



Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Disciplina: Sistemas Operacionais I

Aula 17: Gerenciamento de Memória P1

Prof. Diogo Branquinho Ramos

diogo.branquinho@fatec.sp.gov.br

São José dos Campos - SP

Roteiro

- Conceitos básicos de memória principal
- Gerenciamento de memória
- Hierarquia de armazenamento
- Espaços de endereços lógicos e físicos
- Unidade de Gerenciamento de Memória
 - Registradores de base e de limite
- Estratégias de alocação de memória
- O problema da fragmentação

Conceitos básicos de Memória Principal

- **RAM – *Random Access Memory***
- **Última área de armazenamento (milhões a bilhões de bytes) que a CPU pode acessar diretamente.**
 - Registrador, *cache* e RAM.
- **A RAM é um conjunto de *words* de memória**
 - Cada *word* possui seu próprio endereço.
- **A interação com a CPU é obtida por meio de uma sequência de instruções para carregar ou armazenar bytes em endereços específicos.**
 - LOAD, MOVE, SAVE...

Conceitos básicos de Memória Principal

John von Neumann: 1903-1957, húngaro-americano. Doutor em matemática, gênio, contribuiu para a ciência da computação, mecânica quântica, teoria dos jogos, etc. Inventou o Mergesort em 1945. Propõe que o EDVAC seja executado sem modificações no hardware, através da interrupção e retomada de um programa com dados e instruções na mesma memória através de uma CPU e de outras unidades de execução. É o pai do computador moderno.



- **Contador de programa**
 - Indica a próxima instrução a ser apanhada da memória
 - A rotina e o dado já foram carregados na memória previamente.
- **Velocidade**
 - Acesso a registrador: em um *clock* de CPU (ou menos).
 - Acesso à RAM: pode tomar muitos ciclos.
- **Ciclo típico de execução (von Neumann)**
 - Apanha uma instrução e a coloca no registrador de instruções.
 - A instrução é decodificada, fazendo com que operandos sejam apanhados e armazenados em registradores internos.
 - Após a execução da instrução sobre os operandos, o resultado pode ser armazenado de volta na memória.

Execução de um processo

Perspectiva da memória

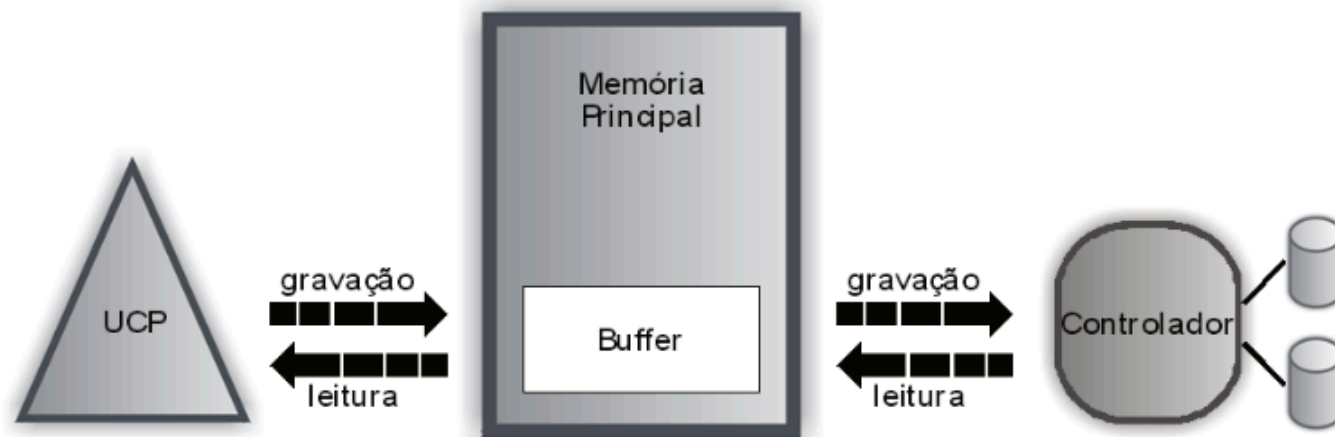
- O programa reside num disco como um arquivo executável binário (ou código relocável).
- Para ser executado precisa ser trazido para a memória e ser caracterizado como um processo
 - Lembre-se do fork () e exec().
 - Ao ser alocado na RAM, transforma-se num processo.
 - À medida que executa, acessa instruções e dados que estão na memória.
- Ao terminar, seu espaço na memória é declarado **disponível**.

Gerenciamento de memória

- **Provê melhor utilização do sistema**
 - Permite o carregamento de vários processos na memória
 - Agilidade no processamento.
 - Proteção da memória
 - Garante a operação correta: cada processo acessa apenas seu intervalo de endereços válidos.
- **Atividades de gerenciamento de memória**
 - Alocar e desalocar espaço de memória conforme a necessidade;
 - Acompanhar quais partes da memória estão sendo usadas atualmente e por quem (processo e usuário);
 - Decidir quais processos (ou partes deles) e dados mover para dentro e fora da memória.

