

LISTA DE EXERCÍCIOS PARA P1 – INCLUI PROVAS ATÉ 2007/2

MULTIPROGRAMAÇÃO

1) Responda sim ou não para as seguintes questões:

- a) A operação "passa para modo usuário" deve ou não ser privilegiada?
- b) A operação "desabilita interrupções" deve ou não ser privilegiada?
- c) A operação "escreve caractere na interface da impressora" deve ou não ser privilegiada?
- d) A operação "desliga o temporizador" deve ou não ser privilegiada?
- e) Pode-se considerar como consequência da multiprogramação "uma pior/melhor utilização do processador"?
- f) Pode-se considerar como consequência da multiprogramação "uma pior/melhor utilização dos periféricos"?
- g) Pode-se considerar como consequência da multiprogramação "uma menor/maior necessidade de memória"?
- h) Pode-se considerar como consequência da multiprogramação "uma menor/maior necessidade de hardware para proteção"?

2) Em qual das situações abaixo não é necessário ocorrer a passagem do processador de modo usuário para modo supervisor ?

- a) Controlador de disco gera interrupção avisando que comando anterior foi concluído.
- b) Processo faz uma chamada de sistema.
- c) Processo chama uma rotina da biblioteca da linguagem.
- d) Processo executa um opcode que na verdade não existe naquele processador.
- e) Timer gera uma interrupção alertando a passagem de mais 10 ms.

3) Os três principais estados de um processo são:

- Esperando pelo processador (ready);
- Executando (running);
- Esperando pela entrada/saída (blocked).

Descreva o evento que faz com que um processo mude de "ready" para "running".

4) O sistema operacional é um programa dirigido por eventos, e esses eventos são sinalizados por interrupções. Para cada uma das três classes de interrupções (periférico, proteção, chamada de sistema), descreva a reação que o sistema operacional deverá ter. Em outras palavras, o que o sistema operacional deverá fazer em função do evento sinalizado.

5) O sistema operacional é um programa dirigido por eventos, e esses eventos são sinalizados por interrupções. O que o sistema operacional deverá fazer em função do evento sinalizado por uma

interrupção de periférico (por exemplo, do controlador de disco) e qual o impacto dela sobre o estado dos processos no sistema ?

6) O sistema operacional é um programa dirigido por eventos, e esses eventos são sinalizados por interrupções. O que o sistema operacional deverá fazer em função do evento sinalizado por uma interrupção do timer e qual o impacto dela sobre o estado dos processos no sistema ?

7) Explique como o mecanismo de modos de execução do processador, associado com o mecanismo de interrupções, pode impedir que um processo executando código de usuário possa, por exemplo, acessar diretamente o controlador do disco.

8) Explique em que situações (que tipos de interrupções) ocorre a passagem do processador de modo usuário para modo supervisor.

9) Explique por que é vantajoso associar a passagem do processador de modo usuário para modo supervisor com o atendimento de uma interrupção.

10) Assim como o processador, outros recursos do computador podem ser gerenciados de tal forma que os mesmos possam ou não possam ser preemptados, isto é, o recurso é retirado temporariamente do processo. Discuta a possibilidade de preemptar a memória de um processo. Em que situação isto é desejável ? Como o processo pode ser retomado mais tarde sem inconsistência ?

11) Considere um microkernel simples, que executa com interrupções desabilitadas, implementa multiprogramação e algumas poucas chamadas de sistema. O algoritmo de escalonamento é prioridade fixa preemptiva. Esboce, através de um fluxograma, o comportamento deste microkernel na ocorrência de interrupção de software (chamada de sistema) e de interrupção de hardware (gerada por periférico). Considere no fluxograma 2 chamadas de sistema: **sleep** (suspende o processo por alguns segundos) e **rxchar** (recebe 1 caractere pela porta serial). Apenas dois tipos de interrupções de hardware são possíveis: uma do temporizador em hardware avisando a passagem de 100ms e outra da porta serial avisando a chegada de um caractere.

Descrever o fluxograma supondo a existência das seguintes funções: `insere_fila_x`, `remove_fila_x`, `salva_contexto`, `coloca_contexto`, `seleciona_processo`. Supor a existência de quaisquer funções auxiliares necessárias.

