



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 AD2 - Segundo Semestre de 2018

**Atenção:** Cada aluno é responsável por redigir suas próprias respostas. Provas iguais umas às outras terão suas notas diminuídas. As diminuições nas notas ocorrerão em proporção à similaridade entre as respostas. Exemplo: Três alunos que respondam identicamente a uma mesma questão terão, cada um, 1/3 dos pontos daquela questão.

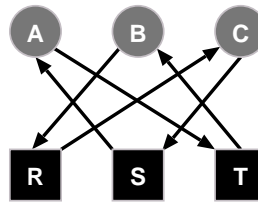
Nome -  
Assinatura -

---

1. (1,5) Suponha que após algumas alocações de recursos tenha sido criado um impasse, representado pelo ciclo orientado  $A \rightarrow T \rightarrow B \rightarrow R \rightarrow C \rightarrow S \rightarrow A$  no grafo de recursos, sendo que A, B e C são processos e R, S e T são recursos. Responda:
  - (a) (0,5) Dê o grafo de recursos supondo que somente as solicitações que geraram o ciclo foram feitas.

**Resp.:** Na figura a seguir damos o grafo de recursos considerando somente as solicitações dadas no ciclo do enunciado, ou seja, que

os processos A, B e C solicitaram e obtiveram, respectivamente, os recursos S, T e R, e que A, B e C foram bloqueados ao solicitarem, respectivamente, T, R e S.



- (b) (1,0) Quantos recursos, no mínimo, devem se tornar preemptivos para evitar um impasse similar ao dado no enunciado (ou seja, um impasse envolvendo todos os recursos)? E se desejarmos evitar completamente os impasses? Justifique a sua resposta.

**Resp.:** -No mínimo um, porque um recurso preemptivo não pode fazer parte de um impasse devido a poder ser removido do processo que o possui sem comprometer a sua computação. Além disso, se também for compartilhável, pode ser alocado a todos os processos que desejem obtê-lo.

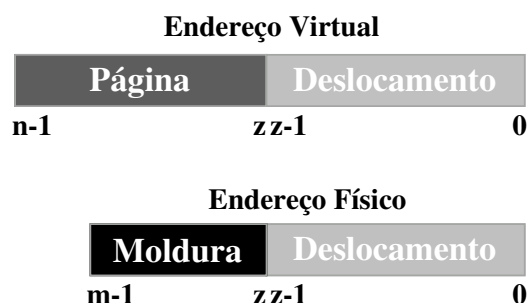
-Agora, para evitar completamente os impasses, como é necessário que um ciclo possua pelo menos dois vértices associados a recursos não-preemptivos, e como temos três recursos, então no mínimo dois recursos devem passar a ser preemptivos.

2. (1,0) Suponha que a memória virtual tenha  $2^n$  bytes, que a memória física tenha  $2^m$  bytes, com  $m \leq n$ , e que a página virtual tenha  $2^z$  bytes, com  $z \leq m$ . Como o endereço virtual, representado em binário, é dividido nos campos número da página virtual e deslocamento? E como o endereço físico, também representado em binário, é dividido nos campos número da moldura de página e deslocamento? Justifique a sua resposta.

**Resp.:** -Em relação à divisão do endereço virtual, como o espaço de endereçamento virtual tem  $2^n$  bytes, então o endereço usa  $n$  bits. Agora,

como o tamanho da página virtual é de  $2^z$  bytes, ou seja, é representado por  $z$  bits, então o campo número da página virtual usa  $n - z$  bits do endereço virtual e o campo deslocamento dentro desta página usa  $z$  bits desse endereço.

-Já em relação ao endereço físico, como o espaço de endereçamento físico tem  $2^m$  bytes, então o endereço usa  $m$  bits. Além disso, como o tamanho da moldura de página é igual ao da página virtual, de  $2^z$  bytes, então o campo número da moldura de página usa  $m - z$  bits do endereço físico e o campo deslocamento dentro desta moldura usa  $z$  bits. Na figura a seguir mostramos a divisão dos endereços virtuais e físicos.



