

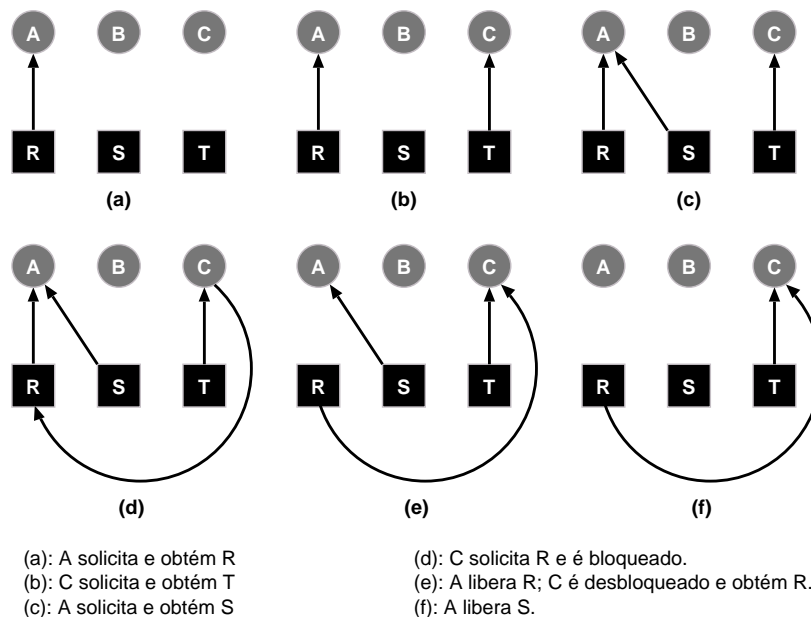
Lista de Exercícios - Sistemas Operacionais

Aula 7: *Entrada/Saída*

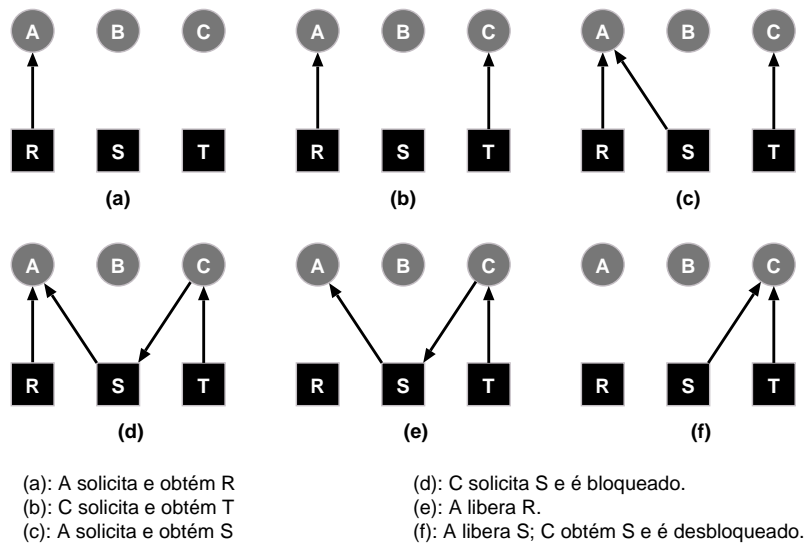
Professores: Felipe M. G. França e Valmir C. Barbosa

