PROVA 02 – 2012/01 (Turma B)

Observação: as respostas abaixo salientam apenas os principais conceitos e não se considera que elas sejam a resposta padrão. Na resposta a prova se espera que essas ideias sejam elaboradas de acordo com o solicitado e, respostas diferentes, bem fundamentadas, são avaliadas de acordo com sua correção e argumentação.

1ª Questão

(a) A sequência de carga de páginas na memória é P2 (t=120); P0 (t=126); P3(t=160);P4(t=230). Após carregadas elas são referenciadas na ordem P1(t=260); P2(t=271); P0(t=279) e P3(t=280). Para essa sequencia de carga e acesso, as páginas vítimas são:

- FIFO: P2, pois foi a primeira a ser carregada na memória;
- LRU: P1, pois foi a referenciada a mais tempo;
- Segunda chance: P0; pois é feita a escolha com base no bit de referência, na ordem em que as páginas foram carregadas, quem tiver o bit de referência em zero é a vítima. Se o bit estiver em 1, na primeira varredura ele é zerado.
- Segunda chance melhorado: P0; pois é feita a escolha com base no bit de referência, na ordem em que as páginas foram carregadas, quem tiver o bit de referência em zero é a provável vítima. Se o bit estiver em 1, na primeira varredura ele é zerado. As páginas com bit de referência em zero são analisadas em função do bit de modificação. A preferência é por aquelas não modificadas, ou seja, bit de modificação zerado.

(b) Amarração dinâmica entende-se a correção de endereços para endereços físicos feitos por uma MMU (Memory Management Unit). Nesse caso, um carregador pode gerar código absoluto, ou seja, o código executável gerado pelo procedimento de ligação não tem nenhuma lista de endereços a serem corrigidos (isso é característico de códigos com relocação). O montador pode gerar tanto código absoluto como código relocável. O montador só precisa ser coerente com a fase de ligação para gerar endereços absolutos não superpostos ou gerar um binário com informação de relocação para o ligador corrigir os endereços na fase de geração do código executável final.

2ª questão

(a) Teoricamente, a memória disponível no sistema é a soma da capacidade da RAM (3 GB) com a área de swap (9 GBB), ou seja, se pode usar no máximo 12 GB de memória. Dessa forma, para um processador de 32 bits, o maior processo é limitado pela capacidade de endereçamento do processador (2^{32}), ou seja, 4 GB. Para o processador de 64 bits, o fator limitante é a capacidade total disponível, ou seja, o maior processo passa a ter um tamanho equivalente de 12 GB. Em ambos os casos, é possível ter seis processos de 2 GB no sistema, pois eles consomem todos os 12 GB disponíveis (2 GB x 6 processos).

(b) SIM. Basta criar segmentos de tamanhos fixos e idênticos. Um segmento de tamanho fixo e idêntico é equivalente a uma página. A tabela de segmentos passa a se comportar como uma tabela de páginas.

3ª Questão

(a) Os mecanismos de alocação com partições fixas ou dinâmicas exigem um espaço contíguo de memória. Já, tanto a paginação como a segmentação, são mecanismos que permitem usar um espaço de endereçamento não contiguo para atender requisições de memória.

(b) Com uma alocação contígua é necessário pré-reservar uma área para o heap no espaço de endereçamento do processo. Se essa pré-reserva não for suficiente o processo parará por erro de alocação de memória. Se for superdimensionado, haverá desperdício de memória. Com a paginação, basta alocar novas páginas de memória para o heap. Se não houver mais páginas disponíveis, o processo para com erro de alocação de memória. Com segmentação, a solução é similar a paginação: enquanto houver memória disponível, se pode alocar novos segmentos para compor a área de heap. (obs.: eventualmente se poderia aumentar o tamanho do segmento, mas, para isso, deve haver espaço livre a partir do último endereço do segmento antes de sua expansão. Não é a melhor solução)

4ª questão

(a) Thrashing é a situação em que ocorre uma paginação (page-in e page-out) excessiva no sistema acompanhado de uma baixa eficiência de uso da CPU (atenção para o atendimento simultâneo dessas duas condições). Isso é causado pelo fato de que os processos tem uma quantidade de quadros alocados menor que a necessidade deles (working set). A detecção pode ser feita analisando o working set de todos os

processos. Se a somatório da cardinalidade dos working set dos processos for maior que a quantidade de quadros do sistema, o mesmo estará em thrashing. A providência a ser tomada é a suspensão (escalonador de médio prazo) de um ou mais processos ou mesmo a eliminação (kill) desses.

(b) O endereço lógico da paginação é dado por $E_L = p + d$; onde $p = E_L \bmod p$ e $d = E_L \text{ div } p$, portanto, conceitualmente, o tamanho da página pode ser qualquer valor. O fato de ser usado potência de 2 é que isso elimina a necessidade de qualquer operação matemática para se calcular as parcelas p e d e facilitar a tradução do endereço lógico para endereço físico. Com páginas em potência de 2, o endereço lógico de n bits é simplesmente dividido em duas partes: página e deslocamento. Isso naturalmente implementa as operações mod e div.

5ª questão

(a) O maior processo, considerando as tabelas, ocupará toda a memória disponível do espaço de endereçamento, ou seja, $2^{32}/2^{12} = 2^{20}$ páginas. O menor processo ocupará três páginas: uma para o diretório de tabela de páginas, uma para uma única tabela de páginas e uma única página para o processo.

(b) Existe apenas uma tabela de diretório de páginas com 1024 entradas.

(c) Existe 1024 tabelas de páginas (corresponde a quantidade de entradas da tabela de diretório de páginas) e cada tabela de páginas possui 1024 entradas.

(d) $t_{médio} = 0{,}95x(10+100) + (0.05)(10+100+100) = 115$ ns; o que corresponde a 95% das vezes se fazer um acesso a TLB (10 ns), recuperar a informação de tradução de página para quadro, e acessar a posição de memória desejada (100 ns); e, em 5% das vezes, se acessa a TLB (10 ns), não se encontra as informações de tradução, sendo necessário acessar a tabela de páginas em memória (100 ns), traduzir o endereço lógico em físico, e então acessar a posição de memória desejada (100 ns).