Gerenciamento de Memória

Por que fazer?

➔Controlar inserção, manipulação, acesso e retirada de dados

A quais requisitos deve atender?

- ➔Garantir integridade de espaços de memória;
- ➔Garantir controle de acesso (processos usuário e de kernel);
- ➔Garantir que processos não acessem ou ocupem espaços de memória destinados a outros processos;
- ➔Processos Kernel devem, no mínimo, poder "dominar" processos de usuário;
- ➔Mapear endereços lógicos em físicos;

Quem faz e como?

Módulo do S.O

"Controle de acesso" a memória

➔É definido pelos **registradores base e limite;**

- ◆Registrador base = primeira posição de memória que o processo pode utilizar;
- ◆Registrador limite = quantidade de memória que o processo pode utilizar

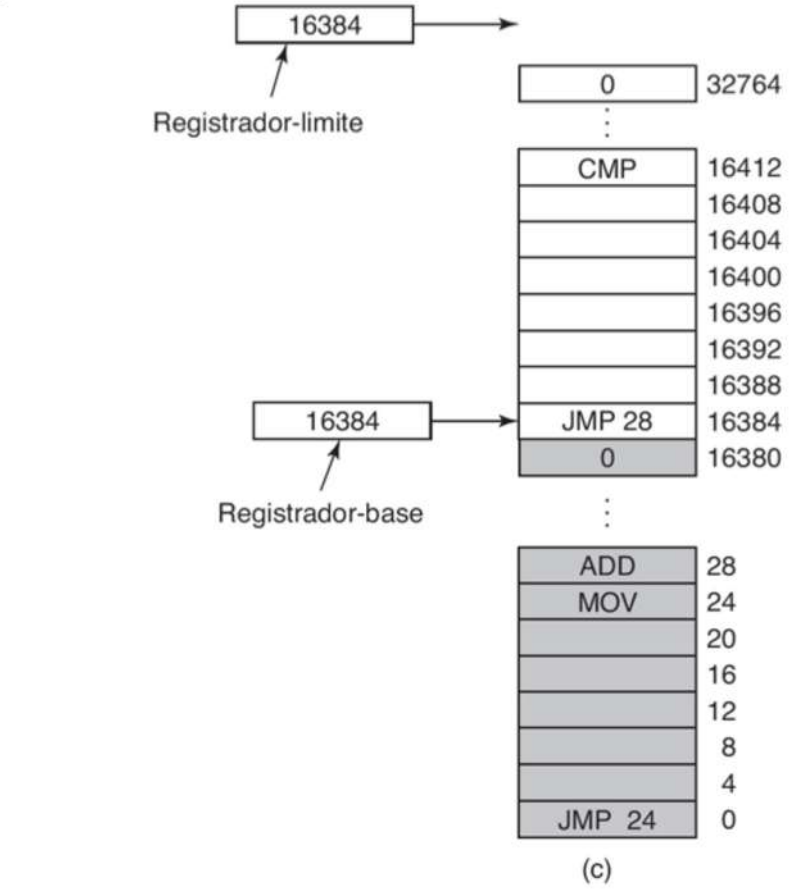
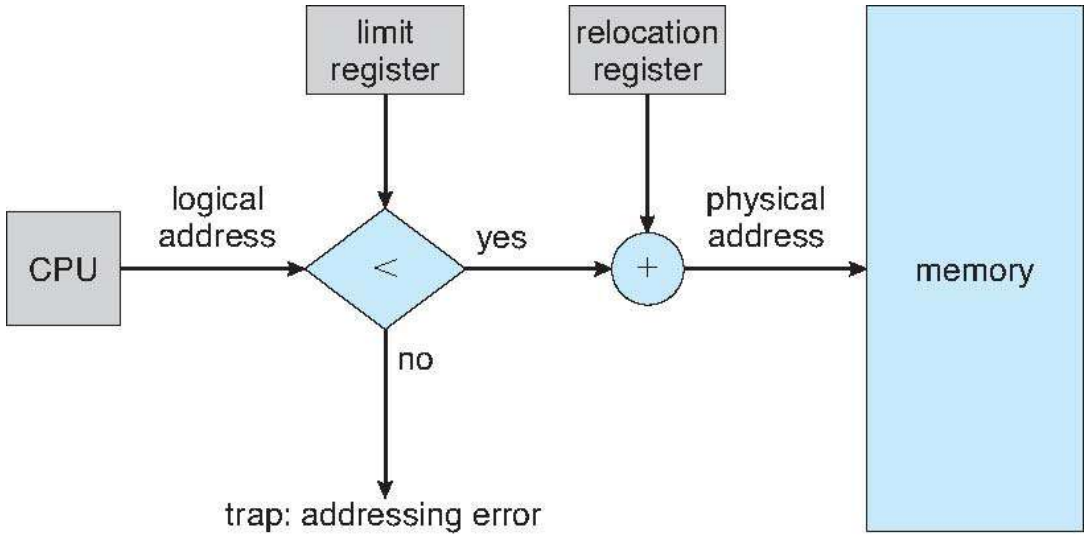


Figura 3.3 O registrador-limite e o registrador-base podem ser usados para dar a cada processo um espaço de endereçamento independente.

Como o hardware faz este controle?