Ações para o gerenciamento

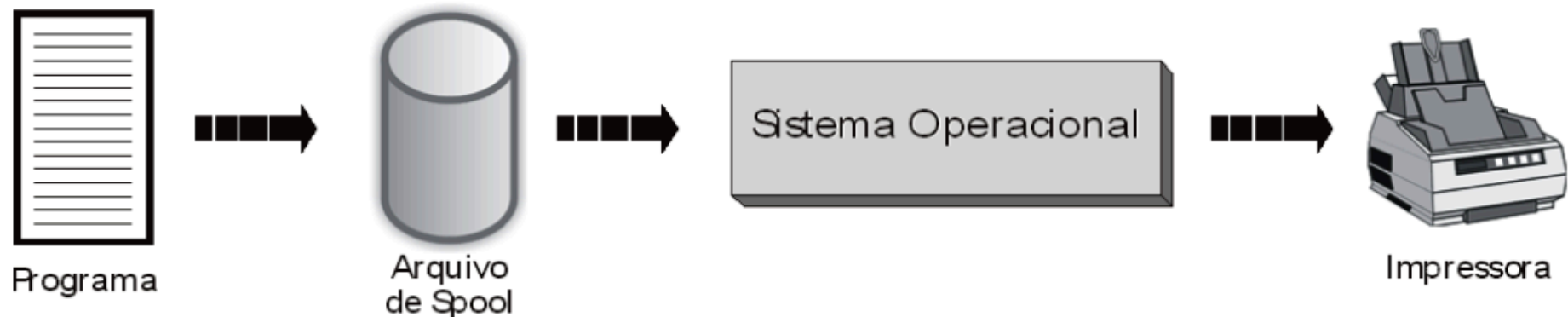
- **Buffering**
 - Armazenamento de dados temporariamente enquanto estão sendo transferidos.



Ações para o gerenciamento

- **Spooling**

- A sobreposição da saída de um job com a entrada de outros jobs.
- Libera a aplicação do hardware.



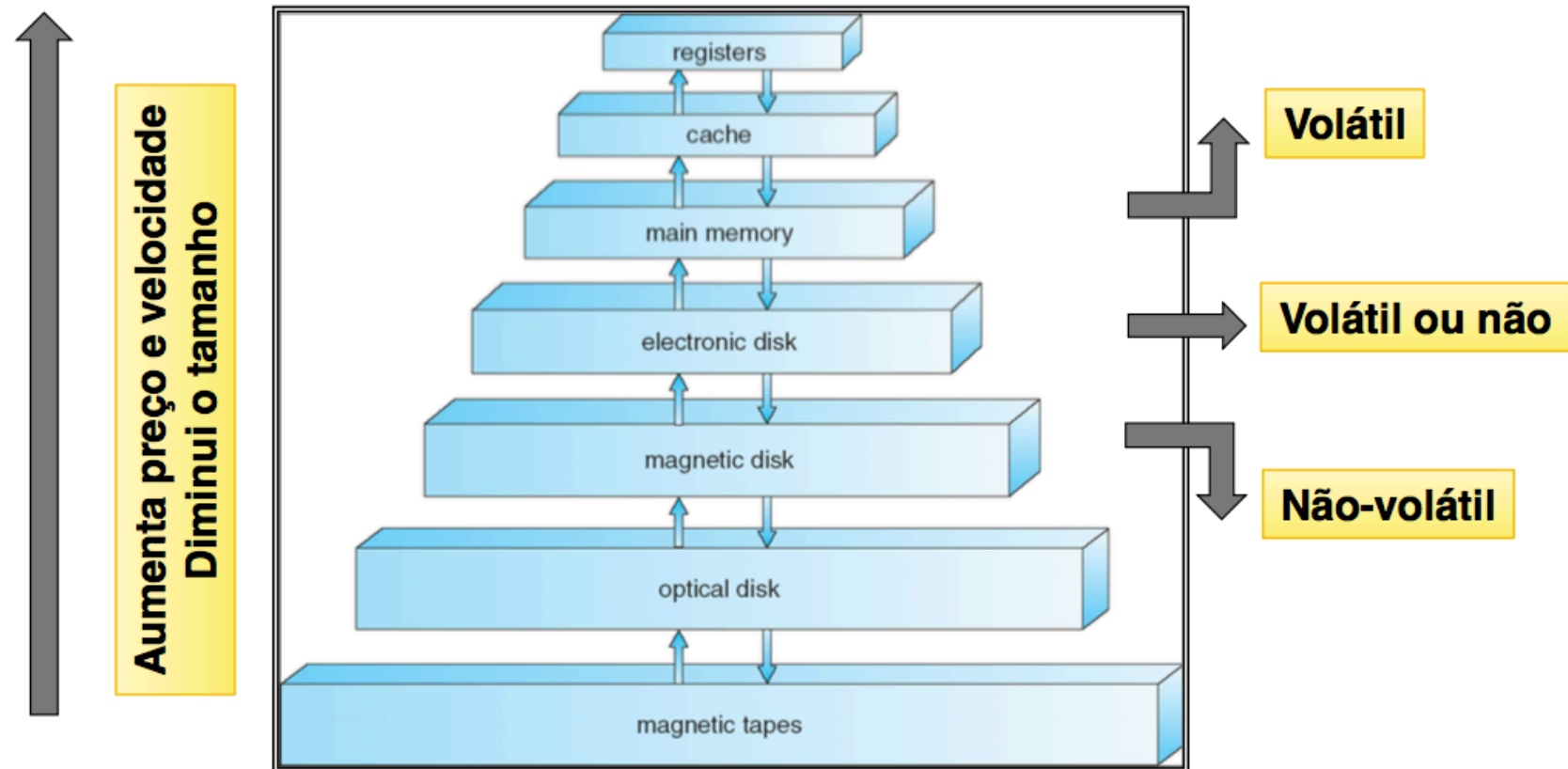
Ações para o gerenciamento

- **Caching**
 - Minimiza o impacto da CPU protelar suas operações quando os dados ainda não estão disponíveis.
 - Informação em uso copiada do armazenamento **mais lento para o mais rápido** temporariamente.
 - Armazenamento mais rápido (*cache*) verificado primeiro para determinar se a informação está lá:
 - Se estiver, a informação é usada diretamente do *cache* (rápido);
 - Se não, é preciso ir à memória.
 - **Isso pode ser pior do que ter cache!**
 - Está ligado a **desempenho**.
- **Tem problemas de projeto similares aos da RAM**
 - Gerenciamento, tamanho, política de substituição, etc.

