

UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL-REI
CIÊNCIA DA COMPUTAÇÃO



Sistemas Operacionais

Documentação primeiro trabalho prático

Matheus Alexandre
Luccas Guidio

Trabalho prático apresentado à disciplina
de sistemas operacionais do curso de
Ciência da Computação pela Universidade
Federal de São João del-Rei.
Professor Dr. Rafael Sachetto.

São João del-Rei
Novembro de 2019

Sumário

| | | |
|----------|----------------------------------|----------|
| 1 | Introdução | 2 |
| 2 | Implementação | 3 |
| 2.1 | Arquivos de Entrada | 3 |
| 2.2 | Validade | 3 |
| 2.3 | Paginação | 3 |
| 3 | Políticas de Substituição | 3 |
| 3.1 | Segunda Chance | 4 |
| 3.2 | NRU | 4 |
| 3.3 | LRU | 4 |
| 4 | Análise de Tabelas | 6 |
| 5 | Bibliografia | 8 |

1 Introdução

O segundo trabalho prático realizado na disciplina de Sistemas Operacionais consiste em simular uma memória virtual. Para funcionar, o simulador recebe uma sequência de parâmetros em sua entrada para reconhecer os endereços de memória acessados por um programa, o tamanho de cada página na memória e também o espaço total que pode ser utilizado. O intuito da atividade é avaliar o desempenho de diferentes configurações da memória virtual em uma série de processos, alternando entre o tamanho de página, memória principal e as políticas de substituição.

2 Implementação

O trabalho foi implementado na linguagem C e deverá ser inicializado da seguinte maneira:

- Uma linha de comando com 5 parâmetros na seguinte ordem: nome do executável, a política de substituição a ser seguida, o arquivo que contém o acesso de memória, o tamanho da página e o tamanho da memória.
ex.: `./tp2virtual lru matriz.log 4 128`

2.1 Arquivos de Entrada

Os arquivos `.log` a serem executados deverão possuir uma série de endereços virtuais em hexadecimal seguidos da forma de acesso. Cada arquivo simula a sequência de acesso à memória executada por um dado processo e, a princípio, não existem tamanhos máximos para a quantidade de tarefas.

2.2 Validade

Assim que o programa inicia, é averiguada a validade dos argumentos; se o arquivo existe, se a quantidade de argumentos está correta, se o tamanho de página e/ou memória não excede o limite definido nas especificações do trabalho. Caso haja algum parâmetro incorreto, o programa é encerrado imediatamente.

2.3 Paginação

A tabela de páginas foi dividida em frames(quadros) organizados por uma Hash. Para decidir o quadro (frame na hash) que será dedicado à uma pagina, é utilizado o endereço virtual desta, a partir dele, são selecionados os bits mais significativos através da função disponibilizada pelo professor na especificação do trabalho prático. Sabendo quais são os bits mais significativos, é realizada a função de tabela Hash para decidir onde cada página será alocada. Caso a hash esteja cheia é chamada a função de substituição a partir da política dada como parâmetro

3 Políticas de Substituição

Após a verificação dos argumentos é iniciada a simulação em si. São criadas variáveis auxiliares para que todas as políticas de substituições sejam realizadas com sucesso. Uma política de substituição será chamada se, e somente se, a memória estiver cheia e ainda for preciso escrever alguma informação nela.

3.1 Segunda Chance

A primeira política a ser discutida é a de segunda chance. Esta consiste em escolher qual página será substituída na memória a partir de uma fila e de atributos auxiliares para saber quando foi a última vez que cada página foi referenciada. Caso a primeira página alocada na memória tenha sido referenciada recentemente, ela é alterada para o final da fila e seu bit de referência passa a ser 0, ou seja, assume que ela não foi mais referenciada (uma vez que não é possível acessar a memória antes de terminar a substituição). Assim que a página que se encontra na primeira posição da fila possuir o bit de referência igual a zero (0), essa página será substituída pela nova página a ser alocada. Sucintamente, a política de segunda chance assemelha-se à política FIFO (First In First Out) porém leva em conta se a página candidata a ser substituída foi referenciada recentemente ou não.