3. (3,0) Suponha que as páginas virtuais 1, 2, 3, 0, 1, 0, 4, 3, 0, 2, 3, 4 tenham sido acessadas, na ordem dada, por dois processos A e B, sendo que A acessou as páginas 2 e 4 e B as páginas 0, 1 e 3. Suponha ainda que quatro molduras, inicialmente vazias, estavam disponíveis para serem usadas por A e B. Se o algoritmo FIFO com uma política de alocação global foi usado, quantas falhas de página foram geradas por A e por B? E se foi usado o algoritmo LRU com uma política de alocação local e igualitária? Justifique a sua resposta.

**Resp.:** -Como vimos na aula 9, no algoritmo FIFO as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo da sua cópia para a memória. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página copiada há mais tempo para a memória. Além disso, quando a política de alocação global é usada, todas as quatro molduras precisam ser consideradas, pois elas

não são alocadas com exclusividade a A ou a B. Na tabela dada a seguir mostramos, em cada linha, o que ocorre ao acessarmos as páginas na ordem dada no enunciado. Para cada uma dessas linhas mostramos, na primeira coluna, a página que é acessada e o processo que a acessa entre parênteses. Já na segunda coluna mostramos a ordem em que as páginas devem ser escolhidas. Finalmente, na última coluna, mostramos se o acesso à página gerou uma falha de página. Como podemos ver pela tabela, foram geradas pelos processos A e B, respectivamente, 2 e 3 falhas.

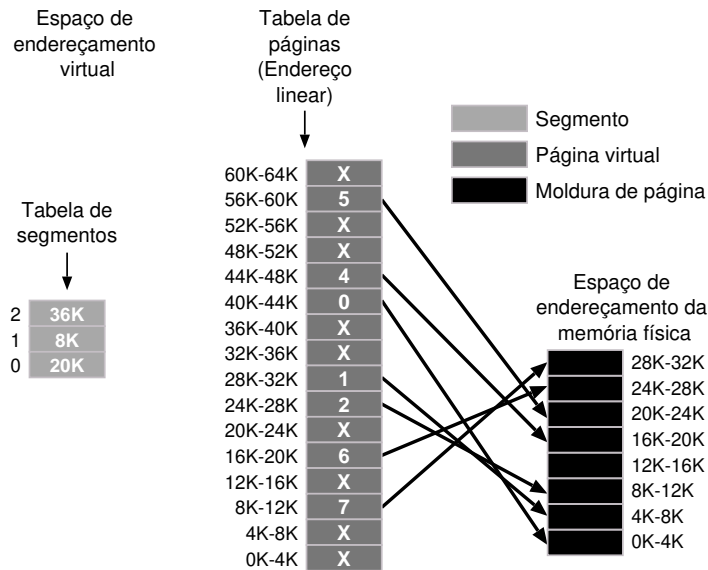
Páginas	Ordenação	Ocorreu uma falha?
1 (B)	1	Sim
2 (A)	1 2	Sim
3 (B)	1 2 3	Sim
0 (B)	1 2 3 0	Sim
1 (B)	1 2 3 0	Não
0 (B)	1 2 3 0	Não
4 (A)	2 3 0 4	Sim
3 (B)	2 3 0 4	Não
0 (B)	2 3 0 4	Não
2 (A)	2 3 0 4	Não
3 (B)	2 3 0 4	Não
4 (A)	2 3 0 4	Não

-Como vimos na aula 9, no algoritmo LRU as páginas são primeiramente ordenadas, em ordem crescente, de acordo com o tempo do seu último acesso. A página a ser substituída é a primeira página segundo essa ordenação, isto é, a página não acessada há mais tempo. Finalmente, quando a política de alocação local e igualitária é usada, 2 molduras de página são alocadas a cada processo, já que temos quatro molduras no total. Para o processo A, como ele somente acessou as páginas 2 e 4 então independentemente do algoritmo de substituição ser o LRU, somente 2 falhas de página foram geradas. Já para o processo B, como ele acessou as páginas 0, 1 e 3, precisamos usar o algoritmo LRU para a sequência de páginas 1, 3, 0, 1, 0, 3, 0, 3. A tabela a

seguir é similar à anterior, mas agora mostra, na segunda coluna, a ordem de escolha segundo o algoritmo LRU. Como podemos ver pela tabela, ocorreram 5 falhas após B acessar as páginas. Logo, podemos concluir que A gerou o mesmo número de falhas e B gerou mais 2 falhas.