Hierarquia de armazenamento

- **Sistemas de armazenamento organizados em hierarquia.**
 - Velocidade
 - Custo
 - Tamanho
 - Volatilidade

Hierarquia de armazenamento



Níveis na hierarquia de armazenamento

Level	1	2	3	4
Name	registers	cache	main memory	disk storage
Typical size	< 1 KB	> 16 MB	> 16 GB	> 100 GB
Implementation technology	custom memory with multiple ports, CMOS	on-chip or off-chip CMOS SRAM	CMOS DRAM	magnetic disk
Access time (ns)	0.25 – 0.5	0.5 – 25	80 – 250	5,000.000
Bandwidth (MB/sec)	20,000 – 100,000	5000 – 10,000	1000 – 5000	20 – 150
Managed by	compiler	hardware	operating system	operating system
Backed by	cache	main memory	disk	CD or tape

Espaço de endereços lógicos e físicos

- **Endereço lógico (ou virtual)**
 - Gerado pela CPU.
 - Endereço da aplicação na perspectiva da CPU.
- **Endereço físico**
 - Endereço da aplicação na perspectiva da unidade de memória.
- **Metáfora**
 - Obra de Machado de Assis na estante!
 - O conjunto de livros é a aplicação (endereços lógicos).
 - A estante é a memória (endereços físicos).

Espaço de endereços lógicos e físicos

- **Espaço de endereços lógicos**

- Conjunto de todos os endereços lógicos gerados por um processo.
- Para o processo, sua RAM é o seu espaço de endereços lógicos.
 - O processo lida com endereços lógicos; ele nunca “vê” os endereços físicos reais.

- **Espaço de endereços físicos**

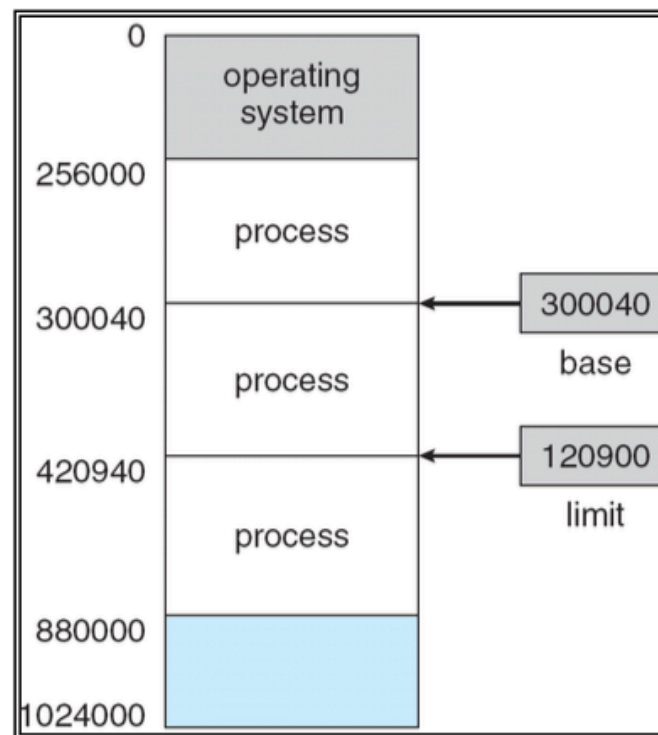
- Conjuntos dos endereços físicos correspondentes a esses endereços lógicos.

Unidade de Gerenciamento de Memória (MMU)

