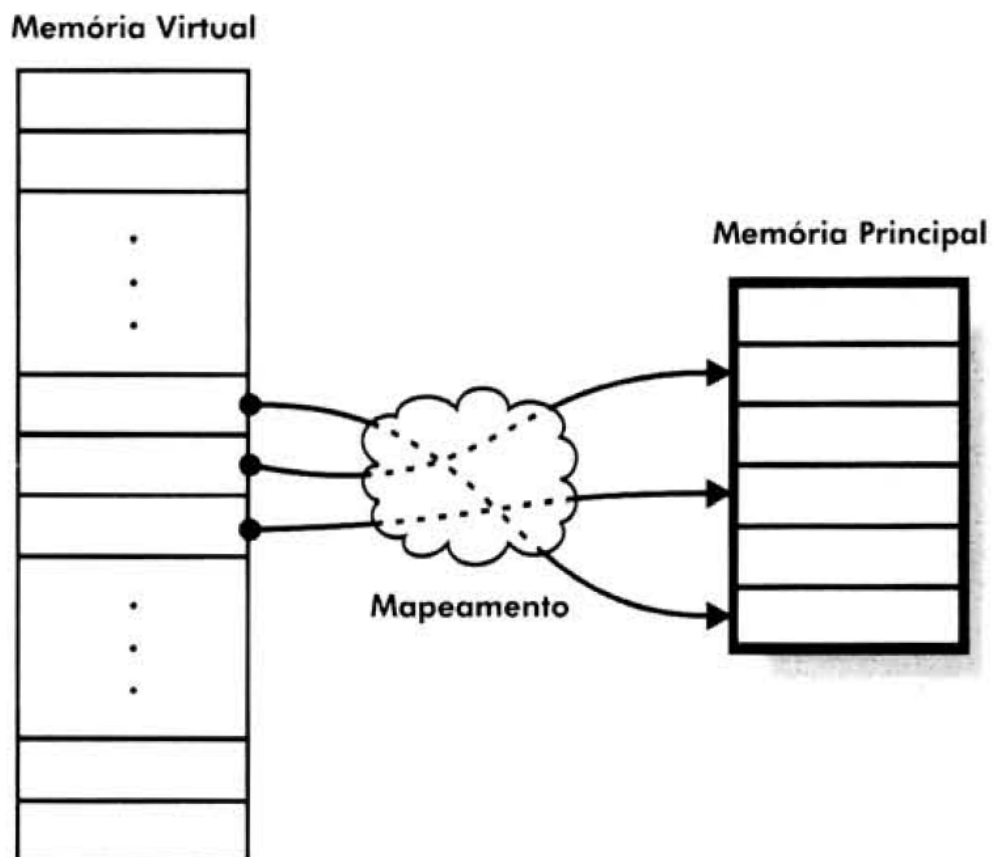
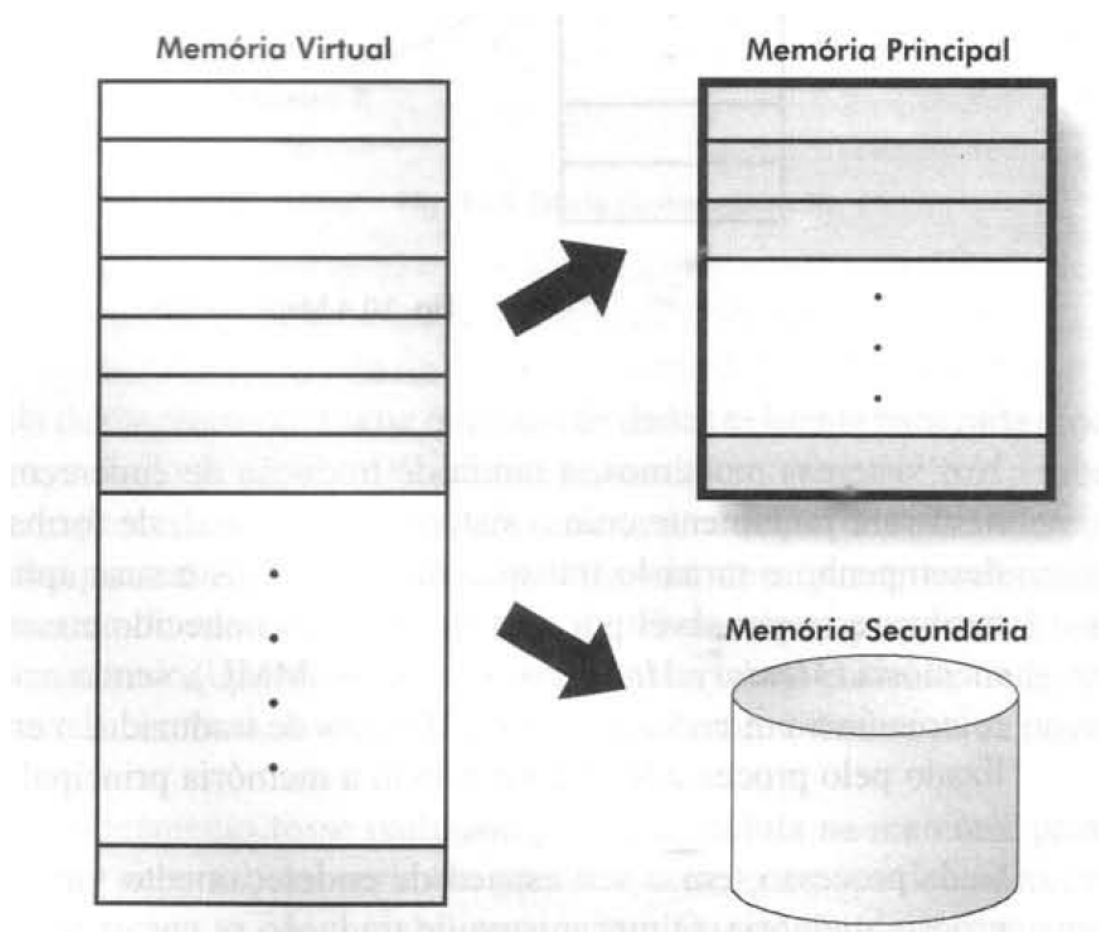