Páginas	Ordenação		Ocorreu uma falha?
1	1		Sim
3	1	3	Sim
0	3	0	Sim
1	0	1	Sim
0	1	0	Não
3	0	3	Sim
0	3	0	Não
3	0	3	Não

4. (1,5) Suponha que um computador utilize uma técnica de segmentação com paginação similar à do processador Pentium da Intel, e que cada endereço virtual bidimensional seja representado como um par (segmento, endereço dentro do segmento). Na figura a seguir mostramos, na tabela de segmentos, o endereço linear inicial no qual cada segmento usado por um processo é mapeado. Nesta figura também mostramos, na tabela de páginas do espaço de endereçamento linear, como as páginas virtuais são mapeadas nas molduras de página da memória física. Usando essas tabelas, e supondo que cada segmento tenha o maior tamanho possível, responda, justificando a sua resposta:



- (a) (0,5) Dê o endereço linear correspondente ao endereço virtual (1, 1234) e, caso tal endereço linear esteja mapeado, o endereço físico no qual ele está mapeado.

**Resp.:** Pela figura, vemos que o segmento 1 está armazenado a partir do endereço linear 8K ou 8192. Logo, o endereço virtual (1, 1234) está associado ao endereço linear  $8192 + 1234 = 9426$ . Como o espaço de endereçamento linear de 64K é dividido em 16 páginas virtuais de 4K então, como o endereço linear 9426 em binário é  $0010 | 010011010010$ , sendo que “|” mostra o ponto de divisão dos campos número da página virtual e deslocamento, vemos que o endereço linear está armazenado na página 2 com o deslocamento de 1234. Agora, como a página 2 está mapeada na moldura de página 7, que começa no endereço físico 28672, então o endereço físico no qual o endereço linear 9426 associado ao endereço virtual (1, 1234) está mapeado é 28672 (o endereço inicial da moldura) mais 1234 (o deslocamento dentro da moldura que é igual ao deslocamento dentro da página), ou seja, 29906.

- (b) (0,5) Se existir algum endereço linear mapeado no endereço físico 16632, então dê esse endereço e o endereço virtual bidimensional

associado a ele.

**Resp.:** Pela figura, vemos que o espaço de endereçamento físico de 32K está dividido em 8 molduras de página de 4K. Agora, usando o equivalente em binário  $100 \mid 000011111000$  do endereço físico 16632, sendo que “|” mostra o ponto de divisão dos campos número da moldura de página e deslocamento, vemos que ele está dentro da moldura 4 com o deslocamento 248. Como, pela figura, vemos que a moldura 4 está mapeando a página virtual 11, que começa no endereço linear 45056, então o endereço físico 16632 está mapeando o endereço linear  $45056 + 248 = 45304$  (pois o deslocamento dentro da página é igual ao dentro da moldura). Finalmente, como cada segmento ocupa todo o espaço disponível, vemos que o endereço linear 45304 pertence ao segmento 2, que começa no endereço linear 36K ou 36864. Logo, podemos concluir que a posição do endereço linear dentro do segmento é  $45304 - 36864 = 8440$ , significando que o endereço linear 45304 está associado ao endereço virtual (2, 8440) o que, por sua vez, significa que o endereço físico 16632 está mapeando o endereço virtual (2, 8440).

- (c) (0,5) Quais faixas de endereços lineares associadas a endereços virtuais do segmento 0 geram falhas de página quando acessadas?