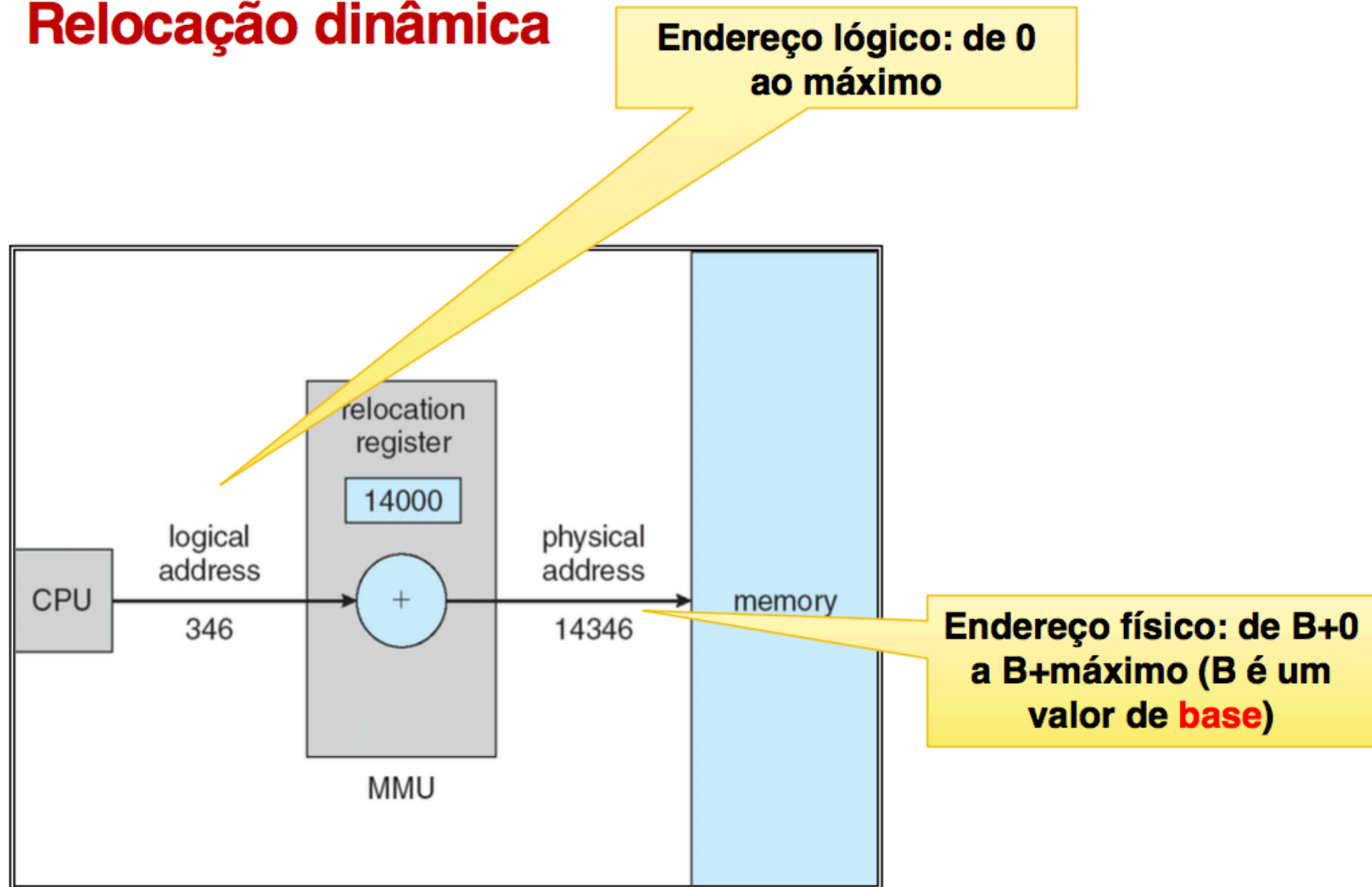
- **Definição**
 - Dispositivo de hardware que mapeia endereço virtual para físico.
- **Principais elementos**
 - Usa um **registrador de base** (também chamado de registrador de relocação) e o **registrador de limite**.
- **Acesso à RAM**
 - O valor no registrador de relocação é somado a cada endereço gerado por um processo.

Registradores de base e limite

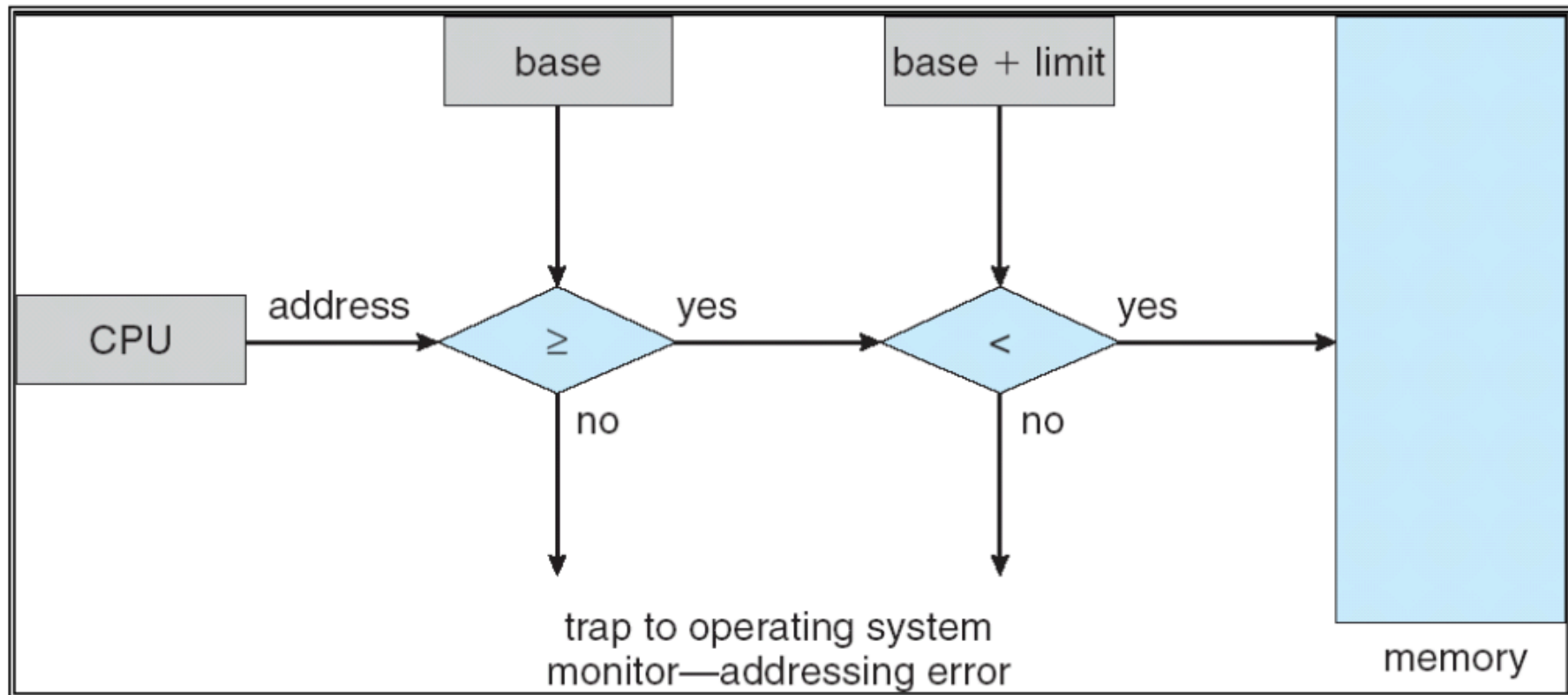
- **Garantindo espaço separado a cada processo**
 - Um par de registradores de **base** e **limite** definem o **espaço de endereços lógicos** no **espaço de endereços físicos**.