Vinculação de endereços

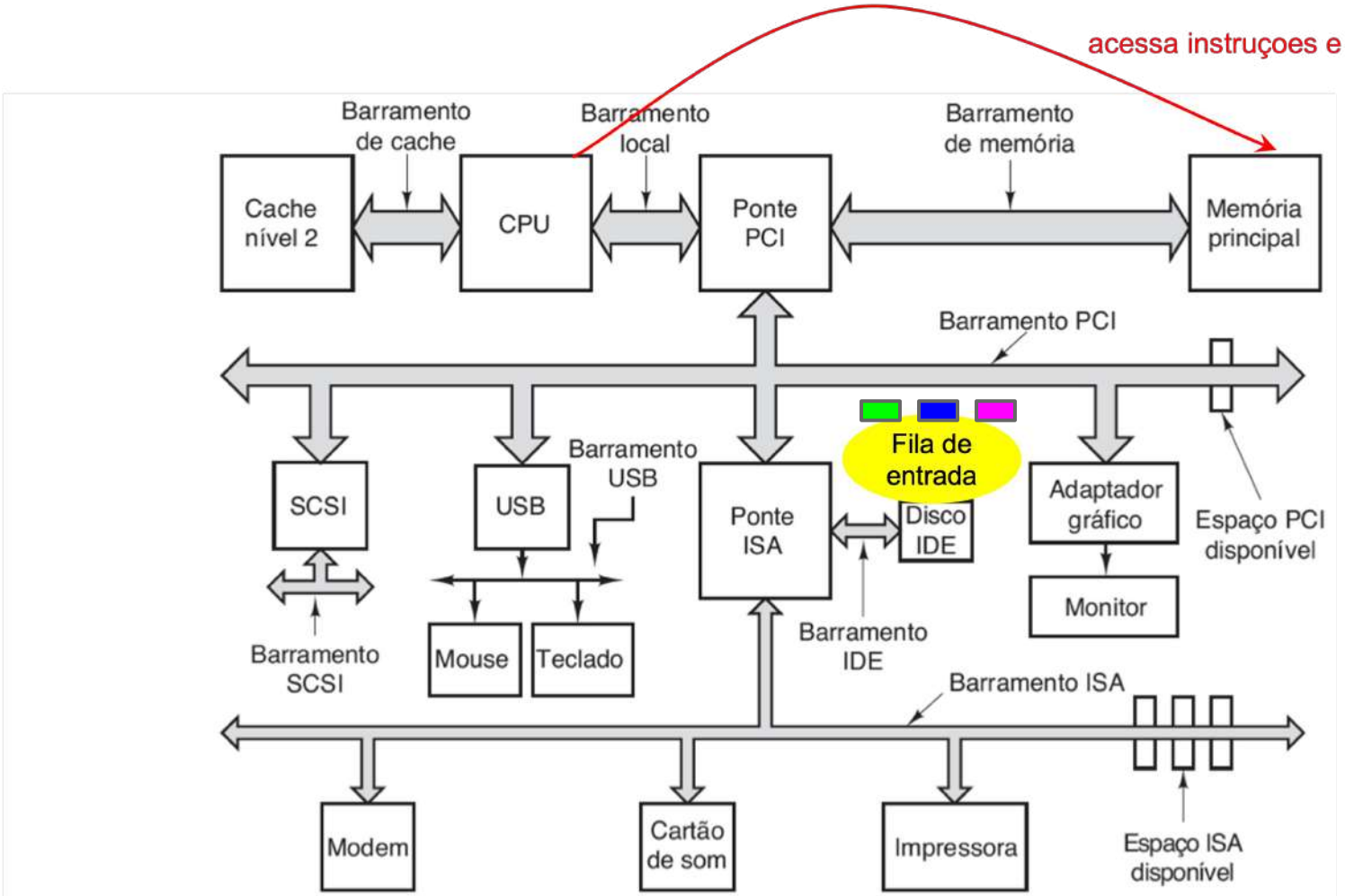
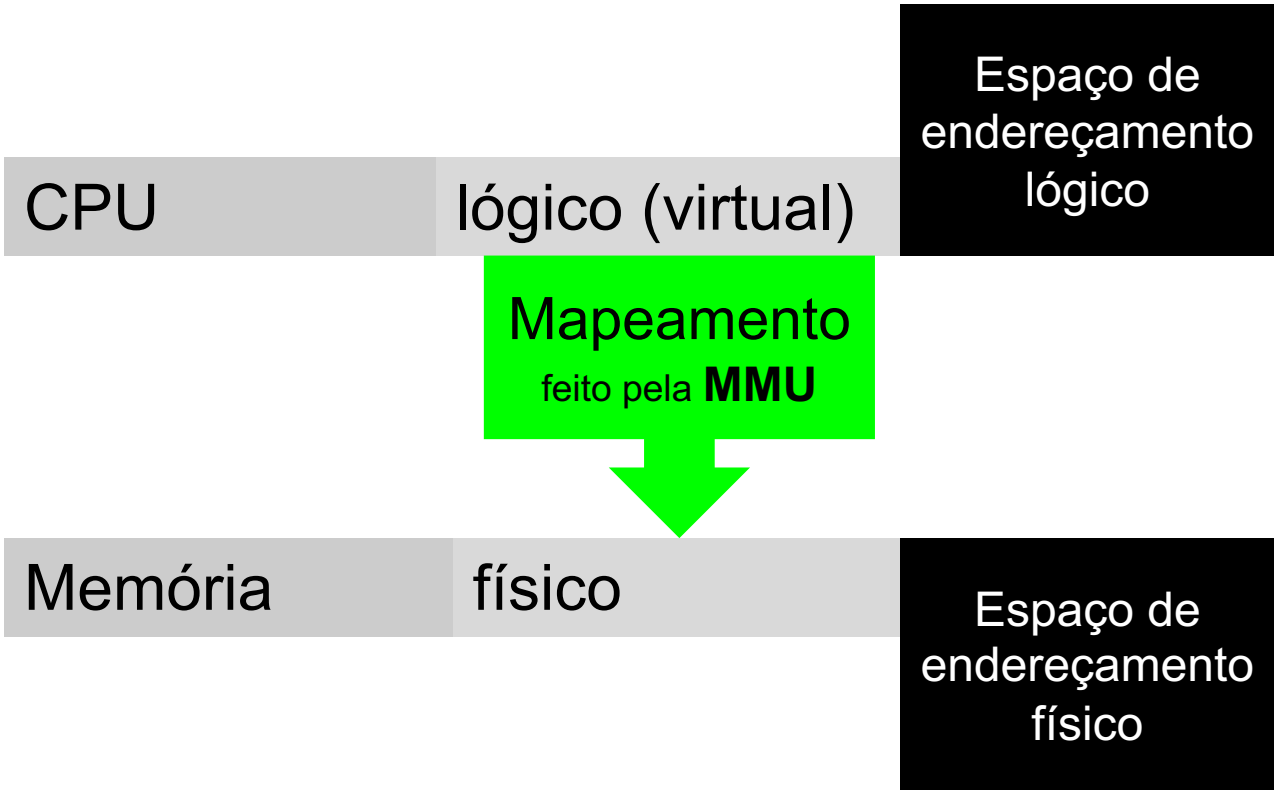
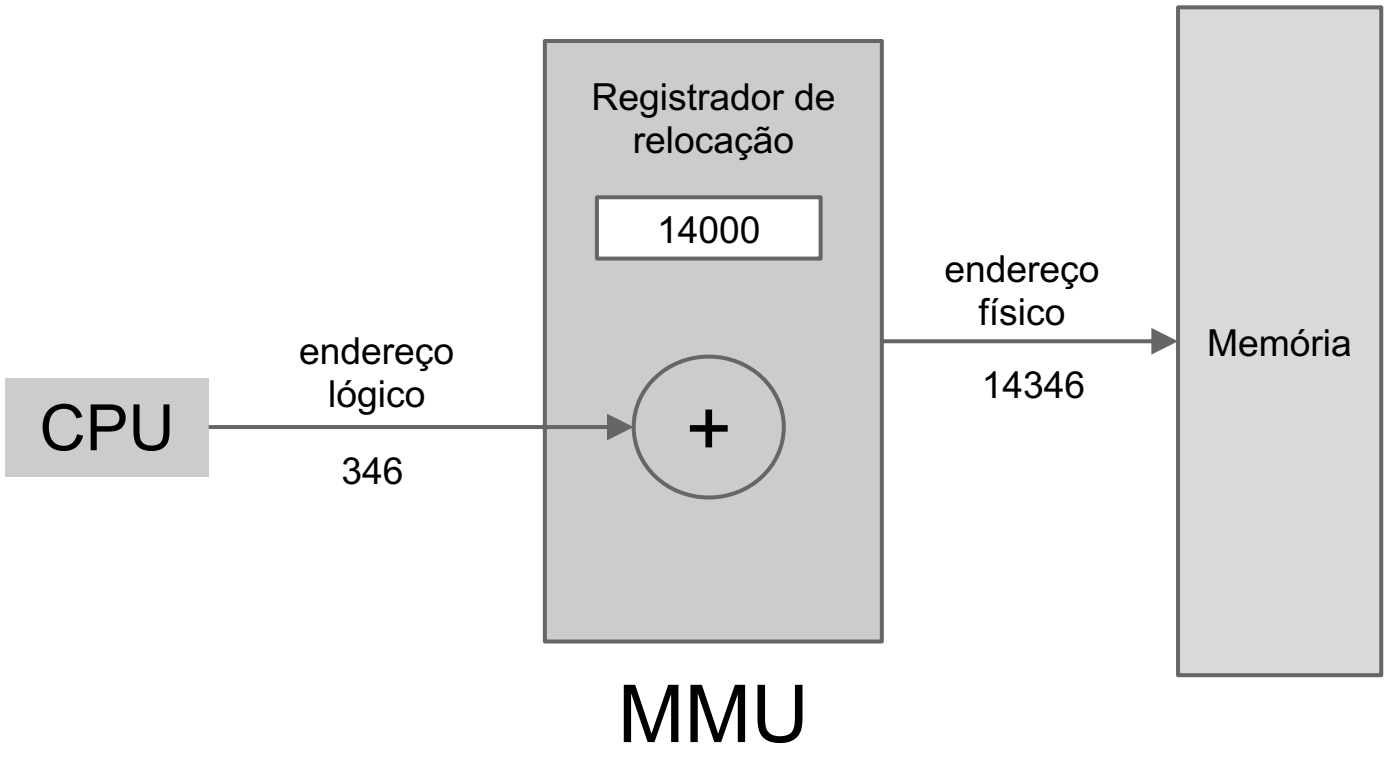


