



Arquitetura de Computadores

Gerenciamento de memória

Quem sou eu



Júlio César Andrade

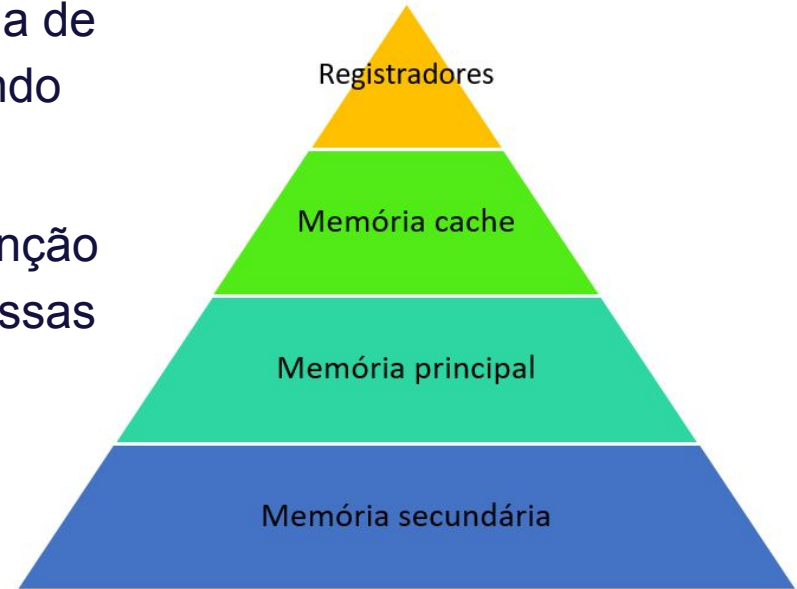
Bacharel em Engenharia de Computação - **UEFS**

Especialista em User Experience - **UNIFACS**

Mestrando em Ciências da Computação - **UEFS**

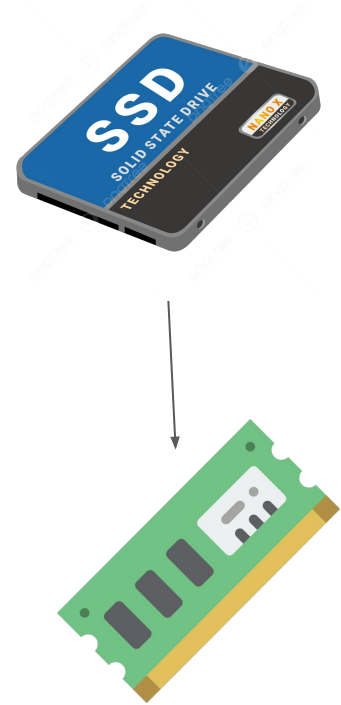
Introdução

- Os computadores utilizam uma hierarquia de memória em sua organização, combinando memórias voláteis e não-voláteis;
- Ao sistema operacional é destinada a função de coordenar e gerenciar a utilização dessas memórias de forma eficiente.



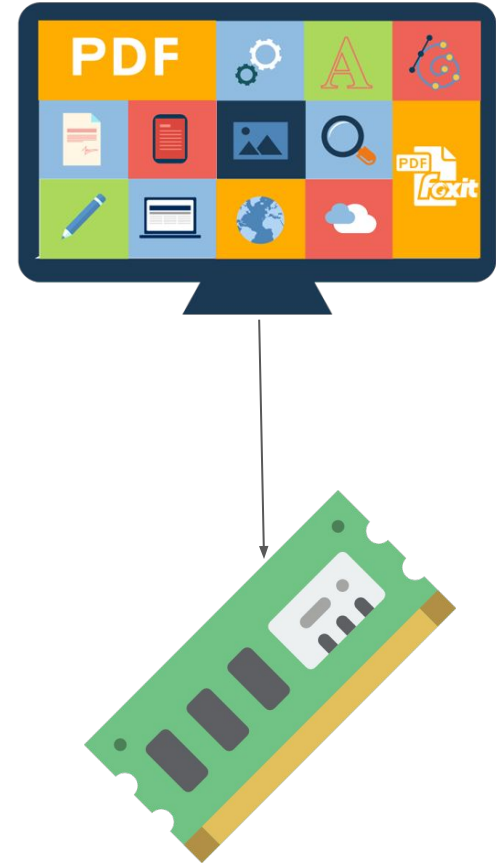
Introdução

- Na memória principal residem os programas em execução;
- Na memória secundária são armazenados os dados de forma permanente;
- Para um programa ser executado, ele deve ser carregado da memória secundária para a principal.



Introdução

Para a eficiência da multiprogramação é necessário que vários programas estejam na memória principal ao mesmo tempo.



