

Época de Recurso 2017/2018

1. @ Isso pode ser garantido pelo sistema operativo ao associar "valid-bits" a cada PTE, desse modo, o processo só pode aceder a uma dada página se o valid-bit de PTE estiver ativo.

Pode também haver um "Page-table length register", com o tamanho de tabela de cada processo, acessos a endereços maiores que a tabela são negados

⑥ Através de "Shared-Pages", vários processos mapeam as páginas partilhadas para a mesma page frame física.

Isto pode ser necessário na implementação de IPCs

2 Não necessariamente, nos algoritmos não-preemptivos, o processo ativo tem de acabar a sua tarefa atual antes de comutar, mesmo que haja uma interrupção de relógio, ele não vai ser trocado se ainda não ~~tem~~ acabado a sua tarefa

3 O TLA serve como um cache que guarda um número relativamente pequeno de PTEs, tornando a pesquisa de páginas bastante mais rápida.

4 Interrupts ou Systemcalls

↓
Sinal para o CPU

↳ Pedido de um programa para uma ação ser executada em Kernel-mode

(I/O, fork(), File write, etc)

que interrompe o código atual
e transfere o fluxo de execução
para outro pedaço de código

S \equiv Swap-out
P \equiv Page Fault
M \equiv Modify bit active

5

	P	P	P	P	P	PS	PS	PS	P	PS	P	PS
FIFO	R4	W6	R3	R5	W3	W1	R2	R5	R6	R4	R1	W3
F0	4	4	4	4	4	1'	1'	1'	1'	1'	1'	3'
F1		6'	6'	6'	6'	6'	2	2	2	2	2	2
F2			3	3	3'	3'	3'	3'	6	6	6	6
F3				5	5	5	5	5	5	4	4	4

9 PF
350

	P	P	P	P	P	PS	PS	PS	P	P	PS	P
LRU	R4	W6	R3	R5	W3	W1	R2	R5	R6	R4	R1	W3
F0	4	4	4	4	4	1'	1'	1'	1'	4	4	4
F1		6'	6'	6'	6'	6'	2	2	2	2	1	1
F2			3	3	3'	3'	3'	3'	6	6	6	6
F3				5	5	5	5	5	5	5	5	3'

10 PF
350

0 \equiv used bit = 0
CLOCK \equiv Clock Position

	P	P	P	P	P	PS	PS	PS	P	P	PS	P
CLOCK	R4	W6	R3	R5	W3	W1	R2	R5	R6	R4	R1	W3
F0	4	4	4	(4)	(4)	1'	1'	1'	1'	(1 ¹⁰)	(1')	1'
F1	0	6'	6'	6'	6'	(6 ¹⁰)	2	2	2	2 ⁰	2 ⁰	3'
F2		0	3	3	3'	3 ¹⁰	(3 ¹⁰)	(3 ¹⁰)	6	6 ⁰	6 ⁰	(6 ⁰)
F3			0	5	5	5 ⁰	5 ⁰	5	(5)	4	4	4

9 PF
250

6

$$EAT = (1-p) \cdot \overset{\substack{\uparrow \\ \text{PF rate}}}{ma} + p \cdot \underbrace{[\text{swap out} + \text{swap in} + \text{OS overhead}]}_{\text{PF Overhead}} \Leftrightarrow$$

$$\Leftrightarrow 11 = (1-p) \cdot (10 + 200) + p \cdot (200 + 200) \Leftrightarrow$$

\uparrow Instruction
 \uparrow Data check

$$\Leftrightarrow 11 = 210 - 210p + 400p \Rightarrow p < 0$$

Não funciona se assumirmos que o tempo de acesso após cache hit é (cache + instruction-fetch)

7 - Cada página tem $\frac{1KB}{2^{2B}} = 2^8$ PTEs

offset
($\log_2(1KB)$)

Com 1 nível, para endereçar todas as páginas, usamos $8 + 10 = 18 < 34$

Com 2 níveis, temos $p_1 = 8, p_2 = 8, d = 10 \Rightarrow 26 < 34$

Com 3 níveis temos $p_1 = 8, p_2 = 8, p_3 = 8, d = 10 \Rightarrow 34 = 34$

R: 3 níveis

8 - $10000 \text{ rpm} = \frac{10000}{60} \text{ rpm}$

$512 \text{ B/sector} = 2^9 \text{ B/sector}$
 $200 \text{ sectors/track}$ \rightarrow Há $200 \cdot 2^9 = 102400 \text{ B num track}$

$$T_{\text{seek}} = 4,9 \text{ ms} = 0,0049 \text{ s}$$

$$T_{\text{rot}} = \frac{1}{2 \left(\frac{10000}{60} \right)} = 0,003 \text{ s}$$

Queremos ler $8KB = 2^{13} \text{ B}$, $T_{\text{transf}} =$

$$T_{\text{transf}} = \frac{\text{size to transfer}}{\text{rps} \cdot \text{track size}} = \frac{2^{13}}{\frac{10000}{60} \cdot 102400} = 0,00048 \text{ s}$$

$$T_{\text{total}} = T_{\text{seek}} + T_{\text{rot}} + T_{\text{transf}} = 0,00838 \text{ s}$$