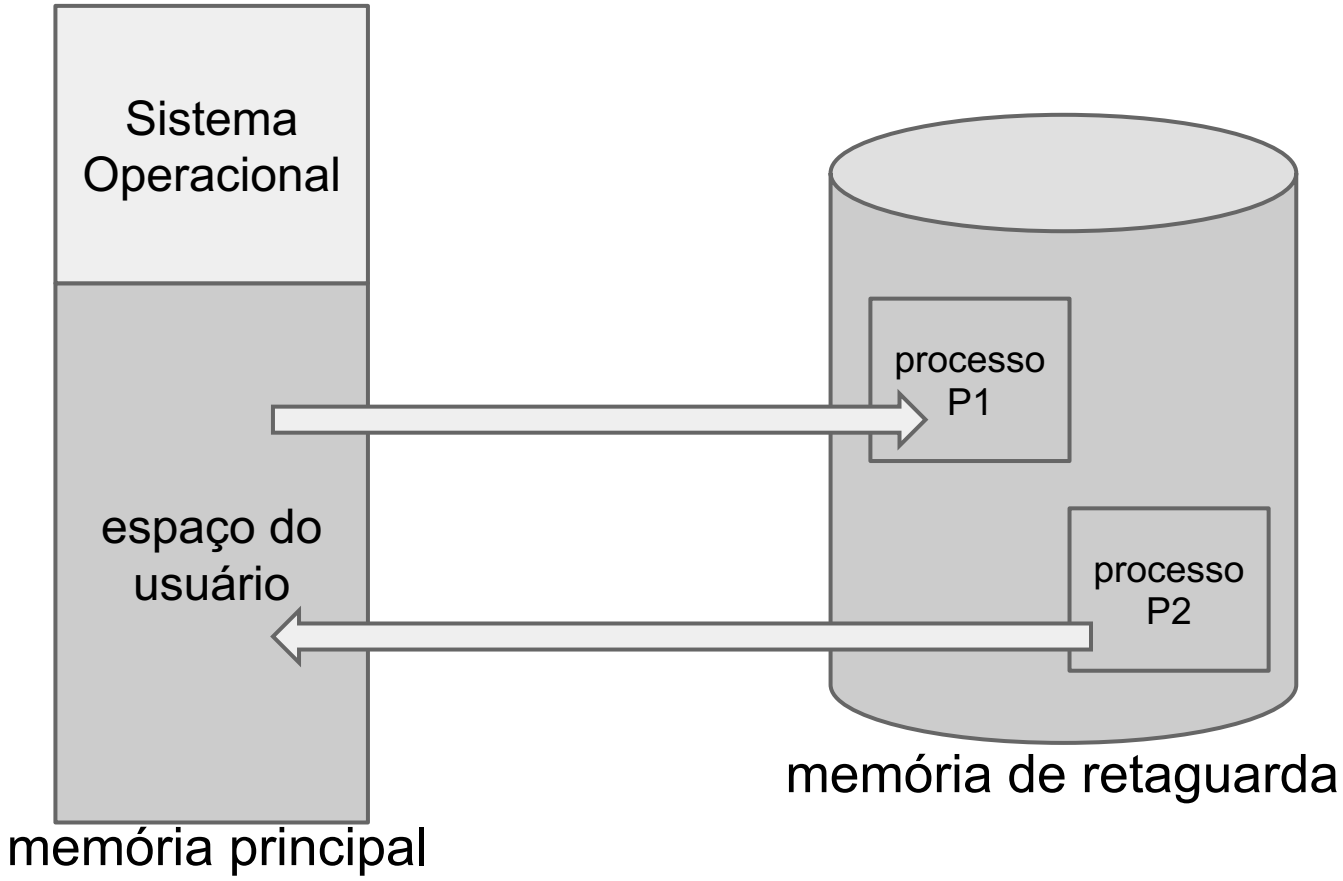
Figura 1.12 A estrutura de um sistema Pentium grande.

Quais tipos de endereços são utilizados?