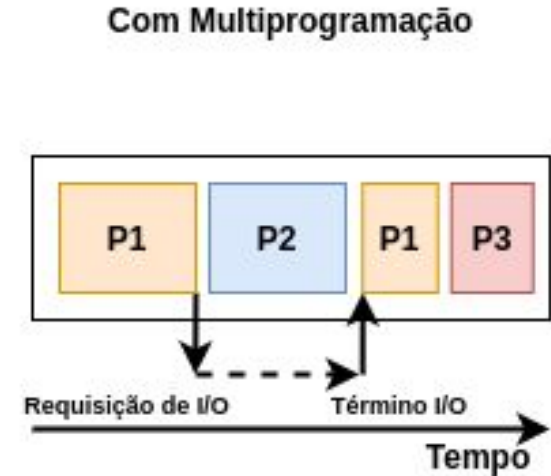
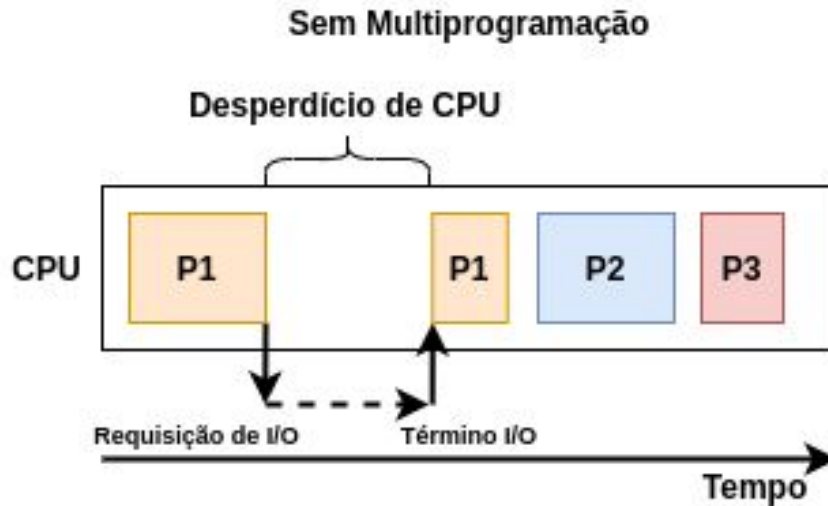


Dúvida comum

Mas, o que é multiprogramação mesmo?

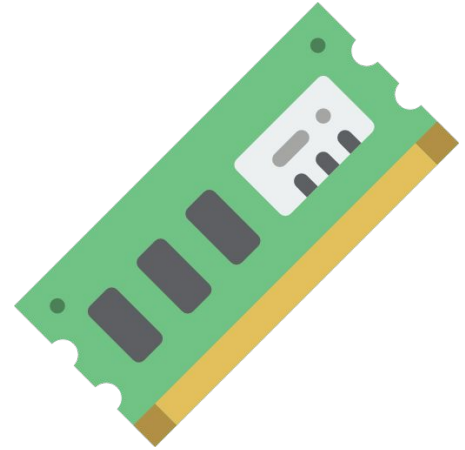
Multiprogramação

O conceito fundamental da multiprogramação é ter vários programas em execução simultaneamente, compartilhando os recursos do sistema.



Gerenciador de memória

O gerenciador de memória desempenha um papel crítico na eficiência e estabilidade de um sistema operacional, **garantindo que os recursos de memória sejam utilizados de maneira eficaz** e que os processos possam ser executados de forma coordenada e segura.



Funções do Gerenciador de memória

- a) Manter registro das partes da memória que estão sendo usadas no momento e por qual processo;
- b) Decidir quais processos deverão ser carregados na memória quando houver espaço disponível;
- c) Alocar e desalocar espaço na memória, conforme necessário;
- d) Tratar do problema do swapping (quando a memória é insuficiente).



Abstração da memória

A abstração de memória é uma parte fundamental do design de sistemas operacionais e é crucial para o funcionamento eficiente e seguro de um computador.



Vantagens da abstração de memória

Isolamento de Processos:

- Cada processo tem seu próprio espaço de endereçamento.
- Evita interferência e corrupção entre processos.

Gestão Eficiente de Recursos:

- Otimização na alocação e desalocação de recursos de memória.



Vantagens da abstração de memória

Facilidade de Programação:

- Programadores lidam com endereços virtuais, enquanto o sistema opera em endereços físicos.
- Simplifica a codificação, abstraindo detalhes de localização física na memória.

Proteção contra Acesso Indevido:

- Atribuição de permissões (leitura, gravação, execução) para regiões de memória.
- Evita operações não autorizadas.



Endereçamento em SO

No contexto de sistemas operacionais, o endereçamento pode ser dividido em dois tipos principais: **endereçamento lógico e endereçamento físico.**

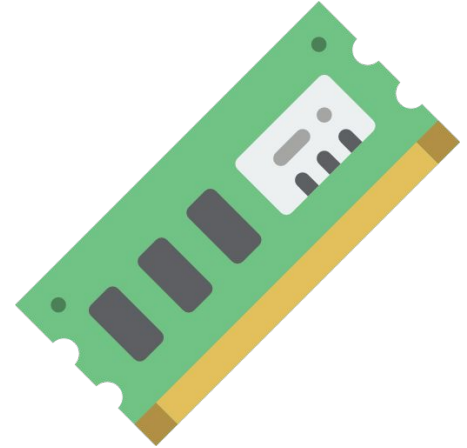
Endereçamento Lógico

- Os programas utilizam endereços lógicos ao acessar a memória, sem conhecer a localização real na memória física.
- Oferece uma visão abstrata e isolada da memória para cada processo.
- Facilita a implementação da memória virtual.



Endereçamento Físico

- Refere-se aos endereços reais na memória física do sistema.
- O hardware traduz endereços lógicos para endereços físicos;
- Envolve a localização real dos dados na RAM.
- É gerenciado pelo sistema operacional para garantir a integridade e proteção dos dados.





