

1ª prova 16/12/2003:

- 1) Qual problema é resolvido(e como é resolvido) pelo overlay? É necessário alguma facilidade especial de hardware para a implementação de overlay? Qual? Cite, exemplificando uma outra solução para o mesmo problema que NÃO necessite de facilidade especial de Hardware.

O problema resolvido é a falta de memória, ou seja, quando o usuário quer executar processos que não cabem na memória da máquina. O overlay divide o processo em partes iguais e aloca na memória apenas instruções e dados que são necessárias em determinado momento. Quando outras instruções são necessárias elas são carregadas no espaço que foi anteriormente ocupado por instruções que não são mais necessárias.

O overlay não precisa de suporte de Hardware sua implementação exige muito do programador. O swap de processos não precisa de suporte especial de HW. Se a memória não tem espaço para executar um processo, um outro processo é removido da memória para um armazenamento auxiliar temporariamente liberado espaço para o novo processo executar e, em seguida, retornando a memória para continuar executando.

- 2) Qual a relação entre a falta de página e página vítima? Essa relação sempre existe? Por quê?

A relação é quando um processo solicita uma página e a página não foi carregada na memória, quando tenta carrega-la não há página livre na memória. Então o SO utilizando um algoritmo de substituição escolhe uma página (página vítima) para salvar em disco, liberando espaço para a página que gerou a falta de página. Essa relação existe, pois se a memória estiver toda ocupada e ocorrer falta de página, o SO tem que escolher uma página vítima para dá lugar a essa página que gerou a falta. Na realidade, o SO sempre deixa algumas páginas livres, quando pode escolher páginas vítimas. Dessa forma otimiza o tempo, ou seja, o tempo que levaria para escolher uma página vítima e fazer todo processo necessário de tratamento dessa página, não seria junto com o carregamento da página que gerou a falta em situações que a memória estaria toda ocupada. Com isso nem sempre a falta vai startar a escolha de página vítima. Se há páginas livres necessariamente não precisa escolher página vítima para gravar no disco.

- 3) Por que o polling (esperaocupada) tem um loop de espera? Qual é (e como é) o mecanismo de HW que possibilita existir uma melhor forma de resolver o problema.

Porque as vezes está em laço, lendo o registrador status várias e várias vezes até que o bit ocupado esteja desativado. Se a controladora e o dispositivo forem rápidos, esse método será razoável. Se o polling se tornar ineficiente, quando testado repetidamente, raramente encontrando um dispositivo pronto para serviço, enquanto outros processadores úteis dfa CPU permanecem por fazer. Em tais casos, pode ser mais eficiente fazer com que a controladora notifique a CPU quando o dispositivo ficar pronto para serviço, em vez de exigir que a CPU faça repetidas consultas para verificar a conclusão de uma operação de I/O . O mecanismo de HW que permite que um dispositivo notifique a CPU é a interrupção.

- 4) A) Diga quais são, exemplificando, os 3 tipos de tempo gastos na leitura de um disco.

B) Diaga quais são, exemplificando, as 3 dimensões que precisam ser identificadas para o acesso ao disco. Explique como este acesso tridimensional pode ser simplificado (use o conceito de cilindro).

O tempo gasto na leitura de um disco é a soma de três tempos:

O tempo de seek: é o tempo gasto para que a cabeça de leitura se desloque até a trilha onde está o setor a ser lido.

O tempo de rotação: é o tempo gasto para que o setor a ser lido passe sob a cabeça de leitura.

O tempo de latência: é o tempo gasto na transferência de um conjunto de bytes do disco para a memória.

Nos discos modernos há vários pratos, um em cima do outro, com um espaçamento entre eles.

Um prato tem 2 superfícies, logo, precisam ser informados 3 informações para ler/grava/apagar um dado: trilha (faixa do disco concêntricas). As trilhas se dividem em setores (uma fatia do disco) **Não sei a definição de setores.**, e superfície.

- 5) Processos podem se comunicar através de mensagens. O envio de mensagens de um processo a outro processo costuma ser implementado através da cópia do conteúdo da mensagem que está em um processo para uma variável do outro processo. Como seria possível, com a paginação, fazer o envio da mensagem SEM a cópia do conteúdo da mensagem.

Um processo A quer mandar uma mensagem para o processo B. O processo A manda a mensagem para a memória física (send (msg, B)) uma página, na qual contém a mensagem e sua tabela de páginas está marcada com o bit de presença ativo para esta página. O processo B compartilha essa página na memória física, com isso o processo A desabilita na sua tabela de página o bit de presença desta página, e o processo B habilita na sua tabela de página (msg = receive());

Prova 15/12/2003

- 1) Como o problema da fragmentação externa pode ser resolvido quando existe paginação?
Ocorre fragmentação externa quando os processos estão alocados na memória de forma não contínua existindo espaço total para alocação de um processo, porém ele não pode ser alocado porque o espaço não é contínuo. Com a paginação a memória é dividida em páginas de tamanho fixo (geralmente 4 KB), onde os processos tem sua tabela de página onde se mapeado o endereçamento lógico e físico de cada página.

