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

Não conseguimos implementar o contador de tempo que está descrito no trabalho.Tentamos colocar uma variável como contador de acessos geral na memória compartilhada mas o problema é que o processo que tem que ficar verificando o contador com um loop é o pai e quando fazemos esse loop,todo o resto do programa para.Tentamos fazer de outra forma mas não encontramos,considerando que o contador tem que ser atualizado pelos filhos e checado pelo pai.Fora isso ele está imprimindo na função trans sempre que a página já está em memória física.Quando não está ele calcula o endereço físico que ela vai entrar(pelo LFU caso a ram esteja cheia).

Exemplo do LOG:

kill number: 0 e process number: 1 // Isso aqui é qual processo mandou o sinal

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:65,0 // O INDEX QUE A PÁGINA FOI ALOCADA NA MEMÓRIA FISICA

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:1970,1

kill number: 0 e process number: 3

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:6410,2

kill number: 0 e process number: 2

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:7958,3

kill number: 0 e process number: 4

kill number: 0 e process number: 1

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:5109,4

kill number: 0 e process number: 3

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:34,5

kill number: 0 e process number: 2

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:14422,6

Process 3, physical address 5 ,R , and offset 8d60 // Quando a página já está lá a trans imprime as informações pedidas

Process 3, physical address 1 ,R , and offset 43c0

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:66,7

Process 2, physical address 2 ,R , and offset fc20

kill number: 0 e process number: 4

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:1511,8

kill number: 0 e process number: 1

kill number: 0 e process number: 3

PHYSICAL ADDRESS NA SHARED MEMORY INDEX:17,9

encheu!!################## // Aqui ele vai chamar o LFU

lost page index: 0
process page owner 1
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 1
kill number: 0 e process number: 2
encheu!!####################
kill number: 0 e process number: 4
lost page index: 6
process page owner 2
tomei um sigusr2 pq perdi a pagina
Process 2, physical address 2 ,R , and offset 7c20
Process 2, physical address 2 ,R , and offset 7c28
Process 2, physical address 2 ,R , and offset 7c28
Process 2, physical address 2 ,R , and offset ff38
Process 2, physical address 2 ,R , and offset 7b48
Process 2, physical address 2 ,R , and offset 7b48
Process 2, physical address 0 ,W , and offset 6f00
kill number: 0 e process number: 3
encheu!!####################
lost page index: 0
process page owner 2
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 2
kill number: 0 e process number: 1
encheu!!####################
lost page index: 0
process page owner 4
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 4
encheu!!####################
Process 3, physical address 9 ,R , and offset 3380
lost page index: 0
process page owner 3
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 1

kill number: 0 e process number: 4
encheu!!####################
kill number: 0 e process number: 3
lost page index: 0
process page owner 4
tomei um sigusr2 pq perdi a pagina
Process 3, physical address 9 ,R , and offset 33a0
Process 3, physical address 9 ,R , and offset 33c0
Process 3, physical address 5 ,R , and offset ccb0
Process 3, physical address 9 ,R , and offset 33e0
Process 3, physical address 9 ,R , and offset 3400
Process 3, physical address 5 ,R , and offset ccc0
Process 3, physical address 9 ,R , and offset 3420
kill number: 0 e process number: 4
encheu!!####################
lost page index: 0
process page owner 1
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 1
kill number: 0 e process number: 3
encheu!!####################
Process 4, physical address 0 ,R , and offset f860
lost page index: 3
process page owner 4
tomei um sigusr2 pq perdi a pagina
kill number: 0 e process number: 4
kill number: 0 e process number: 1
encheu!!####################
lost page index: 0
foi no mesmo
process page owner 4
kill number: 0 e process number: 1
tomei um sigusr2 pq perdi a pagina
encheu!!####################

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).

5. Conclusão

No final a execução ocorreu não ocorreu como prevista para o testes realizados, por alguns motivos.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 só um ou dois processos ficam executando(os outros ficaram presos esperando por um sinal que nunca chega).Vale salientar que o tamanho da memória física é de 10(basta mudar para 256 caso queira testar). 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 numero de acessos e se empatar o Bit M).