3.2 NRU

A política de substituição NRU utiliza classes para decidir qual página será substituída. Cada página alocada na memória possui dois atributos auxiliares que informam se a página foi alterada e não recolocada no disco (dirty bit) e se foi referenciada. As classes são definidas da seguinte maneira:

- Classe 1: não referenciada ($\text{ref} = 0$) e não modificada ($\text{mod} = 0$)
- Classe 2: não referenciada ($\text{ref} = 0$) e modificada ($\text{mod} = 1$)
- Classe 3: referenciada ($\text{ref} = 1$) e não modificada ($\text{mod} = 0$)
- Classe 4: referenciada ($\text{ref} = 1$) e modificada ($\text{mod} = 1$)

O algoritmo verifica se existe alguma página a ser substituída começando pela classe 0. Caso haja mais de um candidato, é escolhido aleatoriamente. Porém, se não houver alguma página que possa ser trocada, a busca passa a ser na classe subsequente. E assim sucessivamente até que se ache uma página. Desta forma, no pior caso, teremos que procurar pelo resultado um total de quatro vezes dentro da tabela de página. Vale ressaltar, também, que o bit de referência é zerado periodicamente com o tempo escolhido de forma arbitrária (no caso desta implementação equivale à 20 operações).

3.3 LRU

A política de substituição LRU substitui páginas utilizando o tempo de referência como parâmetro. Pelo princípio da localidade temporal, se uma página foi utilizada

recentemente há uma grande chance dela ser utilizada novamente. O valor do atributo "tempo" de cada página é equivalente ao período em que a mesma fora referenciada, seja em escrita ou em leitura. Desta forma, a página que possuir o menor valor deste atributo dentre as presentes na tabela será selecionada.

4 Análise de Tabelas

As tabelas a seguir são os resultados obtidos através de testes com um arquivo de um milhão de acessos à memória com variações nos tamanhos de página e tamanho da memória disponível.

| Tam Mem | 4K | | |
|---------|-----------------|----------------|----------|
| | Falta de página | | |
| | LRU | Segunda Chance | NRU |
| 128 | 296074 | 330908 | 321876 |
| 256 | 231892 | 305630 | 292168 |
| 512 | 217654 | 278460 | 254939 |
| 1024 | 211398 | 237151 | 228922 |
| Média: | 224773 | 292045 | 273553,5 |
| | 22,48% | 29,20% | 27,36% |

Figura 1: Tabela de página em 4KB

| Tam Mem | 16K | | |
|---------|-----------------|----------------|--------|
| | Falta de página | | |
| | LRU | Segunda Chance | NRU |
| 128 | 322392 | 400788 | 375195 |
| 256 | 303154 | 355684 | 325169 |
| 512 | 276815 | 306498 | 320681 |
| 1024 | 227703 | 293634 | 281475 |
| Média: | 289984,5 | 331091 | 322925 |
| | 29,00% | 33,11% | 32,29% |

Figura 2: Tabela de página em 16KB

| Tam Mem | 64K | | |
|---------|-----------------|----------------|--------|
| | Falta de página | | |
| | LRU | Segunda Chance | NRU |
| 128 | 388708 | 475113 | 395012 |
| 256 | 343642 | 420674 | 373535 |
| 512 | 300533 | 360953 | 340927 |
| 1024 | 286497 | 325702 | 305272 |
| Média: | 322087,5 | 390813,5 | 357231 |
| | 32,21% | 39,08% | 35,72% |

Figura 3: Tabela de página em 64KB

Averiguando cada uma das tabelas é possível perceber que a política de substituição LRU foi a mais eficiente em todos os casos pois é a que menos realiza troca de páginas. Além disso, também pode-se enxergar que com o aumento no tamanho da memória, menos páginas precisarão ser substituídas, isso é dado pelo fato de que uma memória maior consegue suportar uma maior quantidade de páginas, diminuindo a necessidade de trocas em comparação às memórias menores.

Apesar da necessidade de troca de páginas diminuir conforme o tamanho da memória aumenta, essa mesma necessidade aumenta conforme o tamanho das páginas em si aumentam também, pelo fato de que é suportada uma menor quantidade de páginas na memória e cada acesso vai alocar uma página nova com um tamanho fixo, mesmo que seja utilizado menos do que o tamanho da página.

Portanto é possível concluir que a quantidade necessária de troca de páginas é diretamente proporcional ao tamanho da página e inversamente proporcional ao tamanho da memória, quanto maior a página mais trocas e quanto maior a memória menos trocas.

5 Bibliografia

1. Sistemas Operacionais Modernos - 3ª Ed. por Andrew S. Tanenbaum Data de Acesso: 08/11
2. Gerenciamento de Memória Virtual Algoritmos de Paginação. Data de Acesso: 08/11
3. PDF disponibilizado pelo professor para estudo.