Capítulo 7 – Patterson

7.4 – Memória Virtual

* Memória virtual é uma técnica que usa a memória principal como uma cache para armazenamento secundário.
* Duas motivações para memória principal: Permitir o compartilhamento seguro e eficiente da memória entre vários programas e remover transtornos de programação de uma quantidade pequena e limitada de memória principal.
* A memória principal precisa somente conter as partes ativas dos muitos programas.
* É necessário proteger programa uns dos outros restringindo o acesso a certos espaços de endereçamento.
* Endereço físico, endereço na memória principal.
* Memória virtual gerencial os dois níveis de hierarquia (memória principal e memória secundárias) automaticamente.
* Bloco de memória virtual se chama página.
* Uma falha se chama falha de página.
* Endereço virtual é um endereço que corresponde a um local no espaço virtual e é traduzido pelo mapeamento de endereços para um endereço físico quando a memória é acessada. Traduzido por uma combinação de hardware e software.
* Memória virtual fornece relocação.
* Endereço virtual pe formado pelo número de página virtual e um offset de página.
* O offset de página determina o tamanho da página.
* Decisões para diminuir número de falhas e penalidade de falha: As páginas devem ser grandes o suficiente ara amortizar o longo tempo de acesso. Posicionamento totalmente associativo das páginas. As falhas de página podem ser tratadas a nível de software. Softwares sofisticados de posicionamento são emrpegados já que otempo necessário para rodá-los ainda será menor que a penalidade da falha. Usará o sistema de escrita write-back.

Posicionando uma página e a encontrando novamente

* Projetistas otimizam o posicionamento das páginas.
* Tabela de páginas, é a tabela (ou tabelas) com traduções de endereço virtual para endereço físico em um sistema de memória virtual. A tabela, armazenada na memória, normalmente é indexada pelo número de página virtual; cada entrada na tabela contém o número da página física para essa página virtual se a página estiver atualmente na memória.
* Cada programa possui sua própria tabela.
* Tabela de páginas pode conter páginas não presentes na memória principal.

Faltas de página

* O sistema operacional recebe o controle quando há uma falha de página.

Detalhamento sobre tabela de páginas

* Manter as tabelas de vários programas na memória pode se tornar impraticável.
* Várias técnicas são empregadas para resolver o problema.
* 1) Registrador de limite.
* 2) Crescimento bidirecional.
* 3) Tabela de páginas invertida
* 4)Vários níveis de tabela de páginas.
* 5) Paginação das tabelas de páginas.

E quanto às escritas?

* Escritas no próximo nível da hierarquia( o disco) são muito custosas.
* Esquema write-back é utilizado.

Tornando a tradução de endereços rápida: a TLB.

* TLB: É uma cache especial que monitora os mapeamentos de endereços recentemente usados para evitar um acesso a tabela de páginas.
* As falhas de TLB podem ser tratadas tanto no hardware quanto no software.
* Ampla gama de associatividade é utilizada na TLB.

Integrando memória virtual, TLB e caches

* Cache virtualmente endereçada: O processador pode acessar uma cahce com um endereço parcial ou completamente virtual.
* Cache fisicamente endereçada, cache acessada com endereços físicos.

Implementando proteção com memória virtual

* Proteger processo uns dos outros e proteder o próprio sistema operacional.
* Para permitir a implementação da proteção o hardware precisa fornecer três capacidades básicas: Suportar os modos supervisor e kernel, fornecer uma parte do estado do processador que um processo de usuário pode ler, mas não escrever. Fornecer mecanismos pelo quais o processador pode passar do modo usuário para o modo supervisor e vice-versa.
* Coloca-se as tabelas dos programas no espaço de endereçamento do sistema.

Tratando falhas de TLB e faltas de páginas

* Falha de TLB pode indicar 2 coisas: A página não está na memória principal, e a página não está mapeada na TLB.
* Depois que o sistema reconhecer a falta de página, ele precisa realizar três coisas: 1) Consultar a tabela de páginas para descobrir a localização da pagina em disco. 2) Escolher uma página para ser substituída, obedecendo a regra de write back caso a página tenha sido modificada. 3) Iniciar uma leitura para trazer a página desejada para a memória principal.
* Tornar instruções reinicializáveis.

7.5 – Uma estrutura comum para a hierarquia de memória.

Questão 1: Onde o bloco pode ser colocado?

* Uma gama de esquemas, do diretamente mapeado ao totalmente associativo.
* Aumentar a associatividade diminui a taxa de falhas.
* Associatividade maior tem custo maior e tempo de acesso mais longo.

Questão 2: Como o bloco é encontrado?

* A escolha da associatividade depende do custo da falha em relação a implementação da associatividade.
* A escolha de associatividade total para memória virtual é motivada por: 1) É benéfica em detrimento as falhas que são muito caras. 2) Permite que softwares de alta precisão sejam implementados para evitar falhas. 3) O mapa completo pode ser facilmente indexado. 4) Maior tamanho de página significa overhead relativamente pequeno.

Questão 3: Que bloco deve ser substituído em uma falha de cache?

* Substituição aleatória. Maior associatividade. Pode ser usada na cache.
* Substituição LRU. Mais oneroso. Menor associatividade. Pode ser usada na cache. Usada na memória virtual.

Questão 4: O que acontece em uma escrita?

* Write through – Informações são escritas no nível superior E no nível inferior da hierarquia. As falhas são mais simples e baratas. É mais fácil de ser implementado, embora exija o uso de um buffer de escrita. Cache.
* Write back – Informações são escritas no nível inferior apenas quando um bloco no nível superior precisa ser escrito. Escrita em velocidade de cache. Diversas escritas num bloco exigem apenas uma escrita no nível inferior. Sistema pode fazer uso de uma transferência de alta largura de banda. Cache. Memória virtual só aceita este esquema.

Os 3 Cs

* Falhas compulsórias: Falhas causadas pelo primeiro acesso a um bloco que nunca esteve na cache. Resolvida com: Aumentando o tamanho do bloco, reduzindo a quantidade de referencias feitas para tocar cada parte do programa.
* Falhas de capacidade: São falhas de cache causadas quando a cache não pode conter todos os blocos necessários durante a execução de um programa. As falhas de capacidade ocorrem quando os blocos são substituídos e depois recuperados. Resolvida com: Aumento do tamanho da cache.
* Falhas de conflito: São falhas que ocorrem em caches associativas por conjunto ou diretamente mapeadas quando vários blocos disputam o mesmo conjunto. As falhas de conflito podem ser eliminadas com uma cache totalmente associativa. Resolvida com: Aumento de associatividade.

7.7 – Falácias e armadilhas

* Esquecer de considerar o endereço em bytes ou o tamanho de bloco de cache ao simular uma cache.
* Ignorar o comportamento do sistema de memória ao escrever programas.
* Usar tempo médio de acesso a memória para avaliar a hierarquia de memória de um processador com execução fora de ordem.