12) Tipicamente, na ocorrência de uma interrupção, os processadores automaticamente desabilitam interrupções, para permitir que o tratador da interrupção execute de forma segura. Entretanto, a maioria dos sistemas operacionais torna a habilitar interrupções tão logo quanto possível, mesmo não tendo ainda terminado a execução de todas as ações desencadeadas pela ocorrência da interrupção sendo tratada. Ilustre através de exemplo uma potencial desvantagem de manter as interrupções desabilitadas além do estritamente necessário.

13) É possível usar a própria pilha do processo para salvar seu contexto de execução ? Comente as vantagens e desvantagens de tal solução.

GERÊNCIA DO PROCESSADOR

14) Quatro programas devem ser executados em um computador. Todos os programas são compostos por 2 ciclos de processador e 2 ciclos de E/S. A entrada e saída de todos os programas é feita sobre a mesma unidade de disco. Os tempos para cada ciclo de cada programa são mostrados abaixo:

Programa	Processador	Disco	Processador	Disco
P1	3	10	3	12
P2	4	10	6	8
P3	7	8	7	10
P4	5	12	2	8

Construa um diagrama de tempo mostrando qual programa está ocupando o processador e o disco a cada momento, até que os 4 programas terminem. Suponha que o algoritmo de escalonamento utilizado no processador seja fatia de tempo, com fatias de 4 unidades. Suponha que as solicitações de acesso ao disco sejam atendidas na ordem de chegada (FCFS).

15) Mostre através de um exemplo numérico a razão de ser genericamente melhor favorecer os processos "*i/o-bound*" na disputa pelo processador.

16) Considere um sistema com um escalonador por fatia de tempo e processos em sua maioria interativos (um editor de texto por exemplo). Suponha que, na média, um processo seja executado por 10 ms até gerar uma solicitação de E/S com bloqueio. Discuta qual seria o valor de fatia de tempo mais apropriado ?

17) Considere um sistema com um escalonador por fatia de tempo e processos em sua maioria interativos (um editor de texto por exemplo). Suponha que, na média, um processo seja executado por 10 ms até gerar uma solicitação de E/S com bloqueio. Suponha que 90% dos ciclos de processador tem uma duração entre 8 ms e 15 ms. Do ponto de vista do escalonamento, qual o efeito (overhead e tempo de resposta) de:

- a) Uma fatia de tempo pequena, como por exemplo 1 ms.
- b) Uma fatia de tempo grande, como por exemplo 100 ms.

18) Considerando a fila do processador mostrada abaixo, calcule o instante no qual cada processo conclui o seu ciclo de processador, caso o algoritmo de escalonamento utilizado seja:

- (a) Fatias de tempo com tamanho 3;
- (b) FIFO (first-in-first-out).

Processo Duração do próximo ciclo de processador

A	6
B	5
C	7
D	3
E	8
F	4

19) Considerando a fila do processador mostrada abaixo, calcule o instante no qual cada processo conclui o seu ciclo de processador, caso o algoritmo de escalonamento utilizado seja “Fatias de tempo com tamanho 3”.

Processo Instante de ingresso na fila de aptos Duração do próximo ciclo de processador

A	0	6
B	0	5
C	0	7
D	0	3
E	8	8
F	13	

20) Considere um sistema operacional cujo escalonamento do processador é feito através de prioridades, sendo que a prioridade de um processo é sempre escolhida aleatoriamente entre 1 e 100, toda vez que este processo passa do estado de bloqueado para o estado de apto. Ou seja, cada processo recebe uma nova prioridade aleatória entre 1 e 100 no início de cada um dos seus ciclos de processador.

Responda cada uma das questões abaixo, justificando:

- (a) Existe a possibilidade de postergação indefinida ?
- (b) Esta solução pode ser preemptiva ? Pode ser não preemptiva ?
- (c) Esta solução favorece processos I/O-bound ? Como ela pode ser alterada para favorecer processos I/O bound ? Lembre-se que é preciso identificar automaticamente quem é I/O bound, e também os processos podem mudar de comportamento durante sua execução (entre ciclos de processador).

21) Existe sentido em adotar um mecanismo de envelhecimento e ao mesmo tempo tentar priorizar os processos tipo I/O-bound ? Não tratam-se de objetivos conflitantes ?

GERÊNCIA DE PERIFÉRICOS

22) Mostre, através de um exemplo, como pode ser feita a otimização de tempo de *seek* no acesso a disco e qual o ganho de tempo possível com isso.

23) Mostre como os setores estão distribuídos fisicamente em um disco e como é feita a sua numeração unidimensional, justificando o método usado.

24) Quais são os três componentes principais que formam o tempo de acesso ao disco ?

