



Sistemas Operacionais

CCP/SIF

UNISUL – Tubarão

Cassio Brodbeck Caporal

`cassio{NOSPAM}ostec.com.br`

Agenda

- Revisão;
- Gerência de Memória Virtual;
- Espaço de endereçamento;
- Mapeamento;
- Memória virtual por paginação;
- Políticas de busca de páginas;

Agenda

- Políticas de alocação de páginas;
- Políticas de substituição de páginas;
- Conjunto de trabalho (*working set*);
- Algoritmos de substituição;
- *Translation Lookaside Buffer*;
- Proteção e compartilhamento de memória;

Agenda

- Memória virtual por segmentação;
- Segmentação com paginação;
- *Swapping*;

Gerência da Memória Virtual

- Conceito:
 - Técnica que permite combinar memória principal e secundária, aumentando a capacidade de armazenamento visível para o usuário.
 - Endereços de programas não vinculam diretamente a memória principal;
 - Maior número de programas em memória: somente parte deles são residentes na memória;

Espaço de Endereçamento (virtual)

- Analogia a um vetor (*array*):
 - Quando uma posição é acessada, não é necessário saber a posição de memória daquele elemento;
- Mapeamento:
 - Técnica que permite traduzir endereços virtuais em endereços físicos;
 - EEV e EER;

Espaço de Endereçamento (virtual)

- Utilização da memória secundária como extensão da principal;
- Somente parte de códigos de programas ficam na memória principal:
 - O restante fica na memória secundária e somente é carregado quando referenciado;
 - Aumento do compartilhamento da memória principal entre processos.

Mapeamento

- Novamente:
 - A CPU executa instruções utilizando o *espaço de endereçamento **real***;
 - O mapeamento permite a tradução de endereços do EEV para EER;
- O hardware ajuda no mapeamento através da *Memory Management Unit* (MMU);

Memória Virtual por Paginação

- Espaço de endereçamento virtual e real são divididos em blocos de tamanhos iguais:
 - Páginas virtuais e página reais;
- Mapeamento realizado através de *tabelas de páginas*:
 - Cada processo possui sua tabela de páginas e cada página virtual uma entrada nesta tabela;

Políticas de Busca de Páginas

- Memória virtual permite que um programa execute sem que o código esteja por completo na memória principal;
- As políticas de busca determinam ***quando*** uma página deve ser carregada para MP;
- Paginação por demanda e paginação antecipada;

Políticas de Busca de Páginas

- Paginação por demanda (*demand paging*):
 - A transferência ocorre somente quando a página é efetivamente referenciada;
 - Permite um uso mais racional da memória, evitando que partes desnecessárias (que podem nunca ocorrer) ocupem espaço;

Políticas de Busca de Páginas

- Paginação antecipada (*prepaging*):
 - Transferência da página referenciada e outras páginas que podem **OU NÃO** serem utilizadas ao longo da execução do programa;
 - É bom e ruim! :-)
 - Se o processo precisar, economiza-se tempo na transferência de memória secundária para principal;
 - Se não precisar, perde-se tempo e espaço;

Políticas de Alocação de Páginas

- Objetivo de determinar quantos *frames* cada processo pode ter na MP;
- Alocação fixa:
 - Possui um número máximo de *frames* que podem ser utilizados durante a execução;
 - Pode ser definido de forma igual (mesmo número) para todos os processos ou individualmente;

Políticas de Alocação de Páginas

- Alocação fixa:
 - Substituição de páginas quando houver esgotamento;
 - Limites iguais para processos não é interessante, pois os mesmos tem necessidades diferentes;
 - Os limites são definidos na criação do processo e ficam armazenados no contexto de software;

Políticas de Alocação de Páginas

- Problemas com alocação fixa:
 - Se o número for pequeno, tendência de ter número alto de *page faults*: afeta diretamente o desempenho;
 - Se o número for grande, áreas inutilizadas diminuindo o número de processos residentes e conseqüentemente a multiprogramação.

Políticas de Alocação de Páginas

- Alocação variável:
 - O número de páginas alocadas pode variar durante a execução, com base em métricas de *taxa de paginação* e ocupação da MP;
 - Processos com alta taxa de paginação aumentam o limite diminuindo *page faults*;
 - Processos com baixa taxa, podem ter páginas realocadas para outros processos;

Políticas de Alocação de Páginas

- Problema com alocação variável:
 - O de sempre, com tudo que parece muito bom, grande *overhead*;
 - Exige que o sistema operacional monitore com regularidade taxa de paginação e outros inerentes ao gerenciamento de memória.

