

# Conceitos Fundamentais de Gerenciamento de Memória

## Memória Física e Virtual:

- **Memória Física:** Espaço real na RAM que o sistema operacional e os processos usam para armazenar dados.
- **Memória Virtual:** Um conceito usado pelos sistemas operacionais para dar a impressão de que há mais memória disponível do que fisicamente existe (o sistema mente pro programa que existe memória, mas na verdade ele picota o processo[ex: chrome {seu fdp}]). Os processos utilizam endereços lógicos, que são convertidos em endereços físicos pela MMU (Memory Management Unit).

## MMU (Memory Management Unit):

- Componente de hardware que traduz endereços lógicos (virtuais) em endereços físicos.
- Faz a proteção da memória, impedindo que processos acessem áreas que não foram alocadas para eles (tipo um segurança, decide quem entra e quem sai).

## Hierarquia de Memória:

- A memória é organizada em níveis hierárquicos que variam em velocidade e capacidade, **como cache, RAM e memória auxiliar (disco rígido ou SSD)**. Mais pra cima = mais rápido e menos espaço (CAPACIDADE E VELOCIDADE SÃO INVERSAMENTE PROPORCIONAIS [O cache mais alto é 10mb, RAM é nuns 16GB, disco 1TB])

### Tempo de Acesso e Latência:

- O tempo de acesso varia entre tipos de memória. A RAM tem tempo de acesso muito menor que o disco. O tempo médio de acesso à memória pode ser influenciado pela ocorrência de **faltas de página**.

## Memória Virtual

## Partições e Segmentos:

- **Partições:** A forma mais simples de gerenciar a memória física, onde ela é dividida em blocos contíguos. Cada processo recebe uma partição, e há um registrador base e limite que controla o espaço de cada processo.

Contíguo refere-se a algo que está lado a lado ou que ocupa uma sequência contínua  
Exemplos: Memória contígua:

Um processo que é alocado em uma região contínua de memória ocupa um espaço único  
Memória não contígua:

Quando o espaço de memória é dividido e os blocos não estão lado a lado, ou seja,

No caso de alocação contígua de memória, a simplicidade na organização é uma vantagem

- **Segmentação:** A memória é dividida em segmentos (TEXT, DATA, etc.) de tamanhos variáveis, cada um com políticas de acesso distintas. É mais flexível que a partição, mas também pode causar fragmentação externa.

A segmentação não é necessariamente contígua.

Explicação:

Os segmentos podem ser alocados em locais diferentes da memória física. Isso significa que, por exemplo, você pode ter um segmento de código (TEXT) alocado em uma parte da memória física.

## Paginação:

- A paginação divide a memória física e virtual em blocos de tamanho fixo chamados páginas (memória virtual) e quadros (memória física). A tradução entre eles é feita por uma tabela de páginas.
- Cada processo tem sua tabela de páginas. As páginas que não estão sendo usadas podem ser movidas para o disco para liberar espaço na RAM.
- Tamanho das Páginas e Quadros

Definição: Páginas (memória virtual) e quadros (memória física) são blocos de tamanho fixo. Ambos devem ter o mesmo tamanho, com tamanhos comuns variando entre 4 KB, 8 KB, ou 16 KB. Divisão da Memória: A memória virtual de um processo é dividida em páginas, enquanto a memória física (RAM) é dividida em quadros. Por exemplo, um processo de 8 KB usaria 2 páginas de 4 KB.

Tabela de Páginas: Cada processo possui uma tabela de páginas que mapeia suas páginas virtuais para os quadros físicos correspondentes na memória. Quando um endereço lógico é acessado, ele é dividido em um número de página e um offset. Acesso à Memória: A MMU utiliza a tabela de páginas para traduzir endereços lógicos em endereços físicos. Se uma página não estiver na RAM, o sistema operacional a carrega do disco.

Escolha do Tamanho: O tamanho das páginas e quadros é uma decisão de design que afeta a eficiência da memória. Tamanhos menores podem reduzir a fragmentação interna, enquanto tamanhos maiores podem aumentar a eficiência do gerenciamento da memória.

## TLB (Translation Lookaside Buffer):

- Pequena memória cache dentro da MMU que armazena as últimas traduções de endereços virtuais para físicos, agilizando o processo e reduzindo o tempo de acesso à memória.

## Paginação em Disco

### Falta de Página (Page Fault):

- (geladeira)
- Ocorre quando um processo tenta acessar uma página que não está na memória RAM, resultando em uma interrupção e movimentação da página do disco para a RAM.
  - O sistema pode estar em dois estados: memória saturada (quando é preciso liberar uma página da RAM para carregar uma nova) e memória não saturada (há espaço disponível para novas páginas).

## Swapping e Paging:

- Swapping: Um processo inteiro é movido da memória para o disco (swap out) quando está ocioso e trazido de volta quando necessário (swap in). Usado menos frequentemente hoje.
- Paging: Paginação de partes (páginas) do processo, movendo páginas individuais para o disco quando não estão sendo usadas.

# Critérios para Seleção de Páginas a Serem Movidas para o Disco:

- Idade da página: Páginas mais antigas têm maior chance de serem movidas.
- Frequência de acesso: Páginas menos acessadas recentemente podem ser substituídas.

