha que o aluno está correto basta responder que sim mas, se você acha que o aluno está errado, aponte os erros que existem no grafo.



**Resp.:** O aluno está incorreto porque existem dois erros no grafo. O primeiro erro envolve o processo A e o recurso R, pois existem duas arestas entre eles, uma indicando que R pertence a A, e outra indicando que A deseja R e foi bloqueado. Como R não está alocado a nenhum outro processo, a aresta correta seria a que indica que R está alocado ao processo A, ou seja, aquela orientada de R para A. O segundo erro é que, segundo o grafo, T pertence ao mesmo tempo aos processos B e C, pois existe uma aresta orientada de T para B e uma outra aresta orientada de T para C, o que não é possível, pois T é um recurso não-preemptivo. O correto seria T pertencer a B (indicado pela aresta orientada de T para B), ou T pertencer a C (indicado pela aresta orientada de T para C).

4. (2,0) Considere a execução do algoritmo LRU dada na tabela a seguir. Se o sistema operacional passar a usar o algoritmo FIFO, ocorrerá o mesmo número de falhas de página? Justifique a sua resposta.

Página acessada	Ordenação	Ocorreu uma falha?
0	0	Sim
1	0 1	Sim
0	1 0	Não
2	0 2	Sim
0	2 0	Não
3	0 3	Sim
2	3 2	Sim
3	2 3	Não

**Resp.:** Como vimos na Aula 9, no algoritmo FIFO as páginas são ordenadas em ordem crescente de acordo com o tempo de cópia delas para a memória. Depois disso, o algoritmo escolhe para ser substituída a primeira página segundo essa ordenação, ou seja, a página copiada há mais tempo para a memória. Pela tabela do enunciado da questão, vemos que a ordem de acesso das páginas foi 0, 1, 0, 2, 0, 3, 2 e 3. A seguir mostramos uma tabela similar à dada na questão para o algoritmo FIFO, sendo que agora mostramos, na segunda coluna, as páginas ordenadas em ordem crescente de acordo com o tempo de cópia das páginas

para a memória. Como podemos ver pela tabela, o número de falhas de página não será igual, pois agora ocorrerão 6 falhas ao invés de 5.

Página acessada	Ordenação	Ocorreu uma falha?
0	0	Sim
1	0 1	Sim
0	0 1	Não
2	1 2	Sim
0	2 0	Sim
3	0 3	Sim
2	3 2	Sim
3	3 2	Não

5. (2,0) 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) (0,4) A principal característica do acesso sequencial a um arquivo é que os seus bytes (ou registros) podem ser acessados independentemente uns dos outros.

**Resp.:** F (Falsa), porque somente no acesso aleatório os bytes (ou registros) podem ser acessados independentemente uns dos outros.

- (b) (0,4) Na alocação contígua, os blocos de cada arquivo são distribuídos aleatoriamente no disco.

**Resp.:** F (Falsa), pois na alocação contígua os blocos (lógicos) de um mesmo arquivo são armazenados em blocos consecutivos do disco.

- (c) (0,4) Na técnica por alocação baseada em nós-i, existe um nó-i para cada arquivo armazenado no disco. Cada nó-i armazena todos os atributos do arquivo e seus primeiros blocos lógicos.

**Resp.:** V (Verdadeira).

- (d) (0,4) Se usamos o mapa de bits ao gerenciar os blocos livres do disco, a cada endereço de um bloco é associado um mapa de bits no qual cada bit indica qual byte desse bloco está sendo usado.

**Resp.:** F (Falsa), pois cada entrada do mapa de bits é um bit que indica se o bloco do disco associado a ela está livre ou usado.

- (e) (0,4) Existirá uma inconsistência no sistema de arquivos se existir um bloco do disco não marcado como livre e não usado por um arquivo, ou se esse bloco estiver alocado ao mesmo tempo a dois ou mais arquivos.

**Resp.:** V (Verdadeira).