

Memória Virtual

Conceitos básicos

Paginação por demanda

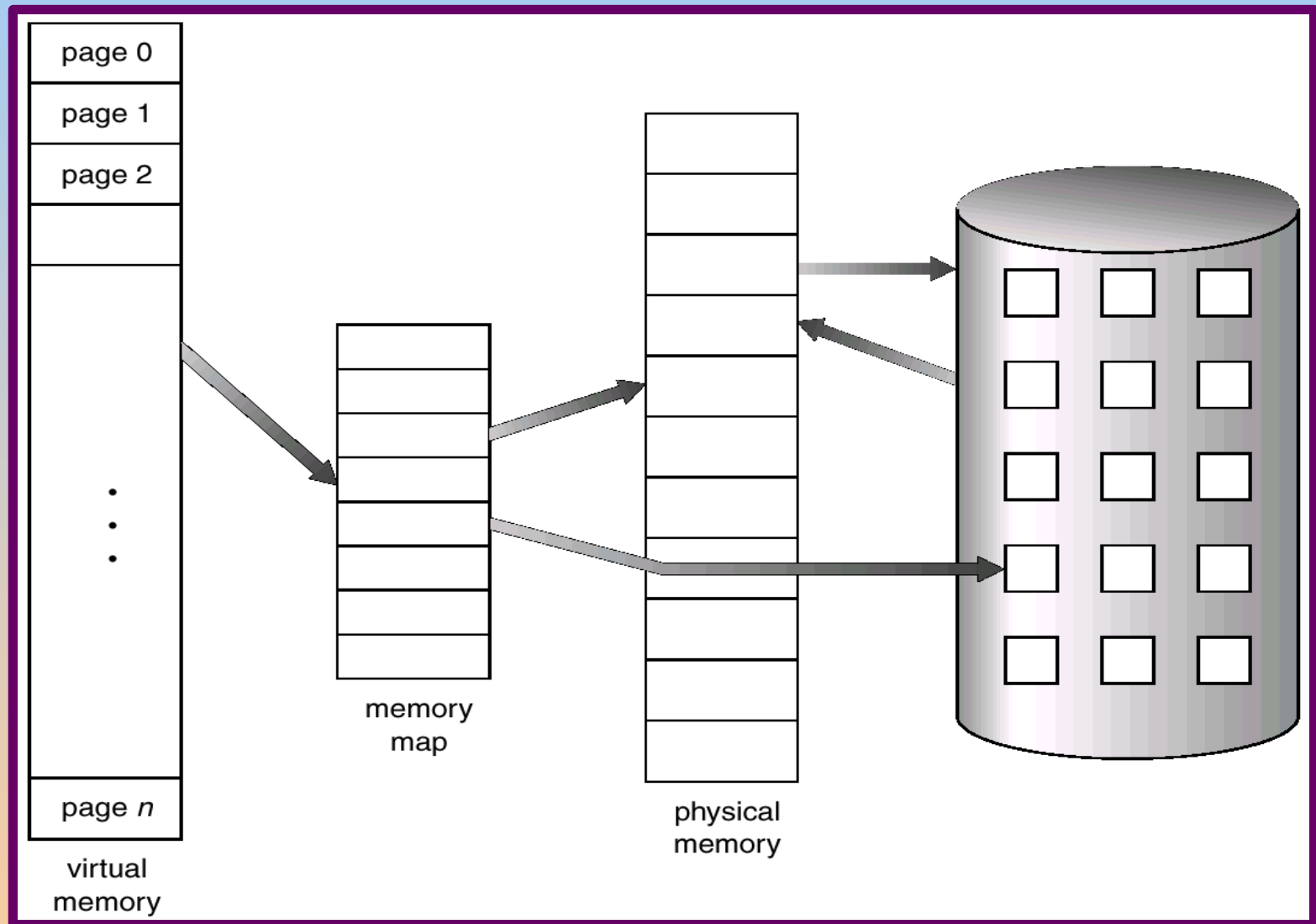
Criação de processos

Substituição de páginas

Conceitos básicos

- **Memória virtual** – separação de memória lógica da memória física.
 - ◆ Somente parte do programa precisa estar na memória para execução.
 - ◆ Espaço de endereçamento lógico pode ser muito maior que o espaço de endereçamento físico.
 - ◆ Permite que espaços de endereçamento sejam compartilhados por vários processos.
 - ◆ Permite uma criação mais eficiente de processos.
- Memória virtual pode ser criada via:
 - ◆ Paginação por demanda (Demand paging)
 - ◆ Segmentação por demanda.

