Relatório Trabalho 2

INF1019 - Sistemas de Computação

2017.2

Matheus Rodrigues de Oliveira Leal - 1511316

Felipe Gustavo Pereira Viberti - 1510384

10 de Dezembro de 2017

1. Introdução

Esse relatório consiste em explicar e analisar os resultados obtidos para o segundo de trabalho de Sistemas de Computadores. O programa consiste em um simulador de memória virtual na linguagem C. Neste simulador é criado as estruturas de dados e os mecanismos de mapeamento de memória virtual para a física necessários para realizar a paginação. Ele possui um processo pai (GM), e 4 processos filho, P1, P2, P3, e P4. Cada um desses lê um arquivo diferente, compilador.log, matrix.log, compressor.log e simulador.log. Também foi implementado um algoritmo de substituição de página global, o Least Frequently Used (LFU), que consiste em selecionar quando os page frames estiverem ocupados e se houver um Page-fault, ou seja, uma nova página precisa ser carregada na memória física.

2.Arquivos que estão sendo enviados

gm.c
gm.h
vm.c
vm.h
semaphore.c
semaphore.h

3.Passos para a execução

$gcc -Wall -o ex semaphore.c gm.c vm.c
$./ex

4. Teste

a) Dados de entrada

Os dados de entrada foram concedidos pelo professor. Em cada arquivo .log as suas linhas contém um endereço de memória acessado (em representação hexa-decimal), seguido das letras R ou W, indicando um acesso de leitura ou escrita, respectivamente.

b) Dados de saída

Realizamos testes para ver quantos pageFaults e Swaps estavam acontecendo no programa e coletamos os dados abaixo:

Testes Realizados(a memoria fisica usada sempre foi de 10,por isso os testes para x,2x e 3x são de 10,20 e 30 ms)

Teste para 10ms:
QTD DE PAGE FAULTS:16
QTD DE SWAPS:3

Teste para 20 ms:
QTD DE PAGE FAULTS:21
QTD DE SWAPS:6

Teste para 30 ms:
QTD DE PAGE FAULTS:28
QTD DE SWAPS:10

Teste de 40 ms:
QTD DE PAGE FAULTS:37
QTD DE SWAPS:18

Teste de 50ms:
QTD DE PAGE FAULTS:44
QTD DE SWAPS:25

Teste de 60 ms:
QTD DE PAGE FAULTS:50
QTD DE SWAPS:35

Teste de 70ms:
QTD DE PAGE FAULTS:56
QTD DE SWAPS:40

Teste de 80 ms:
QTD DE PAGE FAULTS:58
QTD DE SWAPS:42

Teste de 90 ms:
QTD DE PAGE FAULTS:78
QTD DE SWAPS:51

Exemplo do que seria um log(pelo menos o início):
kill number: 0 e process number: 1 //Esses kill todos são para mostrar que os processos estão mandando os sinais apesar de algumas vezes ele não estar recebendo.Descrevo o problema que tivemos com sinais na conclusão do relatório
kill number: 0 e process number: 2
kill number: 0 e process number: 4
kill number: 0 e process number: 3
kill number: 0 e process number: 1
kill number: 0 e process number: 2
kill number: 0 e process number: 4
kill number: 0 e process number: 3
kill number: 0 e process number: 1
kill number: 0 e process number: 4
Process 3, physical address 7 ,R , and offset 8d60 //Trans imprime quando a página já está na memória
Process 3, physical address 3 ,R , and offset 43c0
kill number: 0 e process number: 3
kill number: 0 e process number: 4
encheu!!#################### // Sinal de que a memória está cheia(lembrando que testamos com tamanho 10 de memória)
Processo que perdeu página:4
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 3
encheu!!####################
Processo que perdeu página:1
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 1
encheu!!####################
Processo que perdeu página:4
Atualizou todos os acessos como sendo 0

QTD DE PAGE FAULTS:16
QTD DE SWAPS:3

O log continua mas parei por aqui porque ia ficar muito grande(ele praticamente nunca acaba já que os arquivos de log dados tem 1 milhão de linhas cada).Muitas vezes o programa dá um SegFault depois de algum tempo de execução.Não temos certeza se é por uso excessivo de memória ou por invasão de uma área de memória mesmo(mais provável)

5. Conclusão

No final a execução ocorreu ocorreu razoavelmente como prevista para o testes realizados,mas tivemos alguns problemas.O principal deles é que tivemos um problema gigante com os sinais.A partir de um certo momento ele parece só ignorar alguns dos sinais que sao enviados.Colocamos até um usleep depois do envio dos sinais para retardar que outro sinal fosse enviado mas nao adiantou.Por isso a partir de um certo ponto parece que só um ou dois processos ficam executando(os outros ficaram presos esperando por um sinal que nunca chega).Esse erro não ocorre sempre(parece ser randomico).
Vale salientar que o tamanho da memória física é de 10. O programa gm cria 4 vetores de memória compartilhada chamados de segmentoTabelaPaginas que representam a tabela de páginas de cada um dos processos filhos criados para ler os arquivos .log. Em cada um desses arquivos é lida sua linha que contém um endereço de memória acessado e as letras R ou W, indicando um acesso de leitura ou escrita. Ele usa esse endereço e com a função pegaIndicePagina() pega o índice da página a ser usada. Nos filhos ele chama a função trans que faz a tradução do endereço para obter o endereço físico e manda os sinais(que estão com o problema descrito) para o GM.Quando(Se) o GM recebe o sinal ele verifica se existe espaço na RAM,se tiver ele coloca lá.Se não chama o LFU.O LFU verifica qual página tem que ser substituído de acordo com a forma descrita pelo enunciado(vê o número de acessos e se empatar o Bit M).Temos um contador global para ver quantas páginas foram acessadas e imprimir os valores de PageFault e Quantidade de Swaps que são sempre atualizados na memória compartilhada quando algum dos dois ocorre.