25) Cite uma motivação para exigir de todos os drivers de dispositivo (device-drivers) a mesma interface padrão.

26) Por quê os sistemas operacionais exigem de todos os drivers de dispositivo (device-drivers) a mesma interface padrão ? Não seria mais apropriado deixar cada driver de dispositivo definir as rotinas de interface que fazem sentido para aquele tipo específico de dispositivo ?

27) Um disco possui 20.000 cilindros, 16 cabeças de leitura/gravação (faces) e 63 setores por trilha. O disco gira a uma velocidade de 5.400 rpm. O tempo de seek entre cilindros adjacentes é de 1 ms. Assumindo que a cabeça de leitura/gravação está posicionada no cilindro zero, calcule quanto tempo demora para ler o setor no endereço “cilindro 10.000, face 8, setor 10”. Explique a fórmula usada e qualquer premissa usada.

28) Vamos supor que uma unidade de disco tenha 5000 cilindros numerados de 0 a 4999. A unidade está atendendo no momento um pedido no cilindro 143 e o pedido anterior estava no cilindro 125. A fila de pedidos pendentes, na ordem FIFO, é:

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.

A partir da posição atual da cabeça, qual é a distância total (em cilindros) que o braço do disco se movimentaria para satisfazer todos os pedidos pendentes para cada um dos seguintes algoritmos de escalonamento de disco? (a) FCFS (b) SCAN

29) Vamos supor que uma unidade de disco tenha 5000 cilindros numerados de 0 a 4999. A unidade está atendendo no momento um pedido no cilindro 143 e o pedido anterior estava no cilindro 125. A fila de pedidos pendentes, na ordem FIFO, é:

86, 1470, 913, 1774, 948, 1509, 1022, 1750, 130.

A partir da posição atual da cabeça, qual é a distância total (em cilindros) que o braço do disco se movimentaria para satisfazer todos os pedidos pendentes para cada um dos seguintes algoritmos de escalonamento de disco? (a) FCFS (b) SSJ

30) Discuta quais dificuldades surgiriam se os sistemas operacionais permitissem que cada driver de dispositivo (device-driver) oferecesse uma interface diferente, a mais apropriada para aquele tipo de dispositivo.

GERÊNCIA DA MEMÓRIA

31) Compare partições variáveis, paginação simples e segmentação simples com respeito ao compartilhamento de memória (código). Para cada uma, mostre como é possível implementar ou por que é impossível implementar. Lembre-se de que é sempre necessário manter a proteção entre processos. Suponha que os programas ocupem um espaço lógico contíguo quando segmentação não é empregada.

32) Considere um sistema operacional que trabalha com paginação simples. As páginas são de 1Kbyte. O endereço lógico é formado por 16 bits. O endereço físico é formado por 20 bits. Qual o tamanho do:

- Espaço de endereçamento lógico (maior programa possível) ?
- Espaço de endereçamento físico (memória principal) ?
- Entrada da tabela de páginas, sem considerar bits de proteção ?
- Tabela de páginas (número de entradas necessárias no pior caso) ?

33) Descreva passo a passo o que acontece após uma falta de página.

34) Qual a fragmentação apresentada pelos métodos de gerência de memória baseados em partições variáveis, paginação simples e segmentação simples? Justifique através de um exemplo para cada gerência listada.

35) Existe fragmentação externa nos métodos de gerência de memória baseados em (a) partições variáveis, (b) paginação simples e (c) segmentação simples? Justifique através de um exemplo para cada gerência listada.

36) Como é possível determinar se uma interrupção de proteção acionada pela MMU é devida a um acesso ilegal à memória ou a uma falta de página? Se a MMU informa o endereço lógico que causou a interrupção de proteção, qual o procedimento para decidir isso ?

37) Existe fragmentação interna nos métodos de gerência de memória baseados em (a) partições variáveis, (b) paginação simples e (c) segmentação simples ? Justifique através de um exemplo para cada gerência listada.

38) Uma vez que paginação não apresenta fragmentação externa enquanto segmentação apresenta fragmentação externa, qual motivação existe para justificar a idéia de segmentação ? Que vantagem a segmentação pura apresenta sobre a paginação pura ? Como pode ser eliminada a fragmentação externa, uma vez que optou-se por usar segmentação no sistema ?

