

Sistemas Operacionais

Sexta Lista de Exercícios – Solução

Norton Trevisan Roman
Clodoaldo Aparecido de Moraes Lima

8 de novembro de 2012

1. **Fly-by mode:** Temos 3 operações em cada transferência: solicitação de transferência, feita pela controladora de DMA à controladora do dispositivo, transferência do dispositivo à memória, e confirmação feita pela controladora do dispositivo à controladora de DMA. Cada uma se dá via barramento, ou seja, gastará t_1 para obtê-lo e então t_2 para envio da solicitação/dado/confirmação. Com o modo de uma palavra por vez, isso será feito a cada palavra, gastando $3(t_1 + t_2)$ por palavra, como são 1.000 palavras, teremos um tempo $3.000(t_1 + t_2)$ ns.

Burst mode: A solicitação gastará $t_1 + t_2$. Supondo que todas as 1000 palavras são enviadas em um único surto, o envio gastará $t_1 + 1000 \times t_2$ (no barramento cabe uma única palavra). Por fim, a confirmação levará $t_1 + t_2$, num total de $t_1 + t_2 + t_1 + 1000 \times t_2 + t_1 + t_2 = 3t_1 + 1002t_2$ ns

2. Transferir para a memória os $32 + 1 + 1 = 34$ registradores gastará 340ns. Trazê-los de volta, outros 340ns, num total de 680ns. Supondo que nada seja feito pela interrupção (irreal), um máximo de 10^9 ns/680ns = 1,47 milhões de interrupções podem ser tratadas.

3. A impressora imprime $50 \times 80 = 4.000$ caracteres por página. Com 6 páginas/min temos $6 \times 4.000/60 = 400$ caracteres/s. Isso os leva a $1/400 = 2,5$ ms = 2.500μ s de tempo para processar um caractere, contra apenas 50 para processar a interrupção.

Olhando por outro prisma, seriam 400 interrupções/s, gastando $50 \times 400 = 20.000\mu$ s = 20ms ao todo. Então gastaria 20ms a cada s, restando $1000 - 20 = 980$ ms para outros processamentos. Tem sentido usar sim.

4. Um pacote tem $1024B = 1024 \times 8 = 8192b$. A uma taxa de 10Mb/s (ou seja, $10 \times 1.000.000 = 10.000.000$ b/s), temos que um pacote levará $8192/10.000.000 = 0,000819$ s = $0,819$ ms

Tempo de um pacote:

Chamada ao SO para envio (interrupção): 1ms (1000μ s)

Cópia ao buffer do SO: 1024μ s (1B a cada μ s)

Cópia à controladora: 1024μ s

Envio: 0,819ms (819μ s)

Atraso de rede: 1μ s

Interrupção no destino: 1ms (1000μ s)

Cópia ao SO: 1024μ s

Cópia ao programa do usuário: 1024μ s

Total: 6.916μ s = 6,916ms (0,006916s), para um pacote de 1024B. Então, teremos $1024/0,006916 = 148.062,46B/s$. Note que a capacidade nominal é $10.000.000/8 = 1.250.000B/s$, ou seja, pouco menos de 12% da capacidade nominal.

5. Suponha que transferências se deem 1 página por vez. Como as páginas estão espalhadas, isso exigirá um posicionamento e uma rotação por página. A questão então é qual o tempo de transferência.