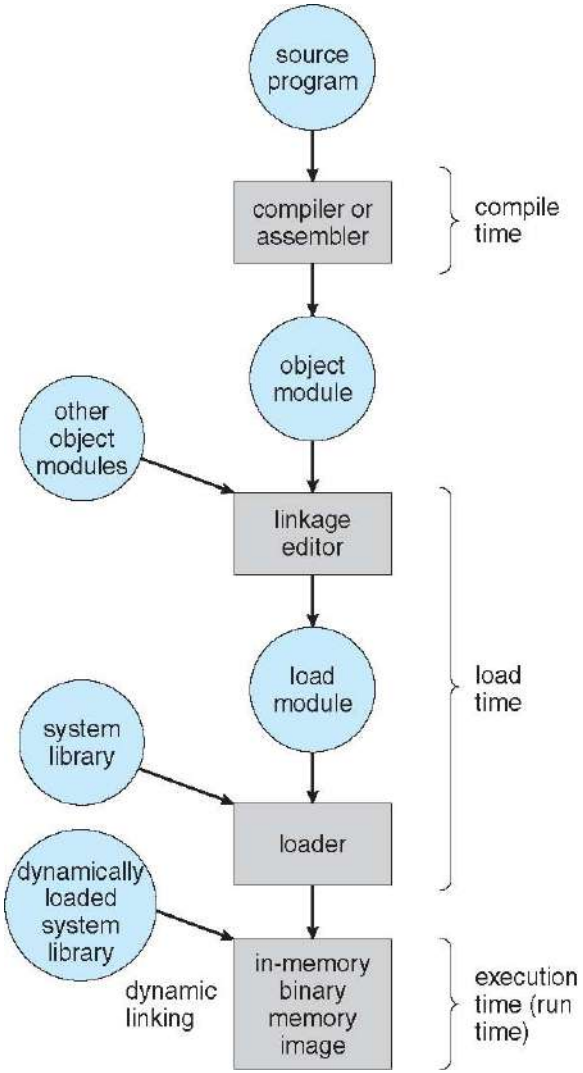



Registrador de relocação
14000
endereço lógico
346
CPU
+
endereço físico
14346
Memória
MMU

Swapping


Sistema
Operacional
espaço do
usuário
memória principal
processo
P1
processo
P2
memória de retaguarda

Momentos em que há alocação de memória



Alocação de memória a processos

Contígua

Partições de tamanho fixo

Ex: cada partição, um processo

Partições de tamanho variável

- Uma tabela indica quais partes da memória estão disponíveis e quais estão ocupadas;
- Brecha = espaço de memória disponível
- Brechas adjacentes formam brechas maiores ("mescladas")

O problema da alocação de memória dinâmica geral

Como uma solicitação de tamanho n pode ser atendida a partir da lista de brechas livre?

- ➔ Primeiro Apto (first-fit)
- ➔ Mais apto (best-fit)
- ➔ Menos Apto (worst fit)

Memória x Tempo

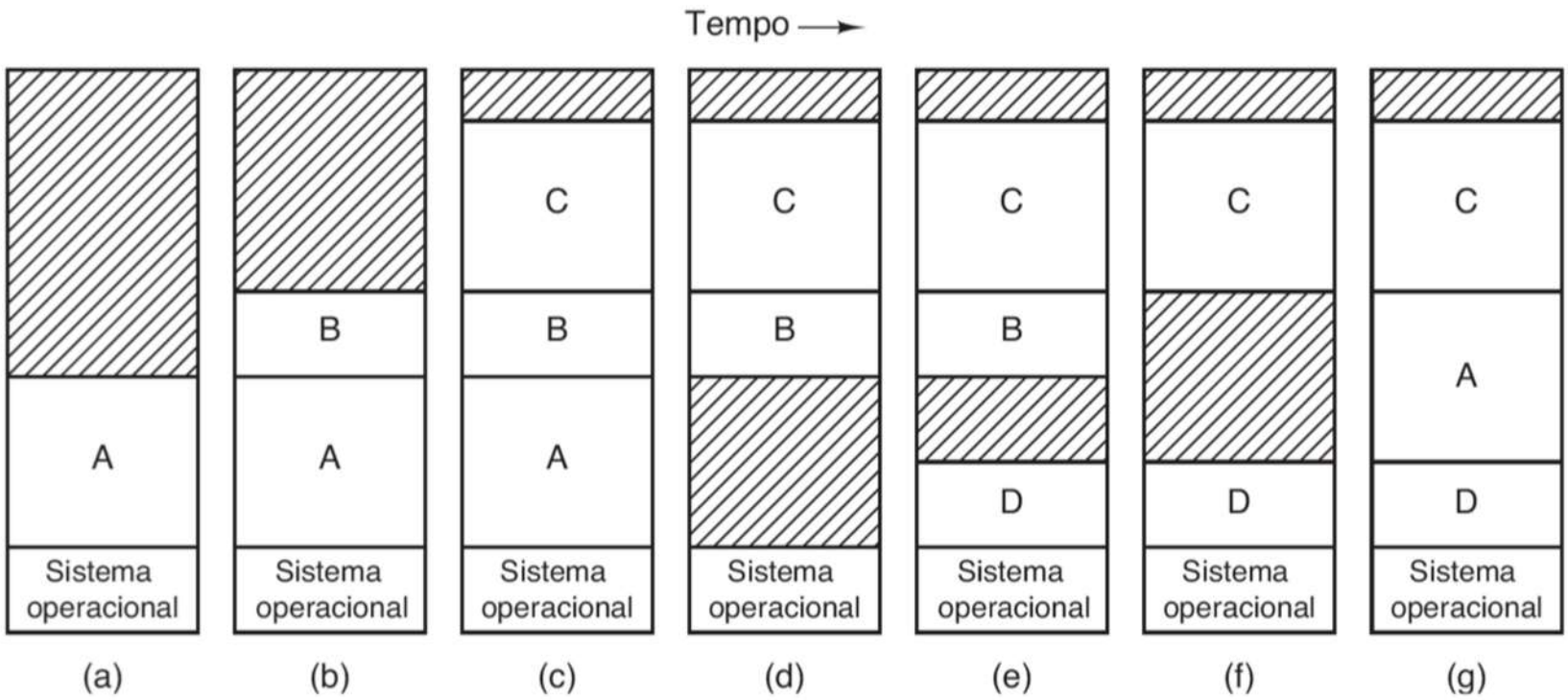
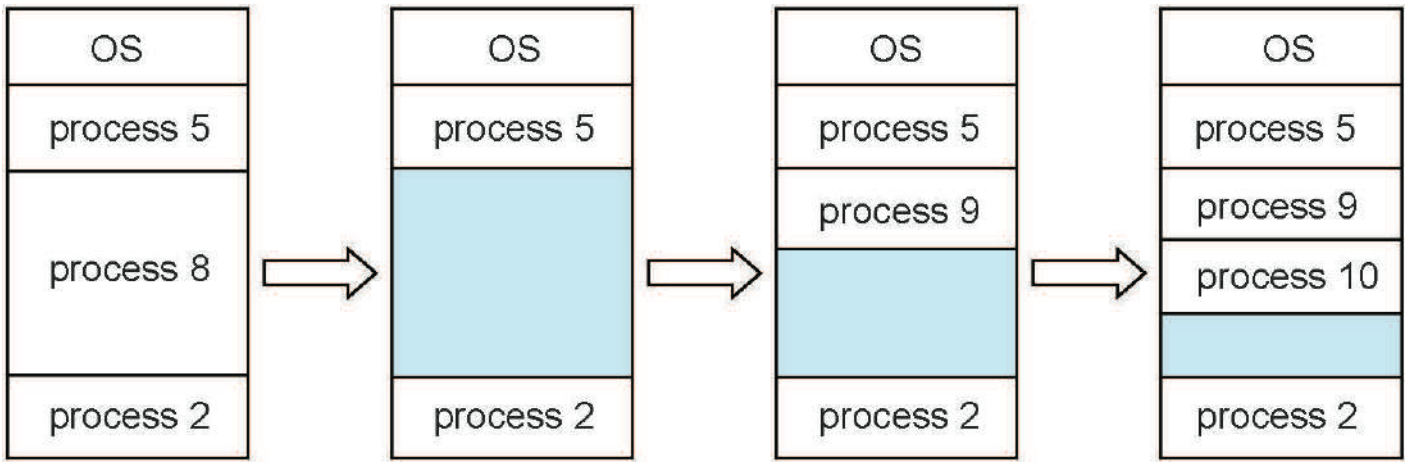