Gerenciamento de Memória

Espaço de Endereçamento Virtual



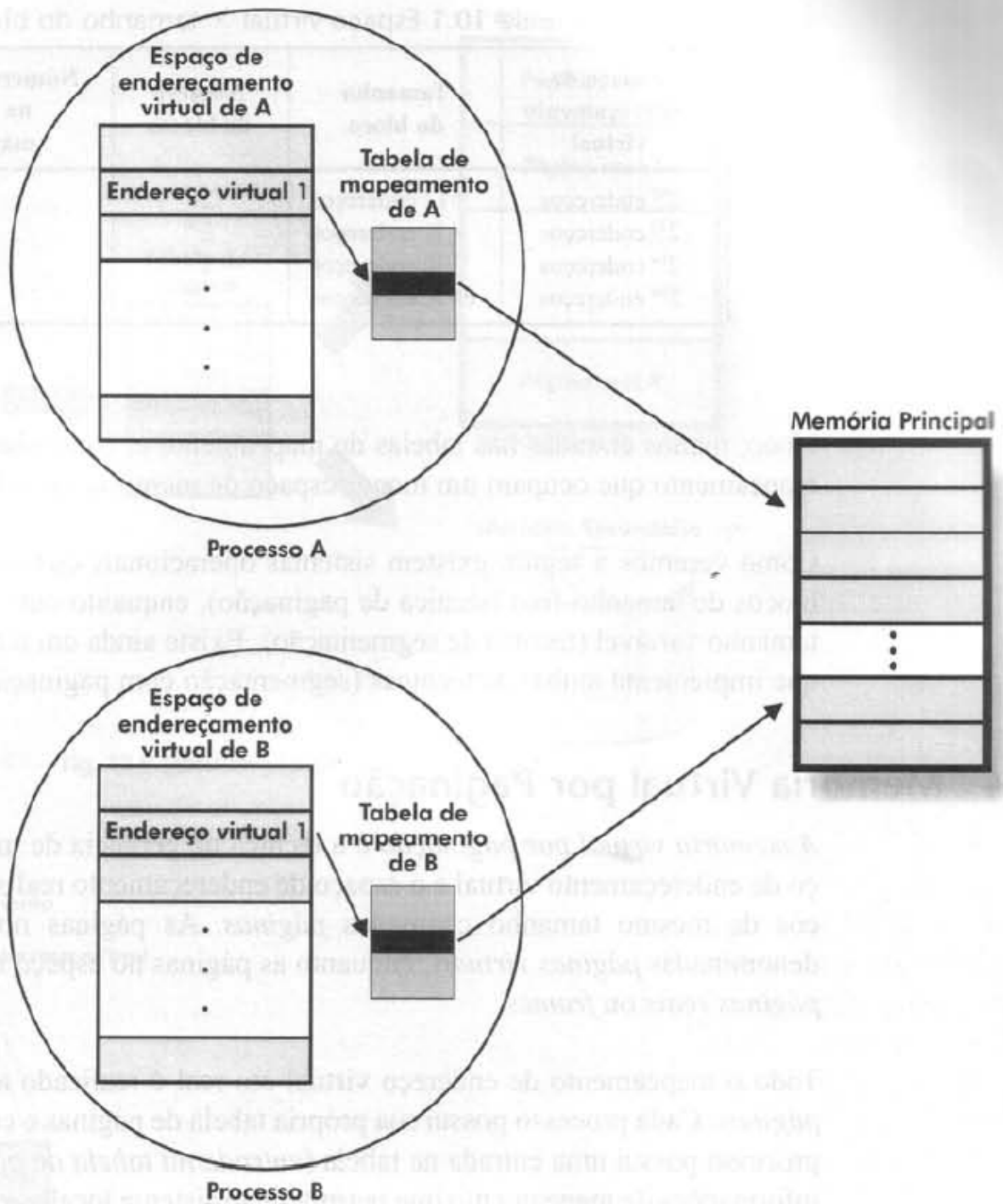


Fig. 10.5 Tabela de mapeamento.