Políticas de Substituição de Páginas

- O que fazer quando um processo atinge o limite de alocação de *frames*?
 - Não confunda com *alocação de páginas*!;
 - Liberar uma página para permitir a realocação;
- Antes de substituir, o sistema operacional tem que verificar se a página foi modificada:

Políticas de Substituição de Páginas

- Antes de substituir, o sistema operacional tem que verificar se página foi modificada:
 - O sistema deve gravar (*page file*) em MS (*page out*);
 - Utilização de um *bit de modificação*;
- Política de substituição local;
- Política de substituição global;

Políticas de Substituição de Páginas

- Política de substituição local;
 - Somente páginas do processo que gerou o *page fault* podem ser realocadas;
- Política de substituição global:
 - Todas as páginas alocadas na MP podem ser utilizadas para substituição, independente de quem gerou o *page fault*;

Conjunto de Trabalho

- Propósito de resolver dois problemas:
 - Um processo não deve manter **poucas páginas** em memória pois isso demandaria diversos *page faults*;
 - Um processo não pode manter **muitas páginas** em memória pois isso diminuiria a utilização da MP por outros processos (multiprogramação);
- *Trashing!*

Conjunto de Trabalho

- O *working set* tem como objetivo reduzir o *trashing*;
 - Princípio da localidade:
 - Localidade espacial: tendência de após uma referência a um determinado endereço de memória, outros endereços próximos também sejam referenciados;
 - Localidade temporal: tendência de ter sucessivas referências a um mesmo endereço em pouco tempo;

Conjunto de Trabalho

- Peter Denning desenvolveu o modelo que posteriormente deve diversas melhorias, desde 1968 até 1980;
- Utiliza o conceito de *janela de páginas*:
 - Permite prever e usar eficientemente o espaço de memória, em um determinado tempo, para um programa em execução.

Algoritmos de substituição

- O grande problema dos algoritmos:
 - Definir quais páginas liberar, não quais carregar (essa informação é conhecida);
 - Precisam selecionar os *frames* que possuem menor chance de serem usados em pouco tempo;
 - O *princípio da localidade* é fundamental para os algoritmos;

Algoritmos de substituição

- Vários sabores de algoritmos, cada um com características específicas:
 - Aleatório;
 - FIFO (com ou sem *buffer* de páginas, circular);
 - LFU (*Least Frequently Used*);
 - LRU (*Least Recently Used*);
 - NRU (*Not Recently Used*);

Tamanho de página

- Relacionado a arquitetura de hardware:
 - Algumas permitem a especificação deste valor;
- Normalmente entre 512 bytes e 16 Kb;
- Determinar o **número ótimo** é uma tarefa muito complexa, pois depende de fatores conflitantes;

Translation Lookaside Buffer

- Mapeamento implica em dois acessos a MP:
 - O primeiro à tabela de páginas e o outro a própria página a ser acessada;
- Com base no princípio da localidade, somente um pequeno tamanho da tabela de mapeamento é realmente necessária:
 - TLB permite mapear endereços virtuais e físicos sem acessar a tabela de páginas.

Translation Lookaside Buffer

- Funcionamento semelhante a uma memória cache:
 - Mantém traduções somente dos endereços virtuais das páginas mais recentemente acessadas/utilizadas;
 - Mapeamento associativo;
 - Sem necessidade de analisar **todo** o TLB;

Translation Lookaside Buffer

- *Timeline* para traduzir um endereço virtual:
 - (1) Verifica o TLB, se o endereço virtual estiver no *cache*, o endereço físico associado é utilizado e a tabela de mapeamento não é vista;
 - (2) Se o endereço não estiver, consulta a tabela de mapeamento;
 - (3) Se a página estiver na MP, a tradução é colocada na TLB e o endereço traduzido;

Translation Lookaside Buffer

- *Timeline* para traduzir um endereço virtual;
 - (4) Se a página não estiver na MP, um *page fault* é gerado, a tabela de mapeamento é atualizada e a informação é enviada para o TLB;
- O TLB permite diminuir o acesso a MP consideravelmente, garantindo menor *overhead* no gerenciamento de memória virtual.

Proteção de Memória

- Objetivo de impedir acessos indevidos a páginas de memória relacionadas ao SO e processos de usuários;
 - Caso contrário, o sistema operacional como um todo poderia ser comprometido;
- Cada processo tem sua própria tabela de mapeamento, e a tradução é feita pelo SO;

Proteção de Memória

- Cada página da memória principal é protegida individualmente através da tabela de mapeamento;
 - Existência de *bits* de proteção (sem acesso, leitura e leitura/gravação);
- Quando uma página é referenciada, estas informações são lidas e validadas.

Memória Virtual por Segmentação

- Ao contrário da paginação, os blocos são de tamanhos **diferentes**;
- Relação direta entre a estrutura do programa e sua alocação na memória:
 - Exemplo: sub-rotinas e estrutura de dados são divididos *logicamente*;
 - Segmentos geralmente são definidos pelo compilador;

Segmentação com Paginação

- Memória dividida em segmentos onde cada segmento é dividido em páginas;
- Objetivo de oferecer em conjunto os benefícios da técnica de segmentação e paginação;
- Endereço virtual formado por NSV, NPV e um *deslocamento (offset)*;