Figura 3.4 Alterações na alocação de memória à medida que processos entram e saem dela. As regiões sombreadas correspondem a regiões da memória não utilizadas naquele instante.


OS
process 5
process 8
process 2
OS
process 5
process 2
OS
process 5
process 9
process 2
OS
process 5
process 9
process 10
process 2

Gerenciamento de Memória com Mapa de Bits

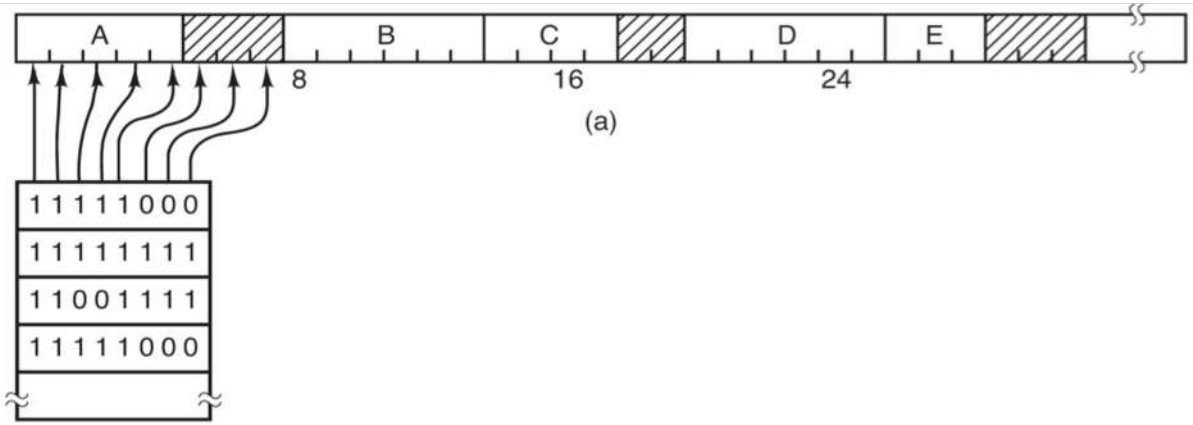


Figura 3.6 (a) Parte da memória com cinco processos e três segmentos de memória. As marcas mostram as unidades de alocação de memória. As regiões sombreadas (0 no mapa de bits) estão livres. (b) O mapa de bits correspondente. (c) A mesma informação como lista.

Como gerenciar memória?

Paginação
Segmentação

Paginação

Paginação

permite que o programa possa ser espalhado por áreas não contíguas de memória, ou seja, é utilizado para um processo, endereços de memória física não contíguos

Quadros = blocos de memória física;
Páginas = blocos de memória lógica;