Endereçamento virtual por Paginação

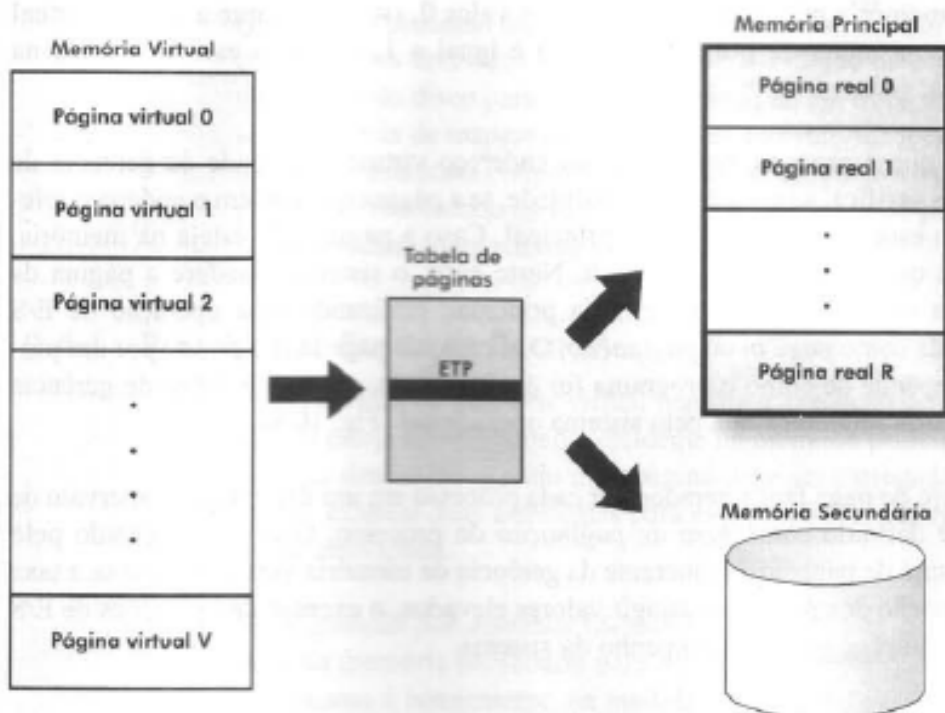


Fig. 10.6 Tabela de páginas.

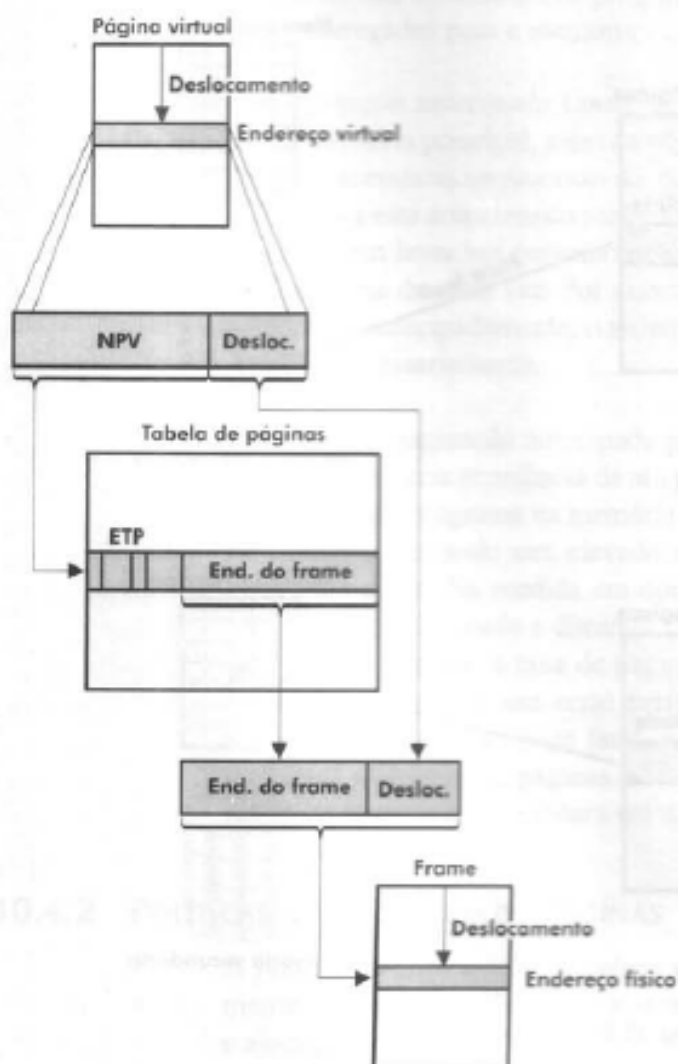


Fig. 10.7 Tradução do endereço virtual.