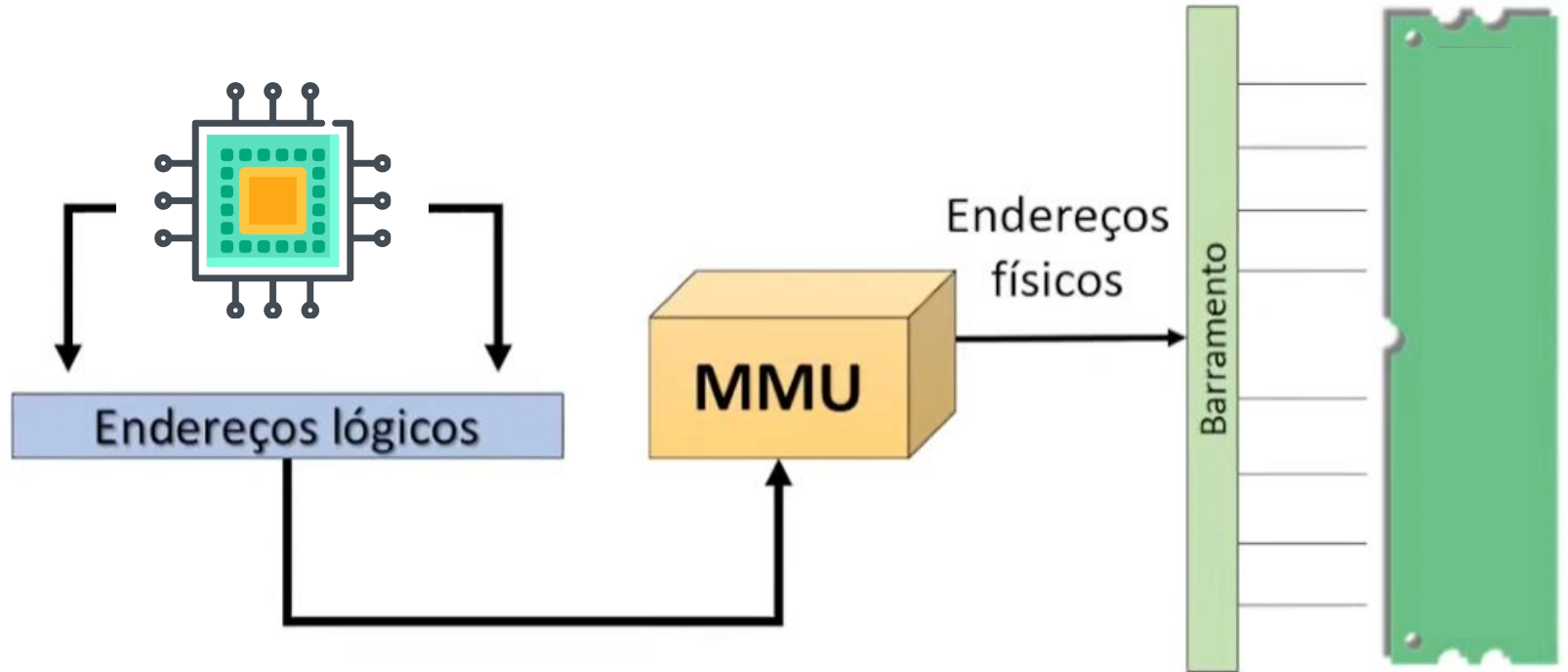
Dúvida

“Hardware traduz endereços lógicos para endereços físicos”?
Qual hardware?

Unidade de Gerência de Memória

(Memory Management Unit (MMU))

A MMU é um módulo de hardware que faz o mapeamento entre os endereços lógicos (end. da memória virtual) e os endereços físicos da memória (RAM), ou seja, é um dispositivo que transforma endereços virtuais em endereços físicos.



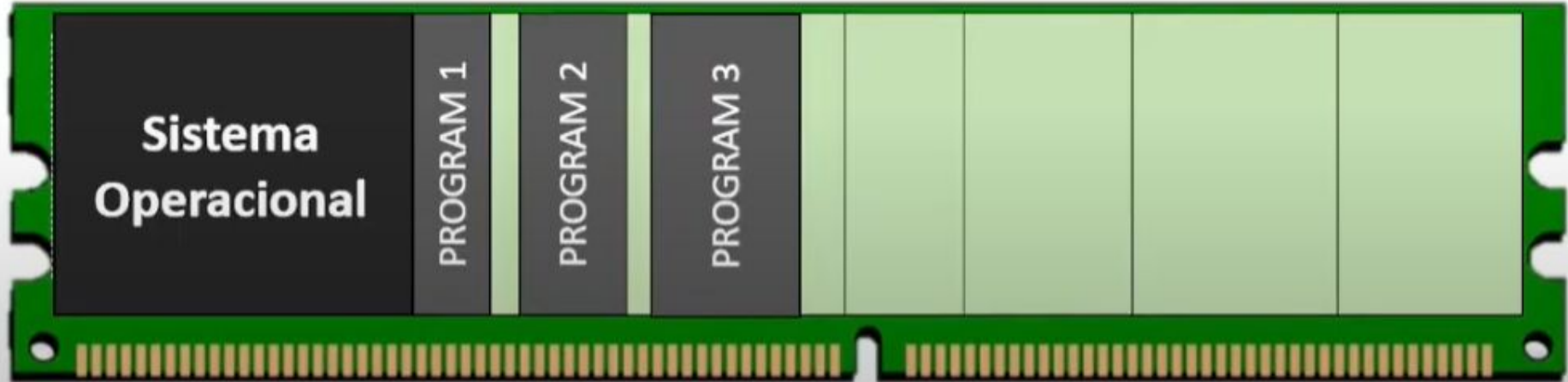
Alocação de memória

A alocação de memória em um sistema operacional refere-se ao processo de reservar e gerenciar espaços de memória para diferentes partes do sistema, como programas em execução.



