

```
// Eric Goulart da Cunha - 2110878
// João Pedro Biscaia Fernandes - 2110361
```

gerenciador de memória virtual (GMV) simulada

Entrada esperada:

```
1 03 R
2 01 W
3 07 R
4 02 R
5 02 R
6 02 W
7 06 W
8 04 R
9 08 R
10 01 R
11 12 W
12 01 R
13 12 R
14 14 W
15 01 W
16 02 W
17 05 W
18 10 R
19 03 R
20 09 R
21 03 R
```

Os arquivos que o programa está lendo devem ter essa formatação, estamos usando três arquivos: **acessos_P1**, **acessos_P2**, **acessos_P3** os três com mais de cem linhas cada.

Funções:

```
void print_table(Tabela *tabela);
void read_and_print_file_line_by_line(const char *filename, char *nome_processo, sem_t *sem_current, sem_t
*sem_next, Tabela *tabela, sem_t *mutex, int process_id, const char *algoritmo, int rodadas, int window_size);
Tabela* trocador_de_paginas(Tabela *tabela, int pagina_virtual, int process_id, char modo, const char *algoritmo,
int window_size);
int troca_NRU(Pagina *tab_paginas, Pagina entrada, int *substituido_id);
void age_pages(Tabela *tabela);
int troca_LRU(Pagina *tab_paginas, Pagina entrada, int *substituido_id);
int troca_second_chance(Pagina* tab_paginas, Pagina entrada, int *substituido_id);
int troca_working_set(Pagina *tab_paginas, Pagina entrada, int window_size, int *substituido_id);
```

Essas são as funções que temos no código.

print_table é responsável por printar a tabela no terminal para irmos verificando se tudo está funcionando corretamente.

read_and_print_file_line_by_line lê os arquivos linha por linha interpretando cada linha como um acesso a uma página virtual de um processo específico. Ela utiliza semáforos para garantir que apenas um processo altere a tabela por vez.

trocador_de_paginas Manipula a tabela de páginas para adicionar ou substituir entradas, conforme necessário, baseado no algoritmo de substituição de páginas especificado (NRU, LRU, Segunda Chance).

age_pages Incrementa a idade de todas as páginas na tabela e reseta o bit de validação (BV) após um número específico de iterações.

troca_NRU Implementa o algoritmo de substituição de páginas "Not Recently Used" (NRU). Seleciona uma página para substituição com base nos bits de validação e modificação.

troca_LRU Implementa o algoritmo de substituição de páginas "Least Recently Used" (LRU). Seleciona a página menos recentemente usada para substituição.

troca_second_chance : Implementa o algoritmo de substituição de páginas "Second Chance". Dá uma segunda chance para páginas que têm o bit de proteção (BP) setado antes de substituí-las.

troca_working_set Implementa o algoritmo de substituição de páginas "Working Set". Mantém páginas ativamente usadas dentro de uma janela de tempo especificada (window_size). Se a página já existe, ela é atualizada. Caso contrário, se a janela estiver cheia, substitui a página candidata que excede a idade permitida,

Agora falando da main. Primeiro pegamos os argumentos passados pelo usuário, inicializamos os semáforos, cada processo filho tem seu próprio semáforo para sincronização e “mutex” para garantir acesso exclusivo a tabela. Em seguida começamos a mexer com memória compartilhada para garantir que os 3 processos tenham acesso a mesma tabela de páginas. Criamos os 3 processos filhos com fork() e cada um deles executa a função **read_and_print_file_line_by_line** com seus respectivos arquivos e semáforos. Quando todos os processos filhos terminam o pai imprime a quantidade de page faults que ocorreu, fecha e desvincula os semáforos, e desassocia e remove a memória compartilhada.

Conclusões:

Tivemos algumas dificuldades para entender como implementar corretamente algumas políticas, mas depois de estudar mais conseguimos fazer os algoritmos.