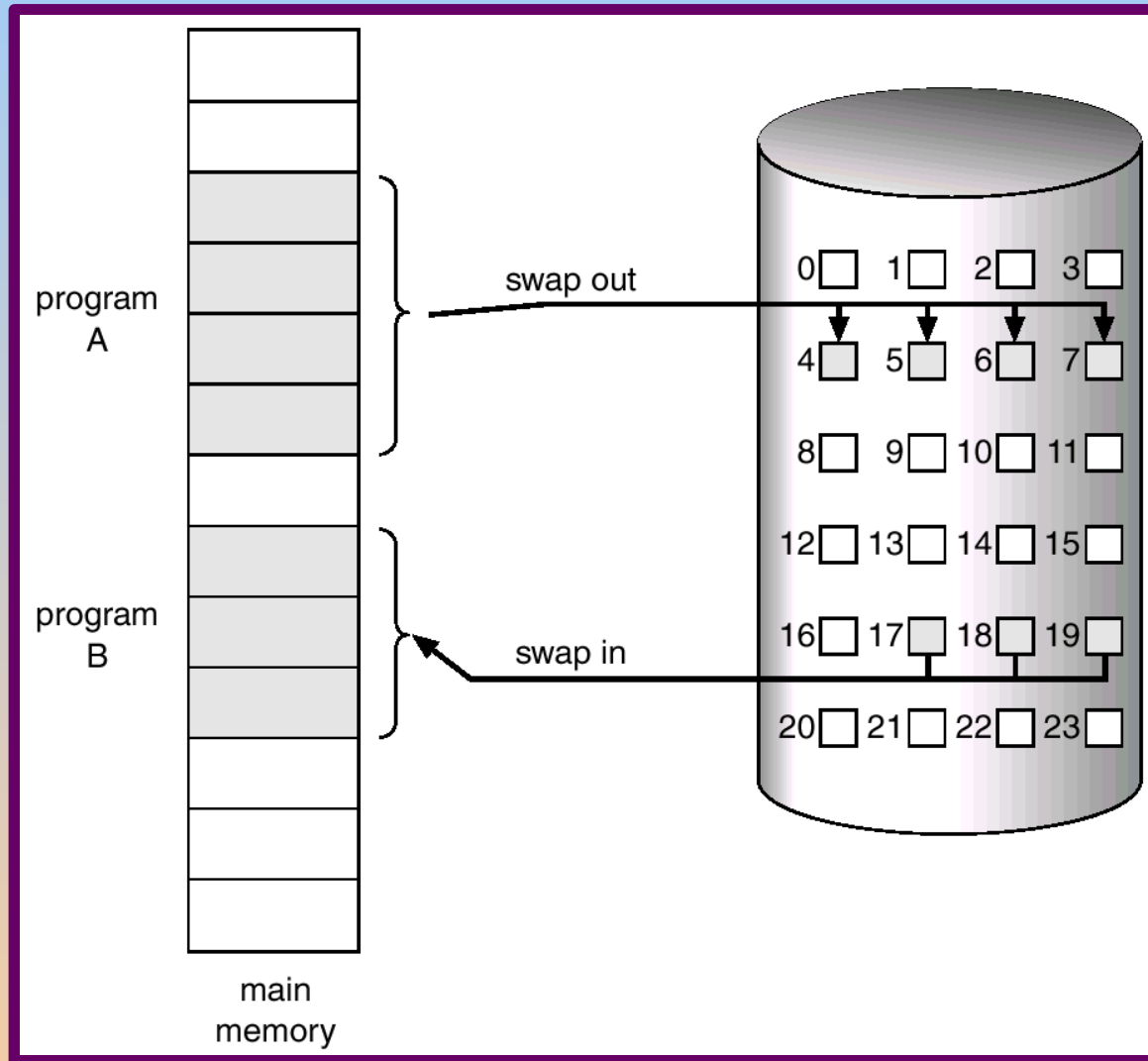
Memória virtual que é muito maior que a memória física