Relocação dinâmica



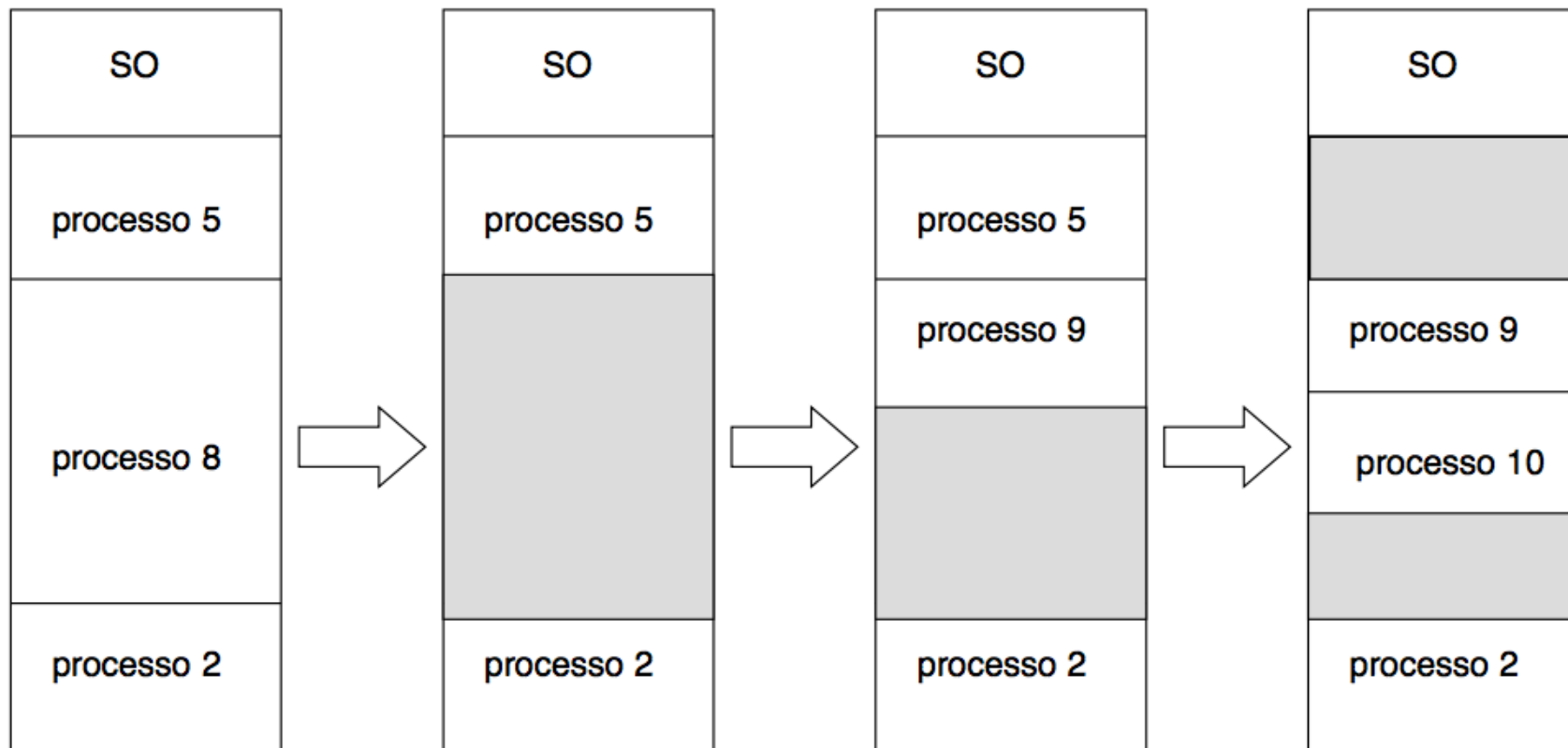
Proteção de endereço de hardware



Alocação de memória

- **Partições fixas (particionamento estático)**
 - Grau de multiprogramação está limitado pelo número de partições.
 - Processos usam múltiplos das partições.
- **Partições variáveis (particionamento dinâmico)**
 - Uso mais livre da memória.
- **SO mantém uma tabela indicando quais partes da memória estão disponíveis e ocupadas.**
- **Espaço: bloco de memória disponível.**
 - Espaços de vários tamanhos estão espalhados pela memória.
 - Quando um processo chega, ele recebe memória de um espaço grande o suficiente para acomodá-lo.

