Época Normal 2014/2015

1- Sem DMA, o CPU era muitas vezes "bottlenecked" pela velocidade de I/O, como era o responsável por esperar por esses eventos de I/O, ia contra os princípios da multiprogramação (ter o CPU sempre ocupado)

1. Vantagem: Mais eficientes e flexíveis, dado que a criação, destruição e mudança entre threads não implicam uma mudança para kernel-mode.

Desvantagem: Quando uma thread faz uma chamada bloqueante, todas as threads param, pois o scheduler não sabe que elas existem

3- bloco = 2KB = 2^{11} B
endereços = 4B = 2^2 B

a) $2^{11} \cdot 10\,B = 20KB$

b) Cada bloco de ponteiros tem $\frac{2^{11}}{2^2} = 2^9$ ponteiros

$2^9 \cdot 2^{11} = 2^{20} = 1MB$

c) $2^9 \cdot 2^9 \cdot 2^{11} = 2^{29} = 512\,MB$

d) $2^9 \cdot 2^9 2^9 \cdot 2^{11} = 2^{38} = 256GB$

4- página = 4KB = 2^{12}
① TLB = 64 entradas

(tecnicamente $\overbrace{N=1, M=1}^{M \geq 1024,\ N \geq 1024}$ está correto, mas parece batota)

Um int no C tem 4 B ⇒ 1 página tem $\frac{4KB}{4B} = 1024$ ints

Para garantir (sempre) TLB miss em cada iteração, temos de aceder a uma página diferente, no total queremos pelo menos 64 páginas diferentes, ou seja

$c \geq 64$
$N \geq c \cdot 1024$
$M \geq \frac{N}{c}$

② Desta vez precisamos que C seja 265, para garantir que no início de cada ciclo exterior, apenas as últimas 64 páginas estejam no TLB, originando misses em cada iteração do ciclo interior

5. $Program = 64\,KB = 2^{16}\,B$
$seek = 5\,ms = 0,005\,s$
$T_{rot} = 5\,ms = 0,005\,s \longrightarrow 0,005 = \frac{1}{2r} \Leftrightarrow r = 100$
$pageSize = 4\,kB = 2^{12}$
Cada pista $= 1\,MB = 2^{20}\,B$

Tempo para carregar uma página:

$$0,005 + 0,005 + \frac{2^{12}}{100 \cdot 2^{20}} = 0,010039062$$

Há $\frac{2^{16}}{2^{12}} = 2^4$ páginas, $\Rightarrow T_{total} = 16 \cdot 0,010039062 = 0,160625\,s$

6

$$Need = \begin{pmatrix} 1 & 1 & 0 & 2 & 1 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 2 & 0 & 3 & 1 \\ 0 & 2 & 1 & 1 & 0 \end{pmatrix}$$

Available: 0 1 0 2 1
↓ P2
0 2 0 3 1
↓ P3
0 2 0 3 2
↓ Deadlock

Há deadlock, tanto o P1 como o P4 precisam de recursos não disponíveis, apenas libertando um dos dois é que o funcionamento pode continuar

7. Com diferentes aneis de proteção, é possível fazer uma divisão clara entre o hypervisor e as máquinas virtuais, sendo que o hypervisor corre num anel mais privilegiado que as VMs, e as aplicações dentro das próprias VMs correm num anel ainda menos privilegiado.