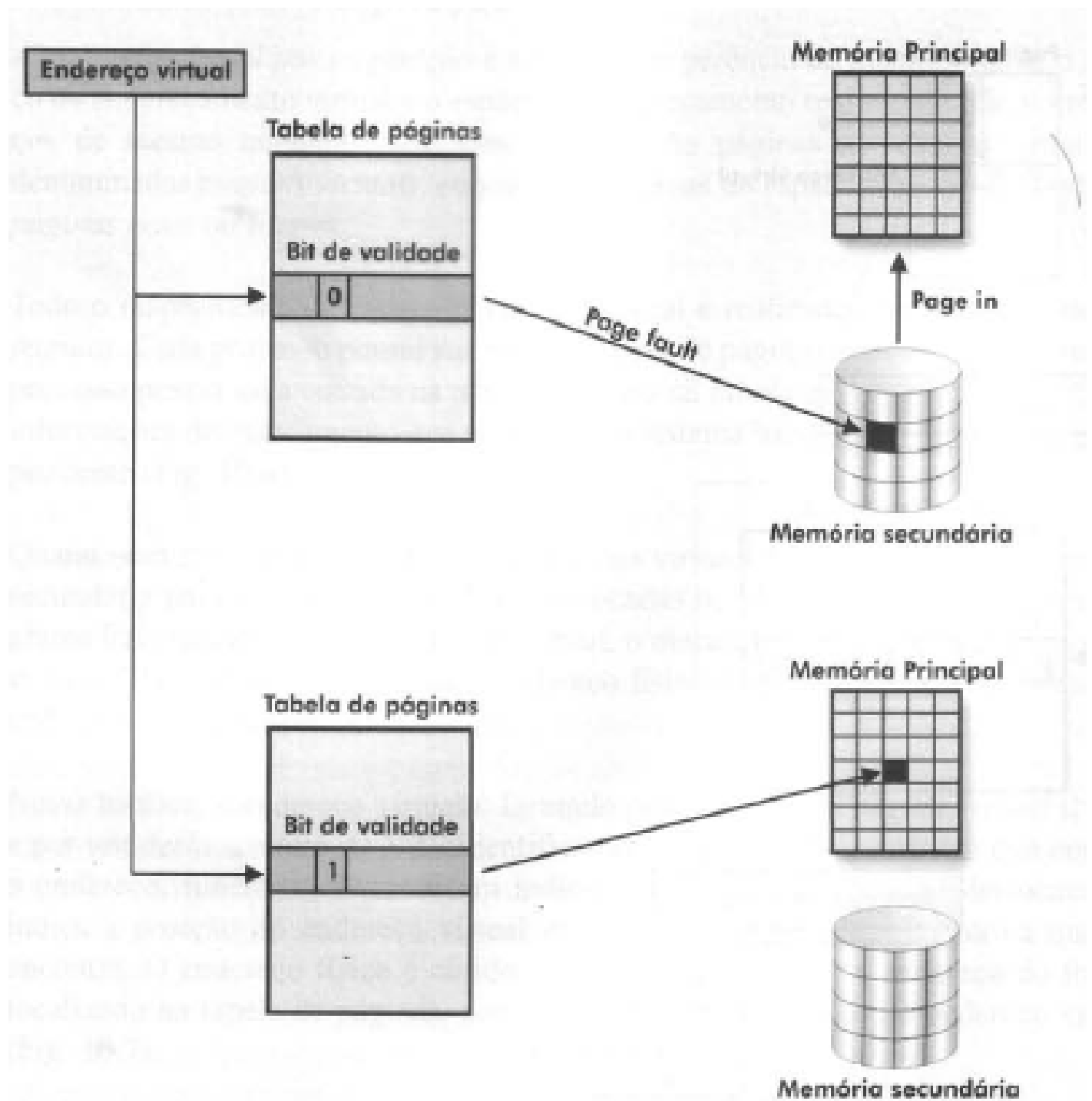


Fig. 10.8 Mecanismo de tradução.

Política de Busca de Páginas

O mecanismo de memória virtual permite a execução de um programa sem que seu código esteja completamente residente na memória principal. A *política de busca de páginas* determina quando uma página deve ser carregada para a memória. Basicamente, existem duas estratégias para este propósito: paginação por demanda e paginação antecipada.

Na *paginação por demanda* (*demand paging*), as páginas dos processos são transferidas da memória secundária para a principal apenas quando são referenciadas. Este mecanismo é conveniente, na medida em que leva para a memória principal apenas as páginas realmente necessárias à execução do programa. Desse modo, é possível que partes não executadas do programa, como rotinas de tratamento de erros, nunca sejam carregadas para a memória.

Na *paginação antecipada* (*anticipatory paging ou prepaging*), o sistema carrega para a memória principal, além da página referenciada, outras páginas que podem ou não ser necessárias ao processo ao longo do seu processamento. Se imaginarmos que o programa está armazenado seqüencialmente no disco, existe uma grande economia de tempo em levar um conjunto de páginas da memória secundária, ao contrário de carregar uma de cada vez. Por outro lado, caso o processo não precise das páginas carregadas antecipadamente, o sistema terá perdido tempo e ocupado memória principal desnecessariamente.

A técnica de paginação antecipada pode ser empregada no momento da criação de um processo ou na ocorrência de um page fault. Quando um processo é criado, diversas páginas do programa na memória secundária devem ser carregadas para a memória principal, gerando um elevado número de page faults e várias operações de leitura em disco. Na medida em que as páginas são carregadas para a memória, a taxa de paginação tende a diminuir. Se o sistema carregar não apenas uma, mas um conjunto de páginas, a taxa de paginação do processo deverá cair imediatamente e estabilizar-se durante um certo período de tempo. Seguindo o mesmo raciocínio, sempre que houver um page fault, o sistema poderá carregar para a memória, além da página referenciada, páginas adicionais, na tentativa de evitar novos page faults e sucessivas operações de leitura em disco.