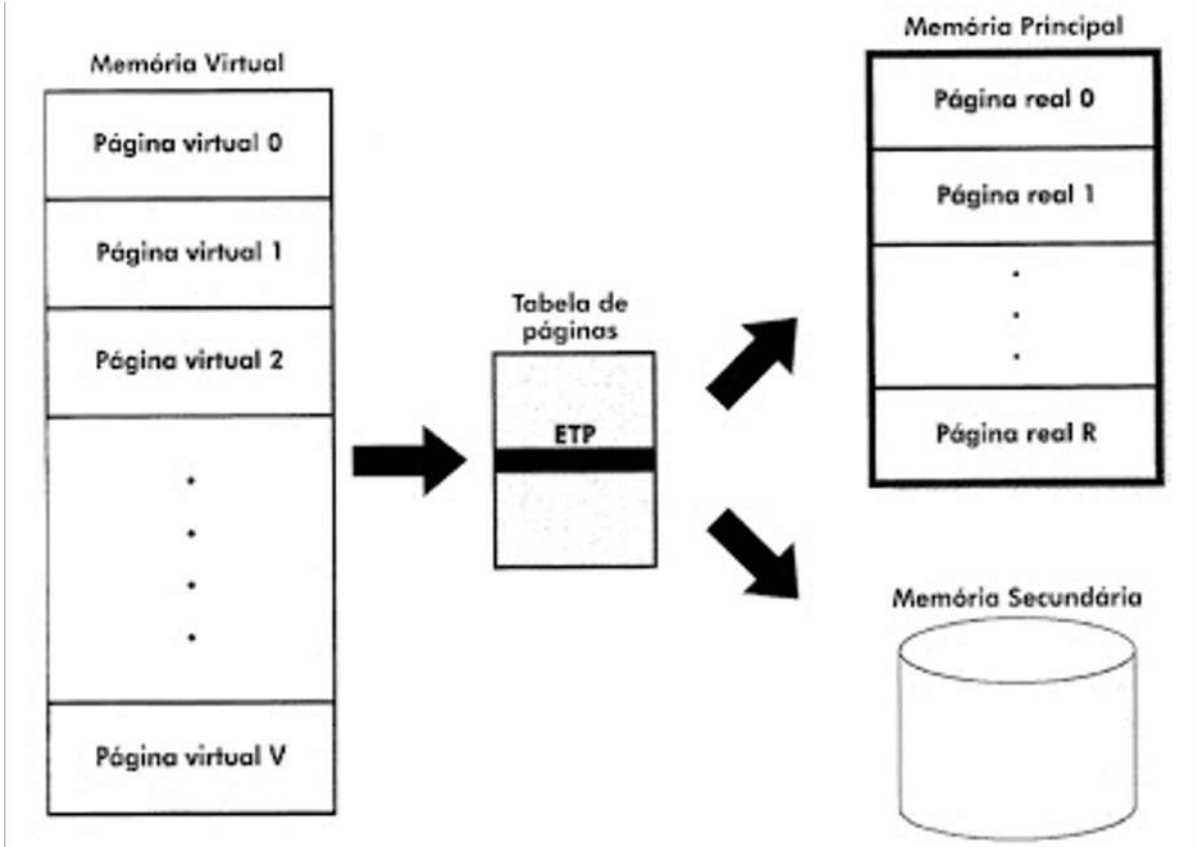


Figura 6: Tabela de páginas

Memória Virtual - Paginação

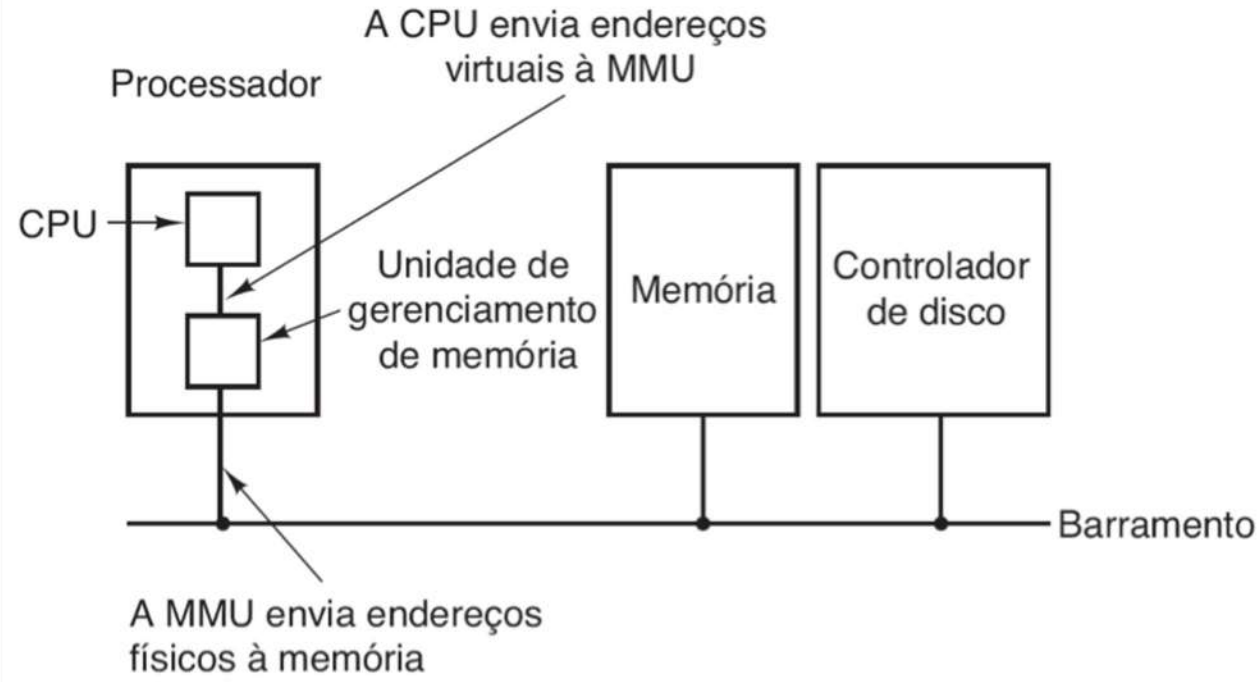


Figura 3.8 A posição e a função da MMU. Aqui a MMU é mostrada como parte do chip da CPU (processador) porque isso é comum atualmente. Contudo, em termos lógicos, poderia ser um chip separado, como ocorria no passado.

End. lógico
x
end. físico

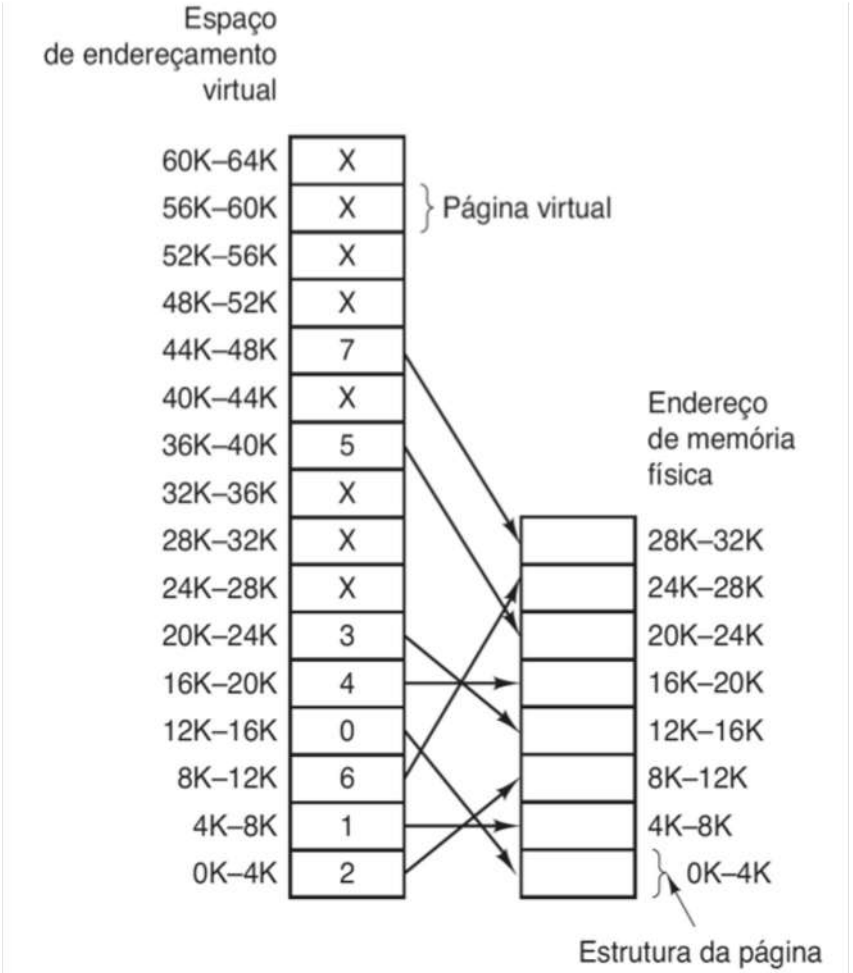
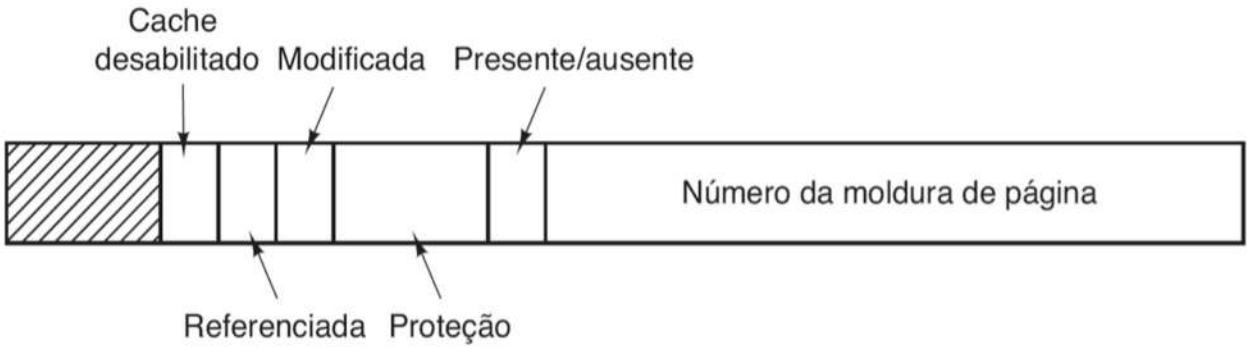


