

Capítulo 09:

- 1) Manter na memória principal o maior número de processos residentes, maximizando o compartilhamento. Swapping: na ausência de espaço livre na memória, o sistema deve permitir que novos processos devam ser aceitos e executados. Permitir a execução de programas que sejam maiores que a memória física disponível, implementado através de técnicas como overlay e memória virtual. Em um ambiente de multiprogramação o sistema operacional deve poder proteger as áreas de memórias ocupadas de cada processo, e também oferecer mecanismos de compartilhamento
- 2) 25% pois a área livre para subutilizar a memória é de 10 kb, o que representa 25% dos 40 kb de espaço total.
- 3) Ele ter um espaço de 50 kb na memória principal, quando o programa for executado, a técnica de overlay vai rodar os módulos individualmente ou simultaneamente conforme for o necessário, e de acordo com o tamanho disponível na memória principal
- 4) Não, pois a técnica de overlay não se utiliza da memória secundária, se o tamanho fosse aumentado pra 40 teria um momento da sobreposição de memória que não teria espaço para rodar o programa (20 kb) e seu modulo de 40 kb.
- 5) Fragmentação interna: quando um programa é alocado numa partição maior que o necessário, assim ocorre a fragmentação interna do espaço. Fragmentação Externa: Quando um programa vai ser alocado mas nenhuma partição possui o tamanho necessário para este, assim se for possível o programa deverá ser alocado em espaços de memória diferentes.
- 8) A grande diferença entre a alocação particionada estática absoluta e a alocação estática relocável é o local na memória principal onde programa é carregado. Na alocação absoluta, um programa pode apenas ser carregado a partir de um único endereço, consequentemente em uma única partição. Na alocação relocável, um programa pode ser carregado a partir de qualquer endereço ou partição.
- 10) Best-Fit: Procura a alocação de memória que mais se adeque ao processo que quer rodar. Worst-Fit: Procura a maior alocação de memória para que o processo rode com certeza.
- 12) a- Best-Fit: Olhara todas as partições e escolherá o de 10kb. Worst-Fit: Olhara todas as partições e escolherá o de 26kb. First-Fit: Olhara a partição de 10 e escolherá esta.

BES-FIT

5 Kb	Programa A
3 Kb	Programa B
6 Kb	Programa D

4 Kb	Livre
6 Kb	Programa C
26 Kb	Livre

5 Kb	Livre
3 Kb	Programa B
6 Kb	Programa D
4 Kb	Livre
6 Kb	Programa C
26 Kb	Livre

5 Kb	Livre
3 Kb	Programa B
6 Kb	Programa D
4 Kb	Programa E
6 Kb	Programa C
26 Kb	Livre

WORST-FIT

5 Kb	Programa A
3 Kb	Programa B
10 Kb	Livre
6 Kb	Programa C
6 Kb	Programa D
20 Kb	Livre

5 Kb	Livre
3 Kb	Programa B
10 Kb	Livre
6 Kb	Programa C
6 Kb	Programa D
20 Kb	Livre

5 Kb	Livre
3 Kb	Programa B
10 Kb	Livre
6 Kb	Programa C
6 Kb	Programa D
4 Kb	Programa E
16 Kb	Livre

FIRST-FIT

5 Kb	Programa A
3 Kb	Programa B
6 Kb	Programa D
4 Kb	Livre
6 Kb	Programa C
26 Kb	Livre

5 Kb	Livre
3 Kb	Programa B
6 Kb	Programa D
4 Kb	Livre

6 Kb	Programa C
26 Kb	Livre

4 Kb	Programa E
1 Kb	Livre
3 Kb	Programa B
6 Kb	Programa D
4 Kb	Livre
6 Kb	Programa C
26 Kb	Livre

- 13) A técnica de swapping foi introduzida para contornar o problema da insuficiência de memória principal. Swapping é uma técnica que permite executar programas e manipular arquivos de dados maiores do que a memória principal.