**Resp.:** Pela figura e pelo enunciado, vemos que o segmento 0 está armazenado nos endereços lineares de 20K ou 20480 (o início do segmento) até 36K-1 ou 36863 (cada segmento tem o maior tamanho possível e o segmento 2 começa no endereço linear 36K). Também pela figura, vemos que os endereços lineares de 20480 até 36863 estão armazenados nas páginas virtuais de 5 até 8, e que as páginas 5 e 8 não estão mapeadas na memória. Logo, ocorrerão falhas de página se forem acessados os endereços lineares de 20K ou 20480 até 24K-1 ou 24575, ou de 32K ou 32768 até 36K-1 ou 36863.

5. (2,0) Suponha que um disco tenha  $2^n$  blocos, numerados de 1 até  $2^n$ ,

com  $n > 1$ , que um arquivo A use  $1/4$  dos blocos ímpares iniciais do disco, e que um arquivo B use  $1/2$  dos blocos pares finais do disco.

**Obs.:** Devido ao enunciado da questão não estar muito claro em relação aos tamanhos dos arquivos A e B serem baseados no tamanho total do disco de  $2^n$  ou no total de blocos pares ou ímpares, ambos iguais a  $2^{n-1}$ , vamos considerar como correta uma das seguintes respostas: (1) a que considere que os tamanhos de A e B são baseados no tamanho total do disco, ou seja, A usa  $1/4$  dos blocos do disco, sendo todos eles ímpares, ou seja,  $2^{n-2}$  blocos ímpares do disco, e B usa  $1/2$  dos blocos do disco, sendo todos eles pares, ou seja,  $2^{n-1}$  blocos pares do disco; ou (2) a que considere que A usa  $1/4$  do total de blocos ímpares do disco, ou seja,  $2^{n-3}$  blocos ímpares, e B usa  $1/2$  do total de blocos pares do disco, ou seja,  $2^{n-2}$  blocos pares.

- (a) (1,0) Dê o mapa de bits do disco, supondo que somente os arquivos A e B estejam armazenados nele.

**Resp.1:** Como o arquivo A usa  $1/4$  blocos do disco, ou  $2^{n-2}$  blocos, e como eles são todos iniciais e ímpares, então foram alocados a A os blocos  $1, 3, 5, \dots, 2^{n-1} - 1$  (a metade dos blocos ímpares). Já o arquivo B, como ele usa  $1/2$  blocos do disco, ou  $2^{n-1}$  blocos, e como todos eles são finais e pares, então foram alocados a B os blocos  $2, 4, 6, \dots, 2^n$  (todos os blocos pares). Agora, como existe no mapa de bits um bit para cada bloco do disco, e como o bit é 0 se o bloco associado ao bit estiver sendo usado ou 1 se ele estiver livre, então os bits  $1, \dots, 2^{n-1}, 2^{n-1} + 2, 2^{n-1} + 4, \dots, 2^n$  serão todos iguais a 0, e os bits restantes do mapa de bits, ou seja, os bits  $2^{n-1} + 1, 2^{n-1} + 3, \dots, 2^n - 1$  serão todos iguais a 1.