Paginação por demanda

- Coloca uma página na memória somente quando é necessário.
 - ◆ Necessita de menos I/O
 - ◆ Necessita de menos memória
 - ◆ Resposta mais rápida
 - ◆ Mais usuários
- Se uma página é necessária \Rightarrow basta referenciá-la
 - ◆ Referência inválida \Rightarrow erro
 - ◆ Não está na memória \Rightarrow carregá-la na memória

Transferência de uma memória paginada para espaço contíguo no disco



Bit Válido-Inválido

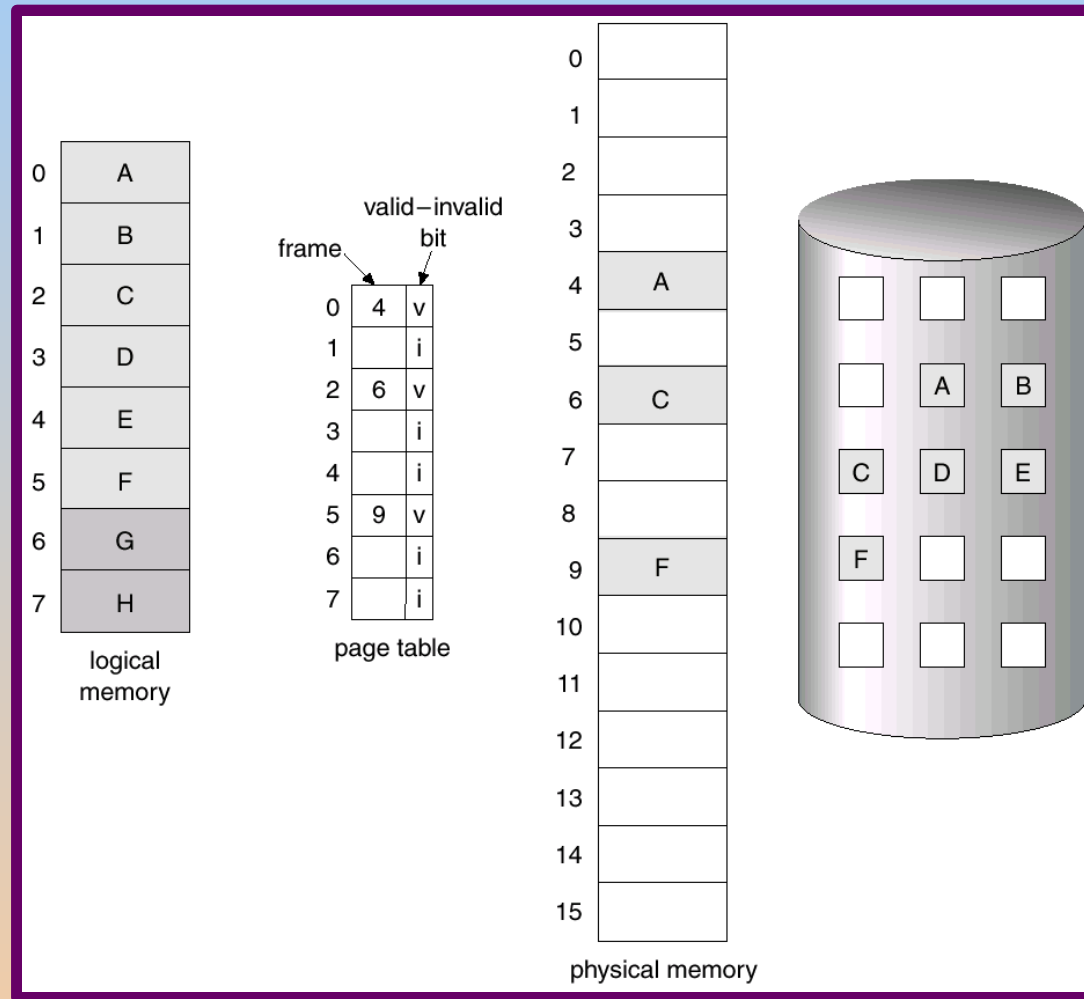
- Com cada entrada na tabela de página um bit válido-inválido é associado (1 \Rightarrow na memória, 0 \Rightarrow não está na memória)
- Inicialmente, todos os bits estão em 0.
- Exemplo:

Quadro #	Bit válido-inválido
	1
	1
	1
	1
	0
⋮	
	0
	0

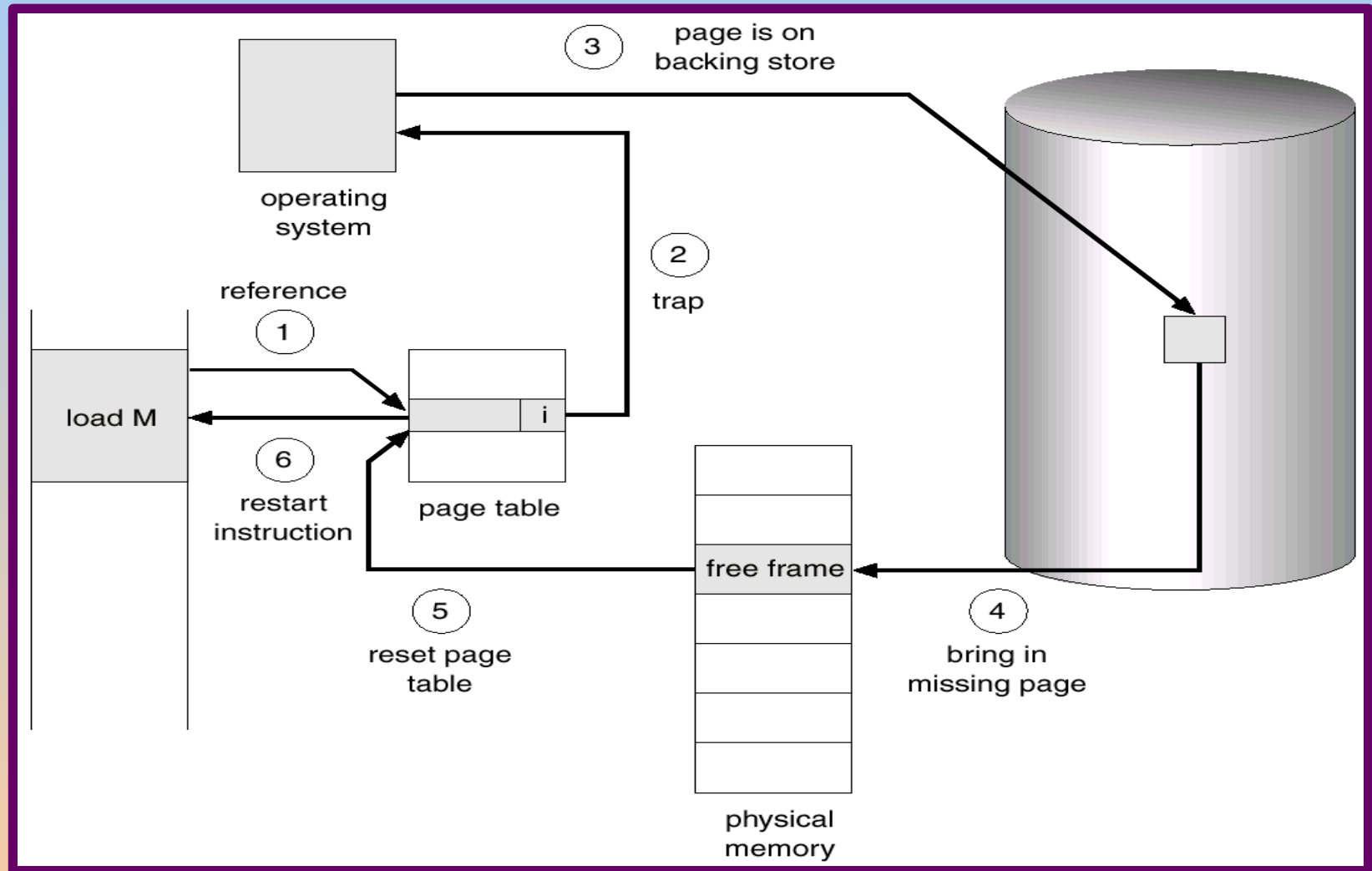
page table

- Durante a tradução de endereço, se o bit válido-inválido na entrada é 0 \Rightarrow falha de página.

Tabela de páginas com algumas páginas que não estão na memória



Passos no tratamento de falha de página



O que acontece se não existem quadros livres?

- Substituição de páginas – encontrar alguma página em memória e armazená-la em disco.
 - ◆ algoritmo
 - ◆ desempenho – queremos um algoritmo que resulte no menor número de falhas de página.
- Algumas páginas podem sair da memória várias vezes.

Criação de processos

- Memória virtual permite outros benefícios durante a criação de processos:
 - Cópia-na-escrita(Copy-on-Write)
 - Arquivos mapeados em memória(Memory-Mapped Files)