O sistema operacional copia o máximo de dados possível na memória principal e deixa o restante no disco. Quando o sistema operacional precisa de dados do disco, ele troca uma parte dos dados (chamados de página ou segmento) na memória principal por uma parte dos dados no disco.

- 14) O loader com relocação dinâmica permite que os programas possam ser retirados da memória principal para a memória secundária e trazidos novamente para a memória principal em qualquer posição.

Capítulo 10:

- 1) A memória virtual é uma técnica que combina as memórias principal e secundária dando a impressão de ter uma capacidade maior do que a real. Desta forma programas e suas estruturas de dados deixam de estar limitados ao tamanho da memória física pois possuem endereços associados a memória secundária. Outra vantagem é a permissão de um número maior de processos compartilhando a memória principal e minimiza o problema de fragmentação da memória principal.
- 2) No momento da execução de uma instrução o endereço virtual referenciado é traduzido para o endereço físico pois o processador manipula apenas posições da memória principal. o mecanismo de tradução do endereço virtual para o endereço físico é denominado mapeamento.

- 3) Porque caso o mapeamento fosse realizado para cada célula na memória principal, o espaço ocupado pelas tabelas de mapeamento seria maior que o de endereçamento virtual de cada processo, o que inviabilizaria a implementação do mecanismo de memória virtual. Um processo em um sistema computacional com arquitetura de 32 bits poderia ter 4G endereços virtuais e, consequentemente, tabelas de mapeamento com 4G entradas.
- 4) A diferença entre os dois sistemas está relacionada a forma como o espaço de endereçamento virtual está dividido logicamente. Na paginação, o espaço de endereçamento está dividido em blocos com o mesmo número de endereços virtuais (páginas), já na segmentação o tamanho dos blocos pode variar (segmentos).
- 5) Página virtual é um conjunto de endereços virtuais que faz parte do espaço de endereçamento virtual de um processo. Página real é um conjunto de endereços reais localizado na memória principal. A página real está sempre associada a uma página virtual.
- 6) São tabelas de mapeamento, utilizadas no mecanismo de memória virtual, que possibilitam que endereços virtuais sejam traduzidos em endereços reais.
- 7) O page fault ocorre todas as vezes que um processo faz referência a um endereço virtual pertencente a uma página virtual que não se encontra mapeada em uma página real, ou seja, não está, no momento, na memória principal. A ocorrência de um page fault é verificada através do bit de validade presente na ETP da tabela de páginas referente à página virtual. Uma elevada taxa de page fault pode comprometer o desempenho do sistema devido ao excessivo overhead de operações de E/S gerados pela paginação.
- 8) Não. A rotina de tratamento de page faults tem que permanecer sempre residente na memória principal, caso contrário não será possível realizar o page in quando necessário (no caso, até mesmo da própria rotina).
- 9) Nem todas as páginas podem ser removidas. Algumas, como as do núcleo de sistema, são marcadas como bloqueadas e não podem ser realocadas. É possível que o processo selecionado sofra um aumento da sua taxa de paginação, em função da redução do número de páginas alocadas na memória.
- 10) A fragmentação interna em um sistema que implementa paginação só é encontrada, realmente, na última página, quando o código não ocupa o frame por completo.
- 11) Na paginação por demanda, as páginas dos processos são transferidas da memória secundária para a principal apenas quando são referenciadas. Este mecanismo é conveniente, na medida em que leva para a memória principal apenas as páginas realmente necessárias à execução do programa. Desse modo, é possível que partes não executadas do programa, como rotinas de tratamento de erros, nunca sejam carregadas para a memória. Na paginação antecipada, o sistema carrega para a

memória principal, além das páginas referenciadas, outras páginas que podem ou não ser necessárias ao processo ao longo do seu processamento. Se imaginarmos que o programa está armazenado seqüencialmente no disco, existe uma grande economia de tempo em levar um conjunto de páginas da memória secundária, ao contrário de carregar uma de cada vez. Por outro lado, caso o processo não precise das páginas carregadas antecipadamente, o sistema terá perdido tempo e ocupado memória principal desnecessariamente.