39) Embora memória virtual possa ser implementada através de um mecanismo de segmentação por demanda, esta solução apresenta um grande inconveniente quando comparada com a paginação por

demanda. Imagine um sistema usando segmentação por demanda e faça uma analogia com o tratamento do page-fault. Discuta as implicações no momento de atender ao segment-fault no sistema imaginado. Suponha que ocorreu um segment-fault. O que ocorre a seguir? O que muda em relação ao page-fault?

40) Em um sistema usando segmentação paginada, o espaço de endereçamento lógico de cada processo consiste de no máximo 16 segmentos, cada um deles podendo ter até 64 Kbytes de tamanho. As páginas físicas são de 512 bytes. Diga quantos bits são necessários para especificar cada uma das grandezas abaixo, explicando de onde veio cada número.

- (a) Número do segmento;
- (b) Número de uma página lógica dentro do segmento;
- (c) Deslocamento dentro de uma página;
- (d) Endereço lógico completo.

41) Considerando os sistemas de memória virtual, sobre a relação entre a quantidade de memória física que um processo dispõe e a sua taxa de falta de páginas:

- a) A taxa de falta de página jamais será zero, não importando quanta memória física o processo disponha.
- b) A taxa de falta de páginas jamais será um, não importando quanta memória física o processo disponha.
- c) A taxa de falta de páginas cresce linearmente com o número de páginas físicas disponíveis.
- d) A taxa de falta de páginas decresce linearmente com o número de páginas físicas disponíveis.
- e) A taxa de falta de páginas decresce de forma proporcional ao quadrado do número de páginas físicas disponíveis.
- f) Após uma certa quantidade de páginas físicas, não é mais possível reduzir a taxa de falta de páginas do processo.

42) Na gerência de memória os tamanhos de páginas são sempre potência de 2 pois:

- a) Na computação existe uma preferência por trabalhar com potências de dois.
- b) Para que o tamanho das páginas lógicas e das páginas físicas possa ser o mesmo.
- c) Para que o deslocamento ocupe um número inteiro de bits dentro do endereço.
- d) Para que o número de páginas físicas possa ser maior que o número de páginas lógicas.
- e) Para que dois processos possam compartilhar páginas na segmentação paginada.

43) Considere que dois processos estão executando o mesmo programa. Este programa possui dois segmentos, um compartilhável e o outro não. Cada segmento é composto por 4 páginas. A gerência de memória emprega segmentação paginada. Faça um desenho, como os utilizados em aula, mostrando:

- o espaço de endereçamento lógico de cada processo;
- as duas tabelas de segmentos;

- as três tabelas de páginas;
- o espaço de endereçamento físico.

44) Suponha que um computador possua, na sua MMU (memory management unit), dois conjuntos de registradores base-limite, os quais são usados para relocar em tempo de execução os endereços lógicos de instruções e de dados, separadamente. No arquivo executável são identificados dois segmentos, um de código e outro de dados. Discuta as conseqüências deste esquema, considerando os seguintes aspectos:

- (a) Fragmentação interna;
- (b) Fragmentação externa;
- (c) Compartilhamento de código;
- (d) Possibilidade de implementar memória virtual.

45) Considere que dois processos estão executando o mesmo programa. Este programa possui dois segmentos, um compartilhável e o outro não. Cada segmento é composto por 4 páginas lógicas. A gerência de memória emprega segmentação paginada. Faça um desenho, como os utilizados em aula, mostrando:

- o espaço de endereçamento lógico de cada processo;
- as duas tabelas de segmentos;
- as três tabelas de páginas;
- o espaço de endereçamento físico.

As páginas lógicas foram colocadas nas seguintes páginas físicas:

Segmento 0 dos processos 0 e 1:	5, 7, 4, 9
Segmento 1 do processo 0:	2, 8, 11, 3
Segmento 1 do processo 1:	0, 1, 6, 10

46) Com respeito a gerência de memória e fragmentação:

- a) Paginação pura causa fragmentação interna e externa.
- b) Paginação pura causa somente fragmentação externa.
- c) Segmentação pura causa fragmentação interna e externa.
- d) Segmentação pura causa somente fragmentação externa.
- e) Segmentação paginada elimina todos os tipos de fragmentação.
- f) segmentação paginada causa fragmentação interna e externa.

47) Considerando os sistemas de memória virtual, sobre a relação entre a quantidade de memória física que um processo dispõe e a sua taxa de falta de páginas:

- a) A taxa de falta de página jamais será zero, não importando quanta memória física o processo disponha.