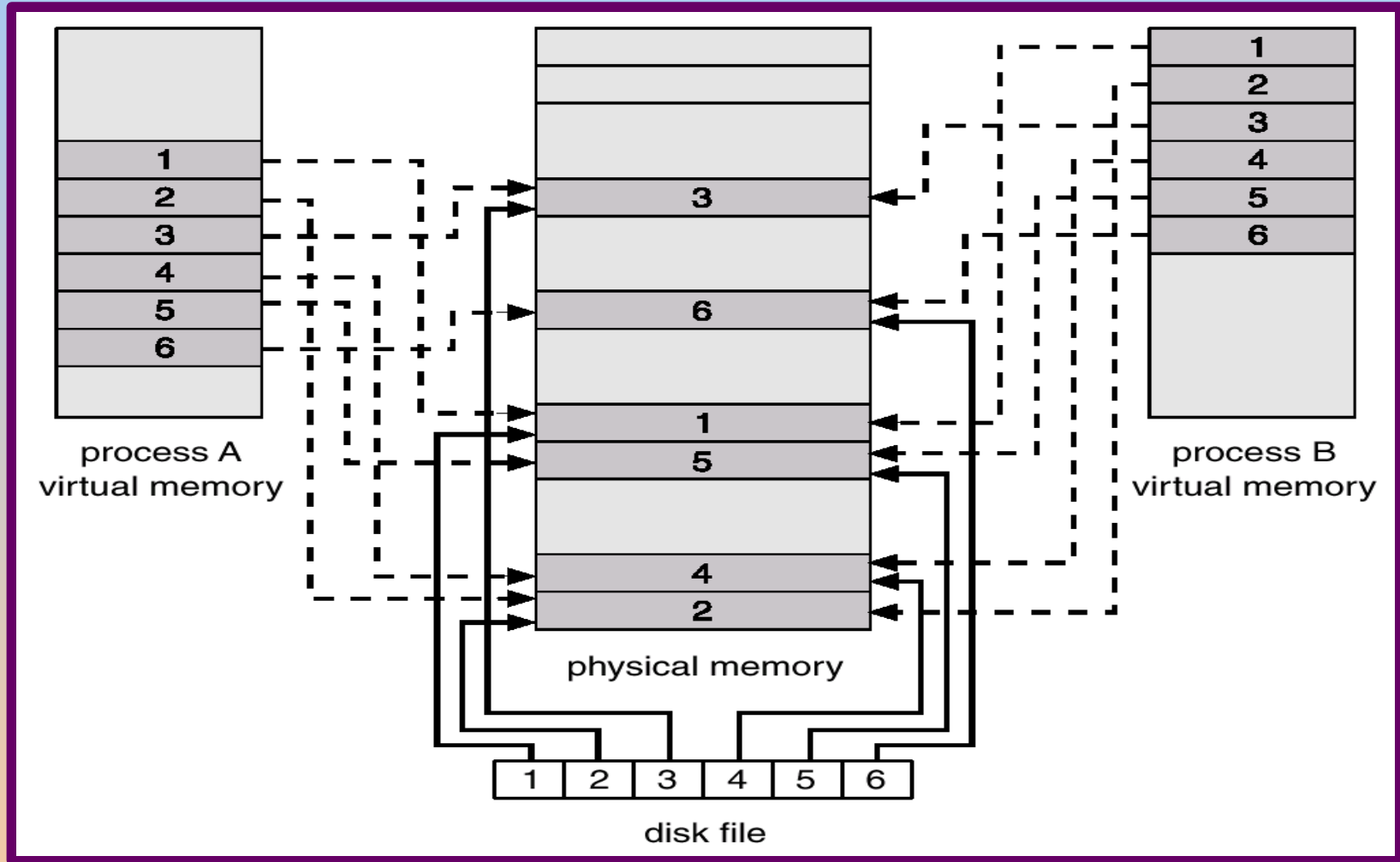
Copy-on-Write

- Copy-on-Write (COW) permite que pai e filho inicialmente compartilhem as mesmas páginas em memória.
- Se um dos processos modifica a página compartilhada, somente a página é copiada.
- COW permite criação mais eficiente de processos, uma vez que somente páginas modificadas sejam copiadas.

Arquivos mapeados em memória

- I/O de arquivos mapeados em memória permitem que I/O em arquivos seja tratada como uma rotina de acesso à memória por mapeamento de um bloco de disco numa página de memória.
- Um arquivo é inicialmente lido usando paginação por demanda. Um trecho do arquivo, do tamanho de uma página, é lido do sistema de arquivos para a memória física. Escritas/leituras subsequentes são tratados como acessos ordinários a memória.
- Simplifica o acesso a arquivos tratando I/O de arquivos através de acessos à memória ao invés de systems calls do tipo **read()** e **write()** .
- Permite, também, que vários processos possam mapear o mesmo arquivo, possibilitando compartilhamento de páginas em memória.

