

Atividade 7 – Sistemas Operacionais (SO)

Júlia Simone Araújo

1.

Todo programa precisa utilizar a memória ao ser executado, sendo visto pelo sistema operacional como uma área exclusiva que só ele o núcleo do sistema pode acessar. A partir dessa afirmativa, temos o endereço inicial (0) e o endereço máximo (n) para o determinado processo, com divisões de faixas de endereços que correspondem a informações de características similares a processos. São elas:

- TEXT: onde armazena-se o código executável do processo, programa principal, onde ficam as instruções (funções, bibliotecas estáticas);

- DATA: onde armazena-se as variáveis globais, locais estáticas e buffers internos;

Em seguida, existe uma estrutura de tamanho flexível que varia de acordo com cada processo. Ela vai a faixa chamada HEAP, variáveis dinâmicas, à STACK, endereços de retorno de chamadas.

2.

Levando em consideração o uso de hardware dedicado, MMU, temos a definição ou resolução de endereço, pelo processador, durante a execução. Isso gera a criação de endereços de memória, chamados endereços lógicos, que não necessariamente são os mesmos que os associados a memória principal, RAM, também chamados de endereços físicos.

O uso de endereços lógicos pelo MMU permite com que quando existir múltiplos processos ativos em um sistema, eles sejam separados e evite a transposição de armazenamento, ou seja, a sobreposição da informação. Isso acontece pois embora eles serão alocados na mesma faixa de endereço, na memória principal serão alocados em lugares diferentes.

3.

A falta de página, também conhecido como Page Fault, é uma exceção realizada pelo hardware quando um programa acessa uma página mapeada na memória virtual que não foi carregada na memória física. Sua causa pode variar, como no caso da página correspondente ao endereço solicitado não está carregada na memória, ou embora esteja carregada o seu estado corrente não foi atualizado no hardware. Ou até em casos de falha de proteção, ex:

- a página não é parte do programa, e portanto não é mapeada na memória do programa;
- o programa não tem privilégios suficientes para ler ou escrever a página, etc.

Para tratar esse problema, o sistema costuma realizar interrupções de execução do processo em caso de acesso ilegal.

4.

A, D e F.

a) Incorreta, pois são gerados os endereços lógicos.

d) Incorreta, pois o armazenamento é apenas das variáveis estáticas inicializadas pelo código-fonte.

f) Incorreta, pois não há threads armazenadas na área de memória STACK pela dinamicidade e atenção requerida.

5.

Páginas de 3 níveis; 100ns de T de acesso; TLB 64 entradas; 98% de taxa de acerto; 10 ns de custo de acerto e 50 ns de erro.

O tempo médio de acesso à memória pode ser determinado pela média ponderada entre o tempo de acesso com acerto de cache e o tempo de acesso no caso de erro. Nesse tempo médio, levaremos em consideração os custos de erro e acerto, de consulta as tabelas e tempo de acesso a RAM.

$$T = 0,98 \times 10 \text{ (ns)} + 0,02 \times (50 \text{ (ns)} + 3 \times 100 \text{ (ns)}) + 100 \text{ (ns)} = 116,8 \text{ ns.}$$

6.

4 físicos disponíveis;

Por FIFO:

PÁGINA VIRTUAL	PÁGINA MEMÓRIA	PAGE FAULT	SUBSTITUIÇÃO
0	N/A	X	N/A
1	0	X	N/A
7	1, 0	X	N/A
2	7,1,0	X	N/A
3	2,7,1,0	X	0
2	3,2,7,1	N/A	N/A
7	3,2,7,1	N/A	N/A
1	3,2,7,1	N/A	N/A
0	3,2,7,1	X	1
3	0,3,2,7	N/A	N/A

Page Fault: 6

Por LRU:

PÁGINA VIRTUAL	PÁGINA MEMÓRIA	PAGE FAULT	SUBSTITUIÇÃO
0	N/A	X	N/A
1	0	X	N/A
7	1, 0	X	N/A
2	7,1,0	X	N/A
3	2,7,1,0	X	0
2	3,2,7,1	N/A	N/A
7	2,3,7,1	N/A	N/A
1	7,2,3,1	N/A	N/A
0	1,7,2,3	X	3
3	0,1,7,2	X	2

Page Fault: 7

7.