Alocação de memória



Alocação de memória

- **Funcionamento**

- A memória é alocada aos processos até os requisitos de memória do próximo processo da fila de entrada não puderem ser satisfeitos (não houver um espaço que o comporte).
- Se o espaço for suficiente, parte é alocada ao processo e a outra parte é devolvida ao conjunto de espaços.
- Se existem espaços vizinhos, eles são mesclados para formarem um espaço maior.

Alocação de memória

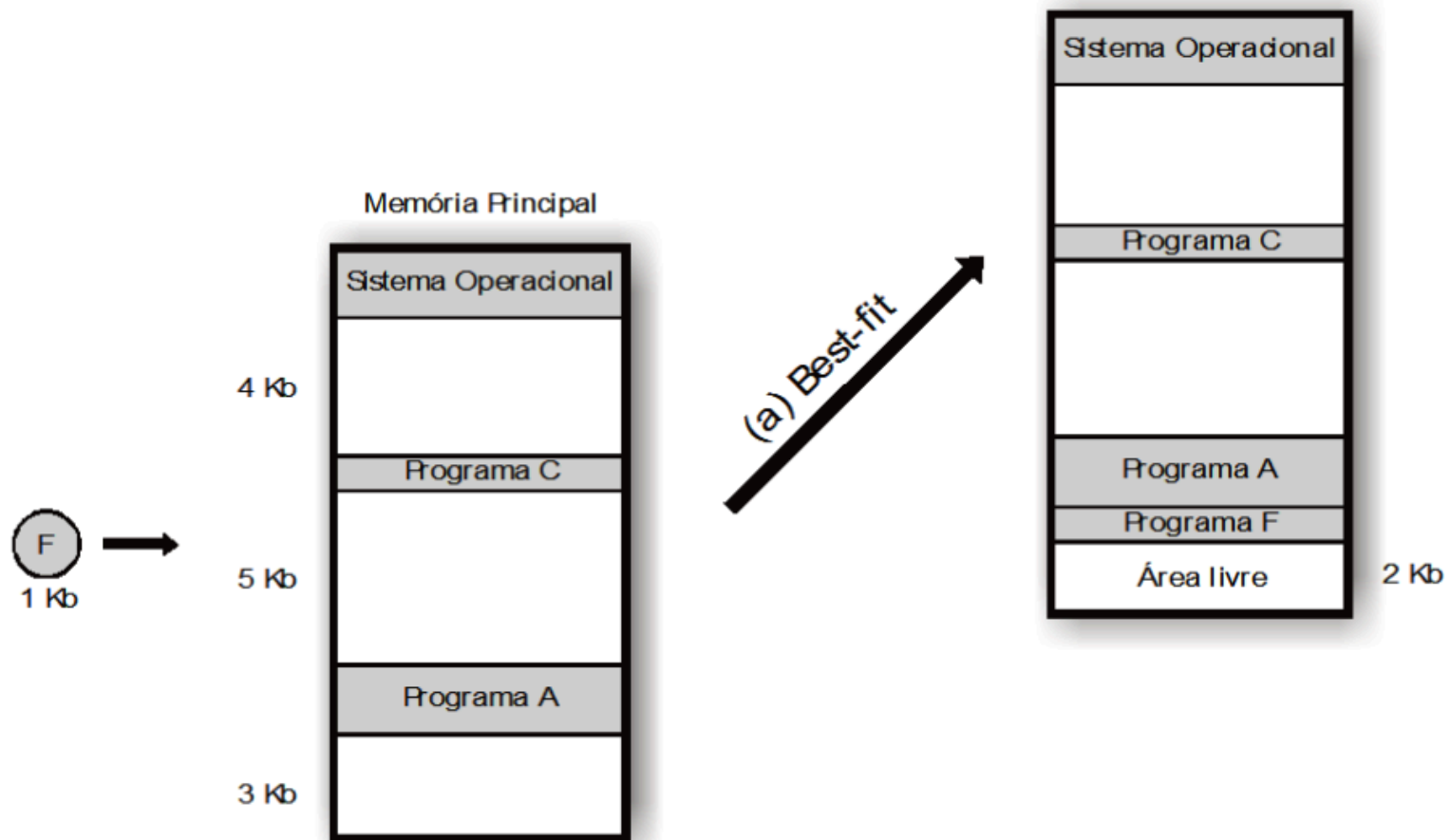
- **Problema**

- Como é feita a alocação de memória?

