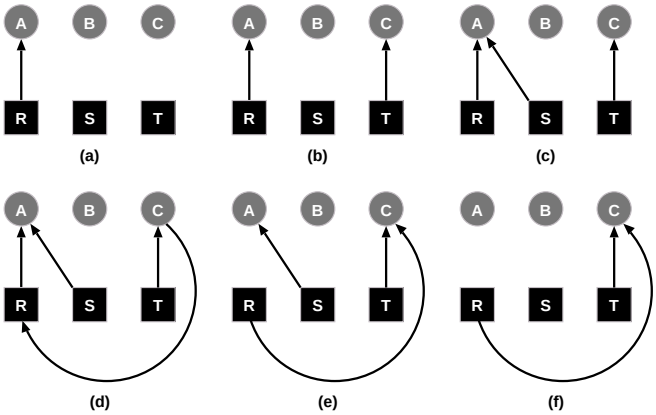
Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
**Professores:** Valmir C. Barbosa e Felipe M. G. França
**Assistente:** Alexandre H. L. Porto

Quarto Período
AD2 - Segundo Semestre de 2007

Nome -
Assinatura -

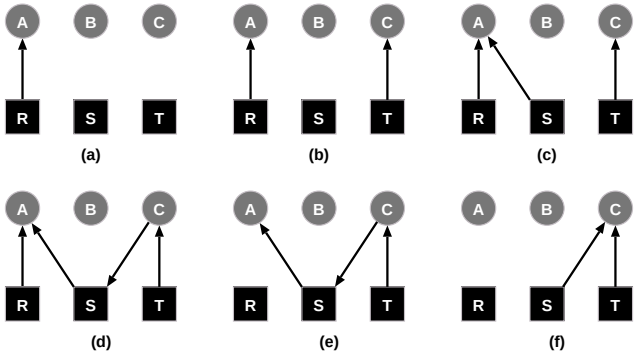
---

1. (2.0) 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 levaria a um impasse? E supondo que ele tenha solicitado tanto $S$ quanto $R$?



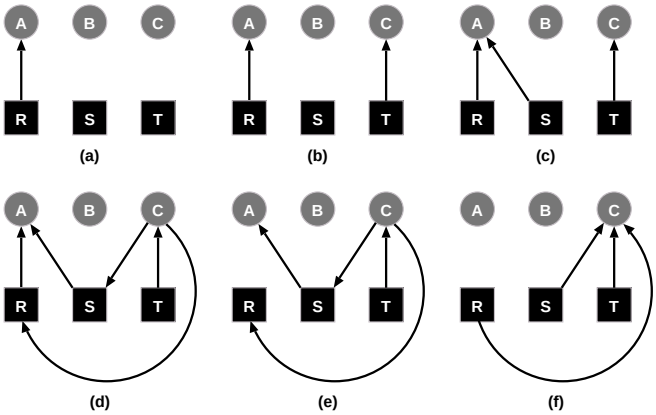

(a): A solicita e obtém R
(b): C solicita e obtém T
(c): A solicita e obtém S
(d): C solicita R e é bloqueado.
(e): A libera R; C é desbloqueado e obtém R.
(f): A libera S.

**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:



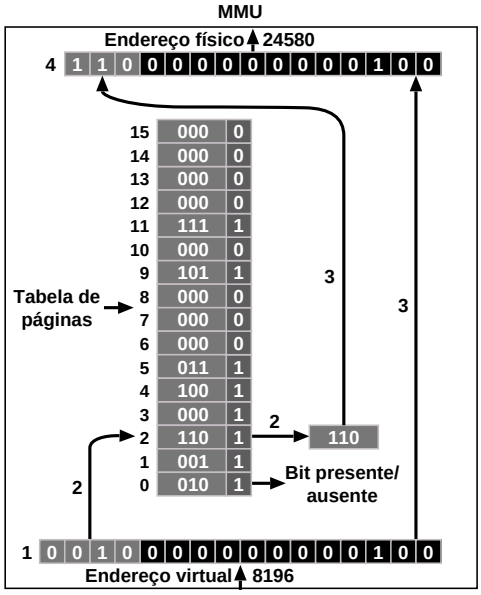

(a): A solicita e obtém R
(b): C solicita e obtém T
(c): A solicita e obtém S
(d): C solicita S e é bloqueado.
(e): A libera R.
(f): A libera S; C obtém S e é desbloqueado.

-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:




(a): A solicita e obtém R
(b): C solicita e obtém T
(c): A solicita e obtém S
(d): C solicita R e S e é bloqueado.
(e): A libera R.
(f): A libera S; C é desbloqueado e obtém R e S

2. (2.0) Na figura dada a seguir (vista no final da Aula 8) o identificador de página do endereço virtual tem 4 bits e o do endereço físico tem 3 bits. Em geral, como se relacionam os tamanhos desses dois identificadores? Discuta a sua resposta.




**1: O endereço virtual (8196) é passado a MMU.**
**2: A página virtual (2) é usada como um índice na tabela de páginas para obter a moldura da página (6).**
**3: A moldura de página (6) e o deslocamento (4) são usados para compor o endereço físico (24580).**
**4: O endereço físico (24580) é passado à memória pela MMU.**

**Resp.:** Se o identificador de página do endereço virtual possuir mais bits do que o do físico, então o espaço de endereçamento virtual será maior do que o físico, e com isso, a paginação será necessária, pois precisaremos mapear endereços virtuais em endereços físicos. Se esse identificador possuir menos bits, a paginação não será necessária, pois como o espaço de endereçamento virtual é menor do que o físico, poderemos sempre carregar todo o processo na memória ao executá-lo (supondo que, se for necessário, sempre teremos espaço na memória para o sistema operacional). Finalmente, se os dois identificadores forem iguais, a paginação será necessária somente se o processo precisar compartilhar a memória com o sistema operacional.

