



Sistemas Operacionais

Simulador de Memória Virtual e Física

Abrahão Picanço e Andreza Gonçalves



- Programas e sistemas modernos exigem mais memória do que os computadores fisicamente possuem.
- **A Questão Central:** Como podemos executar aplicações complexas em um hardware com recursos limitados?
- **A Resposta: A Memória Virtual**, uma das abstrações mais importantes da computação.
- **Foco de hoje** é entender o que é, como funciona, e como nós simulamos seu comportamento para analisar seu desempenho na prática.



- É uma **técnica de gerenciamento de memória** implementada pelo Sistema Operacional.
- Cria a **ilusão** de que o sistema tem muito mais memória RAM do que realmente existe.
- **Como?** Usa o armazenamento secundário (HD ou SSD) como uma extensão da RAM.



1. Processo solicita um dado: A CPU pede um endereço de memória.

2. Verificação na RAM:

- **Se a página está na RAM (HIT):** Ótimo! O acesso é rápido e o programa continua.
- **Se a página NÃO está na RAM (MISS):** Ocorre um **PAGE FAULT**.

3. Ação do Sistema Operacional:

- O SO assume o controle.
- Busca a página necessária no disco.
- **Se a RAM estiver cheia**, um **algoritmo de substituição (FIFO, LRU, Random)** escolhe qual página antiga deve sair.

Nosso simulador modela exatamente este fluxo para contar os *page faults*.



Testes realizados

Algoritmo de substituição fifo (first-in, first-out):

```
● PS C:\Users\andre\Downloads\memory_test> ./simvirtual fifo simulador.log 10 500  
  
Executando o simulador...  
Arquivo de entrada: simulador.log  
Tamanho da memoria: 500 KB  
Tamanho das paginas: 10 KB  
Tecnica de reposicao: fifo  
Total de acessos a memoria: 1000000  
Paginas lidas: 57690  
Paginas escritas: 13333  
○ PS C:\Users\andre\Downloads\memory_test> █
```



Testes realizados

Algoritmo de substituição lru (least recently used):

```
PS C:\Users\andre\Downloads\memory_test> ./simvirtual lru matriz.log 20 240

Executando o simulador...
Arquivo de entrada: matriz.log
Tamanho da memoria: 240 KB
Tamanho das paginas: 20 KB
Tecnica de reposicao: lru
Total de acessos a memoria: 1000000
Paginas lidas: 219134
Paginas escritas: 51885
PS C:\Users\andre\Downloads\memory_test> |
```



Testes realizados

Algoritmo de substituição random com debug:

```
Acesso 198: Endereco 0x5b99b60, Op R -> Pagina 2931  
> Resultado: MISS. Memoria cheia -> PAGE FAULT!  
> Vitima escolhida (quadro 29): pagina 11835
```

```
Acesso 199: Endereco 0x5b99b50, Op R -> Pagina 2931  
> Resultado: HIT no quadro 29
```

```
Acesso 200: Endereco 0x42e320, Op R -> Pagina 133  
> Resultado: HIT no quadro 0
```

```
Executando o simulador...  
Arquivo de entrada: tarefa.log  
Tamanho da memoria: 1600 KB  
Tamanho das paginas: 40 KB  
Tecnica de reposicao: random  
Total de acessos a memoria: 200  
Paginas lidas: 96  
Paginas escritas: 29
```



- **Execução de Programas Maiores que a RAM:** O benefício mais óbvio.
- **Aumento da Multiprogramação:** Mais processos podem ser mantidos na memória simultaneamente, pois apenas partes deles precisam estar lá.
- **Proteção e Isolamento:** Cada processo tem seu próprio espaço de endereço virtual, impedindo que um programa interfira no outro.
- **Compartilhamento de Código:** Múltiplos processos podem compartilhar a mesma cópia de uma biblioteca na memória, economizando espaço.



- **O Custo do Page Fault:** Acessar o disco é **milhares de vezes mais lento** que acessar a RAM. Muitos *page faults* degradam severamente o desempenho.
- **O Perigo do *Thrashing* (Tranco):** É o colapso do sistema quando ele passa mais tempo trocando páginas entre a RAM e o disco do que executando trabalho útil. Ocorre quando não há memória física suficiente para os processos ativos.
- **A Escolha do Algoritmo é Crucial:** Um bom algoritmo de substituição (como o LRU) é fundamental para minimizar os *page faults* e evitar o *thrashing*.
- **O Custo da Página Suja:** Se a página removida foi modificada ('W'), ela precisa ser salva no disco, duplicando o tempo de acesso e a lentidão



- **Sistemas Operacionais Modernos: Windows, macOS, Linux, Android e iOS** dependem totalmente da memória virtual.
- **Aplicações do Dia a Dia:** Permite rodar jogos pesados, editores de vídeo, softwares de engenharia e ter dezenas de abas abertas no navegador.
- **Tecnologias de Servidores:** Essencial para virtualização (VMware, VirtualBox) e contêineres (Docker).



O gerenciamento de memória virtual é um pilar da computação moderna, e a eficiência dos algoritmos de substituição é um fator determinante para o desempenho de todo o sistema.