Políticas de Alocação de Páginas

A *política de alocação de páginas* determina quantos frames cada processo pode manter na memória principal. Existem, basicamente, duas alternativas: alocação fixa e alocação variável.

Na *política de alocação fixa*, cada processo tem um número máximo de frames que pode ser utilizado durante a execução do programa. Caso o número de páginas reais seja insuficiente, uma página do processo deve ser descartada para que uma nova seja carregada. O *limite de páginas reais* pode ser igual para todos os processos ou definido individualmente. Apesar de parecer justo, alocar o mesmo número de páginas para todos os processos, pode não ser uma boa opção, pois a necessidade de memória de cada processo raramente é a mesma. O limite de páginas deve ser definido no momento da criação do processo, com base no tipo da aplicação que será executada. Essa informação faz parte do contexto de software do processo.

Apesar de sua simplicidade, a política de alocação fixa de páginas apresenta dois problemas. Se o número máximo de páginas alocadas for muito pequeno, o processo tenderá a ter um elevado número de page faults, o que pode impactar no desempenho de todo o sistema. Por outro lado, caso o número de páginas seja muito grande, cada processo irá ocupar na memória principal um espaço maior do que o necessário, reduzindo o número de processos residentes e o grau de multiprogramação. Nesse caso, o sistema pode implementar a técnica de swapping, retirando e carregando processos da/para a memória principal.

Na *política de alocação variável*, o número máximo de páginas alocadas ao processo pode variar durante sua execução em função de sua taxa de paginação e da ocupação da memória principal. Nesse modelo, processos com elevadas taxas de paginação podem ampliar o limite máximo de frames, a fim de reduzir o número de page faults. Da mesma forma, processos com baixas taxas de paginação podem ter páginas realocadas para outros processos. Este mecanismo, apesar de mais flexível, exige que o sistema operacional monitore constantemente o comportamento dos processos, gerando maior overhead.

Políticas de Substituição de Páginas

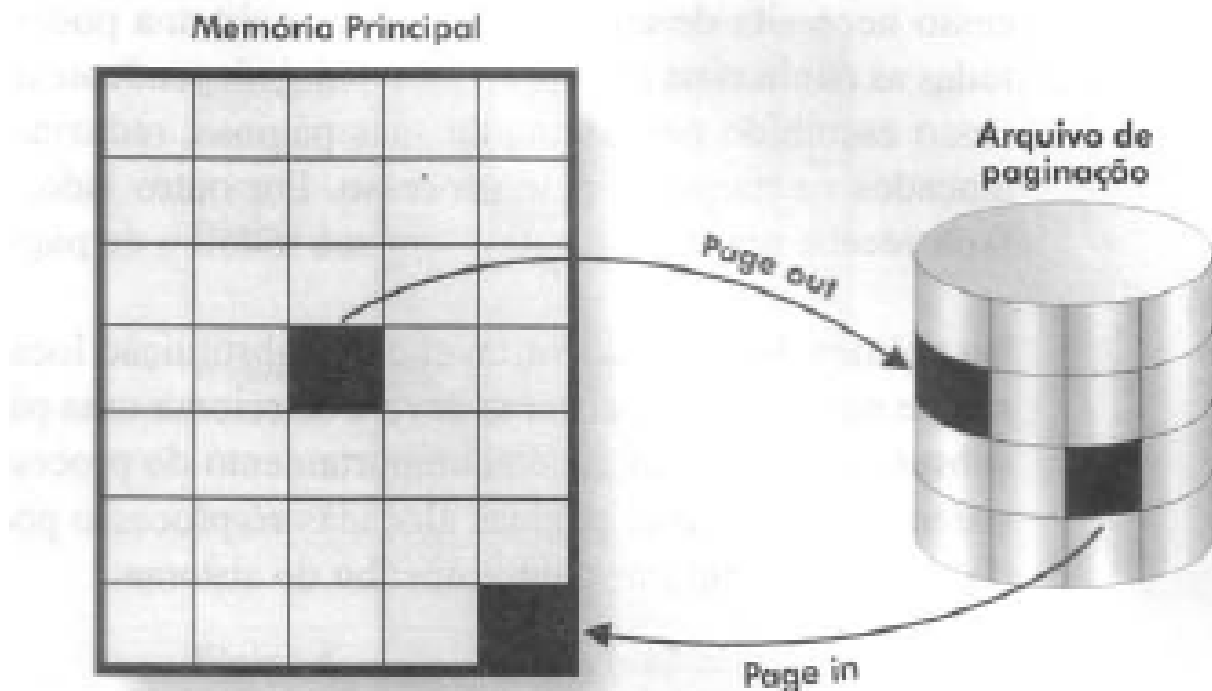


Fig. 10.9 Substituição de páginas.