Assistente: Alexandre H. L. Porto

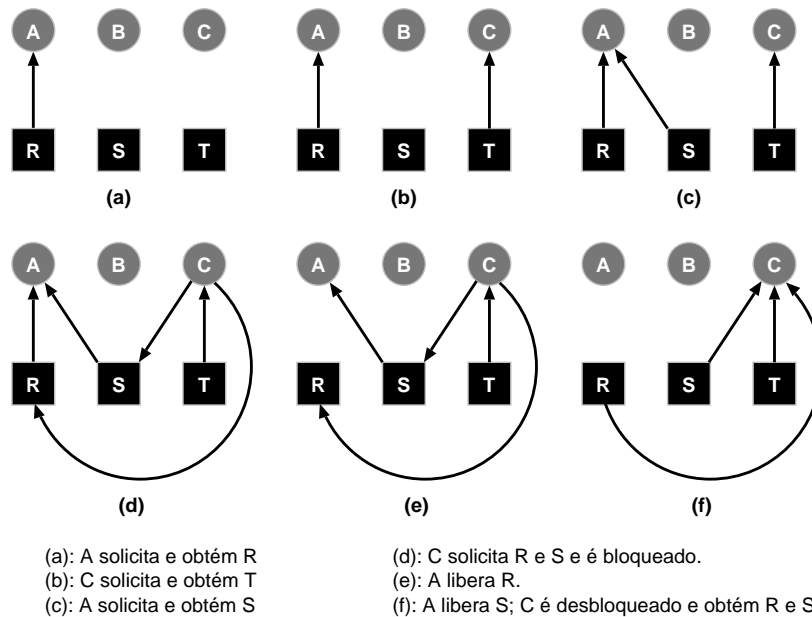
1. Considere a figura dada a seguir em que mostramos um exemplo sem impasses (uma versão animada desta figura foi vista na Aula 7). Suponha que no passo (d) C tenha solicitado *S* em vez de *R*. Isso levará a um impasse? E supondo que C tenha solicitado tanto *S* quanto *R*?



Resp.: -Se C tivesse, no passo (d), solicitado *S* em vez de *R*, não teríamos um impasse, pois existe, por exemplo, a seqüência de execuções descrita na figura dada a seguir, que permite que C obtenha os recursos necessários à sua execução:



-Se C tivesse, no passo (d), solicitado tanto S quanto R , também não teríamos um impasse, pois temos a seqüência de execuções dada a seguir em que C eventualmente obtém os recursos de que necessita:

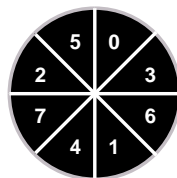


- Em um sistema de transferência eletrônica de fundos, há centenas de processos idênticos que trabalham da seguinte forma. Cada processo lê uma linha de entrada especificando uma quantidade de dinheiro, a conta a ser creditada e a conta a ser debitada. Então, ele bloqueia ambas as contas e transfere o dinheiro, liberando os bloqueios quando tiver

terminado. Com muitos processos executando em paralelo, é possível que, tendo bloqueado a conta x , um processo seja incapaz de bloquear y porque y foi bloqueada por um outro processo agora esperando pelo desbloqueio de x . Esboce um esquema que evite impasses. Não libere um registro de conta até que você tenha completado as transações (em outras palavras, não são permitidas soluções que bloqueiem uma conta e, então, liberem-na imediatamente se a outra estiver bloqueada).

Resp.: Dos métodos vistos na Aula 7, o que satisfaz a condição do problema é o método visto no final da aula, que evita a condição de espera circular através da numeração dos recursos em ordem crescente. Neste método, como vimos, um processo somente poderá solicitar um recurso cujo número é maior do que os de todos os recursos que este processo possui. No problema proposto, os recursos são as contas do banco. Agora, suponha que o processo deseja bloquear duas contas x e y (a transferência pode ser de x para y ou de y para x). Então, o processo bloqueará a conta maior somente quando conseguir bloquear antes a menor. Por exemplo, se x for menor do que y , o processo somente bloqueará y quando conseguir bloquear x com sucesso. Note que a ordem usada é a crescente. Logo, os impasses não poderão mais ocorrer, pois a ordenação dos recursos em ordem crescente, e o impedimento de um processo obter um recurso com número maior do que os dos recursos que ele possui, evitará a condição de espera circular, necessária à ocorrência de impasses.

- Um disco é duplamente intercalado, como na figura dada a seguir, vista na Aula 7. Ele tem oito setores de 512 bytes por trilha e uma velocidade de rotação de 300 rpm. Quanto tempo ele leva para ler todos os setores de uma trilha em ordem, supondo que o braço já esteja corretamente posicionado e $1/2$ rotação seja necessária para que o setor 0 esteja sob o cabeçote? Qual é a taxa de transmissão de dados? Agora repita o problema para um disco não-intercalado com as mesmas características. Compare as taxas de transmissão de dados.



