Sistemas operacionais: Conceitos e Mecanismos

Capítulo 14: Hardware de memória

Integrantes do grupo: Jefferson Botitano e Leonardo Ludvig
Professor: Arliones Stevert Hoeller Junior
Disciplina: Sistemas operacionais

INSTITUTO FEDERAL
Santa Catarina

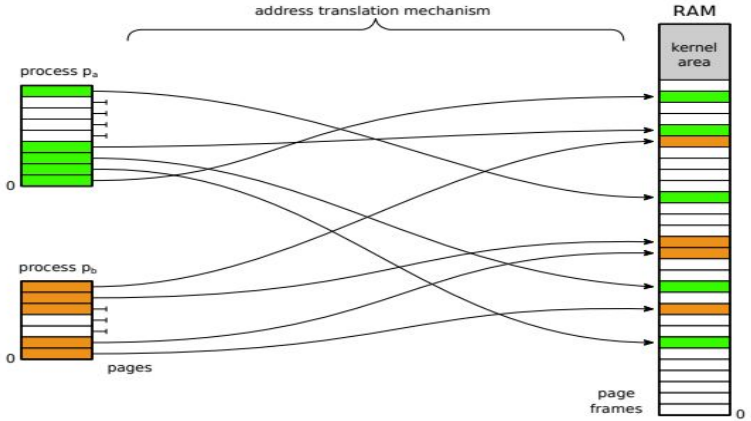
Campus São José

Memória virtual por páginas

As formas de organização de memória apresentadas anteriormente não são muito utilizadas devido a sua complexidade de implementação/compilação e alta suscetibilidade a fragmentação externa, por isso nos sistemas operacionais atuais é utilizado o método de memória paginada, onde o espaço de endereçamento lógico dos processos é mantido linear e unidimensional. Nesse método tanto a memória lógica (Paginas) quanto a física (Quadros) são divididas em N partes geralmente de 4.096 bytes, podendo variar de acordo com a arquitetura abordada, a tradução feita pela MMU utiliza de tabelas de páginas, nas quais cada entrada corresponde a uma página do processo e contém o número do quadro onde ela se encontra.

A tabela de páginas

A figura a baixo representa a tradução realizada pela MMU onde cada processo possui sua própria tabela de páginas, que corresponde ao processo em execução no momento, a qual é referenciada pelo registrador PTBR (Page Table Base Register) esse registrador é atualizado com o endereço da tabela do novo processo em execução a cada troca de contexto. A divisão do espaço de endereçamento lógico de um processo em páginas é feita utilizando os bits menos significativos não utilizados pela página que possui 2^12 de tamanho, então por exemplo em um processador 32 bits os 20 bits restantes seriam utilizados para definir a posição e a quantidade de páginas. Caso o processo tente acessar uma página em branco (sem dados nos bits de localização) é enviado uma mensagem de aviso page fault e gera uma interrupção para abortar o processo ou tomar outra medida



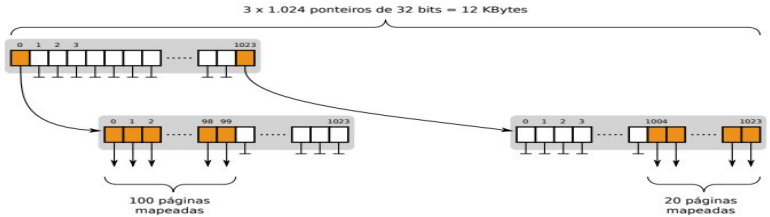
Flags de status e controle

Cada entrada de uma tabela de páginas contém um conjunto de flags de status ou de controle relativos a página,os mais usuais são:

- **Valid:**Sinaliza se a página é válida em bit 1,ou seja, existe espaço de endereçamento daquele processo.Caso bit estiver em 0 uma interrupção a acessos a página será gerada.
- **Writable:**Sinaliza se a página pode ser acessada em leitura/escrita caso estiver 1 senão somente leitura 0.
- **User:**Caso estiver em 1 sinaliza ativo,sendo que o código está em andamento em modo usuário podendo acessar a página caso contrário estiver em 0 sinaliza que a página está apenas acessível a o núcleo do sistema.
- **Present:**Indica se a página está na memória RAM ou foi transferida para um armazenamento secundário sendo o caso de sistemas com paginação em disco.
- **Accessed:**indica se a página foi acessada recentemente e o responsável por ativar esta flag é a MMU por cada acesso a página e podendo ser desativado pelo núcleo quando necessário.Este tipo de informação é usada por algoritmos de paginação em disco.
- **Dirty:**esta flag é ativada pela MMU depois de realizar uma escrita na página para relatar que foi modificada.Este tipo de informação é usada por algoritmos de paginação em disco.

