



# GERENCIAMENTO DE MEMÓRIA

Sistemas Operacionais

Análise e Desenvolvimento de Sistemas

# GERENCIAMENTO DE MEMÓRIA

Um sistema de memória possui pelo menos dois níveis:

- o Memória principal: acessada pela CPU
- o Memória secundária: discos

# GERENCIAMENTO DE MEMÓRIA

- Programas são armazenados em disco
  - o Executar um programa se traduz em transferi-lo da memória secundária à memória primária

# GERENCIAMENTO DE MEMÓRIA

- Qualquer sistema operacional tem gerência de memória
- Monotarefa: gerência é simples
- Multitarefa: complexa
- Algoritmos de gerência de memória dependem de facilidades disponíveis pelo hardware da máquina

# GERENCIAMENTO DE MEMÓRIA

## Memória lógica

- o É aquela que o processo “enxerga”
- o Endereços lógicos são aqueles manipulados por um processo

# GERENCIAMENTO DE MEMÓRIA

## Memória física

- Implementada pelos circuitos integrados de memória.
- Endereços físicos são aqueles que correspondem a uma posição real de memória.

# GERENCIAMENTO DE MEMÓRIA

Espaço lógico de um processo é diferente do espaço físico.

- Endereço lógico: gerado pela CPU (endereço virtual)
- Endereço físico: endereços enviados para a memória RAM

Programas de usuários “vêem” apenas endereços lógicos.

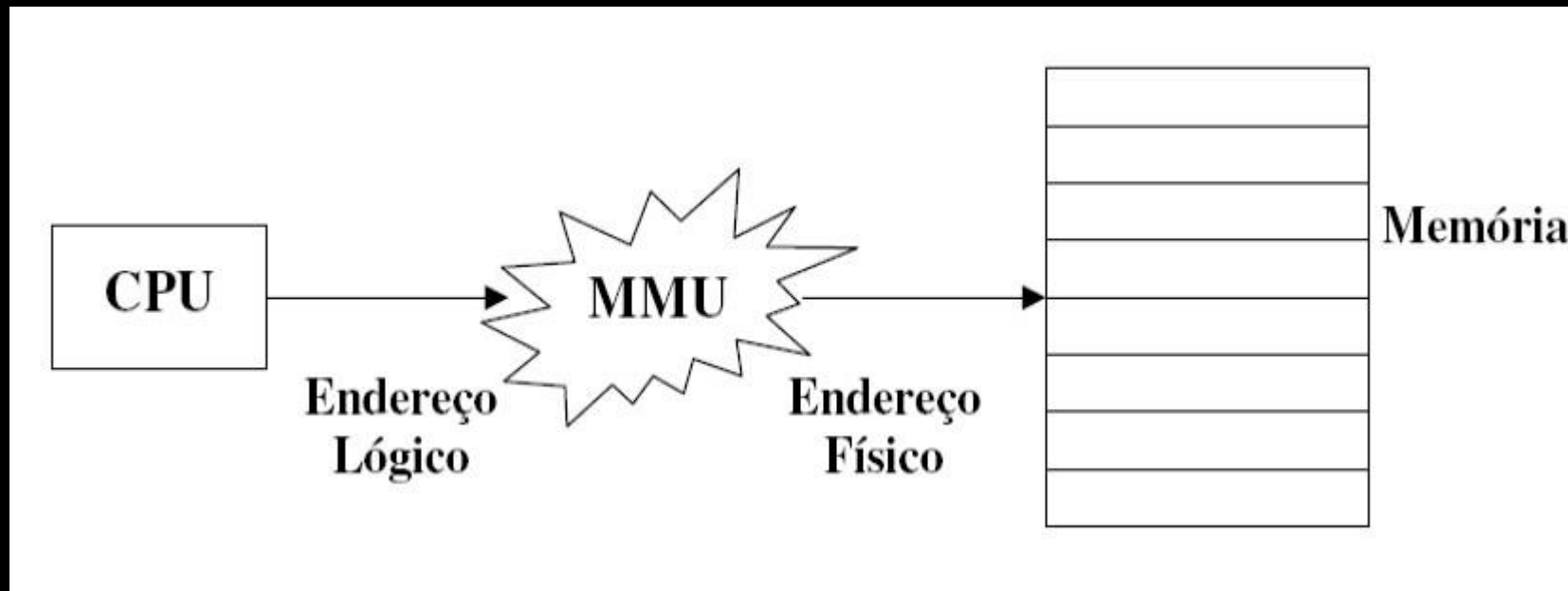
Endereços lógicos são transformados em endereços físicos no momento de execução dos processos.