**Disco duplamente intercalado
com 8 setores por trilha**

Resp.: O tempo de 1 rotação do disco será de 0,2s, pois a velocidade

do disco é de 300 rotações por 60s (1 minuto). O tempo necessário para obter o setor 0 sob o cabeçote será então de 0,1s. Já o tempo para ler um setor do disco será de 0,025s, pois temos 8 setores por trilha, e o tempo de 1 rotação é de 0,2s. Agora, como o disco é duplamente intercalado, o tempo para transferir um setor lido será o dobro do tempo para ler o setor, isto é, será de 0,05s. Logo, o tempo total para ler e transferir um setor do disco será de 0,075s.

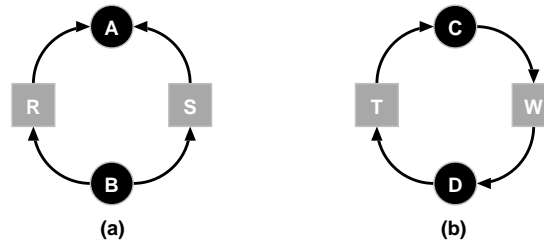
-Como o disco possui intercalação, o próximo setor não será perdido ao transferirmos o setor atual para a memória. Logo, o tempo total de transferência será de 0,1s (o tempo de espera pelo setor 0) mais 8 vezes 0,075 (o tempo para ler e transferir os 8 setores), isto é, será de 0,7s.

-Como transferimos 8 setores de 512 bytes, o número total de bytes transferidos será de 4096. Agora, como o tempo necessário para ler e transferir os 8 setores foi de 0,7s, então a taxa de transmissão de dados será de $4096/0,7$ bytes/s, isto é, será aproximadamente de 5851,428 bytes/s.

-Se o disco não possuir intercalação, então não conseguiremos ler consecutivamente os setores, pois o próximo setor já terá passado sob a cabeça após transferirmos o setor que acabou de ser lido. Logo, depois de ler um setor, deveremos esperar por um tempo de 0,15s, pois como já teremos gasto 0,05s para transferir o setor, não precisaremos esperar por 1 rotação. O tempo total para o setor 0 será o mesmo de antes, isto é, será de 0,1s (o tempo de espera pelo setor 0) mais 0,075s (o tempo para ler e transferir o setor 0), ou seja, será de 0,175s. Já o tempo total para cada um dos outros 7 setores será de 0,15s (o tempo de espera pelo setor) mais 0,075s (o tempo para ler e transferir o setor), isto é, será de 0,225s. Com isso, o tempo total de transferência sem intercalação será 0,175s mais 7 vezes 0,225s, isto é, será de 1,75s. Como a quantidade de bytes transferidos é ainda de 4096, a taxa de transmissão de dados será de $4096/1,75$ bytes/s, isto é, será 2340 bytes/s. Comparada à taxa anterior de 5851,428 bytes/s, vemos que sem o uso da intercalação, a taxa de transferência de dados é reduzida por uma fração aproximada de 0,6, isto é, é reduzida aproximadamente em 60%.

4. Suponha que existam no computador quatro recursos não-preemptivos R, S, T e W, e que um processo possa requisitar mais de um recurso ao mesmo tempo. Suponha ainda, para evitar os impasses, que estes recursos sejam numerados segundo a ordem alfabética de seus nomes, e que um processo somente possa requisitar mais recursos se todos eles possuírem números maiores do que os números de todos os recursos que

ele possui. Os grafos de recursos dados nas partes (a) e (b) da figura a seguir podem ocorrer? Justifique a sua resposta.



Resp.: Como vimos na Aula 7, numerar os recursos em ordem crescente e permitir que um processo somente possa solicitar os recursos cujos números, segundo esta ordenação, sejam maiores do que os números de todos os recursos já alocados a ele evita a condição de espera circular. Esta condição, como vimos na Aula 7, é representada no grafo de recursos por um ciclo orientado, que também identifica um impasse. No grafo da parte (a) da figura temos um ciclo que não é orientado e, portanto, este grafo pode ocorrer. Note que ele somente representa um cenário onde o processo A possui os recursos R e S e o processo B deseja obter estes recursos. Já o grafo da figura (b) não pode ocorrer, pois temos um ciclo orientado que representa uma condição de espera circular (e um impasse que não poderia ocorrer devido a esta condição ser evitada). Note que como o número do recurso T é menor do que o número do recurso W, então D não pode solicitar W e depois disso ser bloqueado.