Tabelas multiníveis

O uso da tabela de páginas por meio dos bits restantes não utilizados para definir o tamanho da página como os 20 bits restantes no caso do processador de 32 bits pode acabar ocupando um espaço muito grande desnecessariamente na memória virtual, pois um processo pequeno que utilize poucas páginas irá utilizar a mesma tabela linear de 20 bits e irá ocupar 4 Mbytes na memória mesmo que utilize apenas algumas páginas e todas as restantes estejam em branco. A solução para esse problema foram as tabelas multiníveis estruturadas em forma de árvores, onde a tabela é dividida no mínimo em duas partes que são usados como índices em cada nível de tabela, até encontrar o número de quadro desejado. Esse método reduz significativamente a quantidade de memória ocupada pela tabela de páginas.



Cache da tabela de páginas

O método de tabelas multiníveis é uma solução eficiente porém com um grave efeito colateral aumentando significativamente o tempo de acesso a memória, isso ocorre pois o processador tem que solicitar um acesso para percorrer cada nível da memória, assim num sistema com tabelas de dois níveis cada acesso à memória solicitado pelo processador implica em mais dois acessos, para percorrer os dois níveis de tabelas, o que gera um tempo de acesso 3 vezes maior. A solução é o uso do TLB (Translation Lookaside Buffer), que salva os últimos pares [página, quadro] obtidos em consultas recentes às tabelas de páginas do processo ativo e zerando esse cache a cada troca de contexto, a consulta ao TLB é realizada antes da consulta a tabela de páginas e caso não seja encontrada a página é feito a busca na tabela e salvo o par no TLB, a eficiência do TLB depende dos seguintes fatores.

- **Tamanho do TLB:** Quanto maior o tamanho do cache maior sua eficiência pois ele irá armazenar mais pares aumentando sua taxa de acerto, porém quanto maior o tamanho maior o valor e volume do hardware.
- **Padrão de acessos à memória:** Caso um processador não possua um padrão ou constância de acesso às mesmas páginas irá reduzir a eficiência do TLB, pois irá gerar muitos erros de TLB.
- **Trocas de contexto:** Como trocas de contexto zeram o cache para receber o novo processo, elas devem ser usadas com cautela.
- **Política de substituição de entradas**: Deve ser implementada uma política eficiente para resolver casos de buffer cheio.

Segmentos e páginas

Como vários processadores permitem a combinação de mais de uma forma de organização, existe a combinação de organização segmentos com a organização por páginas que resulta na flexibilidade oferecida pela organização dos segmentos com a baixa fragmentação oferecida pela organização das páginas.

Mesmo com processadores oferecendo as duas formas de organização de memória a maioria dos sistemas operacionais não utilizam todas as combinações possíveis como é caso da família UNIX(Linux,FreeBSD) que utilizam somente organização por páginas.

Localidade de referências

A propriedade de um processo ou sistema concentrar seus acesso em poucas áreas da memória a cada instante é denominada localidade de referências e existem ao menos três formas de locidade de referências:

- **Localidade temporal:**Recurso que foi utilizado recentemente e em um futuro próximo será utilizado novamente.
- **Localidade espacial:**Recurso será acessado provavelmente caso outro recurso próximo a ele tenha sido acessado.
- **Localidade sequencial:**Caso particular onde após o acesso a um recurso em uma posição p,existe maior chance de acessar um recurso na posição p+1.Um exemplo deste tipo de localidade é útil na otimização de sistemas de arquivo.

A localidade de referência de uma implementação depende de um conjunto de fatores,que incluem:

- **As estruturas de dados usadas pelo programa:**Estruturas do tipo vetor e matrizes tem seus elementos alocados de forma contígua na memória resultando a uma localidade de referências maior que estruturas mais dispersas como estruturas do tipo lista encadeadas e árvores.
- **Os algoritmos usados pelo programa:**A forma que o programa acessa a memória é definido pelo algoritmo implementado.
- **A qualidade do compilador:**Sendo o compilador responsável por analisar quais variáveis e trechos dos códigos são usados com frequência juntos e colocando os organizando nas mesmas páginas de memória, com o intuito de aumentar a localidade de referências do codigo gerado e podendo também alinhar estruturas de dados mais utilizadas em relação às páginas.