# UNIDADE DE GERENCIA DE MEMÓRIA

- *Memory Management Unit (MMU)*
- Hardware que faz o mapeamento entre endereço lógico e endereço físico



# UNIDADE DE GERENCIA DE MEMÓRIA



# GERENCIAMENTO DE MEMÓRIA

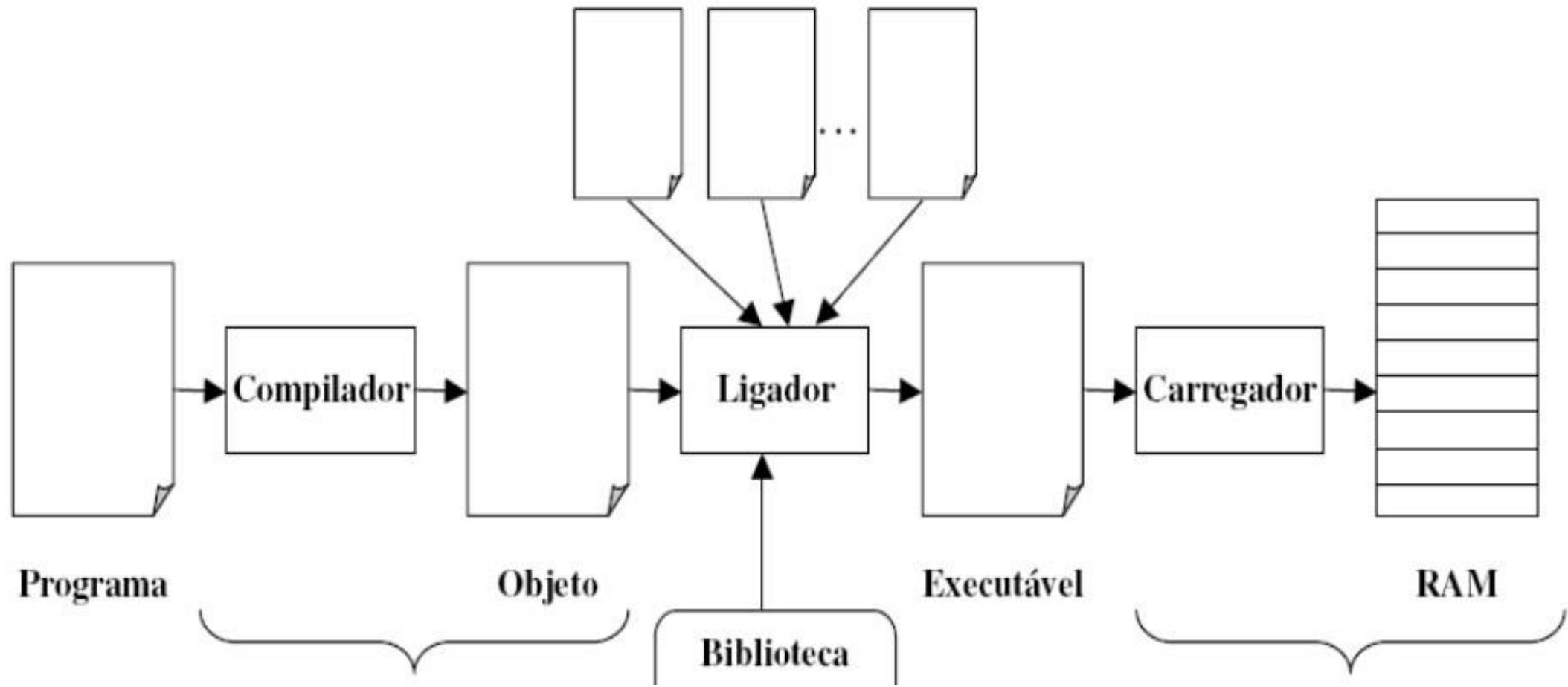
Um programa deve ser transformado em um processo para poder ser executado.

- Alocação de um descritor de processos
- Alocação de áreas de memória para código, dados e pilha

Transformação é feita através de uma série de passos, alguns com a ajuda do próprio programados.