- b) A taxa de falta de páginas jamais será um, não importando quanta memória física o processo disponha.
- c) A taxa de falta de páginas cresce linearmente com o número de páginas físicas disponíveis.
- d) A taxa de falta de páginas decresce linearmente com o número de páginas físicas disponíveis.
- e) A taxa de falta de páginas decresce de forma proporcional ao quadrado do número de páginas físicas disponíveis.
- f) Após uma certa quantidade de páginas físicas, não é mais possível reduzir a taxa de falta de páginas do processo.

48) Um programa escrito em C utiliza as rotinas de biblioteca malloc() e free(), as quais administram uma parte da memória lógica do processo. Malloc(x) serve para o programa alocar uma área contínua de memória com x bytes de tamanho. Free() permite o programa liberar uma área previamente alocada com malloc(). Malloc()/Free() são implementadas pela biblioteca do C.

A parte da memória lógica do processo administrada pelas rotinas da biblioteca malloc()/free() pode ser definida estaticamente (tamanho fixado já no arquivo executável) ou pode ser definida dinamicamente (tamanho do processo é alterado durante a execução através de chamadas de sistema feitas pela biblioteca na medida da necessidade, ou seja, quando é feito um malloc e a biblioteca não tem memória para atendê-lo). Discuta quais as implicações dessas duas abordagens (malloc/free administra área alocada estaticamente ou dinamicamente) considerando que o sistema operacional utiliza para a gerência de memória dos processos:

- (a) Partições variáveis;
- (b) Paginação;
- (c) Segmentação.

SISTEMA DE ARQUIVOS

49) Mostre como pode ser implementado acesso relativo nas três formas básicas de localização do arquivo no disco. Suponha que deve ser lido o terceiro bloco de um dado arquivo. Descreva como este bloco pode ser localizado em cada uma das três formas básicas de localização do arquivo.

50) O sistema operacional LQW-2.0 trabalha com blocos de 4 Kbytes. Endereços de bloco ocupam 4 bytes. Ele utiliza alocação indexada para localizar os arquivos no disco. Cada descritor de arquivo possui uma tabela com 16 endereços de blocos. Os primeiros 12 endereços são diretos (apontam para blocos de dados). Dois endereços são indiretos (apontam para blocos que contém endereços de blocos de dados). Os dois últimos endereços são duplamente indiretos. Qual o tamanho máximo de um arquivo neste sistema?

51) O sistema operacional LQW-2.0 trabalha com blocos de 4 Kbytes. Endereços de bloco ocupam 4 bytes. Esse sistema utiliza alocação indexada para localizar os arquivos no disco. Cada descritor de arquivo possui uma tabela com 10 endereços de blocos. Os primeiros 7 endereços são diretos (apontam para blocos de dados). Dois endereços são indiretos (apontam para blocos que contém endereços de blocos de dados). O último endereço é duplamente indireto. Qual o tamanho máximo de um arquivo nesse sistema?

52) Considere um sistema de arquivos que utiliza alocação indexada para localizar os arquivos no disco. Cada descritor de arquivos contém uma tabela de índices com 5 pointers diretos, 1 pointer indireto e 1 pointer duplamente indireto. São utilizados blocos físicos de 1Kbyte e cada pointer ocupa 4 bytes.

- (a) Qual o tamanho máximo de um arquivo neste sistema ?
- (b) Quantos blocos físicos devem ser lidos do disco caso o processo solicite a leitura completa de um arquivo cujo tamanho total é 512 Kbytes ? Considere que o descritor de arquivos já está na memória principal, além disso existe cache de disco mas ela está vazia.

53) Explique o mecanismo de dois níveis de tabelas para manter as informações sobre arquivos abertos no sistema. Por que não é usada apenas uma tabela geral?

54) Como é possível evitar que um processo abra um arquivo apenas para leitura e depois execute operações de escrita? Quais são as estruturas de dados envolvidas? Faça um diagrama das estruturas de dados e mostre com o kernel pode impedir escritas ilegais em arquivos.

55) Considere um sistema de arquivos que utiliza alocação indexada para localizar os arquivos no disco. Cada descritor de arquivos contém uma tabela de índices com 6 pointers diretos, 2 pointers indiretos e 2 pointers duplamente indiretos. São utilizados blocos físicos de 4Kbyte e cada pointer ocupa 4 bytes. Qual o tamanho máximo de um arquivo neste sistema ?