2)Explique as principais formas de alocação de memória.

INSTITUTO FEDERAL
Santa Catarina

Partições: Uma das formas mais simples de organização da memória e tradução de endereços lógicos em físicos consiste em dividir a memória física em N partições, que podem ter tamanhos iguais ou distintos, fixos ou variáveis. Em cada partição da memória física é carregado um processo. Suas vantagens são simplicidade e rapidez. E as desvantagens são que os processos podem ter tamanhos distintos das partições, os número de processos ≤ número de partições, e dificuldade para compartilhar memória.

Segmentos: O espaço de endereçamento de cada processo não é mais visto como uma sequência linear de endereços lógicos, mas como uma coleção de áreas de tamanhos diversos e políticas de acesso distintas, denominadas segmentos. Cada segmento se comporta como uma partição de memória independente, com seus próprios endereços lógicos.

Páginas: o espaço de endereçamento lógico dos processos é mantido linear e unidimensional. Internamente, de forma transparente para o processador, o espaço de endereçamento lógico é dividido em pequenos blocos de mesmo tamanho, denominados páginas, assim como a memória física é dividida em blocos do mesmo tamanho, denominados de quadros..

3)Por que os tamanhos de páginas e quadros são sempre potências de 2?

INSTITUTO FEDERAL
Santa Catarina

A memória é dividida em blocos de tamanho fixo chamados de frames-molduras de páginas e as páginas tem q ser construídas de forma a se encaixar melhor nessas molduras. Como a memória física é dividida blocos de tamanho fixo através dos bits disponíveis sempre em tamanhos de potência de 2 entre 512 a 8192 bytes. Por isso a memória lógica também deve ser dividida em blocos de mesmo tamanho que a física.

4)Considerando a tabela de páginas a seguir, com páginas de 500 bytes, informe os endereços físicos correspondentes aos endereços lógicos 414, 741, 1.995, 4.000 e 6.633, indicados em decimal. Obs.: Um tamanho de página de 500 bytes permite fazer os cálculos mentalmente, sem a necessidade de converter os endereços para binário e vice-versa, bastando usar divisões inteiras (com resto) entre os endereços e o tamanho de página.

página	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
quadro	3	12	6	–	9	–	2	–	0	5	–	–	–	7	–	1

414: página 0, endereço 1941.

741: página 1, endereço 6241.

1995: página 3, falha de paginação.

4000: página 7, falha de paginação.

6633: página 13, endereço 10133.

5)Explique o que é TLB, qual a sua finalidade e como é seu funcionamento.

INSTITUTO FEDERAL
Santa Catarina

Em um sistema com tabelas de dois níveis, cada acesso à memória solicitado pelo processador implica em mais dois acessos, para percorrer os dois níveis de tabelas. Com isso, o tempo efetivo de acesso à memória se torna três vezes maior. Para atenuar esse problema, consultas recentes à tabela de páginas podem ser armazenadas em um cache dentro da própria MMU, evitando ter de repeti-las e assim diminuindo o tempo de acesso à memória RAM.

O cache de tabela de páginas na MMU, denominado TLB (Translation Lookaside Buffer) ou cache associativo, armazena pares [página, quadro] obtidos em consultas recentes às tabelas de páginas do processo ativo. Esse cache funciona como uma tabela de hash: dado um número de página p em sua entrada, ele apresenta em sua saída o número de quadro q correspondente, ou um erro, caso não contenha informação sobre p.

6)Um sistema de memória virtual paginada possui tabelas de página com três níveis e tempo de acesso à memória RAM de 100 ns. O sistema usa um cache TLB de 64 entradas, com taxa estimada de acerto de 98%, custo de acerto de 10 ns e penalidade de erro de 50 ns. Qual o tempo médio estimado de acesso à memória pelo processador? Apresente e explique seu raciocínio.

INSTITUTO FEDERAL
Santa Catarina

Tmedio=98%*10ns #custo do acerto no TLB

+2%*50ns #custo do erro no TLB

+2%*3*100ns #custo da consulta às 3 tabelas

+100ns #custo do acesso ao quadro

Tmedio= 116,6ns