MONITORIA DE SISTEMAS OPERACIONAIS - 2023.
RESOLUÇÃO DE EXERCÍCIOS;

Monitor: Vinícius de Oliveira Guimarães
E-mail: viniciusguimaraes@estudante.ufscar.br
Professor: Fredy João Valente

QUESTÃO 1

Um computador com um endereço de 32 bits usa uma tabela de páginas de dois níveis. Endereços virtuais são divididos em um campo de tabela de páginas de alto nível de 9 bits, um campo de tabela de páginas de segundo nível de 11 bits e um deslocamento. Qual o tamanho das páginas e quantas existem no espaço de endereçamento?

a. Tamanho pág = 2-KB , número de págs = 2^{23}
b. Tamanho pág = 1000-KB , número de págs = 2^{12}
c. Tamanho pág = 4-KB , número de págs = 2^{20} ✔

Considerando 32 bits, como temos uma tabela de páginas de dois níveis, então no final do segundo nível teremos o endereço físico da página.

Se o primeiro nível tem 9 bits e o segundo nível 11 bits, somando os dois ficamos com 20 bits. Ou seja, como usamos 20 bits para representar os diferentes endereços possíveis de páginas, então temos no total 2^20 páginas.

Verificando a quantidade de bits do deslocamento: 32 - 20 = 12 bits. Como temos 12 bits para o representar o deslocamento dentro da página, então temos 2^12 endereços diferentes, que dá um tamanho total de 2^12 = 4096 bytes.

Resposta: C)

QUESTÃO 2

Uma máquina tem um espaço de endereçamento de 32 bits e uma página de 8 KB. A tabela de páginas é inteiramente em hardware, com uma palavra de 32 bits por entrada. Quando um processo inicializa, a tabela de páginas é copiada para o hardware da memória, a uma palavra a cada 100 ns. Se cada processo for executado por 100 ms (incluindo o tempo para carregar a tabela de páginas), qual fração do tempo da CPU será devotado ao carregamento das tabelas de páginas?

a. 32%
b. 8%
c. 48%
d. 52%✔

Temos 32 bits para representar todos os endereços possíveis.
Cada página tem 8KB de tamanho.
Para descobrir a quantidade de páginas existentes, é necessário dividir a quantidade total de memória pela quantidade de memória para cada página.
Ou seja:
(Total de memória) / (Memória para cada página) = quantidade de páginas
(2^32) / (8 KB) = (2^32) / (2^13) = 2^19 = 524288 páginas diferentes.

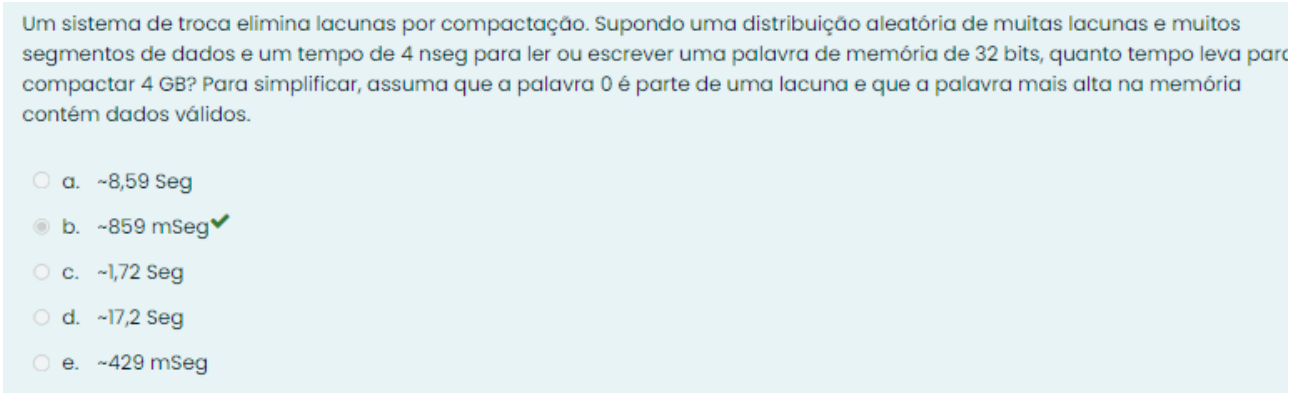
Se temos 524.288 páginas diferentes e o exercício diz que demora 100ns para cada entrada da tabela de páginas, então demora 100ns para carregar cada página.