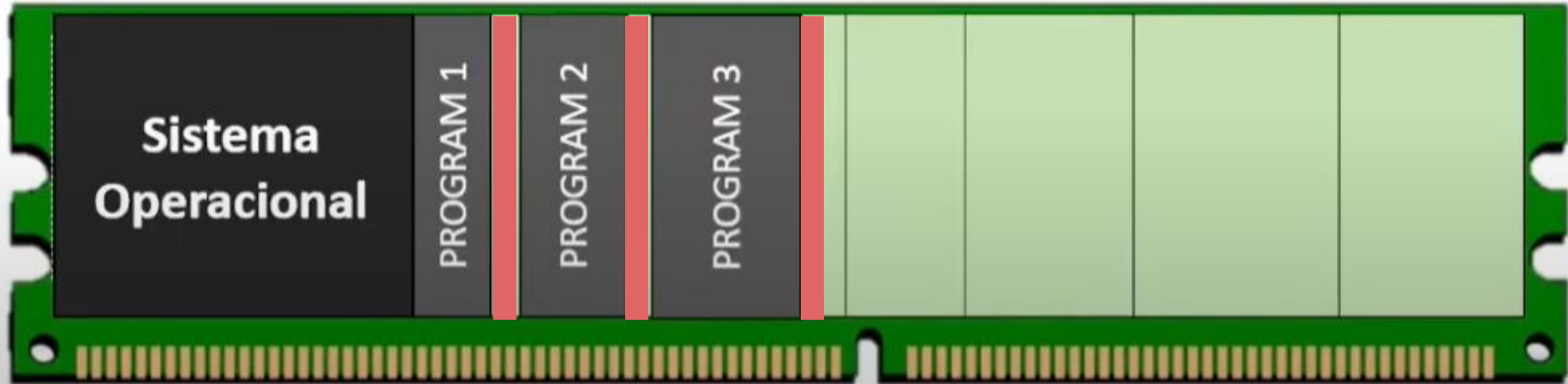
Alocação estática

Alocação estática: Divisão da memória em n partições fixas de tamanhos diferentes. Os jobs são alocados na menor partição capaz de armazená-los.



Fragmentação interna

Pelo fato de usarmos partições de tamanho fixo, todo o restante de **espaço de memória não utilizado pelo job será perdido**. Este desperdício de memória é chamado de **fragmentação interna**.



Fragmentação externa

Imagine que exista duas partições livres, uma de 25 e outra de 100 Kbytes, não contíguas. Nesse instante é criado um processo de 110 Kbytes que não poderá ser carregado em memória pela forma como ela é gerenciada.



Alocação dinâmica

Neste esquema de organização, o tamanho das partições é ajustado dinamicamente às necessidades exatas dos processos.

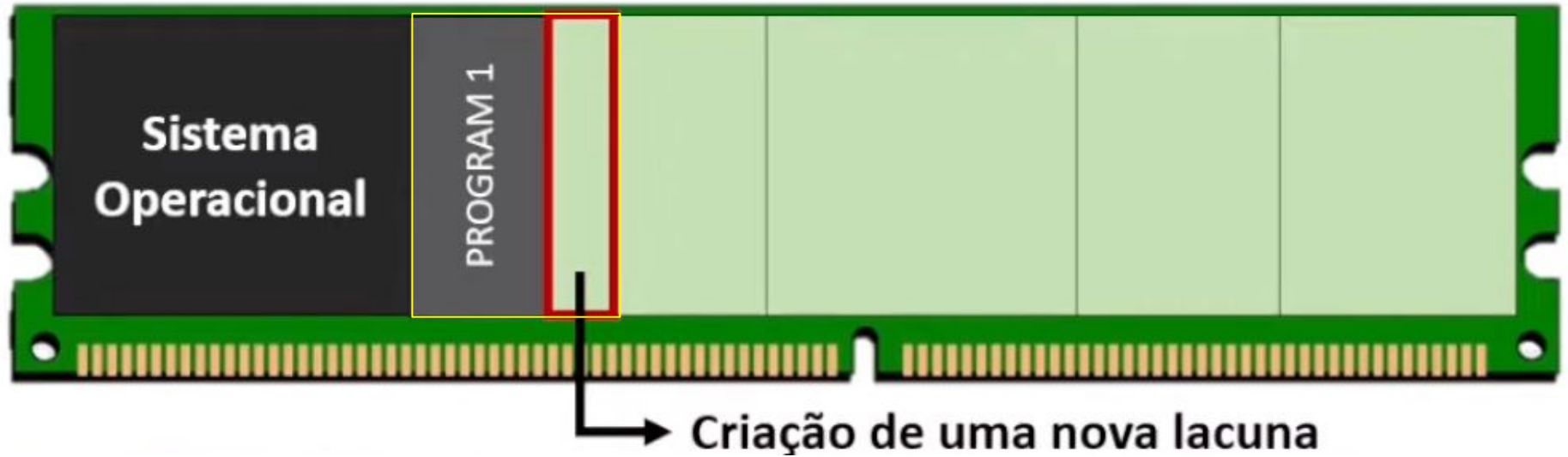


Alocação dinâmica

Como funciona?

