Arquitetura de Computadores 2
Trabalho 3 - Memória Virtual e E/S

Thiago Reis Porto, Mathaus C. Huber

[1]Universidade Federal de Pelotas (UFPEL) – Discentes do Curso Superior de Ciência da Computação
R. Gomes Carneiro, 1 - Centro – 96075-630 – Pelotas – RS – Brazil

trporto@inf.ufpel.edu.br, mchuber@inf.ufpel.edu.br

1. Introdução

2. Memória Virtual

MV(Memória Virtual) é uma técnica onde motivações incluem:

- Possibilidade de oferecer segurança e compartilhamento eficiente de memória para vários programas/processos em paralelo;
- Retirar o compromisso do programador na administração da alocação dos processos e suas partes na memória;
- Permitir que programas maiores do que o permitido na capacidade da memória principal, possam ser executado em um computador de arquitetura Von Neumann, através da utilização do espaço da memória segundaria(e.g. HDD ou SDD).

Utiliza a memória principal de forma semelhante a uma cache, porém com algumas termologias e detalhes distintos. Possui paginas ao equivalente dos blocos da cache e endereços virtuais. Endereço Virtual corresponde a uma localização no espaço virtual, mapeado para um local no espaço físico(endereço físico) quando a memória é acessada. Assim como as faltas de blocos em caches, se uma página não está na memória principal, ocorre uma falta e essa pagina deve ser buscada em um nível inferior, geralmente a memória segundaria.

O processador gera endereços virtuais enquanto a memória é acessada usando endereços físicos. As traduções/mapeamentos ocorrem por meio da unidade MMU (Memory Management Unit), essa unidade é um sistema HW+SW(Hardware e Software) encarregado com o gerenciamento da memória. Cada programa compilado respeita seu espaço de endereçamento, MV simplifica a carga dos programas por meio da relocação, realizada nas páginas permitindo um programa ser carregado em qualquer posição da memória principal. Um endereço virtual ou físico, possui espaço para o número da página e o seu deslocamento, o número é localizado na parte superior do endereço, e o deslocamento da página compõe a parte inferior.

2.1. Faltas de página e a tabela de páginas

A penalidade por faltas de página é muito alto, portanto em um projeto de MV é fundamental soluções que amenizem este problema. Soluções como a utilização de

alta associatividade (geralmente totalmente associativa), processo de substituições controlado por software com algoritmos inteligentes e eficientes(pode se utilizar software porque o ”tradeoff”do custo do software vs custo da penalidade por faltas de páginas, vale a pena). Utilização de uma configuração como a totalmente associativa possui o problema de encontrar as páginas armazenadas, para resolver este problema é utilizado uma estrutura chamada tabela de páginas, que possui as traduções dos endereços virtuais, que estão armazenados na memória principal ou estão na memória segundaria. Bits de controle normalmente são necessários na tabela, por exemplo, para identificar se a página esta presente na memória principal(bit de validação), se foi utilizado recentemente(bit de referência), e se deve ser escrito na memória em caso de substituição(bit sujo/dirty bit).

2.2. Escritas

Como os acessos á memória são lentos, especialmente no pior caso quando ocorre a falta de um página na memória principal, utilizar uma técnica como o WT(Write Through) é pouco factível. WT pode gerar escritas em leituras, e realiza mais acessos á memória do que um WB(Write Back), portando WBs são mais indicados para utilização em uma MV. Com WB a página é escrita na memória segundaria quando for substituída, mesmo WB sendo mais eficiente em comparação a WT, realizar uma escrita continua sendo uma operação de alto custo.

2.3. TLB

Realizar traduções de endereços virtuas requerem dois acessos à memória principal, uma para as tabelas de páginas e outro acesso para de fato buscar o dado. Para reduzir os acessos à memória principal nas traduções é indicado o uso de uma cache. Essa cache de traduções costuma ser chamada TLB(Translation-Lookaside Buffer), e armazena os endereços recentemente traduzidos evitando acessos nas tabelas de páginas. A TLB além de armazenar à tradução e bits de controle das tabelas, por serem uma memória cache necessitam de alguns bits para armazenar a tag, para identificar se uma página está presente. Tag não é necessário na tabela de páginas, pois ela possui uma entrada para cada endereço virtual.

2.4. Projeto

A seguir estão especificados as principais características de um sistema de MV, proposto neste trabalho para uma arquitetura endereçada a byte, com endereços de 32bits, memória de 4GB, Word de 32bits.

2.4.1. Espaço de endereçamento físico

Endereços de memória de 32 bits para endereçar uma memória de 4GB. $2^{32} = 4GB$ de Endereçamento Físico.

2.4.2. Espaço de endereçamento virtual

Para realizar uma ”expansão”de uma memória física de 4GB para uma memória virtual com 8GB, endereços virtuais com 33bits. $2^{33} = 8GB$ de Endereçamento Virtual.

