



Universidade Federal do Rio Grande do Norte - UFRN
Centro de Ensino Superior do Seridó - CERES
Departamento de Computação e Tecnologia - DCT

Curso: Bacharelado em Sistemas de Informação

Disciplina: DCT2101 – Sistemas Operacionais

Professor: João Borges

Data: 05 de julho de 2025

Unidade 3 - Tarefa 1
Laboratório de Memória Virtual

ATENÇÃO 1: Só serão aceitos trabalhos em **duplas** ou **individuais**, mais do que isso invalidará o trabalho;

ATENÇÃO 2: Não serão permitidos plágios entre os trabalhos, sendo punidos, ambos os componentes que tiverem seus trabalhos iguais, com nota 0 (zero)

Objetivo O objetivo deste trabalho é a fixação dos conteúdos teóricos abordados sobre a Gerência de Memória em Sistemas Operacionais Modernos, focando nos mecanismos de suporte à Memória Virtual, em particular o que acontece quando ocorre uma falta de página (*page-fault*) e os algoritmos de substituição de páginas.

1. Esta atividade consiste das seguintes tarefas:
 - (a) Implementação de Algoritmos de Substituição de Páginas em um Simulador Simples de Memória Virtual, baseado em Paginação;
 - (b) Análise comparativa dos Algoritmos de Substituição de Páginas implementados.
2. A implementação deverá ser baseada no Simulador de Memória Virtual utilizado na disciplina de Sistemas de Operação da Universidade Nova de Lisboa, do Prof. Vitor Duarte.
 - (a) O código fonte de base do simulador, e os arquivos de *trace* para execução, estão disponíveis no seguinte repositório:
https://github.com/labepi/virtmem_sim
 - (b) Este simulador já implementa um algoritmo de escalonamento aleatório (randômico) como prova de conceito do simulador.
3. A tarefa consistirá em, com base na implementação citada, implementar os seguintes algoritmos de substituição de páginas:
 - (a) FIFO (*First-In First-Out*);
 - (b) LRU (*Least Recently Used*).
 - (c) SC (*Second Chance*).
 - (d) LFU (*Least Frequently Used*).
 - (e) MFU (*Most Frequently Used*).

4. São permitidas modificações no código original do simulador, a fim de realizar as implementações exigidas, mas o código base deve ser o mesmo passado na atividade.

5. Sobre o Simulador de Memória Virtual

- (a) O arquivo principal do simulador é o `simulador.c`.
- (b) Este simulador irá receber uma sequência de endereços virtuais de referências reais à memória feitas por um processo rodando em uma CPU real. Esta sequência de endereços é passada para o simulador a partir de um arquivo de *trace* de endereços de memória previamente coletados.
- (c) Ao executar, o simulador irá solicitar os seguintes parâmetros:

```
./simulador traceFile numero-de-frames algoritmo
```

Onde:

- `traceFile` é o arquivo de *trace* a ser utilizado na simulação pode ser:
 - i. `trace1.trace`
 - ii. `trace2.trace`
 - iii. `trace3.trace`
 - iv. `trace4.trace`
 - `numero-de-frames` que é o número de *frames* disponível na memória virtual (OBS: não pode exceder o valor de 64).
 - `algoritmo` que indica qual algoritmo deverá ser utilizado para na simulação (OBS: pode ser: `random`, `fifo`, `lru`, `sc`, `lfu`, `mfu`).
- (d) Os arquivos de trace estão disponíveis no diretório `traces`, no repositório da tarefa, e estão compactadas como `.zip`, devendo ser descompactadas antes de usar.
- (e) Cada arquivo de trace possui entradas da seguinte forma:

```
190a7c20 R
3856bbe0 W
...
```

onde,

- a primeira coluna corresponde ao endereço virtual gerado, representados em valores hexadecimais, e
 - a segunda coluna representa o modo de acesso: R (leitura) ou W (escrita).
- (f) O simulador considera os seguintes detalhes para a memória virtual:
- i. Um processo tem um espaço de endereçamento virtual com 32 bits em que 12 bits são para o deslocamento e 20 bits para o número da página virtual.
 - ii. O processo poderá ter no máximo 2^{20} páginas virtuais ($1024 * 1024$).
 - iii. Cada página terá 2^{12} bytes (4096).
 - iv. A tabela de páginas é simulada por um vetor de registros, onde cada página contém:
 - bit de validade
 - número do frame em que a página está carregada
- (g) Para o hardware considerado pelo simulador:
- i. A memória física está dividida em frames com o mesmo tamanho das páginas.
 - ii. O número de frames que o processo pode ocupar é um parâmetro da simulação definido na linha de comando.

- iii. A tabela de frames (tabela de páginas invertida) é simulada por um vetor de registros, onde cada frame contém:
 - o número da página guardada na frame
 - o bit de referência
 - o bit de modificação (dirty bit)
 - a data do carregamento da página virtual na frame
 - a data da última referência à frame
 - (h) As referências às páginas podem ser feitas como leitura ou escrita.
 - (i) O gerenciamento das páginas é feita com paginação sob demanda. Isto significa que inicialmente o processo não ocupa nenhum “frame” na memória física. As páginas vão sendo carregadas à medida que vão sendo referenciadas.
 - (j) Quando não existe nenhum “frame” livre é preciso escolher uma página virtual como vítima, para ser removida.
 - (k) A página PV (página vítima) deixa de estar carregada na memória. A escolha da página vítima é feita por um algoritmo de substituição de páginas, que deverá ser implementado, e será escolhido na execução a partir da linha de comando.
 - (l) O algoritmo `random` já está implementado no simulador, na função `int selectVictimRandom (void)`. Neste, o frame vítima é escolhido aleatoriamente.
 - (m) A sua implementação deverá completar os códigos dos demais algoritmos, nas funções:
 - `int selectVictimFifo (void)`
 - `int selectVictimLRU (void)`
 - `int selectVictimSC (void)`
 - `int selectVictimLFU (void)`
 - `int selectVictimMFU (void)`
 - (n) Cada uma destas funções deve retornar um valor inteiro, que corresponde ao frame (tabela `frameTable`) no qual a página vítima está alocada.
6. Sobre as tarefas exigidas para este trabalho, devem ser seguidas as etapas:
- (a) Entendimento do simulador apresentado;
 - (b) Implementação dos algoritmos de substituição de páginas;
 - (c) Execução do simulador com as entradas dos *traces* cedidos junto com o simulador;
 - (d) Análise da execução do simulador, para os diferentes algoritmos, com relação a:
 - Número de *pages fault*
 - Número de *pages written*
 - (e) Documento com análise comparativa dos algoritmos implementados.
 - Execute os diferentes algoritmos para diferentes valores de frames disponíveis:
 - 2, 4, 8, 16, 32, 64
7. A entrega da atividade deverá ser feita pelo SIGAA, até a data estabelecida da tarefa cadastrada.
8. Deverá ser entregue ao professor, um documento compactado contendo:
- O código-fonte da implementação do simulador com os algoritmos implementados.
 - Um documento PDF contendo os gráficos de comparação das execuções.