1. Os espaços livres na memória são armazenados em uma lista de lacunas;
2. Quando um processo é criado, a lista é percorrida em busca de uma lacuna de tamanho maior ou igual;
3. Se a lacuna for maior, é criada uma nova lacuna com a porção que sobra;
4. O processo recebe o tamanho exato que precisa.

Alocação dinâmica



Alocação dinâmica

Métodos para percorrer a lista de lacunas:

First Fit

Funcionamento: Aloca o primeiro bloco de memória que é grande o suficiente para atender às necessidades do processo.

Vantagens: Simplicidade e eficiência razoável.

Desvantagens: Pode levar a fragmentação, pois espaços menores podem ser ocupados primeiro, deixando lacunas maiores.

Alocação dinâmica

Métodos para percorrer a lista de lacunas:

Best Fit

Funcionamento: Aloca o bloco de memória mais próximo do tamanho solicitado pelo processo.

Vantagens: Tenta minimizar o desperdício de memória ao escolher o bloco mais próximo do tamanho necessário.

Desvantagens: Pode ser menos eficiente, pois tem que percorrer toda a lista.

Alocação dinâmica

Métodos para percorrer a lista de lacunas:

Worst Fit (Ajuste Pior):

Funcionamento: Aloca o maior bloco de memória disponível, mesmo que seja muito maior do que o necessário.

Vantagens: Útil quando temos processos que consomem mais memória com o tempo.

Desvantagens: Pode resultar em um desperdício significativo de memória, especialmente se os blocos disponíveis forem muito maiores que o necessário. Tempo de busca elevado.

Alocação dinâmica

Métodos para percorrer a lista de lacunas:

Circular Fit:

Funcionamento: Similar ao First Fit, mas começa a procurar a lacuna seguinte a última sobra.

Vantagens: Pode ajudar a evitar a fragmentação se houver muitos espaços pequenos no início do espaço de memória.

Desvantagens: Pode introduzir complexidade e potencialmente levar a uma busca mais longa. Desconsidera espaços anteriores.

Sobrecarga de Memória

Manter todos os processos na memória constantemente requer grande quantidade de memória.

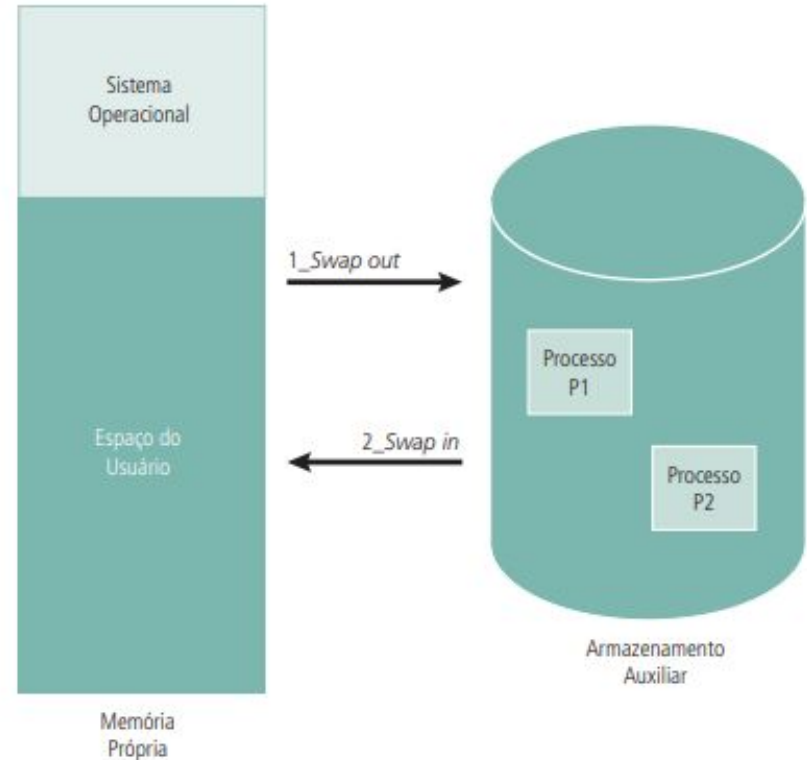
Impraticável na maioria dos casos.

Duas estratégias são comuns:

- Swapping
- Memória Virtual

Swapping (Troca de Processos):

- Processos são trazidos para a memória, executados e colocados de volta no disco.
- Processos ociosos ocupam principalmente espaço em disco, minimizando uso de memória quando não em execução.