2.4.3. Tamanho da página

Para amenizar os altos custos de tempo de acessos, às páginas devem possuir um tamanho razoável(normalmente em um intervalo de 14KB a 63 KB). Aqui escolhemos o tamanho de 16KB, um tamanho interessante para aproveitar a localidade espacial e sem tomar tantos bits do endereço(14 bits), o que não diminui tanto a quantidade de entradas da tabela.

2.4.4. Tamanho da tabela de páginas

Bits para número de páginas virtuais = 19 (33 bits do endereço virtual - 14 bits do tamanho de página).
Bits para número de páginas físicas = 18 (32 bits do endereço físico - 14 bits do tamanho de página).
$2^{19} = 524.288$ Entradas
Entradas da tabela arredonda para 32 bits, para armazenamento de bits de controle e segurança(18 destes bits são para números de páginas físicas).
$2^{19} * 32 = 2MB$ de tamanho.

2.4.5. Tamanho e organização da TLB

Como A TLB é uma cache bastante importante para o desempenho, pois pode evitar acessos a uma tabela de páginas (ou seja, acessos à memória principal). A escolha do tamanho deve considerar que deve possui um tamanho que evite faltas, mas também seja de alto desempenho, outro fator é que como a cache não possui todas entradas de uma tabela, seu número de faltas é maior e uma boa política de substituição é importante. Uma cache muito grande além do custo de desempenho na manutenção, inviabiliza a implementação de uma substituição por LRU devido aos alto custo. Também para reduzir as faltas é indicado utilizar alguma associatividade, se for escolhido utilizar totalmente associativa, também dificulta o uso de LRU devido ao custo.

Escolhemos uma cache relativamente pequena, totalmente associativa e com substituição aleatória:
Tamanho para dados = 1024 linhas * 18 bits(número de pág. física) = 1.8KB
Tamanho da linha = 19(núm. pág. virtual) + 18(núm. pág. física) + 27 bits de controle = 64 bits
Tamanho total = 64 * 1024 = 8KB

2.4.6. Diagrama

Diagrama simples considerando o sistema de MV com uma cache de 64KB de dados e 128KB total, com blocos de 32bits.