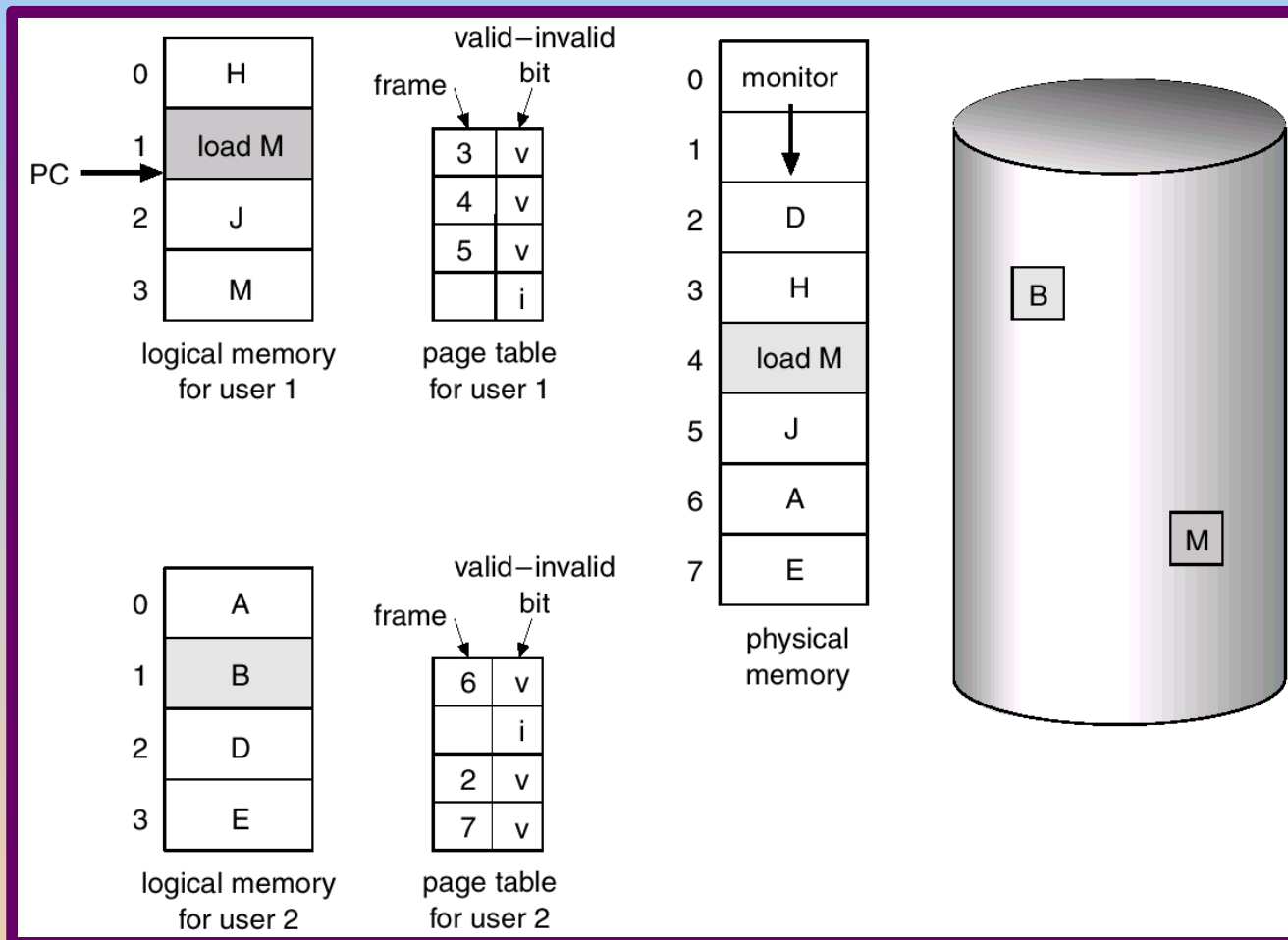
Arquivos mapeados em memória



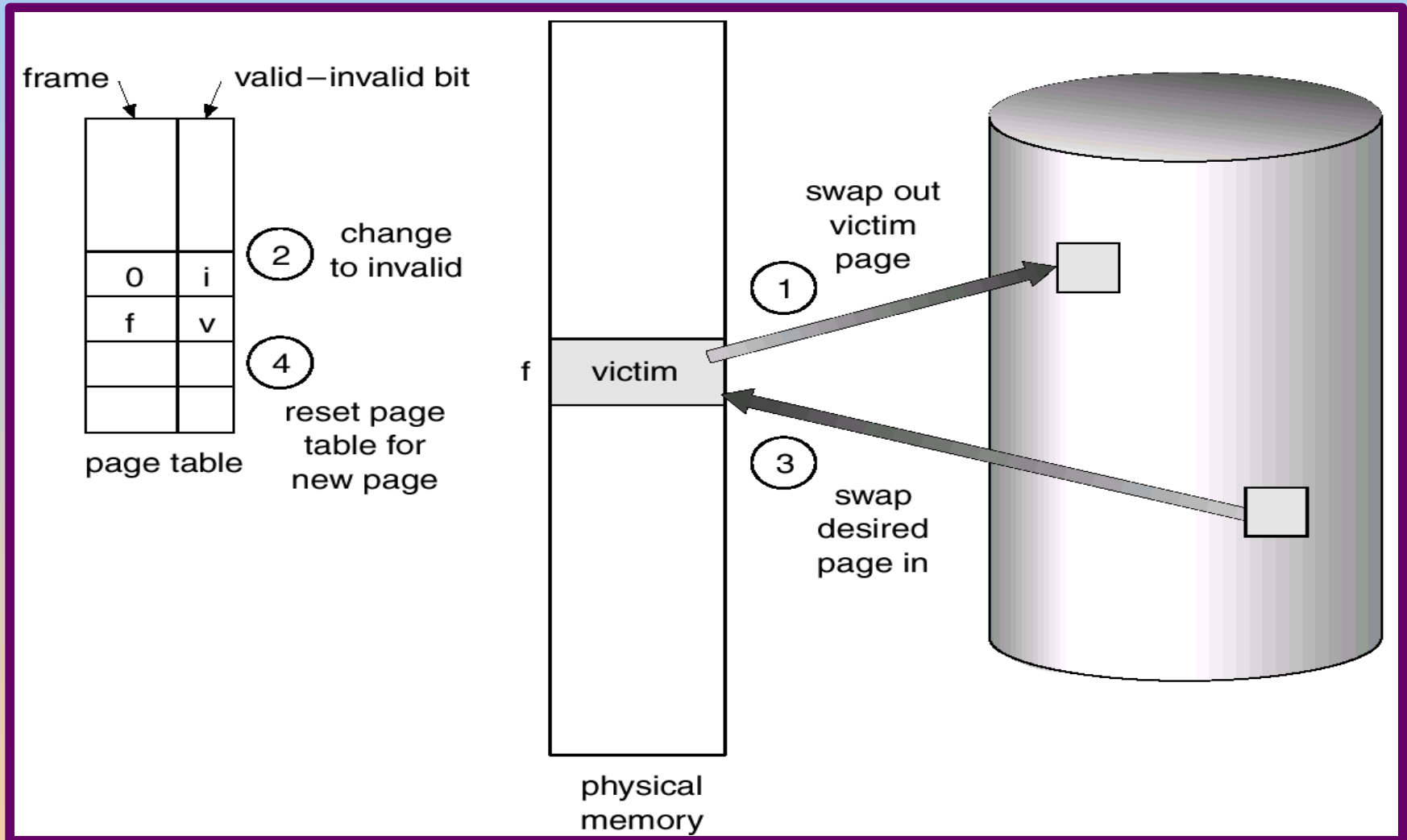
Substituição de páginas

- Previne super-alocação de memória modificando a rotina de serviço de falha de página para incluir substituição de página.
- Usa bit de modificação(dirty bit) para reduzir overhead de transferência de página – somente páginas modificadas são escritas no disco.
- Substituição de páginas completa separação entre memória lógica e memória física. Assim, uma grande memória virtual pode ser mapeada para uma pequena memória física.

Necessidade de substituição de páginas



Algoritmo básico de substituição de páginas



