

# Lista de Exercícios - Memória

Rafael Gonçalves de Oliveira Viana

4º semestre de 2017

1. Em um sistema sem abstração de memória com múltiplos processos em memória, porque é necessário uma forma de realocação ?

**Resposta** Por que cada programa que esta rodando foi compilado como constantes, e para que um novo processo entre em execução é preciso liberar o endereço de memoria, salvando o processo anterior em disco(relaocando ele em disco), e então permitindo que o novo processo manipule a memória, sem prejudicar o programa anterior porém isso e custoso para o S.O.

2. Qual forma de realocação (estática ou dinâmica) é mais fácil de ser usada pelo S.O.? Cite um motivo pelo qual o S.O. não consiga utilizar o método mais fácil.

**Resposta** Realocação dinâmica o S.O e quem é que determina os valores do registrador-base e do registrador-limite, que faz o calculo de posicionamento diferente na memória física. Um dos principais motivos do sistema operacional não a utilizar é porque nesse esquema como o processo está totalmente "carregado"em memória, se o número de processos for muito grande, manter todos eles em memória estouraria sua capacidade, precisando então fazer swap a cada vez que um novo processo não "coubesse"em memória. Com a memória virtual, você só precisa manter partes do processo em memória, e não é necessário estar em partes consecutivas.

3. Suponha que o S.O. utilize realocação estática dos processos e a memória está muito fragmentada. Em um dado momento, o S.O. decide

compactar a memória, movendo os segmentos de processos para o início da memória. Esta operação seria mais eficiente se fosse usada realocação dinâmica? Porque?

**Resposta** Sim. Deixaria menos espaços fragmentados, sendo assim os espaços inutilizados seriam menores. Quando as trocas de processos deixam "muitos" espaços vazios na memória, é possível combiná-los todos em um único espaço contíguo de memória, movendo-os, o máximo possível, para os endereços mais baixos. Essa técnica é denominada compactação de memória. Ela geralmente não é usada em virtude do tempo de processamento necessário.

4. Suponha que três processos executam e necessitam, respectivamente, de 1GiB, 0,5GiB e 1,5GiB de memória. A máquina em que executam tem apenas 2GiB de memória. Qual forma de gerenciamento de memória seria mais eficiente nesta situação, sendo impossível alterar o código dos processos?

**Resposta** Os processos 2 e 3 são executados juntos, quando termina o processo 1 e colocado em memória.

5. Suponha que o S.O. implemente paginação de memória e apenas 10% das páginas de cada processo estejam na memória e todas demais em disco. Se houverem 10 processos prontos que ainda tem muito o que executar, é possível que executem sem que as faltas de páginas se tornem um gargalo de desempenho? Explique.

**Resposta** Sim, Se um programa estiver esperando por outra parte de si próprio ser carregada na memória, a CPU poderá ser dada a outro processo sem que ele fique bloqueado dando continuidade aos demais processos que também serão carregados do disco.

6. Suponha que o acesso ao endereço de memória  $1056 = 10000100000_2$  deva ser traduzido pela MMU em um endereço real de memória. Suponha ainda que a página 1 está na moldura 5 (a página 0 é a primeira). Calcule o endereço real para páginas de 997 bytes e para páginas de 1024 bytes. Você pode calcular em binário se achar mais fácil.

7. Suponha que você está projetando um sistema de paginação para uma máquina de 32 bits cujo endereço das páginas tem 14 bits e cada entrada da tabela de páginas deve conter 6 bits de status além do número da moldura. Se você usar uma única tabela de páginas, qual é o tamanho mínimo da tabela?

**Resposta** Sobram 17 bits, o tamanho de página seria de  $2^{17}$

8. Agora suponha que você está usando tabelas de página em dois níveis e que o endereço da página seja subdividido em duas partes desiguais. A primeira parte, usada para endereçar a tabela de primeiro nível tem 6 bits. A segunda parte tem 12 bits e endereça as tabelas de segundo nível. Qual o tamanho máximo da tabela de páginas (sobram 14 bits para o deslocamento) considerando que cada entrada precisa de 6 bits de status?

**Resposta** 32 bits divididos em um campo PT1 de 6 bits, um campo PT2 de 10 bits e um campo Deslocamento de 14 bits, as páginas são de tamanho 16 KB. Os outros dois campos tem conjuntamente 16 bits o que possibilita um total de  $2^{20}$

9. A ideia por trás do endereçamento em dois níveis é usar menos memória para armazenar a tabela de páginas. Como isso é possível no caso acima descrito?

**Resposta** Simples a ideia é evitar manter na memória todas as tabelas de página o tempo todo, fazendo ela carregar apenas a tabela principal e se necessário trazer para memória a tabela secundária.

10. Suponha que a TLB tenha 32 entradas. Qual o tamanho máximo do espaço de endereçamento que a TLB pode traduzir se o tamanho da página é de 4KiB?
11. Suponha uma máquina de 64 bits com páginas de 4KiB e um total de 8GiB de memória física. O gerenciador de memória do S.O. utiliza uma tabela de páginas invertida que armazena, em cada linha, o número da

página, o número do processo (14 bits) e mais 6 bits de estado. Qual é o tamanho da tabela?

12. Qual é o maior problema com as tabelas de páginas invertidas? Indique algumas maneiras de minimizá-lo.
13. Suponha que uma falta de página leve 10ms para ser resolvida (trazer a página para a memória) e que uma instrução normalmente executa em 1ns (se não houver faltas de páginas). Um determinado programa leva 3 minutos para executar e sofre um total de 15.000 faltas de página. Seu dono decide, então, dobrar a quantidade de memória da máquina e observa que o número de faltas de página reduziu-se pela metade. Quanto tempo o programa demora para executar na nova configuração?
14. Suponha uma máquina com apenas 4 molduras de página e um processo com 8 páginas. Inicialmente todas as molduras estão livres e o processo começa a executar, fazendo acesso às seguintes páginas:

1, 1, 0, 2, 3, 2, 5, 0, 4, 0, 2, 6, 2, 0, 3, 5, 3, 2, 7, 3

Simule a execução do algoritmo FIFO (mais antigo sai) e do relógio. Isto é, determine qual página é removida da memória para dar lugar a cada página ausente.

15. Retome a situação anterior e suponha que a cada 5 referências à memória ocorre um ciclo de atualização de contadores. Simule os algoritmos NFU e WSClock.