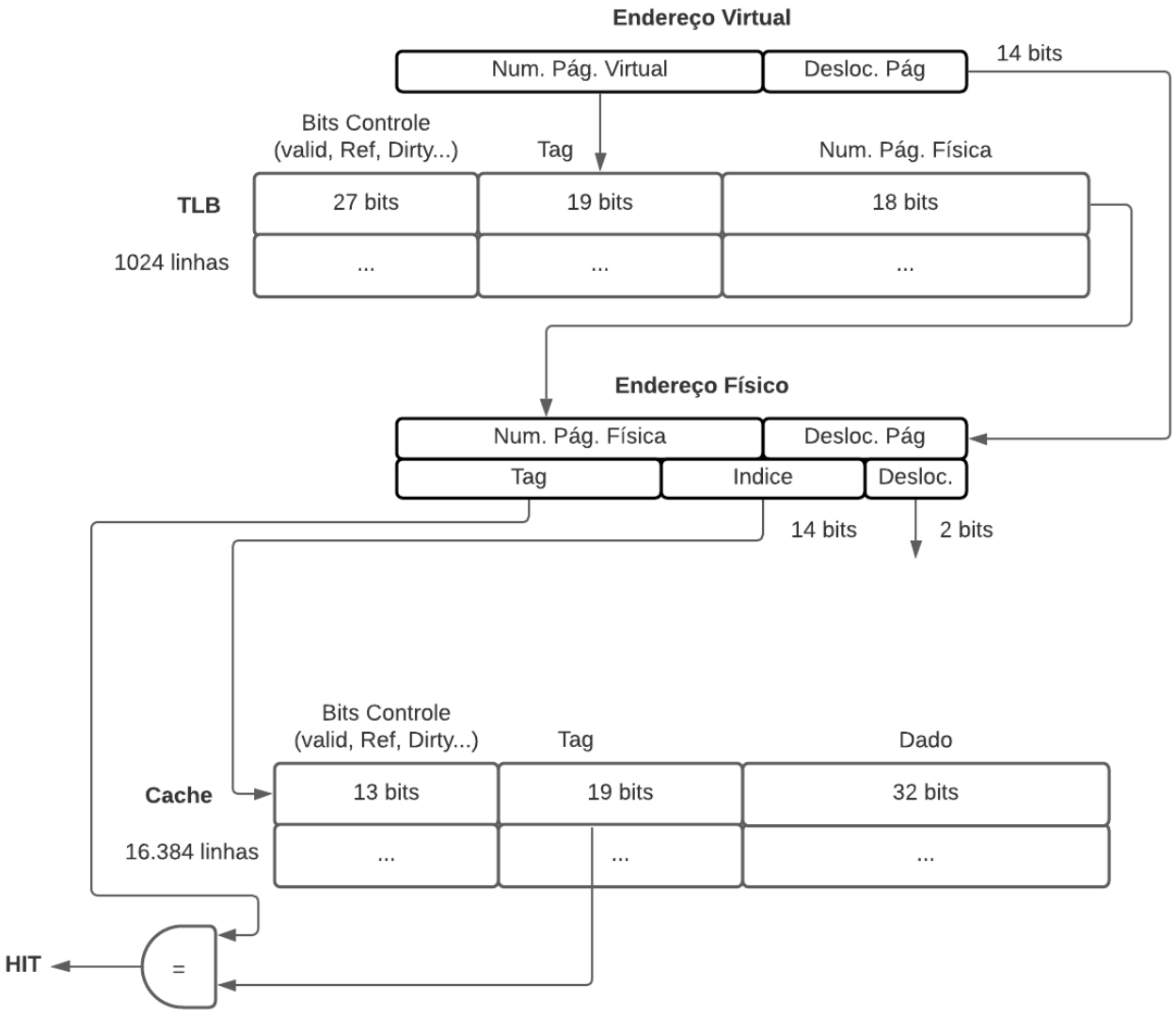

Figura 1. Diagrama do Sistema de MV

2.4.7. Fluxo de acesso de dados

No melhor caso, processador requisita uma tradução do endereço virtual que se encontra na TBL, então o endereço físico é enviado para o sistema de caches. Se for uma escrita, o dado é escrito se for permitido(controlado por um bit de escrita), se não gera uma exceção. Se for uma Leitura, neste caso o dado está na primeiro nível de cache, que então é transmitido para o processador. Então no melhor caso não é feito o acesso na memória principal. Nó pior caso, a tradução não está na TBL, e o endereço não está na memória principal.

Então um acesso deve ser feito na memória segundaria e carregar o endereço na memória principal, atualizar a tabela e TLB. E mais um acesso na memória principal para propagar o dado para o sistema de cache até o processador. Então no pior caso 1 acesso na memória segundaria, 3 acessos na memória principal(carregar endereço, atualizar tabela de páginas e carregar o dado para as caches).

3. Conceitos básicos de dispositivos de E/S

Muitas vezes durante análises de desempenho são apenas consideradas os dados do processador, porém todos os componentes do sistema (e.g. processador, hierarquia de memória, software e dispositivos de E/S) afetam o desempenho do sistema. Muitos dos sistemas atuais utilizam a mesma arquitetura em sua maior parte, se diferenciando apenas nos dispositivos de E/S. Avaliar o desempenho de dispositivos de E/S, pode ser bastante complicada.

Muitas partes dos dispositivos de E/S dependem da tecnologia, além de ser necessário considerar fatores além do desempenho, como a expansibilidade, capacidade de recuperação de erros, em alguns casos até os critérios mudam como a importância do throughput, latência de acesso ou tempo de resposta. Por isso para realizar uma análise de dispositivos de E/S, deve-se considerar as características específicas dos dispositivos e utilizar benchmarks adequados.

3.1. Tipos e características

Dispositivos de E/S podem se diferenciar por características como o Comportamento de Dispositivo(Entrada, Saida e Armazenamento), Parceiro do Dispositivo(Humano ou Máquina) e Taxa de dados(Taxa máxima ou pico de transferência de dados entre os dispositivos e componentes do sistema).
Exemplos:
Mouse: A interface entre o mouse e o sistema pode ser feito por meio de uma série de pulsos gerados pelo mouse quando de seu deslocamento pela superfície, ou pelo incremento/decremento de contadores. O processador pode ler os contadores periodicamente ou contar os pulsos para determinar se o mouse se moveu. O sistema move o cursor pela tela, para refletir a nova posição.O movimento do cursor é suave (velocidade de movimento do mouse é muito menor que a do processador). Mapeamento entre posição do mouse/estado dos botões e o sistema é feito por software.
Disco Magnético: Memória não volátil para armazenamento em massa. Mais eficientes

que discos flexíveis. Para acesso de dados é necessária uma busca(seek), que posiciona a cabeça na trilha apropriada, na trilha correta é necessário esperar o setor esperado atingir a cabeça de escrita, e aguardar o tempo de transferência dos bits.