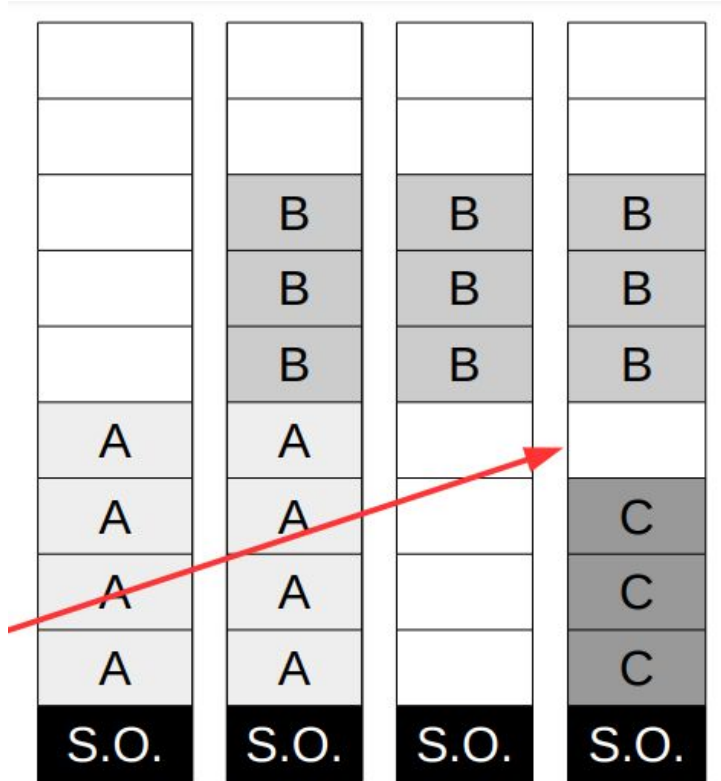
Swapping (Troca de Processos):

1. Carregue todos os processos que couberem na memória;
2. Se faltar espaço retire um processo da memória principal (RAM) e copie seu conteúdo na memória secundária(HD/SSD) (**SWAP OUT**)
3. Quando o processo for executar traga-o de volta da memória secundária(HD/SSD) para a primária(RAM) (**SWAP IN**)

O problema da fragmentação

Depois de várias operações de swap-in e swap-out, a memória pode se tornar fragmentada.

Desperdício



Memória virtual

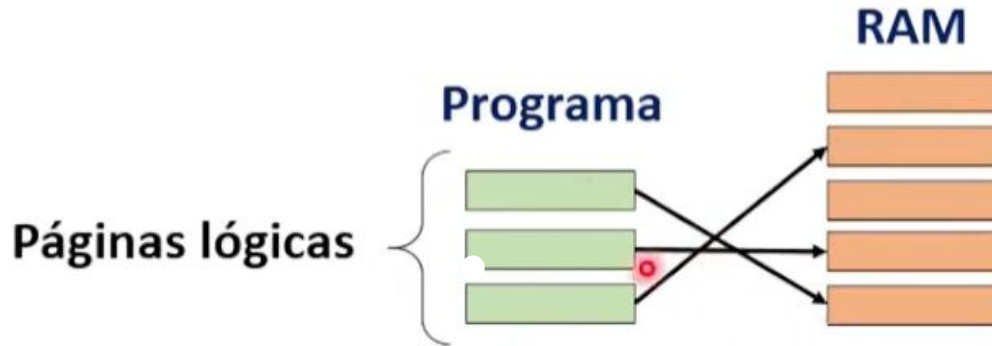
- Permite a execução de programas mesmo parcialmente na memória principal.
- Programas podem ser executados mesmo que não estejam completamente carregados na memória.

Memória virtual

- Fornece espaço de endereçamento maior que a RAM física.
- Espaço de endereçamento do programa é dividido em páginas.
- Não é necessário carregar todas as páginas na memória física de uma vez.
- Páginas são trazidas conforme necessário durante a execução do programa.

Memória virtual: Página

- Unidade de alocação de memória na implementação da memória virtual;
- Cada programa tem seu espaço de endereçamento dividido em páginas;
- Páginas são blocos lógicos;
- Páginas têm tamanhos fixos.



Paginação

Divisão em Blocos Fixos:

A memória é dividida em blocos de tamanho fixo chamados páginas. Da perspectiva do processo, ele enxerga apenas páginas e não tem conhecimento da estrutura física da memória.

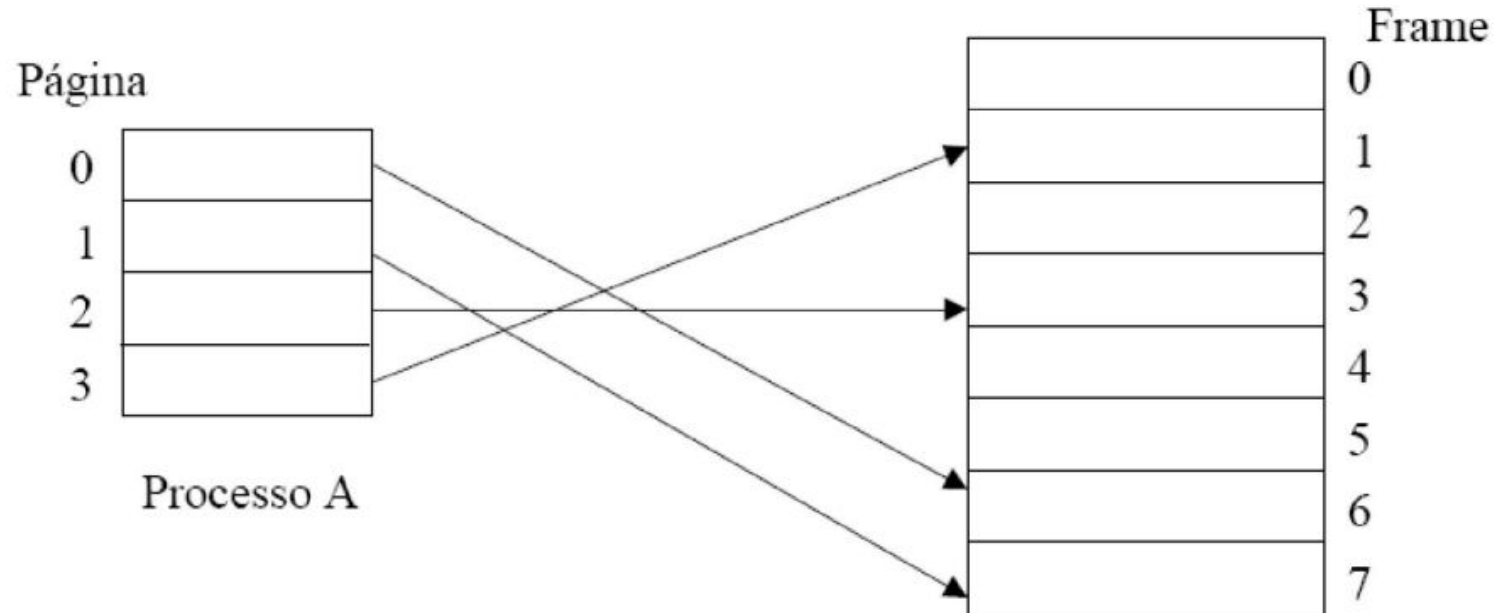
Divisão Lógica e Física:

