

2º Trabalho de Sistemas Operacionais – INF 1316 – 2025.1

Entrega: 24 de junho de 2025, 23:59

Apresentação: 26 de junho de 2025, 11-13 hs

O trabalho consiste em implementar um programa em linguagem C que simule um Gerenciador de Memória Virtual (GMV). Com o simulador você deverá testar e comparar o desempenho de quatro algoritmos de substituição de páginas: o algoritmo **Not Recently Used (NRU)**, o algoritmo de **Segunda Chance**, o algoritmo **LRU/Aging**, e o algoritmo **Working Set(k)**, este último para uma tamanho de janela de k instruções. Dentre esses algoritmos, o **NRU** e o **2nd Chance** deverão ser de substituição global, e o LRU e o WS devem ser de substituição local.¹

Seu programa simulador deve mostrar como quatro processos de aplicação sintéticos, P1-P4, competem pelos quadros de página de uma memória física hipotética composta de apenas 16 quadros de página (page frames). A simulação acontecerá em rodadas, onde em cada rodada cada um dos processos P1, P2, P3 e P4 tem a vez de gerar um endereço randômico de memória virtual. Para tal, implemente um processo unix (e.g. **TodosProcessos**) que cria quatro processos filho P1, P2, P3 e P4 e que se revezam de forma circular (RoundRobin) para transmitir os endereços de memória virtual acessados através de quatro diferentes pipes (ou shmenv) para o GMV.

Portanto, quando for a vez de um processo P_x , ele irá gerar um novo acesso a memória, na forma de um par: <índice da página virtual; tipo_acesso>, onde tipo_acesso pode ser “R” ou “W” indicando se o acesso a memória foi de leitura ou de escrita. Note que basta saber o índice da página e o deslocamento (offset) do endereço virtual nesse caso é desprezível. A sequência desses pares devem ser lidos, um a um, de um arquivo específico para cada processo. Por exemplo, os processos P1 e P3 leem as linhas dos arquivos “acessos_P1” e “acessos_P3”, respectivamente (vide Figura 1). As linhas desses arquivos deverão ser gerados previamente de forma aleatória, devem ter o formato: (page-index R/W), como mostrado abaixo.

acessos_P1	acessos_P2	acessos_P3	acessos_P4
04 R	12 W	23 W	08 W
29 W	24 R	19 R	15 R
22 R	24 R	02 R	10 R
09 R	26 R	00 R	18 R
20 R	10 W	11 W	09 W
...

Figura. 1: Exemplo de arquivos contendo a sequência de acessos a páginas virtuais de cada processo (cada arquivo deve ter pelo menos 100 linhas)

Por exemplo, a execução circular de: P1, P2, P3, P4, P1, P2, P3, P4, ... irá gerar a sequência de endereços (04,R), (12,W), (23,W), (08, W), (29,W), (24,R), (19, R), (15,R),...

¹ Lembrando que, “substituição local” de páginas significa que a página a ser substituída é sempre do mesmo processo que causou o page-fault, enquanto que “substituição global” considera todas as páginas carregadas na memória, seja de qualquer processo.

² O símbolo P_x está no lugar de P1, P2, P3 ou P4

Para esse trabalho assuma que o espaço de endereçamento lógico de cada processo consiste de 32 páginas. Portanto, a imagem completa de cada processo Px por si só já “não caberia” na memória principal. Por isso, todos os índices de página virtual geradas pelos processos estão no intervalo [0-31].

As páginas virtuais geradas por cada processo Px serão escritas na pipe (ou shmem) correspondente e transmitidas para o Gerenciador de Memória Virtual (GMV), que irá ler qualquer uma das pipes que tenha algum novo endereço virtual e tentará alocar um quadro de página da memória física (fictícia) para a nova página virtual.

Ponto extra:

Simule também que sempre que ocorra um page-fault, o processo Px que causou interrupção deverá demorar um pouco mais para concluir o seu acesso à página recém copiada para a memória RAM, no escalonamento RoundRobin.

Funcionamento do GMV

O Gerenciador de Memória Virtual (GMV) deve representar a memória física (RAM) fictícia com 16 quadros de página, e uma tabela de página para cada um dos processos do usuário, digamos, TP_Px. Essa tabela de página deverá ter entradas que comportem todos os bits necessários (presença, pg escrita, etc. , os contadores e endereços utilizados por algum dos 4 algoritmos de substituição de página a serem testados.

Quando um dos processos, digamos Px, gerar uma nova página virtual, p, o Gerenciador de Memória Virtual (GMV) deve decidir como vai tratar esse novo acesso a uma página virtual.

As etapas/opções são:

1. alocar um quadro de página vazio, para a nova página virtual, caso exista um quadro vazio
2. contabilizar um page-fault, e registrar qual o processo originou o page fault
 - 2.1. escolher um quadro de página a ser liberado usando o algoritmo de substituição de páginas, e sobrescrevê-la com a página p (obs: digamos que Py foi o processo que “perdeu o seu quadro de página” para Px)
 - 2.2. se o conteúdo do quadro tiver sido modificado (i.e. bit M=1), então o seu simulador deverá registrar que houve/haverá uma cópia adicional da página “suja” para o disco de swap (hipotético)
3. atualizar a Tabela de Página dos processos Px e Py

Em seu simulador GMV cada um dos algoritmos de substituição de página devem ser implementados através de um procedimento separado, e.g. **select_NRU()**, **select_2nCh()**, **select_LRU()**, **select_WS()**, etc, a ser selecionado/ativado de acordo com o argumento NRU/ 2nCH/ LRU/ WS passados na linha de comando do seu trabalho.

No caso da substituição de páginas usando a técnica de WorkingSet(k) você deve comparar o desempenho da paginação testando diferentes números de janela de working set k, igual para todos os processos.

Sugiro simular para janelas de tamanho k=3 e k= 5.

Saída do Simulador

O seu simulador deve ser interativo. Você deve indicar o algoritmo de substituição que será usado (NRU/ 2nCH/ LRU/ WS) e quantas rodadas ele deve executar. E como resultado o seu simulador deverá imprimir na tela:

Algoritmo de Substituição <parâmetro>

Rodadas executadas: 50

E mostrar a sequência de Page-faults gerados:

Com P_x = gerador do page fault e P_y = processo que perdeu o quadro de página

Além disso, indicar também se o page fault ocasionou uma escrita de página modificada ("suja") na área de swap

Mostrar a Tabela de Processos de cada processo

Obs: <parâmetro> se aplica no algoritmo de Working Set

Relatório

No seu relatório você deverá mostrar o desempenho comparativo da execução dos algoritmos.