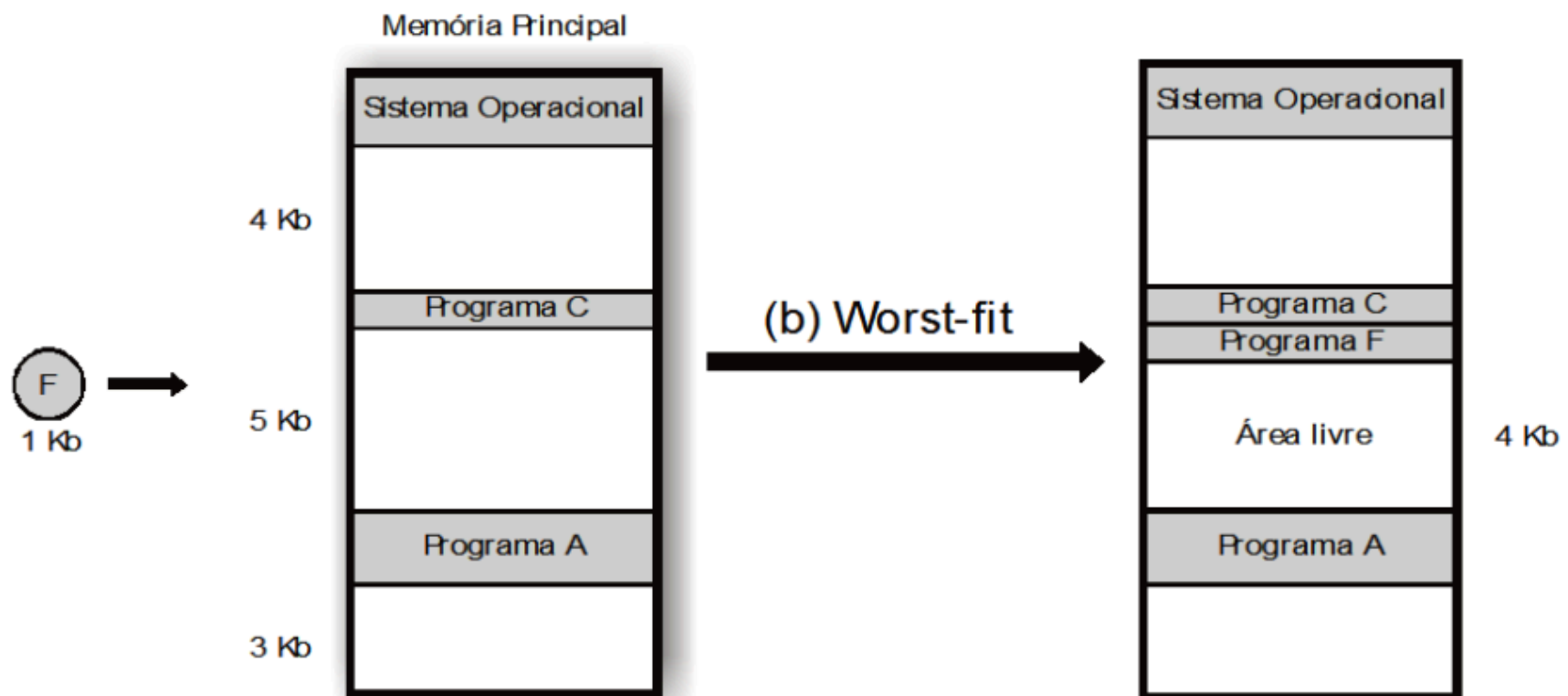
- **Estratégias**

- ***Best-fit.*** Aloca o *menor* espaço com tamanho suficiente; deve procurar lista inteira, a menos que ordenado por tamanho.
 - Produz o menor espaço restante.
 - ***Worst-fit.*** Aloca o *maior* espaço; também deve pesquisar lista inteira.
 - Produz o maior espaço restante. Pode ser mais útil.
 - ***First-fit.*** Aloca o *primeiro* espaço com tamanho suficiente.