O espaço de endereçamento lógico do processo é dividido em páginas, e a memória física é dividida em quadros (frames) de tamanho correspondente às páginas.

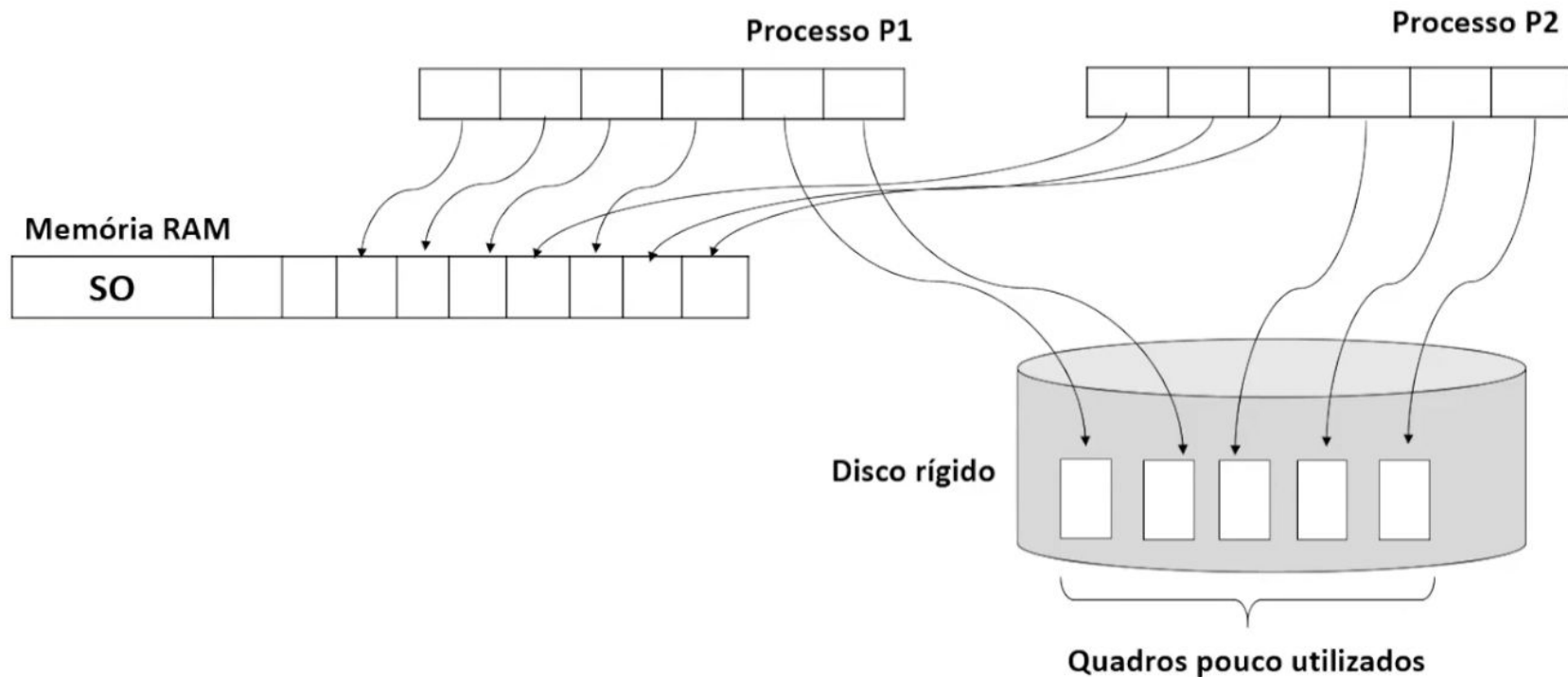
Eliminação de Fragmentação Externa:

Elimina a fragmentação externa, pois as páginas podem ser alocadas de forma não contígua na memória física.