Na *política de substituição local*, apenas as páginas do processo que gerou o page fault são candidatas a realocação. Nesse modelo, sempre que um processo precisar de uma nova página, o sistema deverá selecionar, dentre os frames alocados pelo processo, a página a ser substituída. Os frames dos demais processos não são avaliados para substituição.

Já na *política de substituição global*, todas as páginas alocadas na memória principal são candidatas a substituição, independente do processo que gerou o page fault. Como qualquer processo pode ser escolhido, é possível que o processo selecionado sofra um aumento na sua taxa de paginação, em função da redução do número de páginas alocadas na memória. Na verdade, nem todas as páginas podem ser candidatas a substituição. Algumas páginas, como as do núcleo do sistema, são marcadas como bloqueadas e não podem ser realocadas.

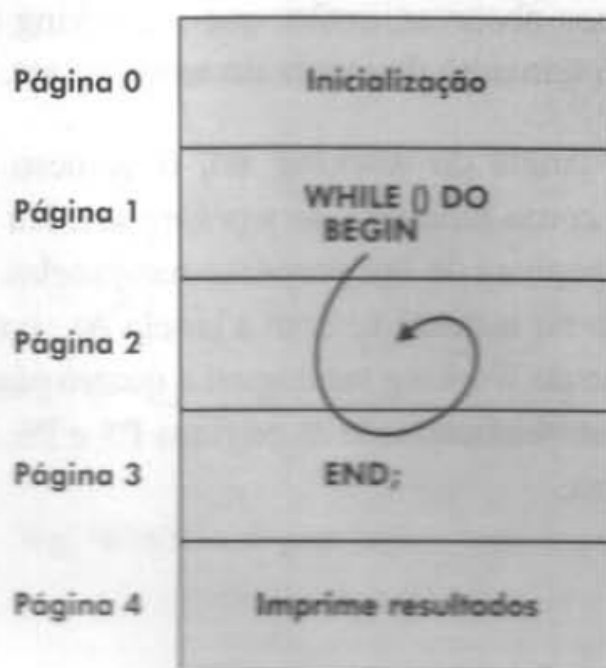
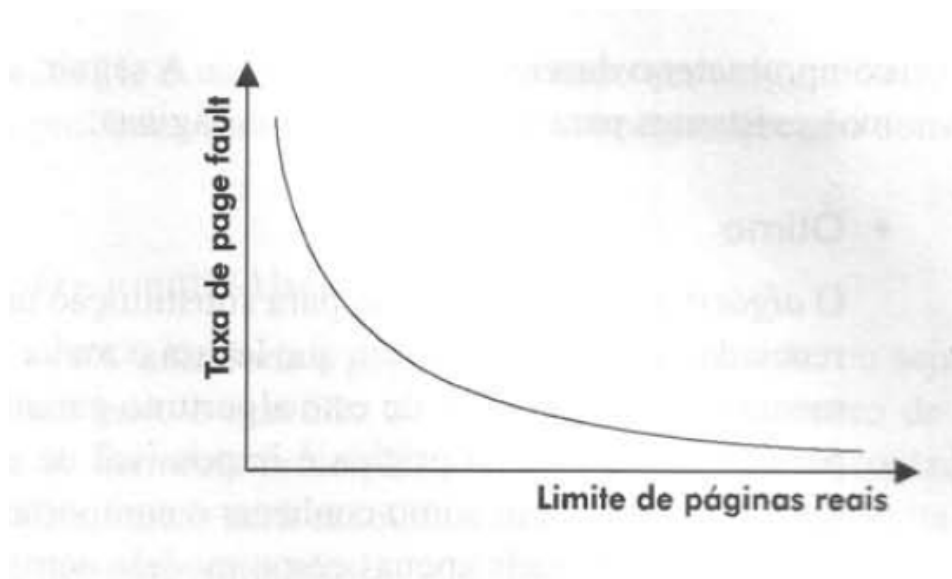


Fig. 10.10 Conceito de localidade.

Fig. 10.13 Taxa de page fault \times limite de páginas reais.

Uma maneira de implementar o modelo de working set é analisar a taxa de paginação de cada processo, conhecida como *estratégia de frequência de page fault*. Caso um processo tenha uma taxa de paginação acima de um limite definido pelo sistema, o processo deverá aumentar o seu limite de páginas reais na tentativa de alcançar o seu working set. Por outro lado, se o processo tem uma taxa de paginação abaixo de um certo limite, o sistema poderá reduzir o limite de páginas sem comprometer seu desempenho.

Algoritmos de Substituição de Páginas

O maior problema na gerência de memória virtual por paginação não é decidir quais páginas carregar para a memória principal, mas quais liberar (Denning, 1968). Quando um processo necessita de uma nova página e não existem frames disponíveis, o sistema deverá selecionar, dentre as diversas páginas alocadas na memória, qual deverá ser liberada pelo processo.

Os *algoritmos de substituição de páginas* têm o objetivo de selecionar os frames que tenham as menores chances de serem referenciados em um futuro próximo; caso contrário, o frame poderia retornar diversas vezes para a memória principal, gerando vários page faults e acessos à memória secundária. A partir do princípio da localidade, a maioria dos algoritmos tenta prever o comportamento futuro das aplicações em função do comportamento passado, avaliando o número de vezes que uma página foi referenciada, o momento em que foi carregada para a memória principal e o intervalo de tempo da última referência.