- 2) Qual o problema que existe em relação aos endereços utilizados quando um programa é carregado na memória e se torna um processo? Como este problema pode ser resolvido quando existe paginação?

A amarração de endereços pode ser feita em diversas fases nas quais preparam a criação de um programa executável: os processo possui um nome e um endereço,

Printf("o valor ..%d",i);

Na linguagem de máquina CALL 73F

quando são carregados a HW fazer a tradução do nome para linguagem de máquina, dependendo do modo que foi carregado (endereço em tempo de carga, endereçamento relativos, registrador-base) vão existir várias correções de endereços durante a execução, carregamento, compilação..

Na paginação o processo é quebrado em páginas de tamanho fixo, todo processo tem seu espaço de endereçamento lógico no qual o processo começa por zero e acha que está sozinho na memória. As páginas são alocadas na memória física quando vão ser executadas, e existe uma tabela para cada processo, que faz o mapeamento do endereçamento lógico para o físico. Não sendo necessário alocação contínua de páginas na memória física.

- 3) Diga quais são, exemplificando, os 6 bits que podem existir na tabela de páginas. Quais são os 3 bits utilizados (e como estes são utilizados) durante o mecanismo de resolução dos problemas de falta de memória.

Bit presente – informa se a página está na memória bit =1 (válido) ou se estiver inválido bit = 0 a página não pertence ao processo, a página pode ter sido escolhida como vítima; a página pode nunca ter sido carregada, está em disco.

Bit acessado (ligado/desligado): bit utilizado na implementação do algoritmo de substituição de páginas de 2ª chance ou do algoritmo relógio. Indicando se a página foi ou não acessada recentemente.

Bit de escrita ou alterável: é aquele que se estiver ligado (bit=1) indica que o conteúdo da página pode ser alterado, se o bit=0 o conteúdo da página não poderá ser modificado.

Bit executável: indica se a página considerada pode ou não ser executada. Nas páginas de código este bit se encontra ligado e nas páginas de dados o mesmo se encontra desligado, para que não se execute uma página que não seja de código.

Bit núcleo: ligado a página pertence ao SO, e por isso os usuários não tem permissão de escrita, modificação, somente o administrador do SO tem este tipo de permissão, desligado não pertence ao SO.

Bit alterado: o bit é ativado pelo HW sempre que qualquer palavra ou byte na página for alterado, indicando que a página foi modificada. Quando uma página é selecionada para substituição, examinamos seu bit de modificação. Se o bit estiver ativado, sabemos que a página foi modificada desde que foi lida do disco. Nesse caso, devemos gravar a página no disco. Se o bit não estiver ativo a página não foi modificada desde que foi carregada na memória.

- 4) O que é e como funciona a TLB? Explique porque seu bom funcionamento é de extrema importância em uma das soluções do problema de tabela de páginas grandes.

É um mecanismo interno a CPU chamado de memória associativa (memória rápida) ou TLB que aumenta a velocidade da tradução dos endereços lógicos para físicos.

Na primeira vez que o HW faz o mapeamento de uma página o HW vai na tabela de páginas gastando 1 acesso a memória. O HW coloca os valores das páginas reais e virtuais numa memória especial, interna ao chip da CPU. Conforme o HW vai fazendo acesso a memória ele vai completando as linhas da TLB. Se mais tarde vier uma instrução que precise de uma determinada página, o HW antes de ir na tabela de página ele verifica se já existe na TLB a tradução da página, se existir o HW não precisa consultar a tabela de páginas. Com essa memória associativa se consegue traduzir o endereço de virtual para físico de forma rápida. Quando a TLB enche, o HW tem que escolher um mapeamento que já foi feito e jogar fora para gravar o mapeamento de novos endereços a serem usados, o que leva um tempo maior pois ocorrerá dois acesso , 1 na TLB e outra na tabela de páginas. A idéia é o HW manter automaticamente na TLB o mapeamento das últimas páginas utilizadas.

A importância da TLB é que existe HW que não possui tabela de páginas e esse mapeamento é feito pela TLB somente.

- 5) O swap de processos resolve 2 problemas em sistemas operacionais diferentes. Como e quais são estes problemas e soluções?

1º problema: falta de memória, neste caso o processo é maior que a área livre disponível na memória, o SO remove um processo da memória temporariamente para um armazenamento auxiliar e, em seguida, carrega o processo que vai ser executado.

2º problema: Thrashing

Este caso é específico para o SO UNIX. Quando ocorre excessiva paginação, o sistema não faz processamento. Todos os processos gastam seu tempo de execução simplesmente fazendo paginação. Nesse caso o SO escolhe um ou mais processos que estão causando o Thrashing e executa um swap de tais processos para o disco, aguarda até que os outros processos terminem e em seguida aloca os processos bloqueados para a execução.