Figura 3.9 A relação entre endereços virtuais e endereços de memória física é dada pela tabela de páginas. Cada página começa com um múltiplo de 4096 e termina 4095 endereços acima; assim, 4K–8K na verdade significa 4096–8191 e 8K–12K significa 8192–12287.

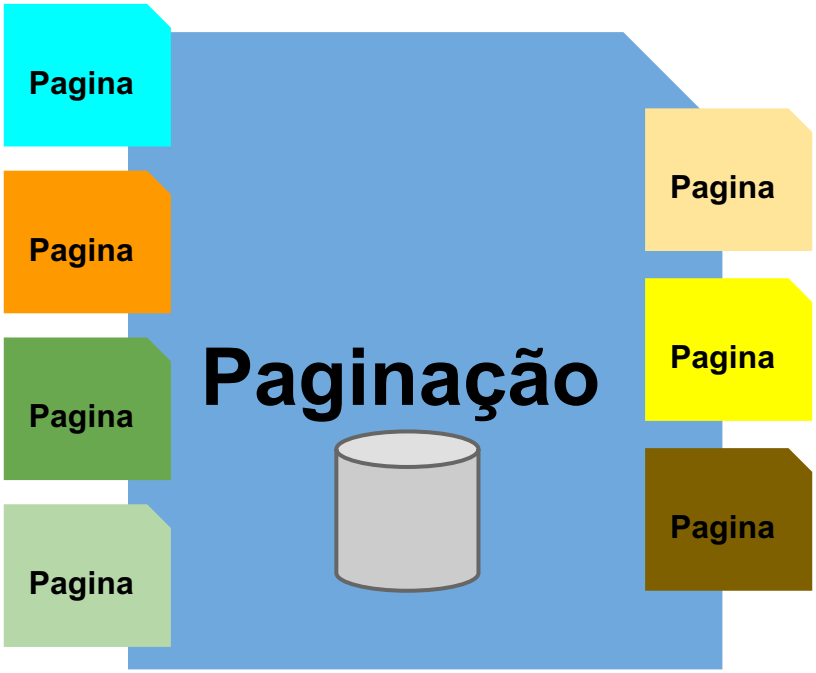
Entrada tabela de páginas



O que pode acontecer durante paginação? Problemas!

- O mapeamento do endereço virtual para o endereço físico deve ser rápido.
- Se o espaço virtual for grande, a tabela de páginas será grande.

Gerenciamento de Memória



Como substituir as páginas?

Algoritmos de subsistuição de páginas

Algoritmo ótimo de substituição de página.

Algoritmo de substituição de página não usado recentemente.

Algoritmo de substituição de página primeiro a entrar, primeiro a sair.

Algoritmo de substituição de página segunda chance.

Algoritmo de substituição de página de relógio.

Algoritmo de substituição de página menos usada recentemente.

Algoritmo ótimo

Seleciona para substituição uma página que não será mais referenciada no futuro ou aquela que levará o maior intervalo de tempo para ser novamente utilizada.

- Difícil de implementar (não há como prever o comportamento futuro das aplicações)
- Utilizado apenas como modelo comparativo na análise de outros algoritmos de substituição.

Algoritmo aleatório

O algoritmo aleatório não utiliza critério algum de seleção.

Todas as páginas alocadas na memória principal têm a mesma chance de serem selecionadas, inclusive os frames que são freqüentemente referenciados.

Embora consuma pouco recurso do sistema raramente é implementada devido sua baixa eficiência

Algoritmo FIFO

- Mesma abordagem de qualquer FIFO (modelo fila): a página que está há mais tempo na memória será selecionada.

- Raramente implementado sem algum outro mecanismo que minimize o problema da seleção de páginas antigas que são constantemente referenciadas.

Algoritmo Segunda Chance

- FIFO com utilização do bit de referência;
  - Bit de referência de página
    - Recurso provido pelo hardware que o posiciona sempre que a página é referenciada;

- Bit de referência = 0, página substituída
- Bit de referência = 1, página mantida (segunda chance)
  - Bit de referênca é alterado para 0
  - Hora de chegada é redefinida para hora corrente