- Compilação, diretivas de compilação e/ou montagem, ligação, etc...
- Amarração de endereços (binding)

## Transformação de programa em processo



# GERENCIAMENTO DE MEMÓRIA

## Amarração de endereços (binding)

Atribuição de endereços para instruções e dados pode ser feita em três momentos diferentes:

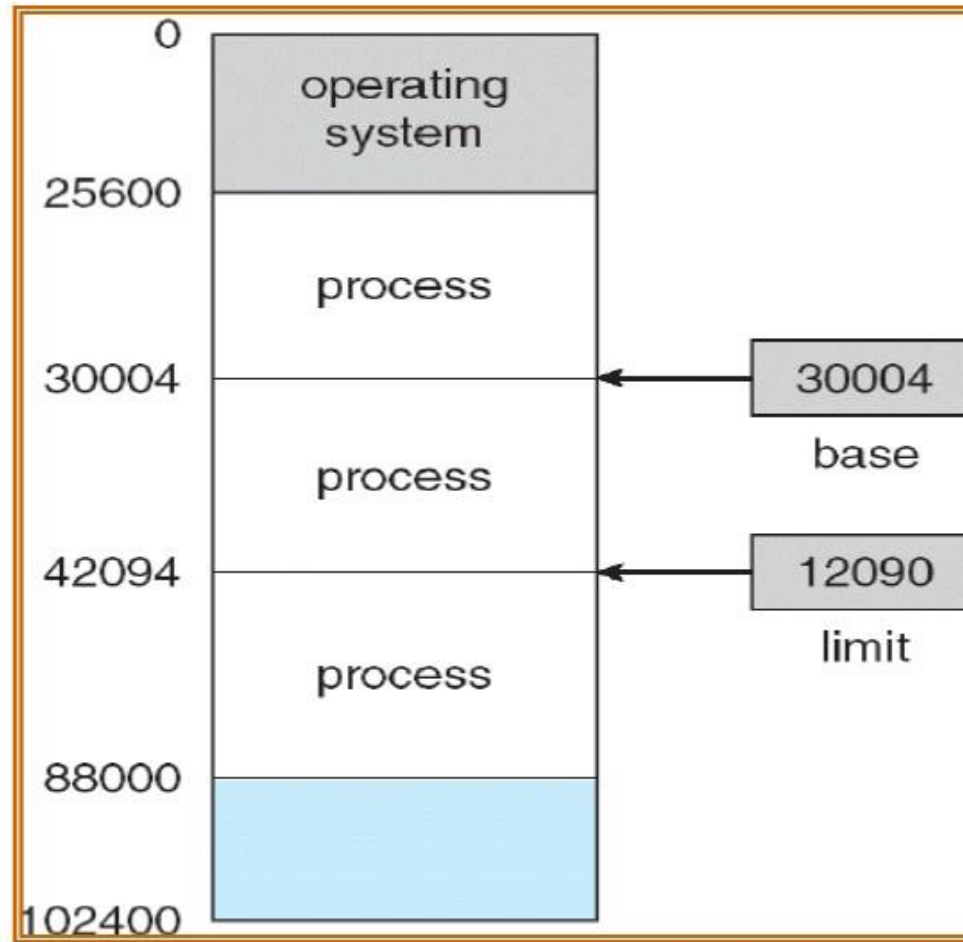
- **Em tempo de compilação:** se a localização da memória é conhecida a priori, código absoluto pode ser gerado; tem que recompilar o código se a alocação inicial mudar
- **Em tempo de carga:** deve gerar código relocável se a localização da memória não é conhecida em tempo de compilação
- **Em tempo de execução:** a atribuição é adiada até o tempo de execução se o processo puder ser movido durante sua execução de um segmento de memória para outro. Precisa de suporte de hardware para mapear endereços (ex.: registradores base e limite)