12) A alocação fixa é simples de ser implementada pelo sistema operacional mas não é sempre uma boa opção pois os processos possuem necessidades diferentes na alocação de memória. A alocação variável é mais flexível mas exige que o sistema operacional monitore constantemente o comportamento dos processos gerando maior overhead.

13)

a)

NPV	Frame
9	4
10	9
34	3
65	7

b)

NPV	Frame
9	4
10	9
12	3
49	0
65	7

17) Sempre que uma página sofre uma alteração, o valor do bit de modificação é alterado, indicando que a página foi modificada. A política de substituição de páginas pode ser classificada conforme o seu escopo, ou seja, dentre os processos residentes na memória principal quais são candidatos a ter páginas realocadas. Em função desse escopo, a política de substituição pode ser definida como local ou global.

18) Para indicar se a página ou segmento foi modificado desde o momento em que foi carregado pela última vez na memória principal.

19) O princípio da localidade é fundamental para qualquer sistema que implemente a gerência de memória virtual, pois reduz a ocorrência de page/segments faults e, conseqüentemente, operações de E/S.

20) Porque o princípio da localidade não se faz presente em códigos desestruturados

21)

FIFO(First In First Out): A página que primeiro foi utilizada será a primeira a ser escolhida. O algoritmo pode ser implementado associando-se a uma estrutura de fila. No caso de uma página que seja constantemente referenciada, como é o caso de páginas que contém dados, o fator tempo torna-se irrelevante e o sistema tem que ser referenciar a mesma página diversas vezes ao longo do processamento

LRU(Least-Recently-Used): o algoritmo seleciona a página na memória principal que está a mais tempo sem ser referenciada. Se considerarmos o princípio da localidade, uma página que não foi utilizada recentemente provavelmente não será referenciada novamente em um futuro próximo. Apesar de ser uma estratégia com uma eficiência comparável ao algoritmo ótimo, é pouco empregada na prática devido ao seu elevado custo de implementação

28)

FIFO gera 6 page fault

Página virtual	Páginas na memória	Page fault	Página a ser substituída
0	-	S	-
1	0	S	
7	1,0	S	
2	7,1,0	S	
3	2,7,1,0	S	0
2	3,2,7,1	N	
7	3,2,7,1	N	
1	3,2,7,1	N	
0	3,2,7,1	S	1
3	0,3,2,7	N	
-	0,3,2,7	-	-

Página virtual	Páginas na memória	Page fault	Página a ser substituída
0	-	S	-
1	0	S	
7	1,0	S	
2	7,1,0	S	
3	2,7,1,0	S	0
2	3,2,7,1	N	
7	2,3,7,1	N	
1	7,2,3,1	N	
0	1,7,2,3	S	3
3	0,1,7,2	S	2
-	3,0,1,7	-	-

29) Um buffer de páginas modificadas permite adiar a gravação de páginas modificadas que foram selecionadas para realocação e seriam gravadas em disco, otimizando o desempenho do sistema.

30) Existe uma relação entre o tamanho da página e o número de operações de E/S que o sistema deverá executar para carregar as páginas da memória secundária para a memória principal. Quanto menor o tamanho da página, maior o número de operações de E/S, aumentando a taxa de paginação. Por outro lado, páginas pequenas oferecem menor fragmentação interna.

31) O endereço virtual é formado por NPV1 com 2 bits, NPV2 com 21 bits e deslocamento com 9 bits.

32) Como existem 12 bits para o deslocamento, temos 212 endereços, ou seja, páginas de 4Kb. Como existem 20 bits para o endereçamento de páginas virtuais, temos 220 páginas possíveis.