Paginação



Como funciona a paginação



Algoritmos de paginação

O melhor algoritmo de substituição de página é aquele em que:

A página retirada da memória para o disco seja a página que está mais longe de ser usada novamente.

Valendo 10 pontos na prova!



Como poderíamos implementar
esse algoritmo perfeito?

Não dá!

Infelizmente, ainda não podemos prever o futuro.

Portanto, este algoritmo como descrito, não pode ser implementado na prática.



Algoritmos de paginação

FIFO (First-In-First-Out):

Este algoritmo substitui a página que está na memória há mais tempo. Funciona como uma fila, onde a primeira página a entrar é a primeira a sair.

LRU (Least Recently Used):

Substitui a página que não foi utilizada por mais tempo. Mantém um registro do tempo em que cada página foi referenciada pela última vez.

LFU (Least Frequently Used):

Substitui a página que foi referenciada menos frequentemente. Mantém um contador para cada página, incrementando-o sempre que a página é referenciada.

Algoritmos de paginação

Clock (Relógio):

Usa uma abordagem semelhante ao ponteiro de um relógio. As páginas são organizadas em um círculo, e um ponteiro avança. Quando uma página precisa ser substituída, o ponteiro encontra a próxima página a ser removida.

NRU (Not Recently Used):

Classifica as páginas em quatro categorias (não referenciada recentemente e não modificada, não referenciada recentemente e modificada, referenciada recentemente e não modificada, referenciada recentemente e modificada). Substitui uma página de uma das categorias menos prioritárias.

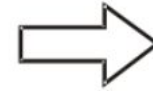
Segmentação

- Visa organizar e gerenciar a memória do sistema de uma maneira mais flexível;
- Em vez de dividir a memória em blocos de tamanho fixo, como na abordagem de paginação, a segmentação divide a memória em segmentos de diferentes tamanhos.

Segmentação

Na segmentação existe uma relação entre a lógica do programa e sua alocação na memória principal

```
PROGRAM Segmento;  
  VAR A: ARRAY...  
  C: ...  
PROCEDURE X;  
END;  
FUNCTION Y;  
END;  
BEGIN  
END.
```

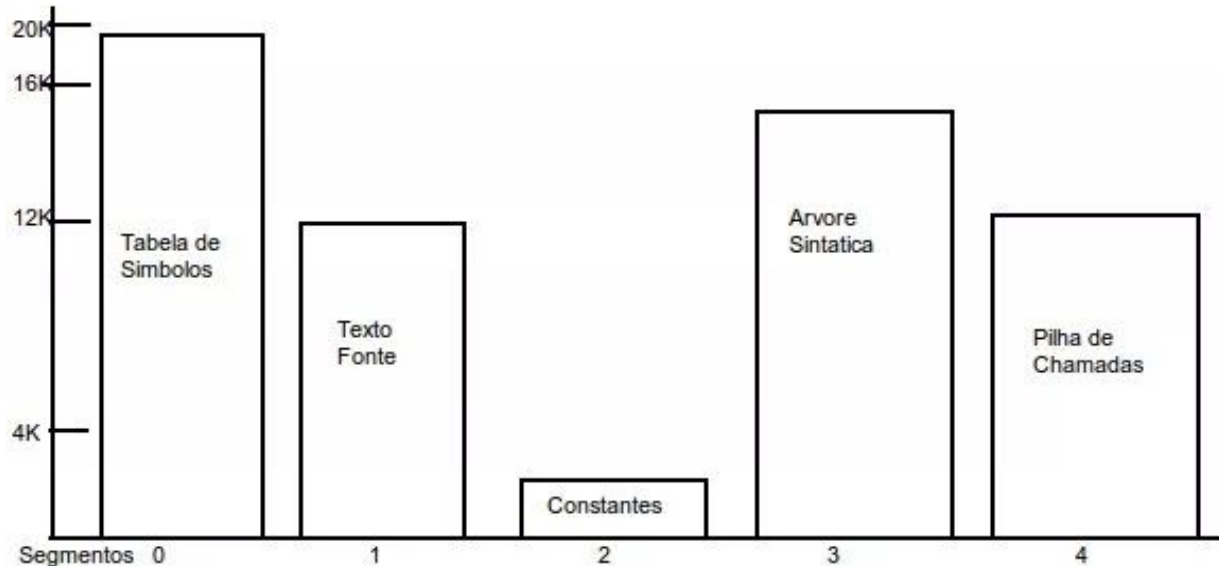


Normalmente, a definição dos segmentos é realizada pelo compilador, a partir do código fonte do programa, e cada segmento pode representar um procedimento, função, vetor ou pilha.



Segmentação x Paginação

A principal diferença entre a paginação e a segmentação é a alocação da memória de maneira não fixa, a alocação depende da lógica do programa.





Vantagens da segmentação

- Sem fragmentação interna.
- A Tabela de segmentos consome menos espaço em comparação à tabela de páginas na paginação.



Desvantagem da segmentação

- Conforme os processos são carregados e removidos da memória, o espaço livre da memória é quebrado em pequenos pedaços, causando fragmentação externa.

Gravou tudinho?



Bibliografia

SILBERSCHATZ, A. & GAGNE, G. & GALVIN, P. B. **Fundamentos de Sistemas Operacionais**. Rio de Janeiro, 2004.

TANENBAUM, A.S. **Sistemas Operacionais Modernos**. 2ª.ed. São Paulo, 2009.