# GERENCIAMENTO DE MEMÓRIA

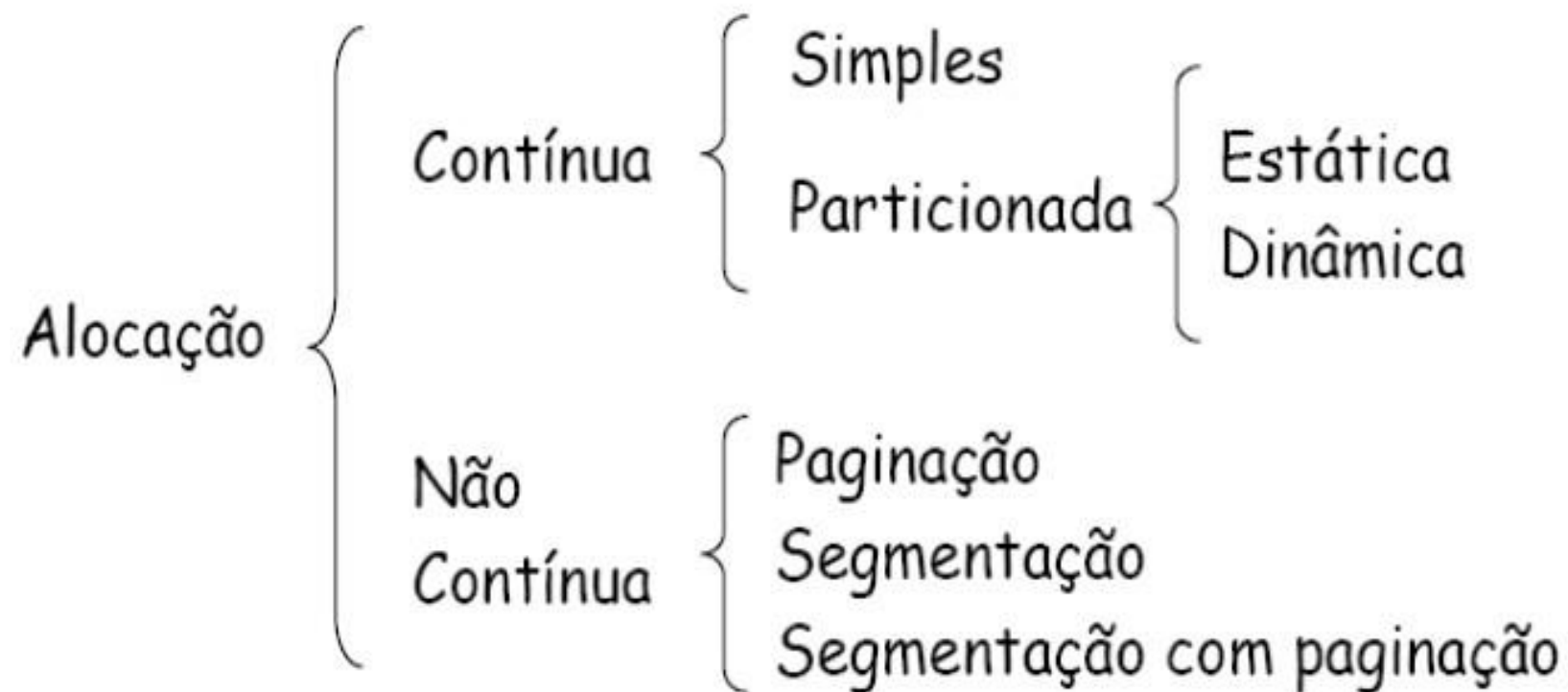
**Registradores de Base e de Limite.**

**Um par de registradores de base e de limite  
definem o espaço de endereço lógico**

endereço lógico



## Mecanismos para alocação de memória



# ALOCAÇÃO CONTÍGUA SIMPLES

- Sistema mais simples
- Memória principal é dividida em duas partições: o Sistema operacional (parte baixa de memória) o Processo de usuário (restante da memória)
- Usuário tem controle total da memória podendo inclusive acessar a área do sistema operacional.  
Ex.: DOS (não confiável)



# ALOCAÇÃO CONTÍGUA PARTICIONADA

- Existência de múltiplas partições
  - Imposta pela multiprogramação
  - Filosofia:
    - o Dividir a memória em blocos (partições)
    - o Cada partição pode receber um processo (programa)
    - o Grau de multiprogramação é fornecido pelo número de partições
- \*\* Importante: não considerando a existência de swapping**