Algoritmo Segunda Chance

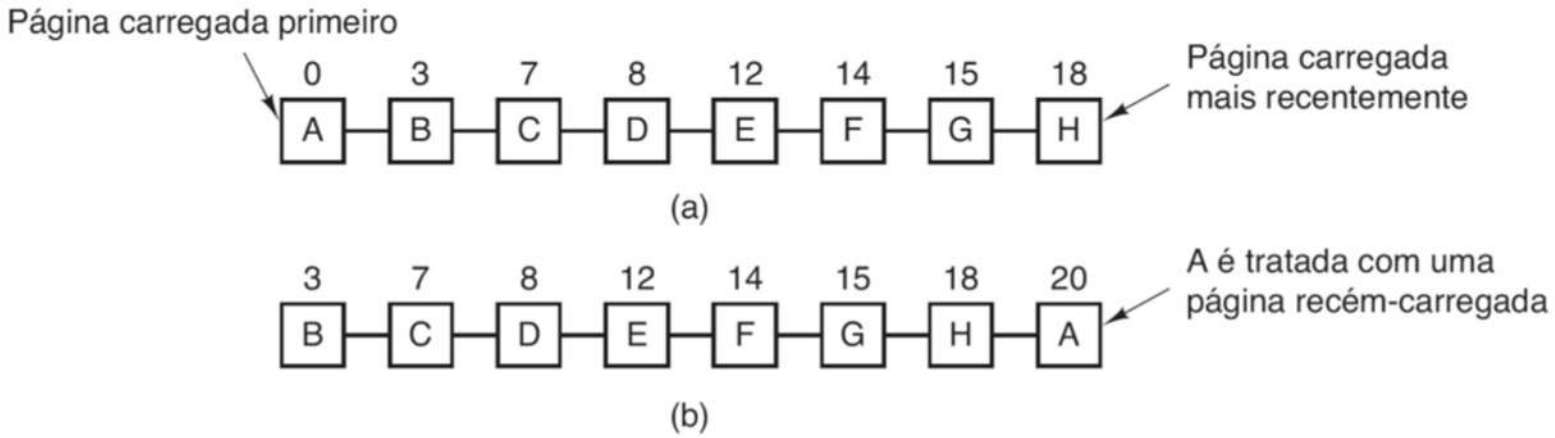


Figura 3.14 Operação de segunda chance. (a) Páginas na ordem FIFO. (b) Lista de páginas se uma falta de página ocorre no tempo 20 e o bit *R* de *A* possui o valor 1. Os números acima das páginas são seus tempos de carregamento.

Página com menor frequência de utilização
LFU (Least-frequently-used)

- ➔ Seleciona a página menos referenciada, ou seja, o frame menos utilizado.
- ➔ A página que possuir o contador com o menor número de referências será a escolhida, ou seja, o algoritmo evita selecionar páginas que são bastante utilizadas.
- ➔ Páginas que estão há pouco tempo na memória principal podem ser são mais prováveis de serem selecionadas pois seus contadores estarão com o menor número de referências.
- ➔ Página muito utilizada no passado e que não seja mais referenciada no futuro pode "ficar"por muito tempo.
- ➔ Serve apenas de base para outros algoritmos de substituição.

Página	No. Ref
A	3
B	4
C	1
D	8
E	9

Menos utilizada recentemente
LRU (Least-recently-used)

- ➔ **Seleciona a página na memória principal que está há mais tempo sem ser referenciada.**
- ➔ Considerando a localidade, uma página que não foi utilizada recentemente provavelmente não será referenciada novamente em um futuro próximo.
- ➔ Pode ser implementada utilizando abordagem de contadores ou pilha;
- ➔ Alto custo de implementação;

Página	Último acesso*
A	MOMENTO 0
B	MOMENTO 6
C	MOMENTO 1
D	MOMENTO 9
E	MOMENTO 4

* atualizado a cada referência a um frame

Não utilizada recentemente
NRU (not-recently-used)

- LRU menos "sofisticada"
- Necessita um bit adicional, conhecido como bit de referência (BR).
- BR indica se a página foi utilizada recentemente e está presente em cada entrada da tabela de páginas.
- BR referência é alterado pelo hardware, indicando que a página foi referenciada (BR = 1). Periodicamente, o sistema altera o valor do bit de referência (BR = 0), e à medida que as páginas são utilizadas, o bit associado a cada frame retorna para 1.
- Assim, é possível distinguir quais frames foram recentemente referenciados.

Não utilizada recentemente
NRU (not-recently-used)

A substituição de uma página, o sistema seleciona um dos frames que não tenha sido utilizado recentemente, ou seja, com o bit de referência igual a zero. O algoritmo NRU torna-se mais eficiente se o bit de modificação for utilizado em conjunto com o bit de referência.

As páginas em quatro categorias, conforme a tabela a seguir:

Categorias	*Bits avaliados*	*Resultado*
1	*BR = 0 e BM = 0*	*Página não referenciada e não modificada*
2	*BR = 0 e BM = 1*	*Página não referenciada e modificada*
3	*BR = 1 e BM = 0*	*Página referenciada e não modificada*
4	*BR = 1 e BM = 1*	*Página referenciada e modificada*

Síntese dos algoritmos de substituição de páginas

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usada recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usada)	Aproximação bastante rudimentar do LRU
Envelhecimento (*aging*)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente

Tabela 3.2 Algoritmos de substituição de página discutidos no texto.

Tratamento de falta de página

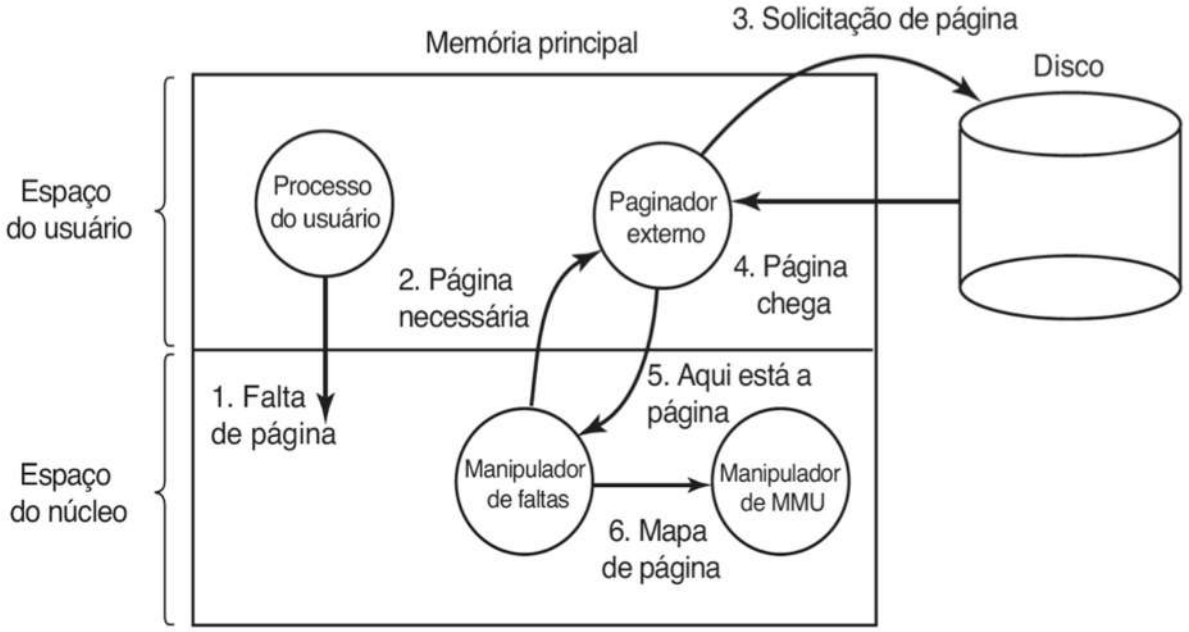


Figura 3.28 Tratamento de falta de página com um paginador externo.