33) O endereço virtual é formado por NPV1 com 8 bits, NPV2 com 6 bits, NPV3 com 6 bits e deslocamento com 12 bits. Se um processo, quando fosse criado, alocasse todas as tabelas de páginas possíveis, o processo teria tabelas que somadas endereçariam 220 posições. Para evitar que a memória principal seja ocupada com páginas que talvez nunca sejam utilizadas, o processo aloca suas tabelas de páginas dinamicamente, conforme a

necessidade. Por exemplo, quando o processo for criado, apenas uma página de nível 1 (28 posições), uma página de nível 2 (26 posições) e uma página de nível 3 (26 posições) são alocadas inicialmente.

34) O tempo de acesso à memória (TAM) é calculado a partir das taxas de acerto e falha na TLB, ou seja, TAM é igual a $(0.98\% * \text{tempo de acesso à TLB}) + (\text{número de níveis} * (0.02\% * \text{tempo de acesso à memória})) = (0.98\% * 0) + (2 * (0.02\% * 100)) = 4\text{ns}$. O tempo total para a execução da instrução é igual a soma do tempo de leitura da instrução (4ns), do tempo de leitura do operando “direto” (4ns) e do tempo de leitura do operando “indireto” (8ns), totalizando 16 ns.

35) Memória virtual por segmentação com paginação é a técnica de gerência de memória onde o espaço de endereçamento é dividido em segmentos e, por sua vez, cada segmento dividido em páginas. Esse esquema de gerência de memória tem o objetivo de oferecer as vantagens tanto da técnica de paginação quanto da técnica de segmentação.

36) Na maioria das políticas, o critério de escolha considera o estado do processo e sua prioridade, buscando dessa forma identificar o processo com as menores chances de serem executados.

37) O problema da fragmentação existe tanto na gerência de memória virtual por paginação quanto na por segmentação. A fragmentação interna ocorre na memória virtual por paginação na última página, caso não seja totalmente ocupada. A fragmentação externa ocorre na memória virtual por segmentação em função dos espaços livres deixados entre segmentos alocados na memória principal.

38) Thrashing é consequência da excessiva paginação/segmentação em sistemas que implementam memória virtual, levando o sistema a dedicar mais tempo com operações relacionadas à gerência da memória do que no processamento das aplicações dos usuários.

Capítulo 11:

1) Lugar destinado a armazenamento de documentos.

2) Pode ser arquivada de maneira sequencial, indexada e relativa, sendo a sequencial a mais simples para armazenamento.

3) Sequencial: O sistema deve identificar o início e final do arquivo, a gravação só é possível ao final do arquivo.

Acesso direto: Permite gravar e ler direto em sua posição sem restrições de ordem.

Acesso indexado: Realiza um acesso direto a partir de um índice que é apontado por um registrador.

4)É quando o programa de computador solicita um serviço do núcleo do sistema operacional sobre o qual ele está sendo executado.

5)Sequencial: O sistema deve identificar o início e final do arquivo, a gravação só é possível ao final do arquivo.

Acesso direto:Permite gravar e ler direto em sua posição sem restrições de ordem.

Acesso indexado:Realiza um acesso direto a partir de um índice que é apontado por um registrador.

6)Alocação Contígua:

Vantagens:

- Simple de implementar e fácil de acessar o arquivo;

- Rápido acesso;

Desvantagens:

- Fragmentação;

Alocação Encadeada

Vantagens:

- Sem fragmentação;

- Simple de aumentar um arquivo, assim como criar e apagar;

Desvantagens:

- Confiabilidade.

7)É a ação de armazenar um arquivo em blocos sequencialmente dispostos no disco. O acesso a arquivos dispostos contiguamente no disco é bastante simples tanto para a forma sequencial quanto para a direta. Seu principal problema é a alocação de espaço livre para novos arquivos.

9)READ/WRITE/EXECUTE/DELETE.

10)Sistema operacional reserva uma area da memoria para que se tornem disponiveis caches utilizados em operações de acesso ao disco.