



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 - Primeiro Semestre de 2010

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. (2.5) Diga se as seguintes afirmativas são falsas ou verdadeiras. Para responder, escreva apenas F ou V para cada item em seu caderno de respostas.

- (a) (0.5) Um sistema operacional, como parte do gerenciamento dos dispositivos físicos do computador, acessa diretamente as partes eletrônicas e mecânicas de cada um destes dispositivos, usando as instruções de E/S do processador.

Resp.: F (Falsa, pois o sistema operacional não acessa diretamente as partes eletrônicas e mecânicas de um dispositivo de E/S. Ao invés disso, ele envia comandos a cada controladora, que então acessa estas partes dos dispositivos ao executar os comandos).

- (b) (0.5) A diferença entre a E/S mapeada em memória e a E/S usando um espaço de endereçamento especial é que, no primeiro caso, os registradores da controladora de cada dispositivo estão na memória podendo, portanto, ser acessados com as instruções de acesso à memória, ao invés de instruções de E/S.

Resp.: V (Verdadeira).

- (c) (0.5) Um recurso nem sempre precisa ser usado exclusivamente por um processo, ou seja, alguns recursos podem ser simultaneamente usados por dois ou mais processos.

Resp.: F (Falsa, pois um recurso sempre deve ser usado exclusivamente por um processo, mesmo que seja preemptivo. Neste último caso, isso simplesmente significa que o recurso pode ser retirado do processo, se for necessário).

- (d) (0.5) Um grafo de recursos é um grafo orientado que representa o estado da alocação dos recursos. Neste grafo existe um vértice para cada recurso e um para cada processo. Existe também uma aresta de um processo para um recurso se o processo deseja obter o recurso, ou uma aresta de um recurso para um processo se o

processo já possui o recurso.

Resp.: V (Verdadeira).

- (e) (0.5) A técnica de prevenção de impasses baseada em numerar os recursos foi desenvolvida para impedir a condição de espera circular. Isto ocorre porque, ao numerar os recursos e impedir que um processo não possa alocar um recurso com um número menor do que os números dos recursos que ele já possui, não são formados ciclos no grafo de recursos.

Resp.: V (Verdadeira).

2. (1.5) Descreva os problemas de realocação e proteção de memória, destacando como os registradores de base (RB) e de limite (RL) podem ser usados para ajudar a resolver estes problemas.

Resp.: Quando usamos os registradores de base e de limite, qualquer acesso a um endereço por uma instrução do programa é convertido, antes de o processador acessar o endereço, no endereço físico igual à soma do endereço acessado ao valor do registrador base. Além disso, o registrador limite é usado para impedir que qualquer instrução acesse um endereço maior ou igual ao endereço correspondente ao registrador limite. Como vimos na Aula 8, o problema de realocação ocorre sempre quando não podemos garantir que as instruções de acesso à memória acessem os endereços físicos corretos. Este problema pode ser evitado se compilarmos o programa sempre a partir do endereço 0 e se, ao carregarmos este programa em uma partição da memória, alterarmos o registrador base para o endereço inicial da partição. Já o problema de proteção ocorre quando não podemos impedir que um programa acesse endereços físicos associados a outros programas. O problema também pode ser evitado se fizermos o mesmo que fizemos para resolver o problema anterior e, adicionalmente, definirmos o valor do registrador limite como o tamanho da partição em que o programa foi carregado. O problema é evitado porque, quando a instrução de acesso à memória acessar um endereço, ele nunca será maior do que o

maior endereço da partição.

3. (1.5) Descreva o algoritmo de idade usado para implementar, em software, uma aproximação do algoritmo LRU, destacando a sua vantagem sobre o algoritmo NFU.

Resp.: O algoritmo de idade escolhe a página a ser substituída do modo descrito a seguir. Ele associa, a cada página na memória, um contador, que é inicializado em 0 quando a página é copiada para uma moldura de página. A cada interrupção do temporizador, para cada página na memória, o seu contador é deslocado em 1 bit para a direita e, depois disso, o valor atual do bit referenciada (R) desta página é copiado para o bit mais significativo do seu contador. A página escolhida para ser a substituída é aquela o menor contador. Se existirem duas ou mais páginas com o mesmo valor mínimo para o seu contador, uma delas é escolhida de modo aleatório. A vantagem sobre o algoritmo NFU é que este último pode deixar de escolher uma página que já não é acessada há algum tempo, escolhendo em vez dela uma que tenha sido acessada recentemente. Isto ocorre porque o algoritmo NFU, ao contrário do algoritmo de idade, adiciona o bit R ao contador e, com isso, uma página não acessada já há algum tempo pode não ser escolhida porque, no passado, foi mais acessada do que as páginas atualmente acessadas. O algoritmo de idade não possui esta desvantagem, porque as páginas acessadas no passado sempre têm os seus bits R iguais a 0 e, com isso, a cada interrupção do temporizador, os seus contadores possuem valores cada vez menores, até eventualmente se tornarem 0.

