



Curso de Tecnologia em Sistemas de Computação
Disciplina de Sistemas Operacionais
Professores: Valmir C. Barbosa e Felipe M. G. França
Assistente: Alexandre H. L. Porto

Quarto Período
AP2 - Segundo Semestre de 2008

Nome -
Assinatura -

Observações:

1. Prova sem consulta e sem uso de máquina de calcular.
 2. Use caneta para preencher o seu nome e assinar nas folhas de questões e nas folhas de respostas.
 3. Você pode usar lápis para responder as questões.
 4. Ao final da prova devolva as folhas de questões e as de respostas.
 5. Todas as respostas devem ser transcritas nas folhas de respostas. As respostas nas folhas de questões não serão corrigidas.
-

1. (1.5) Um dos métodos que vimos para prevenir impasses é os processos requisitarem todos os recursos antes de começarem a sua execução. Quais são os problemas de usarmos este método? Algum dos métodos vistos em aula evita esses problemas?

Resp.: -Existem dois problemas com este método. O primeiro deles é que nem sempre é possível para o programador, ao desenvolver o programa cujo código está sendo executado pelo processo, saber quais recursos serão usados durante a execução. Isso ocorrerá se, por exemplo, o programa usar funções de alguma biblioteca que aloquem recursos para executar as suas tarefas. O outro problema é que os recursos não são necessariamente usados simultaneamente pelo processo e, portanto, isso faria com que cada recurso ficasse ocioso até ser usado, até ser liberado. Isto reduziria a eficiência do sistema em relação ao uso dos recursos pelos processos.

-Sim, poderíamos usar a técnica vista no final da Aula 7, que numera os recursos do sistema. Os dois problemas seriam evitados pois, como esta técnica evita os impasses, o processo poderia requisitar o recurso somente quando ele fosse necessário e depois de usá-lo, liberá-lo. Logo, o programador não precisaria saber, a priori, de todos os recursos usados pelo programa a ser executado. Além disso, a ociosidade de cada recurso poderia ser minimizada, pois para isso bastaria que cada processo requisitasse o recurso imediatamente antes de usá-lo e o liberasse imediatamente após usá-lo.

2. (1.5) Descreva como ocorre o acesso aos endereços da memória principal quando o sistema operacional usa memória virtual, destacando a função da MMU.

Resp.: Se o sistema operacional usa memória virtual, os endereços acessados pelas instruções do processador são endereços virtuais do espaço de endereçamento virtual, e não os endereços físicos da memória. Logo, quando uma instrução que acessa um destes endereços é executada, após decodificar a instrução e descobrir que ela precisa acessar um endereço virtual para ser executada, o processador passa este endereço para a MMU. A MMU, ao receber este endereço, o divide em

dois campos: o número da página virtual e o deslocamento dentro desta página virtual. O número da página virtual será usado para descobrir se a página virtual está na memória e, caso esteja, qual a moldura de página em que ela está mapeada e as suas permissões de acesso. Se a MMU possuir uma TLB, então primeiramente a MMU verificará se o mapeamento da página está nela e, caso esteja, irá compor o endereço virtual e verificar as permissões de acesso, como descrito a seguir. Caso a MMU não possua uma TLB ou o mapeamento não esteja na TLB, o número da página virtual será usado para obter a entrada da tabela de páginas com as informações da página acessada. Note que se o sistema usar uma tabela de páginas multinível, o número da página virtual será adicionalmente dividido para obtermos as entradas em cada uma das tabelas de páginas de cada um dos níveis. Se a página não estiver na memória, uma falha de página será gerada para carregar a página do disco para a memória. Após garantir que a página está na memória, a MMU verifica se o acesso que a instrução deseja fazer ao endereço virtual é válido. Se o acesso não for válido, uma interrupção será gerada para que o sistema saiba que o processo tentou executar uma operação ilegal. Caso contrário, o número da moldura de página é usado para obter, junto com o deslocamento, o endereço físico correspondente ao virtual. Os bits superiores deste endereço serão o número da moldura de página na base binária e os bits inferiores serão o deslocamento na base binária. Finalmente, após obter o endereço físico, ele é colocado no barramento da memória, permitindo que o processador termine a execução da instrução que acessou o endereço virtual.

3. (1.5) Quais são as vantagens de a MMU poder armazenar toda a tabela de páginas do processo em execução em comparação à MMU somente possuir um ponteiro para a tabela de páginas do processo? E quais são as desvantagens?

