

Uma comparação entre algoritmos de substituição de página

Bruno C. P. D. Rosa

¹Departamento de informática
Centro de Tecnologia – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

Resumo. *A técnica de memória virtual abstrai a memória física de modo a poder estendê-la ao ponto de vista do usuário, isto permite que programas que ocupam mais espaço que o espaço físico disponível sejam executados sem problemas. Sendo assim muito importante para um Sistema Operacional, neste trabalho é realizada uma breve introdução ao tópico. Um simulador foi desenvolvido e com ele dois algoritmos de substituição de páginas são implementados. Os experimentos variam em parâmetros e algoritmo utilizados com o intuito de compará-los. O algoritmo Não Recentemente Usadas apresentou o melhor desempenho.*

1. Introdução

O método de memória virtual é um componente chave ao S.O. pois permite que programas sejam carregados parte em memória principal e parte em memória secundária através da paginação. A ideia é permitir que processos ocupem mais espaço que a quantidade de memória física disponível, sendo que neste caso os endereços acessados são virtuais os quais são traduzidos para endereços físicos. Especialmente em computadores mais antigos era muito comum a falta de espaço na memória principal, esta técnica entretanto é essencial aos S.O. atuais também pois trás ganhos ao desempenho do sistema, especialmente durante a execução de programas pesados. Neste contexto o algoritmo de substituição de página pode ter um grande impacto no desempenho deste componente do sistema. Assim, neste trabalho para disciplina de Sistemas Operacionais um simulador de memória virtual foi desenvolvido contendo duas implementações de algoritmos de substituição de página.

2. Referencial teórico

O material utilizado no desenvolvimento do simulador e na implementação dos algoritmos escolhidos foram unicamente as aulas da disciplina de Sistemas Operacionais e informações obtidas em aula. O espaço de endereçamento é dividido em páginas, deste modo as transferências entre a memória principal e secundária sempre ocorrem em páginas. A técnica adotada no desenvolvimento deste trabalho foi a tabela de páginas simples de apenas um nível, a qual fica carregada em memória o tempo todo. É importante lembrar e diferenciar molduras de páginas (frames) de páginas. Os frames são representações de espaço físico na memória, enquanto que as páginas são componentes virtuais da implementação. A substituição de página ocorre quando todos os frames estão "ocupados" e uma página precisa ser carregada em memória, nesta situação uma das páginas carregadas deverá ser descartada para que seu frame seja liberado para a nova página. Os algoritmos de substituição de páginas escolhidos foram: Não Recentemente Usado (NRU) e Primeiro a entrar, primeiro a sair (First in, first out - FIFO).

O algoritmo NRU utiliza dois bits adicionais, os quais são "acoplados" a tabela de páginas, o primeiro bit é o bit de referenciamento, seu valor é 1 sempre que a página em questão foi referenciada por uma leitura, sendo que, após algum intervalo de tempo, podendo ser ciclos de clock, ele volta a ter valor 0. O segundo bit é o bit de modificado, este tem valor igual a 1 quando uma escrita foi realizada naquela página e só volta a ter valor 0 quando ela for salva em memória secundária no caso de substituição. Em alguns hardwares estes bits são armazenados em registradores para este propósito, falta deste componente em hardware o S.O. se responsabiliza pelo seu gerenciamento. Neste algoritmo ambos bits são usados para controlar quais páginas são descartadas na ocorrência de necessidade de substituição de página, com dois bits formam-se quatro estados possíveis (00, 01, 10, 11) a quais chamamos de classes, a página a ser substituída é sempre a de menor estado.

O algoritmo FIFO é mais simples, sua lógica consiste na utilização de uma fila para as entradas das páginas e sempre que uma substituição de página ocorrer a primeira página da fila será escolhida para ser descartada. Sua simplicidade pode ser considerada sua melhor qualidade visto que na prática sua falta de sofisticação representa uma ausência de mecanismos para proteger páginas que estão sendo constantemente acessadas e obviamente não deveriam ser removidas pois isto só causa mais atraso no sistema.

3. Desenvolvimento

O simulador foi desenvolvido na linguagem C, principalmente pela liberdade de gerenciamento de memória que a linguagem proporciona. Para obtenção e tratamento da parametrização do programa foi utilizada a biblioteca getopt, além disso bibliotecas comuns como stdio, stdlib foram utilizadas para propósitos gerais. Como a entrada para a simulação eram arquivos de textos crus foi utilizada a biblioteca string para tratamento do texto e tradução em dados inteiros dentro do programa. Os parâmetros do programa são:

- Número de moldura de páginas: Sendo obrigatoriamente maior que um, este parâmetro afeta diretamente nos resultados por influenciar diretamente a necessidade de substituição de páginas.
- Tamanho da moldura: Este parâmetro controla o tamanho de cada página em KBs, quanto maior o tamanho menor será o número de páginas virtuais necessárias para mapear toda a memória.
- Algoritmo: Por fim o parâmetro do algoritmo seleciona qual algoritmo de substituição de página será utilizado durante a execução do programa.

4. Experimentos e Resultados

Os experimentos realizados compreenderam a variação dos parâmetros citados anteriormente com o propósito de investigar a influencia destes na execução, e quanto ao parâmetro de algoritmo este é especialmente importante pois avaliamos o desempenho de cada um. A tabela abaixo apresenta os resultados baseado no número de substituição de páginas necessárias (Page Faults). A entrada para a simulação foram instruções de leitura e escrita, a primeira entrada possui 1000 instruções, a segunda 10000 e a terceira 81976.

Algoritmo	Numero de frames	Tamanho da pagina	Entrada	Page Faults
NRU	4	4KB	81976	63
NRU	8	4KB	81976	59
NRU	8	8KB	81976	36
FIFO	4	4KB	81976	81
FIFO	8	4KB	81976	64
FIFO	8	8KB	81976	36

Claramente o algoritmo NRU apresenta um melhor desempenho, o que já era esperado devida a sua lógica mais elaborada que classifica as páginas por utilização permitindo que páginas frequentemente acessadas se mantenham em memória por mais tempo, evitando transferências desnecessárias.

5. Conclusão

O mecanismo de memória virtual permite que a memória possa ser abstraída e até expandida de certa forma do ponto de vista do usuário. Sua implementação e o algoritmo de substituição de página utilizado podem afetar diretamente o desempenho do sistema, sendo assim um aspecto muito importante no desenvolvimento de um Sistema Operacional.