Algoritmo First-In-First-Out (FIFO)

- Referências: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- 3 quadros (3 páginas, no máximo, na memória)

1	1	4	5
2	2	1	3
3	3	2	4

9 falhas de página

- 4 quadros

1	1	5	4
2	2	1	5
3	3	2	
4	4	3	

10 falhas de página

Algoritmo ótimo

- Substituir uma página que não será usada por um longo período de tempo.
- Exemplo (4 quadros)

1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

1	4
2	
3	
4	5

6 falhas de páginas

- Como obter uma aproximação deste algoritmo ?

Algoritmo Least Recently Used (LRU)

Página menos recentemente usada

- Seqüência de páginas: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

1	5
2	
3	5 4
4	3

- Implementação do contador:
 - ◆ Toda entrada de página tem um contador; cada vez que uma página é referenciada, copiar o clock no contador.
 - ◆ Quando é necessário substituir uma página, localizar a que possui o contador mais antigo.

Algoritmos baseados em contagem

- Manter um contador com o número de vezes que a página foi referenciada.
- Algoritmo LFU(Least Frequently Used) : substitui a página com menor contador.
- Algoritmo MFU(Most Frequently Used) : substitui a página com maior contador. É baseada no argumento que uma página com o menor contador foi recentemente colocada na memória mas ainda não foi usada.