- (a) Páginas de 2KB. Em uma trilha, temos $32/2 = 16$ páginas. Supondo uma distribuição uniforme dos blocos na trilha (normalmente verdadeiro), o tempo de posicionamento médio de 10ms corresponde ao tempo de acesso ao elemento do meio da trilha. Assim, para correr a trilha inteira gastamos $2 \times 10 = 20$ ms. Com 16 páginas na trilha, cada página leva $20/16 = 1,25$ ms para ser lida. Seu tempo de acesso total (posicionamento + rotação + transferência) será $T_t = 10 + 10 + 1,25 = 21,25$ ms. Como temos $64KB/2KB = 32$ páginas no programa, o tempo de carregamento deste será de $21,25 \times 32 = 680$ ms.
- (b) Páginas de 4KB. Em uma trilha, temos $32/4 = 8$ páginas, o que nos leva a um tempo de transferência de $20/8 = 2,5$ ms para cada página. O tempo de acesso total (posicionamento + rotação + transferência) será $T_t = 10 + 10 + 2,5 = 22,5$ ms. Como temos $64KB/4KB = 16$ páginas no programa, o tempo de carregamento deste será de $22,5 \times 16 = 360$ ms.
6. $7.200\text{rpm} \Rightarrow 7.200/60 = 120$ rotações/s (0,12 rotações em 1 ms). Em 1ms (tempo de posicionar o braço), o disco rodou 0,12 volta, ou seja, $0,12 \times 200 = 24$ setores se passaram. Então a torção cilíndrica será de 24 setores.
7. 120 rotações/s $\Rightarrow 120 \times 200$ setores/s $\Rightarrow 120 \times 200 \times 512 = 12.288.000B/s \Rightarrow 11,72MB/s$.
8. Talvez, se a maioria dos arquivos estiverem armazenados em setores consecutivos, pode vale a pena entrelaçar os setores para dar aos programas tempo de processar os dados recebidos. Se isso é realmente válido depende do tipo de programa rodado.
9. **Entrelaçado:** Posicionamento: 4 setores (1/2 volta).
 A cada setor lido, pula outros 2 (exceto o último lido). Então, cada setor, à exceção do último, vale por 3. Como temos 8 setores na trilha, corrê-los em ordem faz com que percorra $3 \times 7 + 1 = 22$ setores ao todo (2,75 voltas)
 Como leva 0,5 volta para chegar ao setor 0, ao todo gasta $2,75 + 0,5 = 3,25$ voltas. A 300rpm temos $3,25/300 = 0,0108\text{min}$ gastos $\Rightarrow 0,0108 \times 60 = 0,65s$. Levou então 0,65s para efetivamente ler $8 \times 512B = 4096B = 4KB \Rightarrow \frac{4KB}{0,65s} \approx 6,15KB/s$
Não entrelaçado: Posicionamento: 4 setores (1/2 volta).
 Correr os 8 setores em ordem faz com que percorra 8 setores ao todo (1 volta)
 Ao todo, deu $1 + 0,5 = 1,5$ voltas. A 300rpm temos $1,5/300 = 0,005\text{min}$ gastos $\Rightarrow 0,005 \times 60 = 0,3s$. Levou então 0,3s para efetivamente ler $8 \times 512B = 4096B = 4KB \Rightarrow \frac{4KB}{0,3s} \approx 13,3KB/s$
10. Possivelmente. Entrelaçamento duplo corresponde a uma torção cilíndrica de 2 setores. Se isso for o suficiente, então não há tal necessidade. Senão, há.
11. (a) A capacidade do disco será a soma de todos os setores (ou blocos) em todos os cilindros e faces. Para um determinado setor isso será $\text{cilindros} \times \text{setores} \times \text{tamanho_setor} \times \text{número_cabecas}$. Então temos $100 \times (160 + 200 + 240 + 280) \times 512 \times 16 = 687,5MB$
 (b) Com 7.200rpm, o disco faz $7.200/60 = 120$ rotações/s, ou $120/1.000 = 0,12$ rotação a cada ms (tempo em que o braço se move de um cilindro a um adjacente).
Zona 1 (160 setores): Durante 0,12 volta, $0,12 \times 160 = 19,2$ setores passam $\Rightarrow 19,2$ setores de torção.
Zona 2 (200 setores): $0,12 \times 200 = 24$
Zona 3 (240 setores): $0,12 \times 240 = 28,8$
Zona 4 (280 setores): $0,12 \times 280 = 33,6$
 (c) A maior taxa se dará na trilha com maior número de setores, ou seja, trilhas da zona 4. Como o disco roda 120 vezes a cada segundo, nesse tempo passaram $280 \times 120 = 33.600$ setores e $120 \times 280 \times 512 \approx 16,41MB \Rightarrow 16,41MB/s$
12. (a) Iniciando no 20, andará $|20-10| + |10-22| + |22-20| + |20-2| + |2-40| + |40-6| + |6-38| = 10 + 12 + 2 + 18 + 38 + 34 + 32 = 146$ cilindros $\Rightarrow 146 \times 6 = 876ms$

