



90 minutos

Nome Tiago Jorge Coimbra da Silva nº 2022 216215

Regras

Qualquer tentativa de fraude conduzirá à anulação da prova para todos os intervenientes.
As respostas devem ser obrigatoriamente efectuadas na folha fornecida.

1 Questão

Nos primeiros sistemas operativos, a leitura ou escrita de cada byte era da responsabilidade do CPU (i.e. não existia DMA). Que implicações teve esta realidade nos primeiros sistemas com suporte de multiprogramação? Justifique.

Uma vez que as operações I/O tinham de ser administradas pelo CPU, este ficava extremamente sobrecarregado. Em sistemas com suporte de multiprogramação, uma vez que cada byte tinha de ser processado pelo CPU, se existissem vários processos a competir pelo mesmo dispositivo, a latência era enorme e a eficiência reduzida.

2 Questão

Qual a principal vantagem de implementar threads em espaço de utilizador? E qual a principal desvantagem?

Comutação entre threads é rápida porque não implica uma mudança de privilégios mas, por outro lado, se 1 thread bloquear todas bloqueiam.

3 Questão

Considere um sistema UNIX que possui blocos de 2^{11} KB e endereços de disco de 2^2 bytes. Qual será o tamanho máximo de um ficheiro neste sistema se cada i-node tiver: a) 10 ponteiros diretos; b) 1 ponteiro indireto; c) 1 ponteiro "duplo" indireto; d) 1 ponteiro "triplo" indireto

a) $10 \times 2^{11} = 5 \times 2^{12} = 20 \text{ KB}$ | entradas: $2^{11}/2^2 = 2^9$
b) $2^9 \times 2^{11} = 2^{20} = 1 \text{ MB}$
c) $2^9 \times 2^{20} = 2^{29} = 512 \text{ MB}$
d) $2^9 \times 2^{29} = 2^{38} = 256 \text{ GB}$ | Total: $5 \times 2^{12} + 2^{20} + 2^{29} + 2^{38}$

4 Questão

Considere o seguinte programa em C:

```
int x[N];
int step = M; /* M e N são constantes predefinidas */
for (int i = 0; i < N; i += step)
    x[i] = x[i] + 1;
```

- 1) Se este programa for executado num computador com páginas de 4 KB e uma TLB de 64 entradas, que valores de M e N irão originar um "TLB miss" em cada execução do ciclo? Justifique.

Inteiros por página: $\frac{2^{12}}{2^2} = 2^{10}$ inteiros

$$C \geq 1 \quad N \geq C \times 1024, \quad M \geq \frac{N}{C}$$

- 2) Se o ciclo presente no programa fosse executado mais do que uma vez, como é ilustrado no segmento de código que se segue, a sua resposta seria a mesma da pergunta 4.1 (pergunta anterior)? Justifique.

```
int x[N];
int step = M; /* M e N são constantes predefinidas */
for (int j = 0; j < 10; j++)
    for (int i = 0; i < N; i += step)
        x[i] = x[i] + 1;
```

Neste caso para haver sempre TLB miss, a cada passo teremos que percorrer o limite dos 64 páginas desta forma:

$$C \geq 65, \quad N \geq C \times 1024, \quad M \geq \frac{N}{C}$$

5 Questão

Quanto tempo seria necessário para carregar de disco para memória um programa de 64 KB, considerando que o disco possui um seek time médio de 5 milissegundos, que a latência rotacional é de 5 milissegundos e que cada pista contém 1 MB de dados. Considere ainda que as páginas têm uma dimensão de 4 KB e que se encontram dispersas pelo disco de forma aleatória, pelo que a probabilidade de 2 páginas partilharem o mesmo cilindro (serem carregadas num só movimento da cabeça de leitura) é negligenciável.

$$\frac{1}{2R} = 0,005 \Rightarrow R = \frac{1}{2 \times 0,005} = 100 \text{ RPM}$$

$$\text{Tempo carregar 1 página} = \frac{2^{12}}{100 \times 2^{20}} + 0,010 = 0,010039$$

$$16 \times T_{\text{página}} = 0,1606252$$

6 Questão

A

C

Considere o seguinte estado de um sistema que possui 4 processos em execução e 5 tipos de recursos.

Recursos alocados por processo

P1	0	1	1	1	2
P2	0	1	0	1	0
P3	0	0	0	0	1
P4	2	1	0	0	0

Recursos necessários por processo

P1	1	1	0	2	1
P2	0	1	0	2	1
P3	0	2	0	3	1
P4	0	2	1	1	0

Recursos disponíveis

R1	R2	R3	R4	R5
0	1	0	2	1
0	2	0	3	2

Utilizando o **algoritmo para detecção de deadlocks**, verifique se o sistema se encontra em **deadlock**. Justifique apresentado os passos do algoritmo e a sua conclusão.

Sim, os processos P1 e P4 encontram-se em deadlock

7 Questão

Explique de que forma os anéis (rings) de proteção da arquitetura x86 podem ser utilizados no suporte à virtualização.

A virtualização permite que vários SO's rodem simultaneamente no mesmo hardware físico. Para este fim, os anéis de proteção são usados para garantir a segurança e eficiência do processo. Dependendo da necessidade, eles operam em diferentes anéis. Por exemplo: o hypervisor opera normalmente no anel 0 para ter acesso ao hardware enquanto VMs e as aplicações correm normalmente num anel superior (normalmente 3)