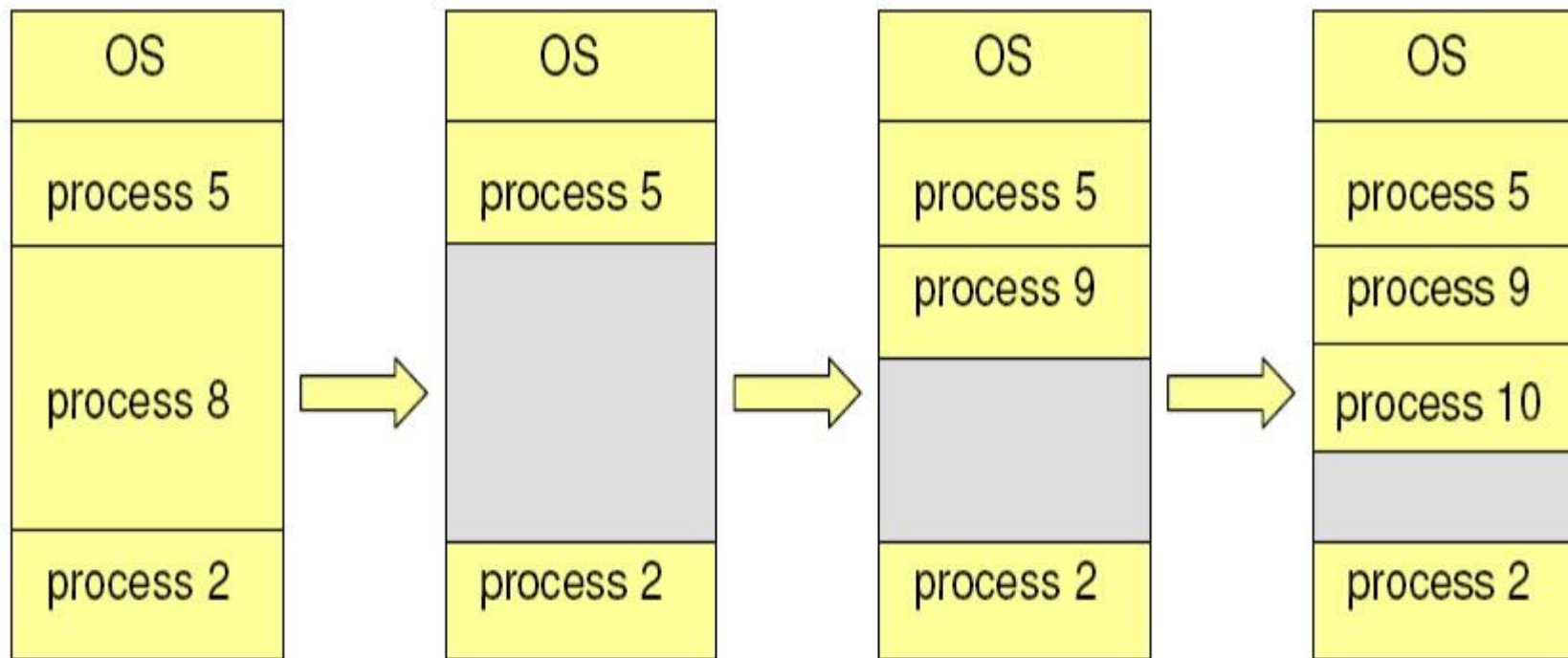
# ALOCAÇÃO CONTÍGUA PARTICIONADA

- O sistema operacional é responsável pelo controle das partições mantendo informações como:
  - o Partições alocadas
  - o Partições livres
  - o Tamanho das partições

# ALOCAÇÃO CONTÍGUA PARTICIONADA FIXA

- **Memória disponível é dividida em partições de tamanho fixo que podem ser do mesmo tamanho ou não**

# ALOCAÇÃO CONTÍGUA PARTICIONADA DINÂMICA



# ALOCAÇÃO PARTICIONADA DINÂMICA

## Fragmentação externa

**Exemplo:**

Criação processo 120K

SisOp	
Processo 1	320 K
Processo 4	128 K
	96 K
Processo 3	288 K
	64 K

# ALOCAÇÃO PARTICIONADA DINÂMICA

Soluções possíveis fragmentação externa

- Reunir espaços adjacentes de memória
- Empregar compactação
- Relocar as partições de forma a eliminar os espaços entre elas e criando uma área contígua

Desvantagem:

Consumo do processador

Acesso a disco

- Acionado somente quando ocorre fragmentação
- Necessidade de código relocável

# ALOCAÇÃO PARTICIONADA DINÂMICA

## Gerenciamento de partições dinâmicas

- Determinar qual área de memória livre será alocada a um processo
- Sistema operacional mantém uma lista de lacunas o Pedacos de espaços livres em memória
- Necessidade de percorrer a lista de lacunas sempre que um processo é criado