3. (2.0) Se a substituição de página FIFO for utilizada com quatro molduras de página e oito páginas, quantas falhas de página ocorrerão com a cadeia de referências 0172327103 se as quatro molduras estão inicialmente vazias? Agora repita esse problema para LRU.

**Resp.:**-Ao usarmos o algoritmo FIFO, teremos um total de 6 falhas de página, como podemos ver pela tabela dada a seguir, em que as páginas são mostradas, na segunda coluna, na ordem em que foram carregadas:

| Página referenciada | Conteúdo das molduras de página | | | | Ocorreu uma falha? |
|---|---|---|---|---|---|
| 0 | 0 | | | | Sim |
| 1 | 0 | 1 | | | Sim |
| 7 | 0 | 1 | 7 | | Sim |
| 2 | 0 | 1 | 7 | 2 | Sim |
| 3 | 1 | 7 | 2 | 3 | Sim |
| 2 | 1 | 7 | 2 | 3 | Não |
| 7 | 1 | 7 | 2 | 3 | Não |
| 1 | 1 | 7 | 2 | 3 | Não |
| 0 | 7 | 2 | 3 | 0 | Sim |
| 3 | 7 | 2 | 3 | 0 | Não |

-Já ao usarmos o algoritmo LRU, teremos um total de 7 falhas de página, como podemos ver pela tabela dada a seguir, em que as páginas são mostradas, na segunda coluna, na ordem em que foram referenciadas pela última vez:

| Página referenciada | Conteúdo das molduras de página | | | | Ocorreu uma falha? |
|---|---|---|---|---|---|
| 0 | 0 | | | | Sim |
| 1 | 0 | 1 | | | Sim |
| 7 | 0 | 1 | 7 | | Sim |
| 2 | 0 | 1 | 7 | 2 | Sim |
| 3 | 1 | 7 | 2 | 3 | Sim |
| 2 | 1 | 7 | 3 | 2 | Não |
| 7 | 1 | 3 | 2 | 7 | Não |
| 1 | 3 | 2 | 7 | 1 | Não |
| 0 | 2 | 7 | 1 | 0 | Sim |
| 3 | 7 | 1 | 0 | 3 | Sim |

4. (2.0) Foi observado que o número de instruções executadas entre falhas de página é diretamente proporcional ao número de molduras de página alocadas para um programa. Se a memória disponível é dobrada, o intervalo médio entre falhas de página também dobra. Suponha que uma instrução normal leve 1 microssegundo, mas se uma falha de página ocorre, ela leva 2001 microssegundos (isto é, 2ms para tratar a falha). Se um programa levasse 60s para executar, tempo durante o qual ocorressem 15000 falhas de página, quanto levaria para executar se o dobro da memória estivesse disponível?

   **Resp.:** Como tivemos 15000 falhas de página, e como o tempo para tratar de cada falha é de 2000 microssegundos, então gastamos um total de 30000000 de microssegundos, ou 30 segundos, para tratar das falhas de página. Logo, o tempo gasto somente para executar as instruções do programa foi também de 30 segundos. Agora, com o dobro de memória gastaremos a metade do tempo gasto anteriormente para tratar das falhas, ou seja, 15 segundos, já que temos a metade do número de falhas. Com isso, o tempo total de execução do programa será agora de 45 segundos, pois o tempo de execução das instruções do programa não varia com o aumento da memória.

5. (1.0) A alocação contígua de arquivos leva à fragmentação de disco, como mencionado na Aula 11. Essa fragmentação é interna ou externa? Faça uma analogia com algo discutido nas aulas de gerenciamento de memória.

   **Resp.:** Na alocação contígua cada arquivo é armazenado em um conjunto consecutivo de blocos do disco. Logo, a fragmentação é externa, pois a contínua criação e deleção dos arquivos tenderá a fazer com que os arquivos estejam dispersos por todo o disco, separados por pequenos conjuntos contíguos de blocos. Como vimos nas aulas sobre gerenciamento de memória, este tipo de fragmentação ocorre no gerenciamento por troca e por segmentação, devido à contínua alocação e desalocação de, respectivamente, partições e segmentos.

6. (1.0) Quando um arquivo é removido, seus blocos geralmente são devolvidos à lista de livres, mas eles não são apagados. Você acha que

seria uma boa idéia ter o sistema operacional apagando cada bloco antes de liberá-lo? Considere os fatores segurança e desempenho em sua resposta e explique o efeito de cada um.

**Resp.:** Em relação à segurança do sistema, isto seria o ideal, pois caso o bloco devolvido à lista de livres pertencesse a um arquivo com informações confidenciais, estas ficariam expostas aos usuários mal intencionados. Neste caso, um usuário mal intencionado poderia obter parte desta informação caso o bloco fosse de algum modo alocado a um arquivo criado por este usuário. Mas o desempenho do sistema seria degradado, pois como precisaríamos fazer uma escrita de um bloco no disco sempre que este bloco fosse devolvido à lista de livres, o número de escritas no disco aumentaria consideravelmente. Uma idéia interessante, implementada no mainframe Burroughs, foi a de apagar somente os blocos dos arquivos marcados como críticos pelo usuário, o que permite manter o sistema seguro sem reduzir muito o seu desempenho.