56) Com respeito às formas básicas (puras) de alocação de espaço em disco, está errado afirmar que:

- a) Alocação contígua apresenta o menor gasto de memória e o acesso aleatório mais rápido.
- b) Apenas alocação indexada define um tamanho máximo para o arquivo, independentemente do tamanho do disco em questão.
- c) O uso de uma pequena cache no sistema de arquivos elimina o problema de acesso aleatório encontrado na alocação encadeada.
- d) O uso de apontadores indiretos, com vários níveis de indireção, permite que na alocação indexada sejam indexados arquivos de grandes proporções.
- e) Alocação contígua gera fragmentação externa e dificulta o crescimento dos arquivos.

57) Considere um sistema de arquivos em um disco que tem tamanho de bloco de 512 bytes e endereçamento de blocos de 32 bits. O último acesso a um bloco de um arquivo foi o bloco lógico 10, quantos blocos de discos devem ser lidos para acessar o bloco lógico 4 em seguida ? Discuta isto considerando:

- (a) alocação contígua;
- (b) alocação encadeada.

58) Considere um disco de 80 Gbytes e uma unidade de alocação de espaço em disco (bloco físico) de 8 Kbytes. Alocação encadeada será usada.

a) Caso os apontadores sejam mantidos em uma tabela separada (FAT), qual será o tamanho desta tabela em número de entradas ? Qual a vantagem e a desvantagem de blocos físicos menores ?

b) Suponha que a FAT é mantida no disco e apenas as partes necessárias dela são lidas para a memória principal, quando necessário. Suponha que estas partes são mantidas na cache do sistema de arquivos, caso sejam necessárias novamente em seguida. Discuta como o nível de espalhamento dos arquivos no disco pode aumentar/reduzir a eficiência do emprego da cache para manter as partes usadas da tabela de apontadores. Considere que o disco é lido em termos de blocos.

59) Compare as alocações encadeada e indexada com respeito ao espaço em disco gasto para localizar o conteúdo de um determinado arquivo. Procure quantificar sua resposta.

60) Considerando a alocação de espaço em disco, tanto na alocação encadeada como na indexada, qualquer bloco físico serve para guardar qualquer bloco lógico de qualquer arquivo. Em função da dinâmica do sistema, naturalmente os arquivos acabam espalhados pelo disco. Desfragmentar o disco significa trocar o conteúdo dos blocos no disco de tal forma que blocos lógicos de um mesmo arquivo sejam armazenados em blocos físicos próximos, objetivando a redução do tempo de seek no acesso aos arquivos.

Esta redução no tempo de seek é garantida ou provável (determinista ou probabilista) ? Justifique a sua resposta através de um exemplo de processos acessando arquivos no disco.

61) Considerando a alocação de espaço em disco, na alocação indexada qualquer bloco físico serve para guardar qualquer bloco lógico de qualquer arquivo. Em função da dinâmica do sistema, naturalmente os arquivos acabam espalhados pelo disco. Pode-se desfragmentar o disco de duas formas:

(a) Eliminar as lacunas existentes entre as partes ocupadas do disco;

(b) Remanejar os blocos de dados de sorte que os dados de um mesmo arquivo fiquem próximos entre si.

Estas duas ações irão sempre reduzir o tempo de seek do disco ? ou existem casos particulares onde o tempo de seek aumenta após aquele tipo de desfragmentação ?

62) Considere um disco de 80 Gbytes e uma unidade de alocação de espaço em disco (bloco físico) de 8 Kbytes. Alocação encadeada será usada. Os apontadores serão mantidos em uma tabela separada, a qual será mantida no disco e também na memória principal. Acesso relativo será realizado através do caminhamento sobre a cópia da tabela em memória. É necessário escolher o tamanho de cada entrada da tabela de apontadores entre as opções “2 bytes” e “4 bytes”. Comente esta escolha.

63) Considere um sistema de arquivos onde é feita periodicamente uma compactação do disco. A compactação é composta por 2 fases. Na fase 1 todos os blocos ocupados são copiados para um “lado do disco”, deixando todos os blocos livres do “outro lado”. Na fase 2 é repetidamente feita uma troca de lugar entre dois blocos ocupados, de tal forma que ao final todos os blocos de um arquivo fiquem contíguos no disco. Avalie as conseqüências das duas fases da compactação quando o sistema de arquivos utiliza:

(a) Alocação contígua (b) Alocação encadeada (c) Alocação indexada

Discuta os benefícios (ou não) que tal compactação trará para os processos que acessam os arquivos.