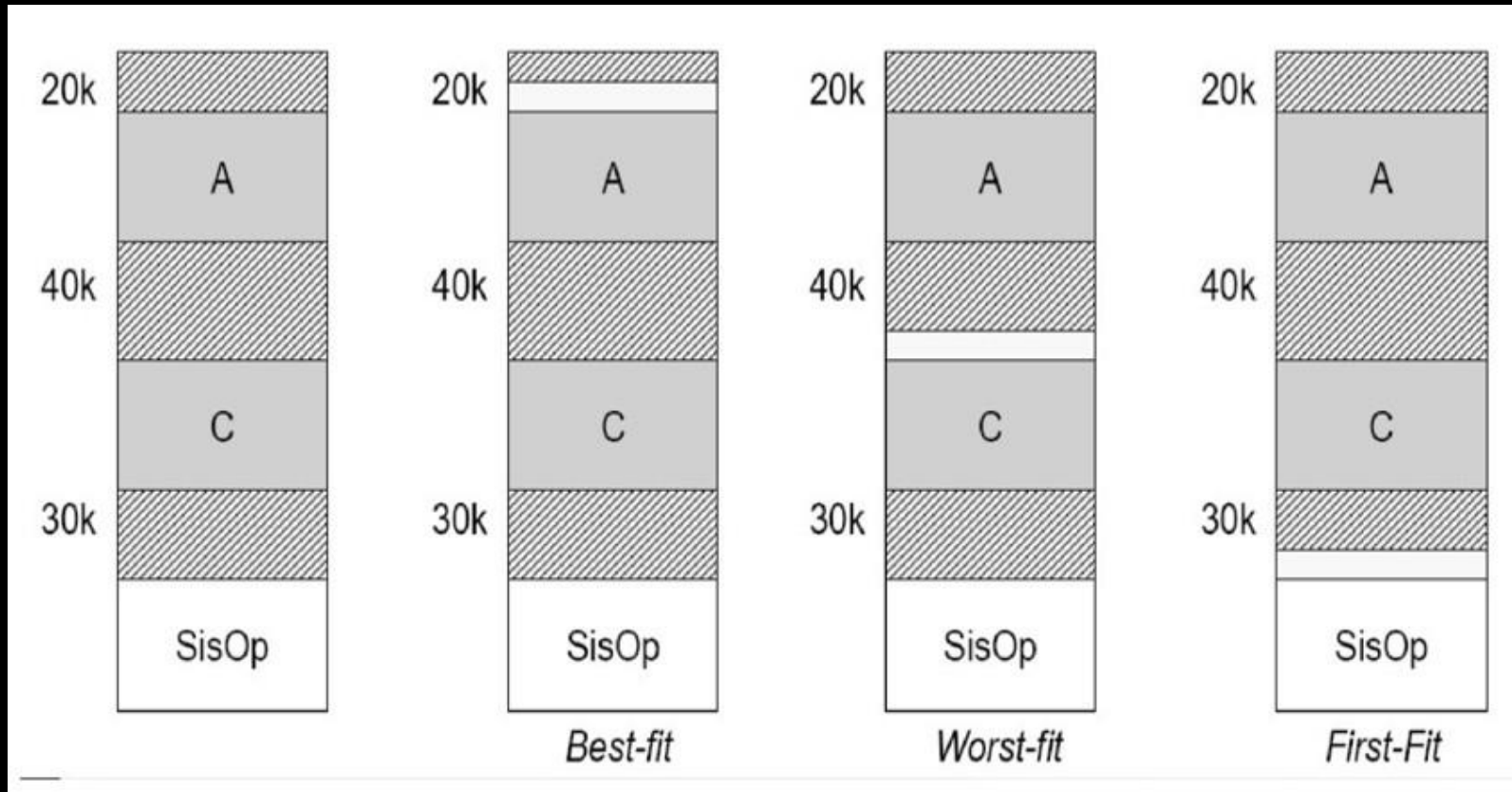
# ALOCAÇÃO PARTICIONADA DINÂMICA

## Algoritmos para alocação contígua dinâmica

- **Best fit**  
Minimizar  $\text{tam\_processo} - \text{tam\_bloco}$  o Deixar espaços livres os menores possíveis
- **Worst fit**  
Maximizar  $\text{tam\_processo} - \text{tam\_bloco}$  o Deixar espaços livres os maiores possíveis
- **First fit**  
 $\text{tam\_bloco} > \text{tam\_processo}$
- **Circular fit**  
Variação do first-fit