Resp.: -Existe somente uma vantagem em colocar a tabela de páginas na MMU: a de que o mapeamento das páginas virtuais em molduras de página pela MMU será bem mais rápido. Neste caso, a tabela de páginas do processo será armazenada em registradores da MMU ao invés da memória do computador. Agora, acessar os registradores da MMU é muito mais rápido do que acessar as posições da memória do

computador, o que seria necessário se a MMU possuísse somente um ponteiro para a tabela de páginas do processo.

-Existem duas desvantagens em colocarmos a tabela de páginas do processo na MMU. A primeira é a de que cada troca de contexto será bem mais lenta do que quando temos somente um ponteiro para a tabela de páginas do processo na MMU. Isso ocorrerá porque quando o escalonador for chamado, toda a tabela de páginas do processo em execução deverá ser salva na memória e, depois disso, toda a tabela de páginas do processo escolhido para ser executado deverá ser copiada da memória para os registradores da MMU. Agora, se a MMU possuir somente um ponteiro para a tabela de páginas, bastará que o mudemos para o ponteiro da tabela de páginas do processo que foi escolhido. O outro problema é que é muito mais custoso (hardware) armazenar toda a tabela de páginas nos registradores da MMU, ao invés de armazená-la na memória, cujo custo de armazenamento é menor, do que colocar na MMU somente um registrador apontando para o início desta tabela de páginas.

4. (1.5) Suponha que o computador usa a técnica de segmentação. Faz sentido colocar uma TLB na MMU? E se o computador usar a técnica de segmentação com paginação?

Resp.: -Neste caso, não faz sentido colocarmos uma TLB na MMU, porque tabelas de páginas não são usadas na segmentação, e cada entrada da TLB guarda o mapeamento de uma página virtual do endereço virtual para a moldura de página que contém o endereço físico correspondente a este endereço. Porém, como, na segmentação, o acesso a um endereço de um segmento é convertido para um endereço físico a partir do endereço inicial do segmento na memória, poderíamos usar uma estrutura similar, mas não idêntica a uma TLB, para armazenar os endereços iniciais dos segmentos mais acessados.

-Neste caso, teria sentido colocar uma TLB na MMU pois, nesta técnica, o espaço de endereçamento linear de cada segmento é dividido, assim como o espaço de endereçamento virtual na paginação, em páginas. Logo, a TLB é necessária para armazenar o mapeamento das páginas virtuais mais acessadas para as molduras de página correspondentes. Note que agora a TLB também deverá ter, em cada uma de suas en-

tradas, o número do segmento, pois o espaço de endereçamento linear de cada segmento é dividido em páginas virtuais.

5. (1.5) Qual é a principal vantagem da técnica de alocação contígua sobre as outras técnicas de alocação de blocos do disco? E qual é a principal desvantagem?

Resp.: -A principal vantagem é a de que na técnica de alocação contígua, todos os blocos lógicos de um arquivo são blocos consecutivos do disco, o que faz com que o acesso ao arquivo seja mais rápido, porque é mais rápido ler blocos consecutivos do disco. Nas outras técnicas, como vimos na Aula 11, blocos lógicos consecutivos do arquivo não são necessariamente blocos consecutivos do disco. Note que isso ocorre porque precisamos executar a operação de busca no disco somente para o primeiro bloco, no caso da alocação contígua, enquanto que isso pode ser feito para cada bloco, no pior caso, nas outras técnicas de alocação. -A principal desvantagem advém do fato de um arquivo usar blocos que são consecutivos no disco. Isso não somente gera uma fragmentação externa no disco, conforme os arquivos são criados e removidos, como limita o tamanho do arquivo ao maior número de blocos consecutivos livres no disco. Nas outras técnicas de alocação, um arquivo pode usar todos os blocos livres do disco, pois não é necessário que o arquivo esteja armazenado em blocos consecutivos do disco.

6. (2.5) Suponha que o sistema operacional usa uma cache de blocos do disco. O uso desta cache pode gerar inconsistências no sistema de arquivos? Justifique a sua resposta.

Resp.: Poderemos ter inconsistências no sistema de arquivos se a cache possuir blocos cujos conteúdos são essenciais à estrutura do sistema de arquivos. As inconsistências ocorrerão se estes blocos não forem imediatamente salvos após serem alterados, isto é, se o sistema não usar uma cache de gravação intermediária, em que as alterações de um bloco são imediatamente salvas no disco. Se um destes blocos for modificado e não for salvo no disco porque o sistema operacional parou de funcionar (por exemplo, a energia acabou), isso certamente

causará uma inconsistência no sistema de arquivos, pois este bloco foi modificado devido a alguma alteração nesta estrutura do sistema de arquivos. Note que se todos os blocos da cache pertencerem a arquivos, não teremos uma inconsistência no sistema de arquivos e sim, perda de dados em um ou mais arquivos do sistema de arquivos.