4. (1.5) Descreva como a TLB é usada para acelerar o mapeamento dos endereços virtuais nos endereços físicos em um sistema operacional que use a técnica de segmentação com paginação.

Resp.: Assim como ocorre na paginação, a TLB é usada para converter, sem a necessidade de serem acessadas as informações dos segmentos e as tabelas de páginas, os endereços virtuais bidimensionais

associados aos pares (segmento, número da página virtual) mais usados. Cada entrada da TLB é então composta, pelo menos, por cada par (segmento, número da página virtual) mais usado, e pelo número da moldura de página na qual a página virtual do par está armazenada (em geral, teremos outras informações como, por exemplo, as informações de proteção que definem como a página pode ser acessada). Quando a TLB é usada, ao convertermos um endereço virtual bidimensional (segmento, endereço dentro do segmento), primeiramente precisamos descobrir qual é o número da página virtual associada a este endereço. No MULTICS, este número é obtido ao dividirmos o espaço de endereçamento do segmento em páginas. Porém, nos processadores Pentium da Intel, primeiramente devemos mapear o endereço dentro do segmento, usando a base do segmento, em um espaço de endereçamento comum, no qual todos os segmentos do processo em execução são mapeados, para depois obtermos o número da página. Depois de descobrirmos o número da página virtual, então verificamos se o par (segmento, número da página virtual) está na TLB. Se o par estiver na TLB, então obtemos o número da moldura de página diretamente da entrada da TLB com este par. Caso o par não esteja na TLB, então devemos obter o número da moldura do modo normal, usando as informações do segmento e as tabelas de páginas. Em ambos os casos, após obtermos o número da moldura, usamos este número e o deslocamento dentro da página virtual para obter o endereço físico associado ao endereço virtual bidimensional.

5. (1.5) Descreva como funciona a abertura de um arquivo nos sistemas UNIX, ou seja, como é obtido o nó-i associado a este arquivo.

Resp.: Primeiramente, se o caminho dado para o arquivo a ser aberto for relativo, deveremos convertê-lo para um caminho absoluto, usando o diretório de trabalho do processo que deseja abrir o arquivo como a base para este nome relativo. Então, supondo que temos um nome de caminho absoluto para o arquivo a ser aberto, primeiramente dividimos este nome em componentes, usando o separador “/” dos diretórios do UNIX. Agora, começando com o nó-i associado ao diretório raiz, cujo ponteiro fica armazenado em um local pré-determinado, vamos executar o algoritmo descrito a seguir. Para cada componente do caminho,

sendo que os componentes são considerados da esquerda para a direita, inicialmente procuramos o componente no diretório associado ao nó- i que estamos avaliando. Se o componente não estiver no diretório, ou seja, se não existir uma entrada para ele no arquivo especial que representa o diretório, então o caminho é inválido, e devemos gerar um erro. Em caso contrário, a partir da entrada do componente no diretório, obtemos o seu nó- i . Se este componente não for o último do caminho, então deveremos repetir o algoritmo descrito para o próximo componente, usando agora o nó- i recém obtido. Se for o último, o nó- i obtido é o do arquivo a ser aberto, e pode então ser usado para abrir o arquivo e permitir o acesso ao seu conteúdo.

6. (1.5) Quando um arquivo é removido, seus blocos geralmente são devolvidos à lista de livres, mas eles não são apagados. Você acha que seria uma boa idéia ter o sistema operacional apagando cada bloco antes de liberá-lo? Considere tanto os fatores segurança como desempenho em sua resposta e explique o efeito de cada um.

Resp.: Em relação à segurança do sistema, isto seria o ideal, pois caso o bloco devolvido à lista de livres pertencesse a um arquivo com informações confidenciais, estas ficariam expostas a usuários mal intencionados. Neste caso, um usuário mal intencionado poderia obter parte desta informação caso o bloco fosse de algum modo alocado a um arquivo criado por este usuário. Mas o desempenho do sistema seria degradado, pois como precisaríamos fazer uma escrita de um bloco no disco sempre que este bloco fosse devolvido à lista de livres, o número de escritas no disco aumentaria consideravelmente. Uma idéia interessante, implementada no mainframe Burroughs, foi a de apagar somente os blocos dos arquivos marcados como críticos pelo usuário, o que permite manter o sistema seguro sem reduzir muito o seu desempenho.