Assim, o tempo total em ns para carregar todas as páginas é 524288 * 100 = 52428800ns
Transformando nanosegundos em milissegundos temos 52428800/10^6 = 52.429 milissegundos. (1ms = 10^6ns)

Ou seja, 52.429 milissegundos são necessários para carregar todas as páginas.
Como o tempo total de execução do processo é 100ms, então a fração do tempo gasto para carregar as páginas é 52.429/100 = 0.524 = 52%

Resposta: D)

QUESTÃO 3



Inicialmente temos que lembrar que a compactação serve para diminuir/eliminar as questões de fragmentação da memória.

Queremos compactar 4GB de memória.
Como cada palavra tem 32 bits = 4 bytes, então vamos ter 4GB/4B palavras diferentes.
4GB = 4294967296.
4294967296 / 4 = 1073741824 palavras diferentes
1073741824 /2 = 536870912

536870912 * 8 = 4294967296
4294967296/10^6 = 4294.967ms

Na compactação geralmente temos que realizar leitura de uma palavra e escrita em outro local.
Considerando que para cada palavra serão realizadas no mínimo uma leitura e uma escrita, então temos 4ns + 4ns = 8ns para cada palavra.

O tempo total da compactação vai ser: (quantidade total de palavras) * (tempo para cada palavra)
Tempo total da compactação: 1073741824 * 8ns = 8589934592ns
Transformando nanosegundos em milissegundos: 8589934592 / 10^6 = 8589.935ms ~= 8.59 seg

OBS: Considerando que apenas metade das palavras fossem mudadas de posição, então teríamos ~= 4seg.

QUESTÃO 4

Dê um exemplo simples de uma sequência (abaixo) de referências de páginas onde a primeira página selecionada para a substituição será diferente para os algoritmos de substituição de página LRU e de relógio. Presuma que 3 quadros sejam alocados a um processo, e a sequência de referências contenha números de páginas do conjunto 0, 1, 2, 3.

Na sequência : 0, 1, 2, 0, 2, 1, 3.

a. LRU: pág 1 pela 3, Relógio: pág 0 pela 3
b. LRU: pág: 0 pela 3, Relógio: pág: 2 pela 3 ✔
c. LRU: pág 2 pela 3, Relógio: pág 1 pela 3

LRU: Least recently used ⇒ Vai substituir a página mais antiga pela nova
Temos 3 quadros alocados a um processo.

Para LRU:
Sequência 0, 1, 2, 0, 2, 1, 3.
[vazio, vazio, vazio] ⇒ Estado inicial
[0, 1, 2] ⇒ Inserção do 0, 1 e 2 (0 é o mais antigo)
[0, 1, 2] ⇒ Como 0 já está no quadro, então permanece a mesma coisa (1 é o mais antigo)
[0, 1, 2] ⇒ Como 2 já está no quadro, então permanece a mesma coisa (1 é o mais antigo)

[0, 1, 2] ⇒ Como 1 já está no quadro, então permanece a mesma coisa (0 é o mais antigo)
[3, 1, 2] ⇒ Chegou a página 3 e o 0 é o mais antigo, logo, substitui a página 0 pela 3.
Resultado final: [3, 1, 2] com substituição da página 0 pela 3.