## Algoritmos de Substituição de Páginas

- FIFO (First In, First Out): As páginas são substituídas na ordem em que foram carregadas. Simples de implementar, mas pode não ser eficiente, pois não leva em conta se as páginas antigas ainda estão sendo usadas.
- LRU (Least Recently Used): Substitui a página que não foi utilizada há mais tempo, com base na suposição de que páginas usadas recentemente continuarão sendo usadas. É um dos algoritmos mais eficientes, mas consome mais recursos, pois precisa manter o controle de acesso a cada página.
- Algoritmo Ótimo (OPT): Substitui a página que será utilizada mais tarde no futuro, mas é apenas teórico, pois o sistema não pode prever o futuro.
- Segunda Chance (Relógio): Variante do LRU, menos complexa. Dá uma "segunda chance" para páginas usadas recentemente, evitando a remoção imediata.
- NRU (Not Recently Used): Classifica as páginas em quatro categorias com base em se foram referenciadas e se são modificadas. O sistema escolhe uma página de uma categoria que não foi referenciada para substituição.
- Random: Substitui uma página aleatoriamente. É simples de implementar, mas pode ser ineficiente, pois não leva em conta o uso das páginas.
- LRU Aproximado: Usa técnicas de amostragem para estimar quais páginas não foram usadas recentemente, reduzindo a complexidade do LRU completo. Exemplos incluem usar contadores de tempo ou bits de referência.
- Algoritmo do Envelhecimento: Uma forma de LRU que usa um contador de idade para cada página. O contador é deslocado periodicamente e as páginas com contadores mais antigos são substituídas.
- Páginas Modificadas (Dirty Pages): Substitui páginas que foram modificadas e precisam ser escritas de volta ao disco. O algoritmo considera se uma página é "suja" (modificada) ao decidir qual substituir.
- **FILO (First In, Last Out)**: É um conceito que se refere a uma estrutura de dados do tipo pilha, onde o último item adicionado é o primeiro a ser removido. Não comumente utilizado substituição de páginas, o FILO pode ser aplicado teoricamente em cenários de gerenciamento de memória onde a última página carregada é a primeira a ser substituída. A abordagem é simples e fácil de implementar, mas pode ser ineficiente, pois não considera padrões de uso de páginas que poderiam ser mais relevantes para o processo, levando a uma alta taxa de faltas de página em ambientes dinâmicos.
- Algoritmos como **LRU (Least Recently Used)** e **FIFO (First In, First Out)** são mais frequentemente preferidos devido à sua eficácia em gerenciar a memória.

## Fragmentação de Memória

- Fragmentação Interna: (bagageiro do busão) Ocorre quando um processo não usa todo o bloco de memória que lhe foi alocado, resultando em desperdício dentro do bloco. É quando sobra bytes dentro de uma página, tornando esses espaços inutilizáveis.
- Fragmentação Externa: (meias no roupeiro) Acontece quando a memória está fragmentada em blocos não contíguos. Mesmo que haja espaço livre suficiente, ele não está contíguo para alocar novos

processos. é quando existem páginas avulsas entre páginas ocupadas, assim acabando por, se necessitar ocupar mais de uma página de espaço, o processo é dividido entre páginas com endereços longes um do outro.

- **Alocador Buddy:** Técnica que aloca blocos de memória em potências de 2 para reduzir a fragmentação externa, mas pode causar fragmentação interna. Por exemplo, se um processo requisitar 258 KB, será alocado 512 KB.

## Cálculos Importantes

- **Tempo Médio de Acesso à Memória:** O tempo médio de acesso é influenciado pelo tempo de acesso à RAM e ao disco, e pela frequência de faltas de página. Quando há memória saturada, o tempo de acesso ao disco se torna mais relevante, pois o SO precisa retirar uma página para liberar espaço.
- **Número de Faltas de Página:** Algoritmos como LRU e FIFO podem ser usados para determinar o número de faltas de página com base em uma sequência de acessos.
- **Memória Virtual**
- **Endereço virtual:** 10 bits
  - Total de endereços virtuais =  $(2^{10} = 1024)$  endereços.
- **Tamanho da página:** 10 bytes
  - Número de páginas =  $(\frac{1024 \text{ bytes}}{10 \text{ bytes/página}} = 102.4)$  (arredondado para 102 páginas).
- **Memória Virtual Total** =  $(1024 \text{ bytes} = 10 \text{ KB})$ .
- **Memória Principal (RAM)**
- **Endereço físico:** 8 bits
  - Total de endereços físicos =  $(2^8 = 256)$  endereços.
- **Tamanho da página:** 10 bytes
- **Memória Principal Total** =  $(256 \text{ endereços} \times 10 \text{ bytes/página} = 2560 \text{ bytes} = 2.5 \text{ KB})$ .
- **Resultados**
- **Memória Virtual:** 10 KB
- **Memória Principal (RAM):** 2.5 KB

## Pontos Importantes para a Prova:

- Entenda a diferença entre segmentação, paginação e partições.
- Saiba como funciona a TLB e a MMU.
- Esteja familiarizado com algoritmos de substituição de páginas como FIFO, LRU e o algoritmo ótimo (OPT).
- Saiba calcular o tempo médio de acesso à memória em diferentes cenários de saturação.
- Estude as causas e o tratamento de faltas de página.