



Pontifícia Universidade Católica de Minas Gerais
Departamento de Ciência da Computação
Disciplina: Sistemas Operacionais - TP
Professor: Pedro Ramos

VALOR: 30 PONTOS (10 código + 20 entrevista)

ENTREGA NO CANVAS: 04/12/2025

1 Objetivo

Neste trabalho prático, você deverá implementar um simulador do sistema de gerenciamento de **Memória Virtual** utilizando **Paginação por Demanda** em um **único processo**. O foco é a simulação e comparação do desempenho de quatro políticas distintas de substituição de páginas, contabilizando o número de *Page Faults* (falhas de página) para cada cenário, bem como o tempo decorrido em cada processamento.

2 Especificações e Entrada

Seu programa deverá processar as configurações do sistema e as sequências de requisições a partir da entrada padrão (stdin), e escrever o resultado do processamento na saída padrão (stdout) conforme exemplo a seguir em Java e C/C++:

```
1 # em Java
2 $ javac Main
3 $ java Main > saida.txt < entrada.txt
4
5 # em C++
6 $ g++ -o exec main.c
7 $ ./exec > saida.txt < entrada.txt
```

2.1 Entrada de Configuração

A primeira parte da entrada consistirá nos seguintes parâmetros, fornecidos sequencialmente:

1. **Tamanho da Memória Física (M):** Um valor inteiro que representa a capacidade total da memória principal (RAM) em **bytes**.
2. **Tamanho da Memória Virtual (V):** Um valor inteiro que representa o tamanho total do espaço de endereçamento virtual em **bytes**. Assume-se que $V \geq M$.
3. **Arquitetura (A):** Uma string indicando a arquitetura de endereçamento (*x86* ou *x64* bits).
4. **Número de Páginas Virtuais (P):** Um valor inteiro que representa a quantidade total de páginas no espaço de endereçamento virtual. Os índices das páginas variam de 0 a $P - 1$.
5. **Número de Sequências (N):** Um valor inteiro que indica a quantidade de sequências de requisições de páginas a serem processadas pelo simulador.

2.2 Entrada de Requisições

A partir do parâmetro N , o programa deverá ler N blocos de sequências de requisições:

6. Para cada uma das N Sequências:

- Primeiro, um número inteiro (**R**) que indica o total de requisições nessa sequência.
- Em seguida, R números inteiros (índices de página), variando de 0 a $P - 1$, que representam as páginas virtuais requisitadas em ordem cronológica.

2.3 Cálculos e Parâmetros Derivados

O simulador deve inferir certos parâmetros para funcionar corretamente. Por exemplo:

1. Qual é o **Tamanho da Página (S_P)**? Você deve assumir que S_P é uma potência de 2.
2. Quantos **Frames (N_{frames})** ou *registros* há na memória física? Considere o tamanho da página S_P neste cálculo.
3. Qual o **Tamanho Mínimo de Swap**? O espaço de armazenamento secundário necessário (em bytes) para manter todas as páginas virtuais que não cabem na memória física?

3 Políticas de Substituição de Páginas

Para **cada uma** das N sequências de requisições de páginas, seu programa deve executar **quatro simulações distintas**. Antes de cada simulação de política, o estado inicial da memória física deve ser reinicializado (vazio).

As políticas a serem implementadas são:

1. **LRU (Least Recently Used)**: Substitui a página cujo último acesso ocorreu no tempo mais distante.
2. **FIFO (First-In, First-Out)**: Substitui a página que reside na memória há mais tempo (a primeira a entrar).
3. **Aleatório (RAND)**: Substitui um frame escolhido aleatoriamente da memória física.
4. **MIN / Ótima (OPT)**: (Política teórica). Substitui a página que não será utilizada pelo período de tempo mais longo no futuro. Esta política requer o conhecimento total da sequência de requisições.

4 Saída (Output)

Para cada sequência de requisições processada, o simulador deve emitir os seguintes resultados:

1. Parâmetros Derivados do Sistema:

- Tamanho da Página Inferido: S_P bytes
- Número de Frames de Memória Física: N_{frames}
- Tamanho do Swap Necessário: (Tamanho do Swap) bytes

2. Resultados da Simulação por Sequência:

- Sequência de Requisições: [...] (Listar a sequência de entrada)
- Então, para **cada política**, listar:
 - Tempo de execução total do processamento (em segundos, número inteiro mais próximo);
 - Número de Page Faults;
 - Estado do Swap ao final do processamento;

5 Exemplo de Entrada

O exemplo de entrada será composto por múltiplas linhas. As primeiras 4 linhas indicam, respectivamente:

- Tamanho da memória física F
- Tamanho da memória virtual M ($M \geq F$)
- Arquitetura (x86 ou x64)
- Número de páginas P

Seguidas de uma linha em branco, e um outro número N que indica o numero de sequências de requisições de páginas devem ser processadas. Cada sequência é listada a seguir, separadas por uma linha em branco. Cada sequência é composta por 2 linhas: o número de requisições e a sequência de requisições.

A seguir, um exemplo de arquivo de entrada (arquivo de texto UTF-8) formatado conforme as especificações.

```

1 4096
2 16384
3 x86
4 16
5
6 2
7
8 8
9 0 1 2 3 0 1 4 0
10
11 10
12 0 1 2 0 3 4 5 6 7 0

```

Neste exemplo, o tamanho da memória física é 4096 bytes; o tamanho da memória virtual é de 16384 bytes; a arquitetura é de 32 bits; o número de páginas é 16. Duas sequências de requisições de páginas serão processadas. A cada sequência, considere a memória virtual inicialmente vazia. No exemplo acima, a primeira sequência tem 8 requisições e a segunda sequência tem 10 requisições.

6 Exemplo de Saída

Para um exemplo simplificado com apenas uma sequência (entrada.txt) abaixo:

```

1 4096
2 16384
3 x86
4 16
5
6 1
7
8 8
9 0 1 2 3 0 1 4 0

```

A saída (saida.txt) deve ser no seguinte formato:

```
1 1024
2 4
3 12288
4
5 1
6
7 0 1 2 3 0 1 4 0
8 FIFO
9 0
10 6
11 1
12 RAND
13 0
14 5
15 2
16 LRU
17 0
18 5
19 2
20 MIN
21 0
22 5
23 3
```

Na saída acima, as primeiras três linhas indicam que a página tem 1024 bytes, há 4 registros (frames) na memória, e o espaço de Swap deve ser no mínimo igual a 12288 bytes. Uma sequência foi processada.

A seguir, a sequência já foi processada e os dados das 4 políticas são listados na ordem indicada: FIFO, RAND (aleatório), LRU, MIN.

Em FIFO, por exemplo, o tempo decorrido foi de 0 segundos (por se tratar de uma sequência pequena), e houve 6 *page faults*. O estado do Swap contém a página 1.

7 Critérios de Avaliação

O programa pode ser implementado em Java, C ou C++ **somente**. O trabalho pode ser realizado em **dupla, trio ou individualmente**.

1. Código (10 pontos): você deve gerar exemplos pequenos, médios e grandes para seu simulador. Utilize um script python disponibilizado com a especificação.
2. Entrevista (20 pontos) (04/12): Você deve ser capaz de explicar ao professor quais foram as estruturas de dados usadas e porquê, e quais decisões de implementação foram tomadas para cada uma das políticas.

Bom trabalho!