3.2. Barramentos

É necessário um meio para conectar os diversos componentes de um sistema, sendo normalmente feito por barramentos. Barramento é um canal de comunicação compartilhado entre os componentes. Esse compartilhamento reduz custos, barramentos também são bastante versáteis quando é necessário adicionar/remover dispositivos. Barramentos podem possuir funções distintas como comunicação de dados, comunicação de endereços e comunicação de controle. São classificados principalmente em Barramento processador-memória, Barramento de E/S e Barramento do backplane. Barremento processador memória precisam serem de alto desempenho, são curtos e minimizam a banda memória-processador. Barramento de E/S são longos, geralmente possuem muitos dispositivos conectados, antedem ampla variedade de bandas, sem conexão direta com a memória, padronizados, interface simples e de baixo nível. Barramentos Backplane projetados para permitir que processador, memória e dispositivos de E/S possam coexistir em um único barramento físico, balanceiam as demandas de comunicação processador-memória com as demandas de comunicação dispositivos de E/S-memória, muitas vezes são construídos diretamente no backplane da máquina, padronizados e necessitam de uma lógica adicional para interface barramento de backplane-dispositivo.

3.3. Métodos de comunicação

O Sistema Operacional é o principal responsável pelo gerenciamento dos dispositivos de E/S, administra os acessos dos usuários, fornece rotinas de manipulação das operações de baixo nível e trata as interrupções. Os principais métodos de comunicação são o polling, interrupção, e acesso direto à memória (DMA - Direct Memory Acess).
Transferência com Polling: Os dispositivos de E/S executam a operação requisitada, sinalizam seu termino e carregam o valor apropriado no registrador de estado. O processador não é alertado sobre o término da operação, precisa verificar periodicamente o registrador estado. Alto custo e overhead, porém é necessário se um dispositivo pode iniciar operações de entrada e saída de maneira independente.
Transferência por Interrupção: Processador envia um comando de leitura para o dispositivo de E/S, prossegue para executar outras instruções, ao final de ciclos verifica instruções pendentes, se encontrar uma interrupção salva o contexto e a executa. Os dispositivos recebem o comando de leitura e recebem o dado dos periféricos, quando o dado estiver pronto no seu registrador requisita uma interrupção. Ganho de desempenho em relação ao polling, por evitar o overhead de fazer o processador esperar o resultado do dispositivo mesmo quando não está pronto para transferência.
DMA: Utiliza um hardware especial, e libera o processador de acompanhar as etapas de transferências. Suas desvantagens incluem o overhead da controladora de DMA e o custo do hardware extra.

3.4. Comunicação eficiente para E/S

3.4.1. Mouse

O metódo de comunicação mais eficiente para o mouse seria o Polling, pois é um dispositivo que pode iniciar operações de entrada e saída de maneira independente

Calculo de desempenho:
Considerando o caso do mouse ser verificado 30 vezes/seg.

ciclos = 30 x ciclospooling= 30 x 400 = 12000 ciclos
FraçãoCPU = 12000x102/500x106 =0.002%

3.4.2. Disco rígido

O método de comunicação mais eficiente para o disco rígido seria o Acesso Direto à Memória (DMA), pois não exige que o processador esteja envolvido em todos os ciclos de clock durante a transferência.

Calculo de desempenho:
Considerando o caso do disco rígido transferir blocos de 4 palavras a uma taxa de 4MB/-seg.

Blocos = 8KB
Transmissão em 100% do tempo
Setup = 1000 ciclos
Overhead(interrupção) = 500 ciclos
TempoDMA= 8KB/(4MB/seg)= 2x10-3 seg
ciclos/seg= (1000+500)/ 2x10-3 = 750x103
FraçãoCPU = 750x103 x102/500x106 = 0,2%

4. Conclusão

Antes dos sistemas operacionais serem multi-tarefas os computadores tipicamente tinham apenas um ou dois processos em execução ao mesmo tempo. O Sistema MS -DOS da Microsoft é um exemplo perfeito porque o único processo funcionando era o programa que interpretava os comandos do usuário. Com a chegada do método de memória virtual os arquivos da MV ocuparam uma quantidade maior da RAM, mas em compensação permitiu mais programas em execução ao mesmo tempo. Entre as vantagens de se utilizar essa técnica, juntamente com a razão porque foi inventado, é que ele permite mais dados para permanecer em uso ao mesmo tempo do que a memória física é capaz de realizar. Contudo, o computador pode ser mais lento quando a memória virtual está em uso, pois os dados são armazenados no disco rígido em vez de na memória física, então o tempo de demora para alcançar os dados é mais longo.

No que se trata de dispositivos de entrada e saída, essa função de conexão foi desenvolvida basicamente para que seja possível a comunicação entre vários dispositivos.

Essa interface é responsável por conectar fisicamente o processador e a memória do sistema ao barramento. Em outras palavras ”Entrada”indica a inserção de dados por meio de um código ou programa, para algum hardware, e ”Saída”indica o retorno dos dados como resultado de alguma operação, ou seja, o resultado de alguma entrada.