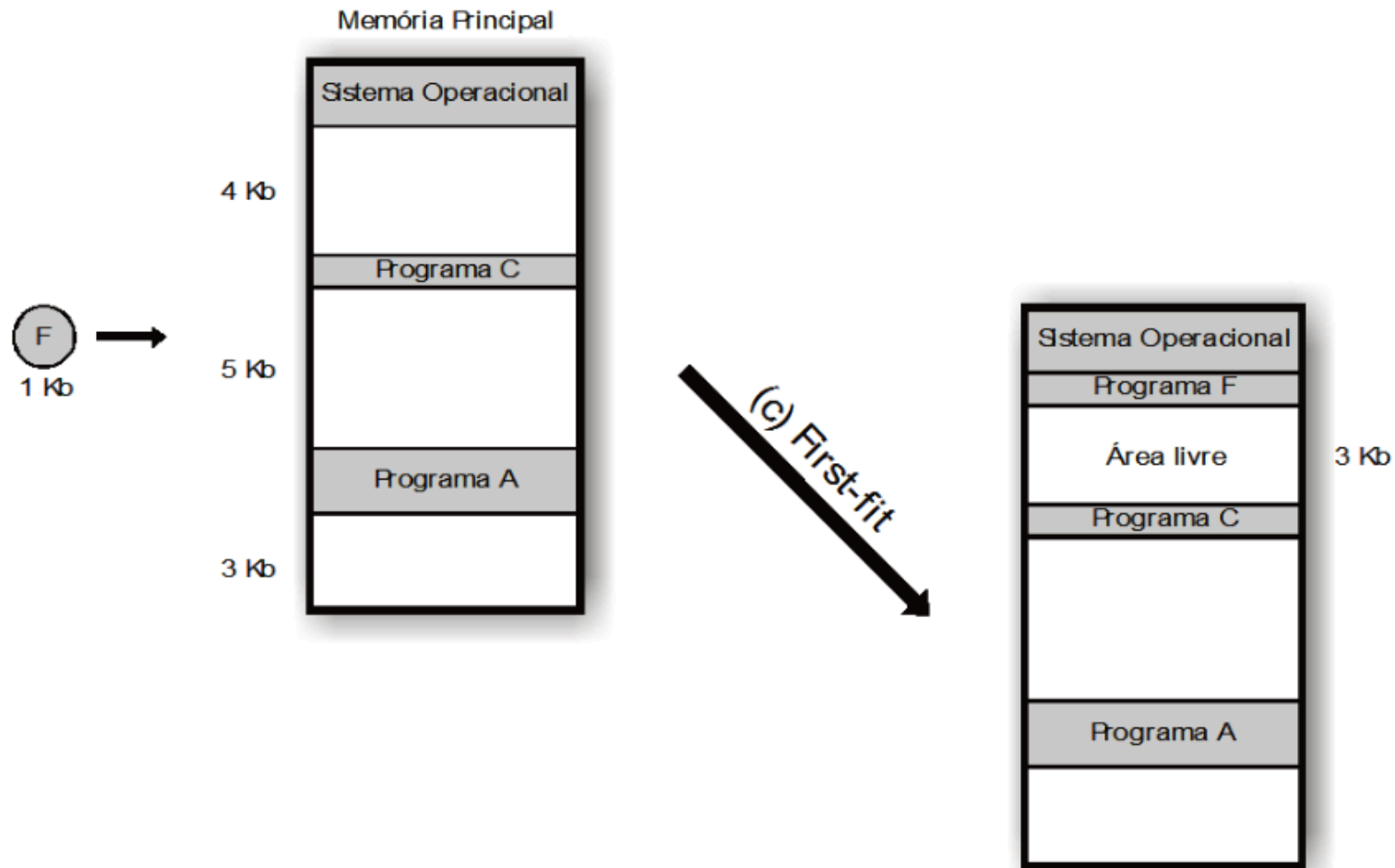
Alocação de armazenamento dinâmico



Alocação de armazenamento dinâmico



Alocação de armazenamento dinâmico



Fragmentação

- ***First-fit e best-fit* geram fragmentação externa.**
 - À medida que processos são carregados e removidos da memória, o espaço livre é repartido em pequenas partes.
 - Perde 50% da memória em média!
- **Fragmentação externa**
 - Existe espaço de memória total para satisfazer uma solicitação, mas **não é contíguo**.
- **Fragmentação interna**
 - Memória alocada pode ser ligeiramente maior que a memória requisitada. Espaço: 18.464 bytes; Processo: 18.462 bytes.
 - Essa diferença de tamanho é memória interna a uma partição, mas que não está sendo usada.

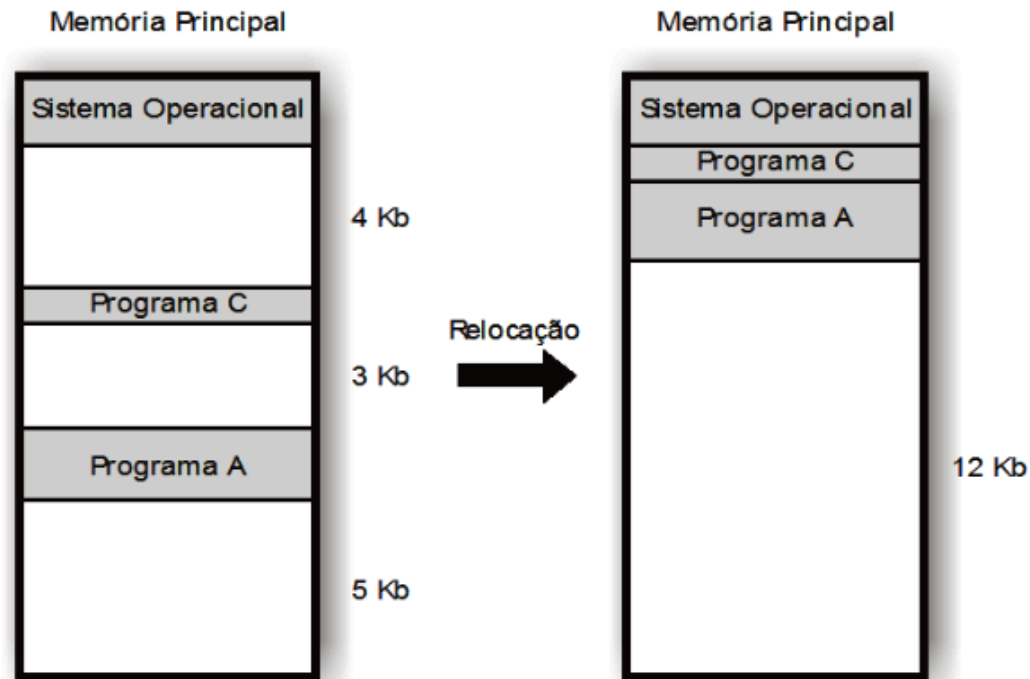
Fragmentação

- Qual a solução?

- Solução 1

- Reduza com a **compactação ou relocação**

- Misture o conteúdo da memória para colocar toda a memória livre junta em um bloco grande.
- A compactação só é possível se a relocação for dinâmica, e é feita em tempo de execução.
- É dispendioso! Ideia ruim!



Fragmentação

- **Solução 2**

- Permitir que o espaço de endereços lógicos de um processo seja não-contíguo.
 - O processo vai receber memória física onde estiver disponível.
- Paginação e Segmentação!