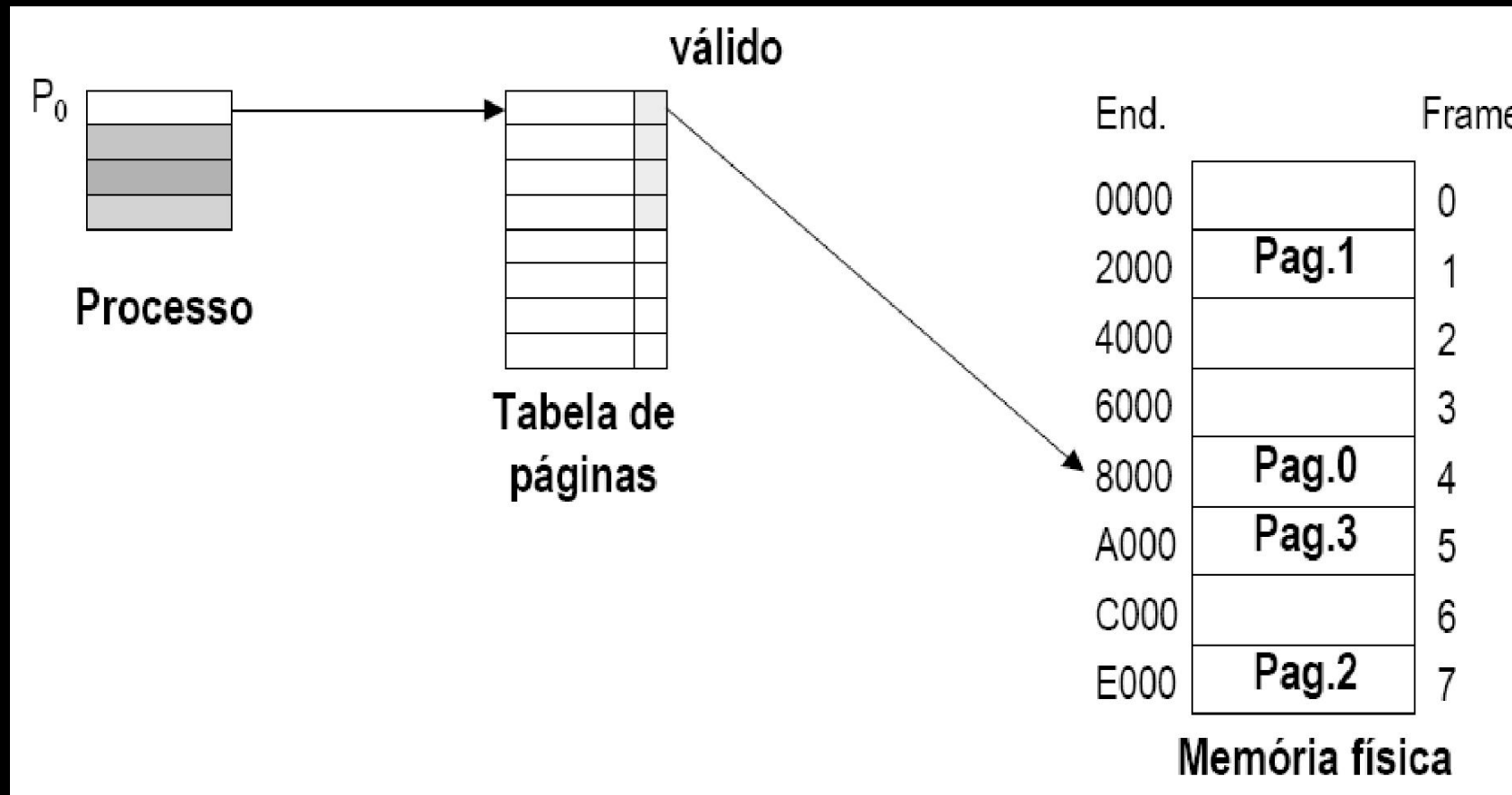
# ALOCAÇÃO PARTICIONADA DINÂMICA



# PAGINAÇÃO DE MEMÓRIA

- O Espaço de Endereçamento lógico de um processo pode ser não contínuo; aloca-se memória física ao processo sempre que esta é disponível.
- A memória física (sistema) e a memória lógica (processo) são divididos em blocos de tamanho fixo e idênticos:
  - o Memória física dividida em blocos de tamanho fixo denominados de frames
  - o Memória lógica divide em blocos de tamanho fixo denominados de páginas