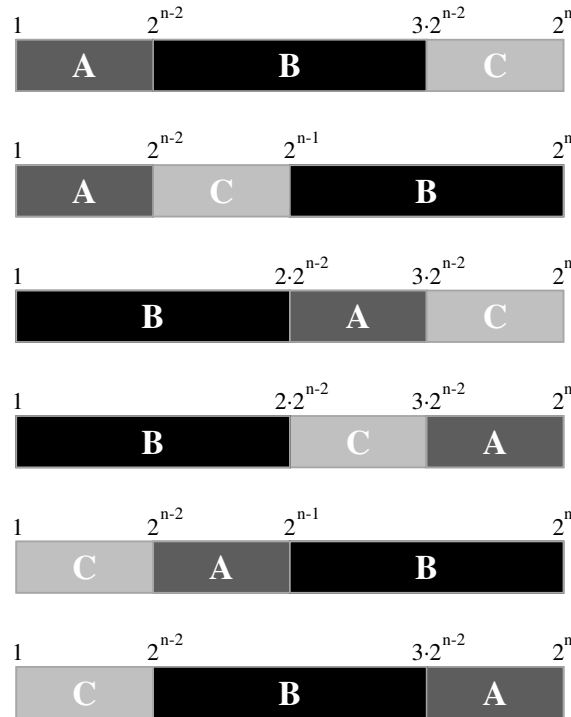
**Resp.2:** Como o arquivo A usa  $1/4$  blocos ímpares iniciais do disco, ou  $2^{n-3}$  blocos, então foram alocados a A os blocos  $1, 3, 5, \dots, 2^{n-2} - 1$ . Já o arquivo B, como ele usa os  $1/2$  blocos pares finais do disco, ou  $2^{n-2}$  blocos, então foram alocados a B os blocos  $2^{n-1} + 2, 2^{n-1} + 4, 2^{n-1} + 6, \dots, 2^n$ . Agora, como existe no mapa de bits um bit para cada bloco do disco, e como o bit é 0 se o bloco



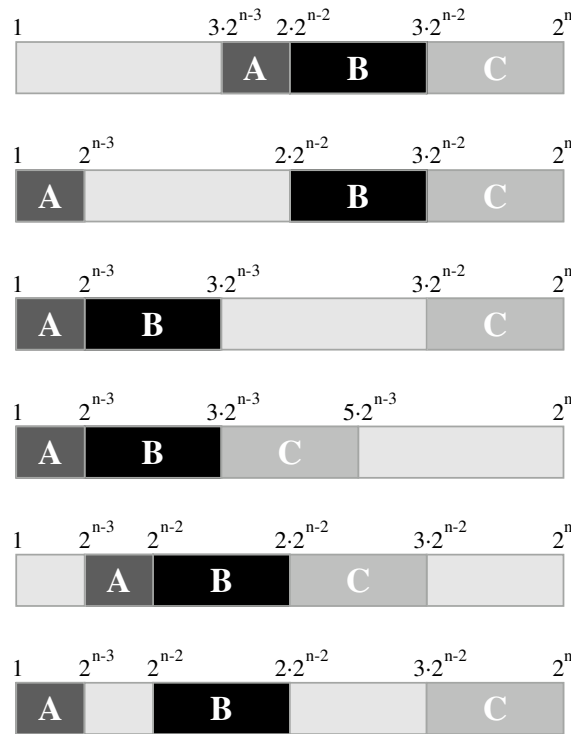
associado ao bit estiver sendo usado ou 1 se ele estiver livre, então os bits  $1, 3, 5, \dots, 2^{n-2} - 1$  e  $2^{n-1} + 2, 2^{n-1} + 4, 2^{n-1} + 6, \dots, 2^n$  serão todos iguais a 0, e os bits restantes do mapa de bits, ou seja, os bits  $2, 4, 6, \dots, 2^{n-2}$  e  $2^{n-1} + 1, 2^{n-1} + 3, 2^{n-1} + 5, \dots, 2^n - 1$  serão todos iguais a 1.

- (b) (1,0) Se a alocação contígua passar a ser usada, diga todos os modos de armazenar A e B que permitam a um novo arquivo C usar 1/4 dos blocos do disco.

**Resp.1:** Como A usa 1/4 dos blocos do disco, e como B usa 1/2 dos blocos disco, implicando em que existem exatamente 1/4 (o tamanho de C) de blocos livres no disco, então deveremos armazenar A, B e C consecutivamente em qualquer ordem a partir do início do disco. Nas figuras a seguir mostramos todos os modos de armazenarmos A, B e C no disco, sendo que os blocos consecutivos de cada arquivo são identificados pelo seu nome.