Para Relógio:
[0, 1, 2] ⇒ Ponteiro do mais antigo aponta para o 0
[0, 1, 2] ⇒ Chegou o 0, então agora o ponteiro do mais antigo para o 1
[0, 1, 2] ⇒ Chegou o 2, o 2 não era o mais antigo, então o ponteiro do mais antigo continua no 1
[0, 1, 2] ⇒ Chegou o 1, o 1 era o mais antigo, agora o ponteiro avança para o 2
[0, 1, 3] ⇒ Chegou o 3, que não estava no quadro, então pega a página apontada pelo ponteiro do mais antigo e substitui pela nova. Assim, a página 2 é substituída pela página 3.

Resposta: B)

QUESTÃO 5

Observou-se que o número de instruções executadas entre faltas de páginas é diretamente proporcional ao número de quadros de páginas alocadas para um programa. Se a memória disponível for dobrada, o intervalo médio entre as faltas de páginas também será dobrado. Suponha que uma instrução normal leva 1 ms, mas se uma falta de página ocorrer, ela leva 2001 μs (isto é, 2 ms) para lidar com a falta. Se um programa leva 60 s para ser executado, tempo em que ocorrem 15.000 faltas de páginas, quanto tempo ele levaria para ser executado se duas vezes mais memória estivesse disponível?

- a. 50 seg
- b. 30 seg
- c. 45 seg ✔
- d. 55 seg

Número de instruções executadas ⇒ diretamente proporcional ⇒ número de quadros de páginas.
Memória disponível for dobrada ⇒ intervalo médio entre as faltas páginas também será dobrado

Tempo instrução normal: 1ms
Tempo falta de página: 2ms

O programa leva 60s para rodar.
Tivemos 15.000 faltas de páginas, logo, o tempo decorrido para falta de páginas foi de 15000*2 = 30000ms = 30s.

Se o programa rodou no total 60s e tivemos 30s para faltas de páginas, então os 30 segundos restantes foram gastos executando instruções normalmente.
Se tivemos 30 segundos executando instruções normalmente e cada instrução demora 1ms, então tivemos 30.000 instruções sendo executadas corretamente.
Ou seja, 30000 instruções executadas sem ter falta de página e 15000 faltas de páginas.

Sabemos que quando duplicamos a memória, a quantidade de faltas de página cai pela metade.
Logo, ao duplicarmos a memória vamos ter 15000/2 = 7500 faltas de página.
Como cada falta de página leva 2ms, então vamos ter 2*7500 = 15000ms = 15s decorridos por falta de página.
As 30000 instruções (normais sem falta de página) vão estar sendo executadas normalmente.

Logo, 30s (das instruções executadas sem falta de página) + 15s das faltas de página = 45s.
Assim, o programa vai executar por 45 segundos.

Resposta: C)

QUESTÃO 6

Um computador cujos processos têm 1024 páginas em seus espaços de endereços mantém suas tabelas de páginas na memória. O custo extra exigido para ler uma palavra da tabela de páginas é 5 ns. Para reduzir esse custo extra, o computador tem uma TLB, que contém 32 pares (página virtual, quadro de página física), e pode fazer uma pesquisa em 1 ns. Qual frequência é necessária para reduzir o custo extra médio para 2 ns?

- a. 75% ✔
- b. 100%
- c. 50%
- d. 25%

Com TLB ⇒ Tempo é 1ns
Sem TLB ⇒ Tempo é 5ns

O que queremos saber é a frequência ou probabilidade de ocorrência das páginas na TLB para que o custo extra médio seja 2ns.

Ou seja, temos que fazer uma média ponderada de acordo com a probabilidade da página estar ou não na TLB

Custo extra médio esperado = probabilidade_na_tlb * 1ns + probabilidade_não_tlb *5ns
Vamos chamar essa probabilidade de estar na TLB de p.
Se a probabilidade de estar é p, então a probabilidade de não estar é (1 - p)
2ns = p * 1 + (1-p)*5
2ns = p + 5ns - 5p
4p = 3
p = 3/4
p = 75%
Ou seja, para que o custo extra médio esperado seja 2ns, a probabilidade da página estar na TLB tem que ser 75%

Resposta: A)