# PAGINAÇÃO DE MEMÓRIA



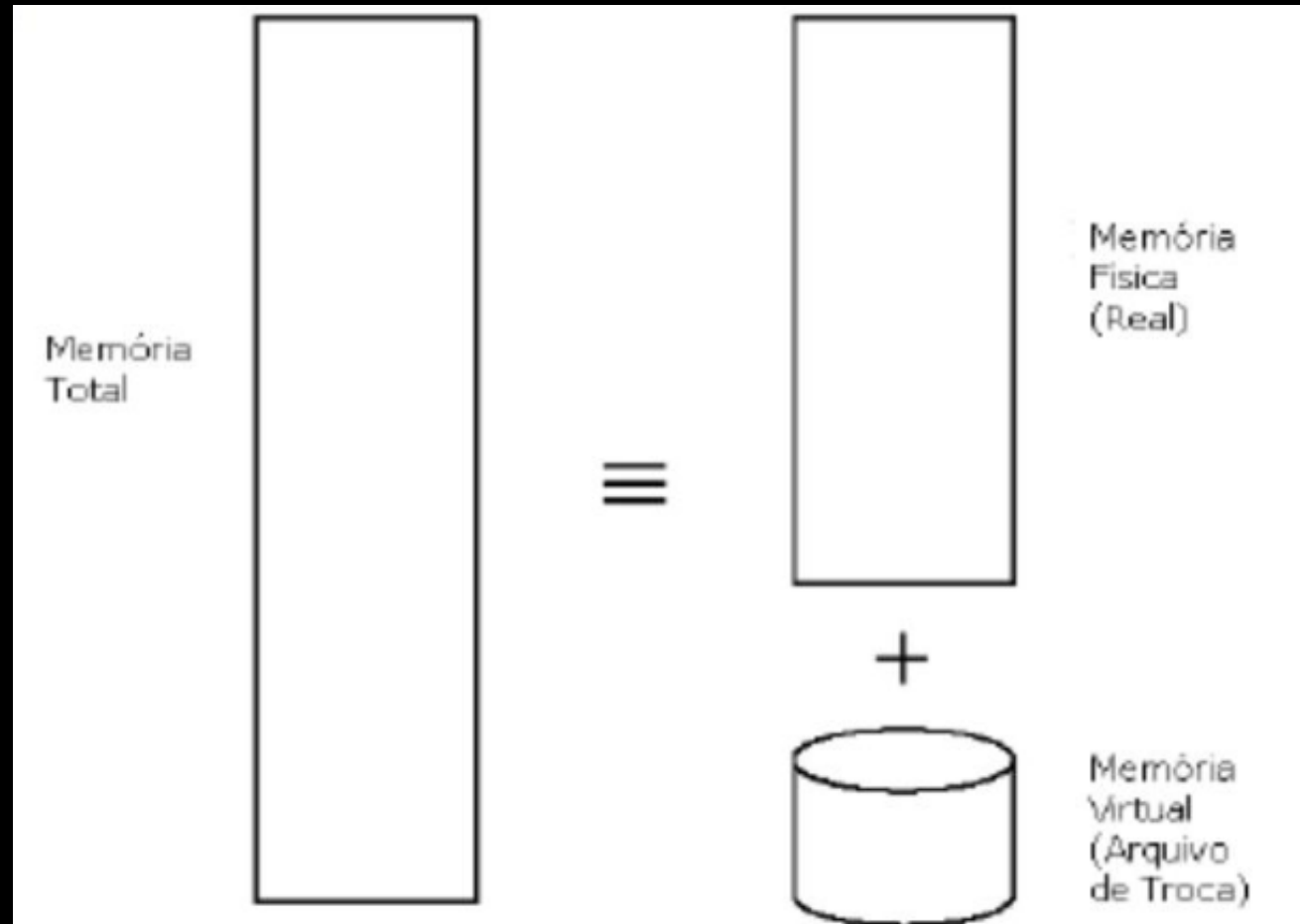
# SEGMENTAÇÃO DE MEMÓRIA

- Técnica de gerência de memória, onde os programas são divididos logicamente e em sub-rotinas e estruturas de dados e colocados em blocos de informações na memória
- Segmentos – blocos de tamanhos diferentes com seu próprio espaço de endereçamento.
- Respeita a visão do programador.

# MEMÓRIA VIRTUAL

- Desvincula o endereçamento feito pelos programas aos endereços físicos de memória
- Programas não limitados ao tamanho total da memória física
- Permite ao SO endereçar muito mais memória do que a instalada

# MEMÓRIA VIRTUAL





# MEMÓRIA VIRTUAL

- A memória virtual de um SO é o seu arquivo de swap (troca) gravado no HD
- Memória total de um SO é a soma da memória física (fixa) com a virtual
- No windows a memória virtual (arquivo de paginação) pode ser definida como 1,5 x a quantidade de memória física

# SWAPPING

- Processo necessita estar na memória para ser executado
- Se não há mais espaço em memória é necessário fazer um rodízio de processos em memória
- Um processo pode ser temporariamente movido para fora da memória (backing store) e então trazido de volta para a memória para continuar a execução



# SWAPPING

sistema operacional mantém uma fila de processos que estão prontos para executar, que possuem imagens da memória em disco