**Resp.2:** Como A usa  $1/4$  dos blocos ímpares do disco ou  $1/8$  dos blocos do disco, como B usa  $1/2$  dos blocos pares do disco ou  $1/4$  dos blocos do disco, implicando em que existem  $5/8$  de blocos livres no disco, e como C precisa usar  $1/4$  dos blocos do disco então, após armazenarmos A, B e C, teremos  $3/8$  de blocos livres no disco. Logo, deveremos armazenar A, B e C em qualquer ordem no disco, com o espaço livre distribuído entre as posições nas quais A, B e C foram armazenados no disco. Como nesta resposta podemos ter muitos modos de armazenar os arquivos no disco, os professores decidiram, devido à opção de resposta 1 ter somente seis modos, aceitar também seis dos possíveis modos nesta resposta. Nas figuras a seguir mostramos seis dos possíveis modos de armazenarmos A, B e C, sendo que os blocos consecutivos de cada arquivo são identificados pelo seu nome.



6. (1,0) Suponha que o sistema operacional use o mecanismo de proteção por lista de controle de acesso ao gerenciar o acesso aos objetos do

sistema de arquivos pelos processos. Descreva o significado de cada uma das permissões dadas na tabela a seguir.

Objeto	Lista com os dominios e as operações
notasad1.txt	(* , alunos, R-), (* , prof, RW-), (guest, *, ---)
aula12.ppt	(* , *, R-X), (professor, cederj, RWX)
AD2.pdf	(linux, so, --X), (aluno, cederj, -W-), (aluno,so, -WX)
linux.exe	(* , so, --X), (* , adm, RWX)

**Resp.:** Como vimos na aula 12, quando uma lista de controle de acesso é usada, cada entrada da lista representa um objeto do sistema, e o seu conjunto de ternos informa o que certos usuários do sistema podem fazer com o objeto. Cada terno é composto por um usuário do sistema, o grupo ao qual esse usuário pertence, e as operações que podem ser executadas sobre o objeto. Quando o usuário for “\*”, isso significa que as operações são válidas para todos os usuários do grupo. Já se o grupo for “\*”, isso significa que as operações são válidas para todos os grupos do mesmo usuário (note que um “\*” no usuário e no grupo indica que as operações são válidas para todos os usuários do sistema). Finalmente, o campo de operações é composto por uma string com três caracteres, sendo que o primeiro caractere indica se o objeto pode (R) ou não (-) ser lido, o segundo caractere indica se o objeto pode (W) ou não (-) ser escrito e, finalmente, o terceiro caractere indica se o objeto pode (X) ou não (-) ser executado. A seguir mostramos o significado das entradas para cada objeto, que no nosso caso são todos arquivos:

- Arquivo notasad1.txt: o primeiro terno diz que qualquer usuário do grupo alunos pode somente ler o arquivo. Já o segundo terno diz que qualquer usuário do grupo prof pode ler ou escrever no arquivo. Finalmente, o terceiro terno diz que o usuário guest de qualquer grupo não pode fazer nenhum tipo de acesso ao arquivo. Além disso, nenhum outro usuário, além dos descritos, pode ler, escrever ou executar o arquivo (note que isso implica que o terceiro terno não é necessário).
- Arquivo aula12.ppt: o primeiro terno diz que qualquer usuário de qualquer grupo pode ler ou executar o arquivo, enquanto que o segundo terno diz que o usuário professor do grupo cederj pode ler, escrever e executar o arquivo (este terno substitui o primeiro terno mais genérico).

- Arquivo AD2.pdf: o primeiro terno diz que usuário linux do grupo so pode somente executar o arquivo. Já o segundo terno diz que o usuário aluno do grupo cederj pode somente escrever no arquivo. Finalmente, o terceiro terno diz que o usuário aluno do grupo so pode escrever e executar o arquivo. Assim como antes, nenhum outro usuário, além dos descritos, pode executar operações no arquivo.
- Arquivo linux.exe: o primeiro terno diz que qualquer usuário do grupo so pode somente executar o arquivo, enquanto que o segundo terno diz que qualquer usuário do grupo adm pode ler, escrever ou executar o arquivo. Finalmente, como antes, nenhum usuário, além dos descritos, pode ler, escrever ou executar o arquivo.