- (b) $|20-20|+|20-22|+|22-10|+|10-6|+|6-2|+|2-38|+|38-40| = 0+2+12+4+4+36+2 = 60 \Rightarrow 60 \times 6 = 360\text{ms}$
- (c) Ordenando os cilindros, temos 2, 6, 10, 20, 22, 38, 40. Iniciando do 20: $|20-20|+|20-22|+|22-38|+|38-40|+|40-10|+|10-6|+|6-2| = 0+2+16+2+30+4+4 = 58 \Rightarrow 58 \times 6 = 348\text{ms}$
13. Não necessariamente. Um programa que leia 10.000 blocos vai executar os pedidos um por vez, bloqueando após cada um deles até que se complete. O driver viu efetivamente um único pedido por vez. O aluno deveria iniciar muitos processos paralelos, ao mesmo tempo, para realmente testar o algoritmo.
14. O relógio, a cada segundo, gastará $60 \times 2 = 120\text{ms}$ de processamento $\Rightarrow \frac{120}{1000} = 0,12 \Rightarrow 12\%$ da CPU
15. $500\text{MHz} \Rightarrow$ decrementa o contador 500.000.000 de vezes em um segundo.
- (a) Em 1ms, o contador terá oscilado $500 \times 10^6 \cdot 1 \times 10^{-3} = \frac{500 \times 10^6}{10^3} = 500 \times 10^3 \Rightarrow 500.000$ vezes. Então esse deve ser o valor no contador
- (b) $100\mu\text{s} = 0,1\text{ms}$. Então temos $500 \times 10^6 \cdot 1 \times 10^{-4} = \frac{500 \times 10^6}{10^4} = 50.000$
16. **5000**
 Tempo atual: 5000
 Próximo sinal: 8
 Cabeçalho: 8 (5008) \rightarrow 4 (5012) \rightarrow 3 (5015) \rightarrow 14 (5029) \rightarrow 8 (5037)
5005
 Tempo atual: 5005
 Próximo sinal: 3
 Cabeçalho: 3 (5008) \rightarrow 4 (5012) \rightarrow 3 (5015) \rightarrow 14 (5029) \rightarrow 8 (5037)
5013
 Tempo atual: 5013
 Próximo sinal: 2
 Cabeçalho: 2 (5015) \rightarrow 14 (5029) \rightarrow 8 (5037)
5017
 Em 5017, quando o novo sinal chega, a situação será:
 Tempo atual: 5017
 Próximo sinal: 12
 Cabeçalho: 12 (5029) \rightarrow 8 (5037)
 O sinal para 5033 entrará entre o de 5029 e 5037, deixando a fila
 Cabeçalho: 12 (5029) \rightarrow 4 (5033) \rightarrow 4 (5037)
5023
 Tempo atual: 5023
 Próximo sinal: 6
 Cabeçalho: 6 (5029) \rightarrow 4 (5033) \rightarrow 4 (5037)
17. Com 32 bits armazenamos $2^{32} = 4.294.967.295\text{s} \Rightarrow \frac{4.294.967.295}{3600} = 1.193.046,47\text{h} \Rightarrow \frac{1.193.046,47}{24} = 49.710,27\text{d} \Rightarrow \frac{49.710,27}{365,25} = 136,1$ anos.
- Como o UNIX começa a contar o tempo desde 01/01/1970, teremos o ano de $1.970 + 136 = 2.106$, mês de $0,1 \times 12 = 1,2 \Rightarrow$ fevereiro (início). Com o advento de 64 bits, não acontecerá.