Tempo acesso a RAM = 5 ns; Acesso ao disco = 5 ms; Falta de página (page fault) = 1 a cada 10^6 .

Levando em consideração o tempo médio de acesso à memória pode ser determinado pela média ponderada $T = (1 - p_a) t_e + p_a \cdot t_a$, temos: a soma entre o tempo de acesso, sem page fault, com o tempo de acesso ao disco, sobre o page default. Assim:

$$T = (10^6 - 1) \times 5 \times 10^{-9} \text{ (ns)} + 5 \times 10^{-6} \text{ (ms)} / 10^6$$

$$T = 0,004999995 + 0,000005 / 1.000.000$$

$$T = 0,000000005004995, \text{ ou seja, } T = \text{aproximadamente, } 5 \text{ ns.}$$

8.

a) A página 3 está mapeada no frame 10, que é o 11 frame da memória principal. A partir disso, o endereço físico é: $(11 \times 2^{11}) - 1 = 22.527$

b) A página 2 está mapeada no frame 100, que é o 101 frame da memória principal. Dito isso, o endereço físico: $(100 \times 2^{11}) = 204.800$

c) A página 3 está sendo mapeada no frame 10, que é o 11 frame da memória principal somado ao deslocamento. O endereço físico então é: $(10 \times 2^{11}) + 10 = 20.490$

d) Estarão as páginas: 0, 1, 2, 3 e 6.

9.

Letra B, instalação de mais memória principal. Pois já que o servidor possui taxas de UCP (Unidade Central de Processamento) e do disco iguais a 10% e 96,7%, para que haja aumento na taxa de uso da UCP.

10.

4 quadros de memória;

Baseando no algoritmo de envelhecimento, segue o contador:

T1, 0101:

Q0	0	00000000
Q1	1	00000000
Q2	0	00000000
Q3	1	00000000

T2, 0011:

Q0	0	00000000
Q1	0	10000000
Q2	1	00000000
Q3	1	10000000

T3, 1110

Q0	1	00000000
Q1	1	01000000
Q2	1	10000000
Q3	0	11000000

T4, 1100:

Q0	1	10000000
Q1	1	10100000
Q2	0	11000000
Q3	0	01100000

T5, 1001:

Q0	1	11000000
Q1	0	11010000
Q2	0	01100000
Q3	1	01100000

T6, 1011

Q0	1	11100000
Q1	0	01101000
Q2	1	00110000
Q3	1	10110000

T7, 1010

Q0	1	11110000
Q1	0	00110100
Q2	1	10011000
Q3	0	11011000

T8, 0111

Q0	0	11111000
Q1	1	00011010
Q2	1	11001100
Q3	1	01101100

T9, 0110

Q0	0	01111100
Q1	1	10001101
Q2	1	11100110
Q3	0	10110110

T10, 0111

Q0	0	00111110
Q1	1	11000110
Q2	1	11110011
Q3	1	01011011

Tfinal:

Q0	N/A	00011111
Q1	N/A	11100011
Q2	N/A	11111001
Q3	N/A	10101101

Q0 será substituído por ser o de menor valor.

11.

Tc = 220;

Em t = 50, P0 = 220 - 142 = 78 (substitui). Ou seja, página p0 será substituída.

Já em t = 100: p0 = 220 - 142 = 78 (passa); p1 = 220 - 197 = 23 (passa); p2 = 220 - 174 = 46 (passa); p3 = 220 - 46 = 174 (substitui). Ou seja, página p3 será substituída.