

1. Quais os problemas existentes com a utilização da técnica de *swapping*?

Resposta:

- Se o processo crescer demais, pode não haver espaço suficiente e sendo impossível funcionar
- Problemas com o tempo gasto em cada carregamento, que pode ser resolvido com uma implementação no Hardware dos computadores para permitir que a relocação seja realizada durante a execução do programa.
- Elevados custos das operações de E/S (Entrada e Saída).

2. Sobre a técnica de *overlays*:

A. Explique o funcionamento da técnica.

Resposta:

É uma técnica de separação do programa em módulos lógicos, tal que caibam na memória, mas o programa pode se constituir de quantos overlays forem necessários, sendo armazenada normalmente em disco. Também podem ser feitos overlays utilizando os slots.

B. Pense em um exemplo e descreva-o.

Resposta:

A grande maioria dos jogos em linguagem de máquina usa este método, pois é comum vermos um programa principal dividido em 6 ou mais overlays de 16 Kbytes de extensão.

C. Qual a vantagem dessa técnica em relação à do *swapping*?

Resposta:

Overlays exigem planejamento cuidadoso de consumir tempo, enquanto em *swapping* não é necessária e precisam emitir chamadas de sistema (System calls) para solicitar e liberar memória que não é em caso do Overlay.

D. Quais os problemas dessa técnica? Cite ao menos dois.

Resposta:

Tarefa complexa que tornasse cara para organizar um programador para desenvolvê-la e as funções não podem chamar outras que não estejam no mesmo módulo.

3. Elabore um parágrafo que explique o funcionamento da técnica de memória virtual. Este parágrafo deve conter os seguintes termos: **memória principal, acesso à memória física, acesso à memória virtual, páginas, quadros, MMU e processos.**

Resposta:

A técnica da memória virtual permite execução de processo que estão parcialmente na memória, assim garantindo que existam programas maiores que a memória física (o HD), assim os programadores estarão livres de limitações do armazenamento de memória, parecendo com que tivessem uma memória principal enorme, essa ilusão que dá se o nome de paginação (Frames). Essa paginação contém alguns conceitos que relacionam entre si que podemos denomina-los de endereços físicos, sendo o espaço real disponível em memória, e o espaço de endereços virtuais que são os endereços virtuais do programa (podendo ser maiores do que os espaço do HD). Para que eles possam se relacionar os endereços, uma tabela de páginas ou mapa de memória fica a serviço para gerenciá-los facilitando a comunicação entre a memória principal e a memória física, esse mecanismo ou serviço é chamado de unidade de gerência de memória (MMU), ele é quem associa as páginas lógicas aos quadros de página físicos na memória. Todas as técnicas que envolve o gerenciamento de memória têm o mesmo princípio, manter muitos processos na memória simultaneamente, para permitir a multiprogramação, como por exemplo, a utilização da técnica de overlay que contornar o problema do tamanho dos programas, porém é de difícil implementação na prática e nem sempre é uma solução garantida e eficiente. O conceito de memória virtual baseia-se em não vincular o endereçamento feito pelo programa aos endereços físicos da memória principal tendo como vantagem desta técnica permitir um número maior de processos compartilhando a memória principal, já que apenas partes de cada processo estarão na memória. Levando a uma utilização mais eficiente do processador, além de minimizar (ou quase eliminar) o problema da fragmentação, a técnica da memória virtual fez com que programadores parassem de focar na memória física e se concentrassem no problema a ser programado.

4. Complete a tabela abaixo de características da memória virtual de sistemas:

Quadros = Mem. Física/Págs

Mem. Máx. = 2^X , onde X = bits End.

Index Páginas = Log na base (2) Y.

#bits end.	#Págs.	Mem Física	Mem. Máx.	Tam. Págs.	# Quadros
16	16	32K	64K	32	2
17	8	16K	128K	16	2
23	128	512K	8M	256	4
24	64	2M (2048K)	16M	2048	32
28	4096	16M (16384K)	256M	16384	4
32	1M	2G (2048M)	4G	2048	2048
48	8M	8G (8192M)	256T	8192	1024
64	2048(2M)	128G (131072M)	16E	131072	64

5. Complete a tabela de valores de acesso à memória abaixo, para um sistema com endereçamento de 16 bits e 16 páginas de memória virtual por processo:

12000 em binário é igual a $\underbrace{0010}_{\text{Nº Páginas}} \underbrace{1110\ 1110\ 0000}_{\text{pos. das páginas (Acesso)}}$

Decimal	Binário (16bits)	Página	Acesso
47895	1011101100010111	11	2839 (101100010111)
26157	0110011000110111	6	1591 (11000110111)
53354	1101000001101010	13	106 (000001101010)
56167	1101101101100111	13	2919 (101101100111)
41100	1010000010001100	10	140 (000010001100)

32303	0111111000101111	7	3631 (1110 0010 1111)
19087	0100101010001111	4	2703 (1010 1000 1111)
60576	1110110010100000	14	3232 (1100 1010 0000)
44538	1010110111111010	10	3578 (1101 1111 1010)
1933	0000011110001101	0	1933 (111 1000 1101)

6. Suponha um sistema com memória virtual cuja CPU possua 11 bits para endereçamento de memória, com 1K de memória física instalada, e cada página possua 128 palavras de tamanho. Logo que iniciado, não há nenhuma página em quadro algum. O sistema então lança 3 processos, sendo que cada um deles realiza os seguintes acessos à memória, na ordem indicada:

Ordem	Pid	Acesso	Ordem	Pid	Acesso
1	P1	1632	7	P2	988
2	P2	499	8	P3	1407
3	P3	1894	9	P3	1334
4	P1	653	10	P1	1482
5	P1	1108	11	P2	616
6	P2	1824	12	P3	117

Desenhe as tabelas de páginas de cada processo, além da organização das páginas em quadros, e simule o funcionamento desse sistema.

7. O processador da Intel 8086 não implementa memória virtual. No entanto, algumas desenvolvedoras de computadores da época produziram sistemas com este processador e com suporte à memória virtual, mesmo utilizando processadores 8086 comuns, sem modificações internas. Pense a respeito e descreva como isso foi possível.

Resposta:

Por exemplo o 80286, que podia executar vários programas de forma simultânea (Multitarefa), ele utilizava o recurso de memória virtual aonde que aparentava ter uma memória principal maior do que realmente tinha, ele revezava os dados e códigos entre memória principal e secundária dependendo do acesso ao processador. E simulava o 8086 na unidade de endereços à unidade que realizava tarefas complicadas de cálculo de endereços, que ficava isolada da memória virtual e de outras novidades que o 8086 não tinha.

8. Considere o seguinte programa:

15	<code>public static void main (String args[]) {</code>
16	<code> int X[] = new int[N];</code>
17	<code> int passo = M;</code>
18	<code> for (int i = 0; i < N; i += passo)</code>
19	<code> X[i] = X[i] + 1;</code>
20	<code>}</code>

Suponha um sistema cuja CPU possua 12 bits para endereçamento de memória, e cada página possua 512 palavras de tamanho. Indique, explicando o motivo, quantas falhas de página (*page-faults*) ocorrerá no laço for, caso:

- A. $N = 400$ e $M = 1$;
- B. $N = 3500$ e $M = 4$;
- C. $N = 4000$ e $M = 1024$.

Resposta: Irá apenas passar 3 vezes no laço faltando algumas páginas para serem completas.