- Ótimo
- Aleatório
- FIFO (First-In-First-Out)
- LFU (Least-Frequently-Used)
- LRU (Least-Recently-Used)
- NRU (Not-Recently-Used)

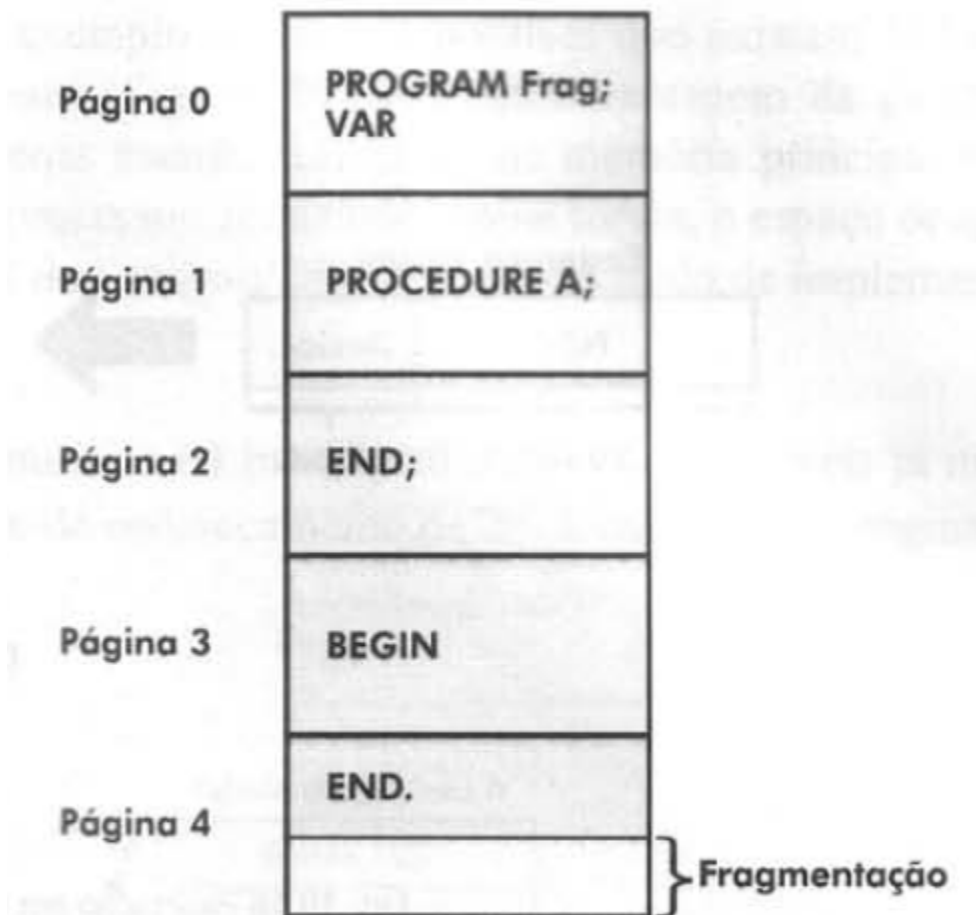
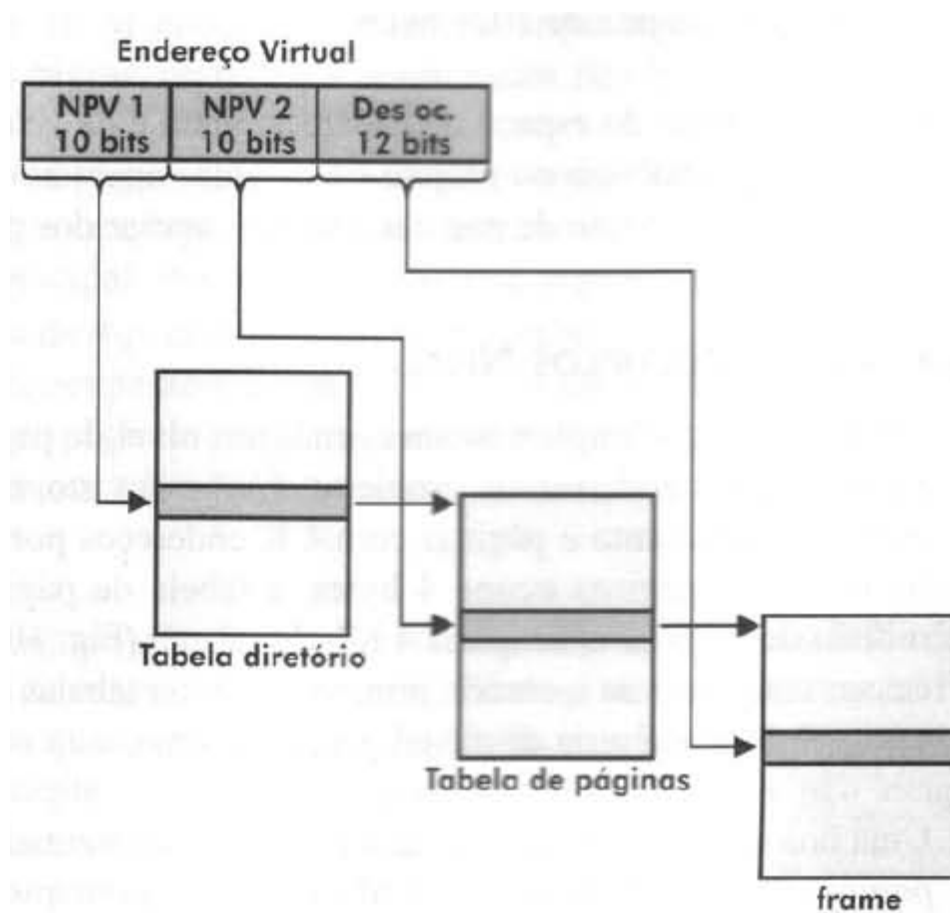
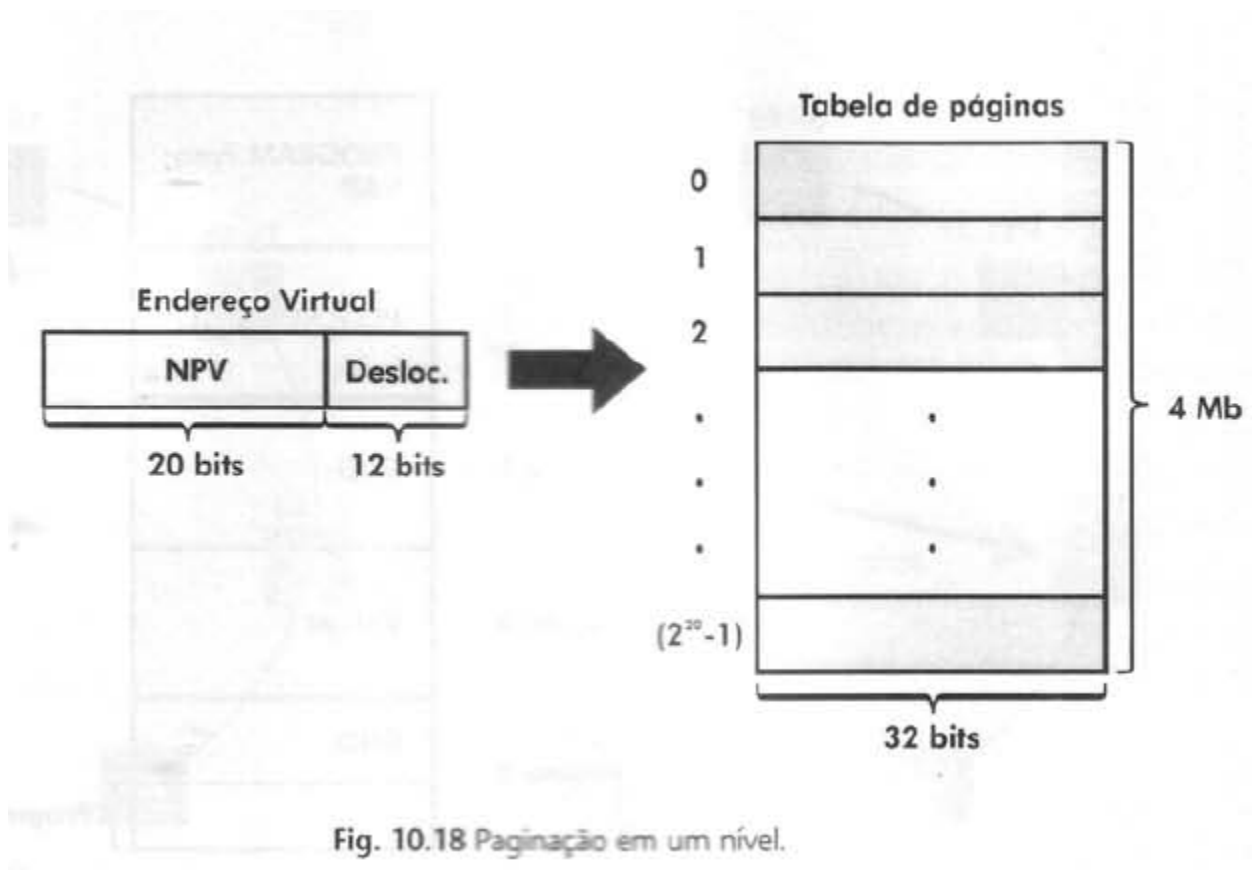


Fig. 10.17 Fragmentação interna.

Tabela 10.1 Espaço virtual × tamanho do bloco

Espaço de endereçamento virtual	Tamanho do bloco	Número de blocos	Número de entradas na tabela de mapeamento
2^{32} endereços	512 endereços	2^{23}	2^{23}
2^{32} endereços	4 K endereços	2^{20}	2^{20}
2^{64} endereços	4 K endereços	2^{52}	2^{52}
2^{64} endereços	64 K endereços	2^{48}	2^{48}

Paginação em um ou dois níveis



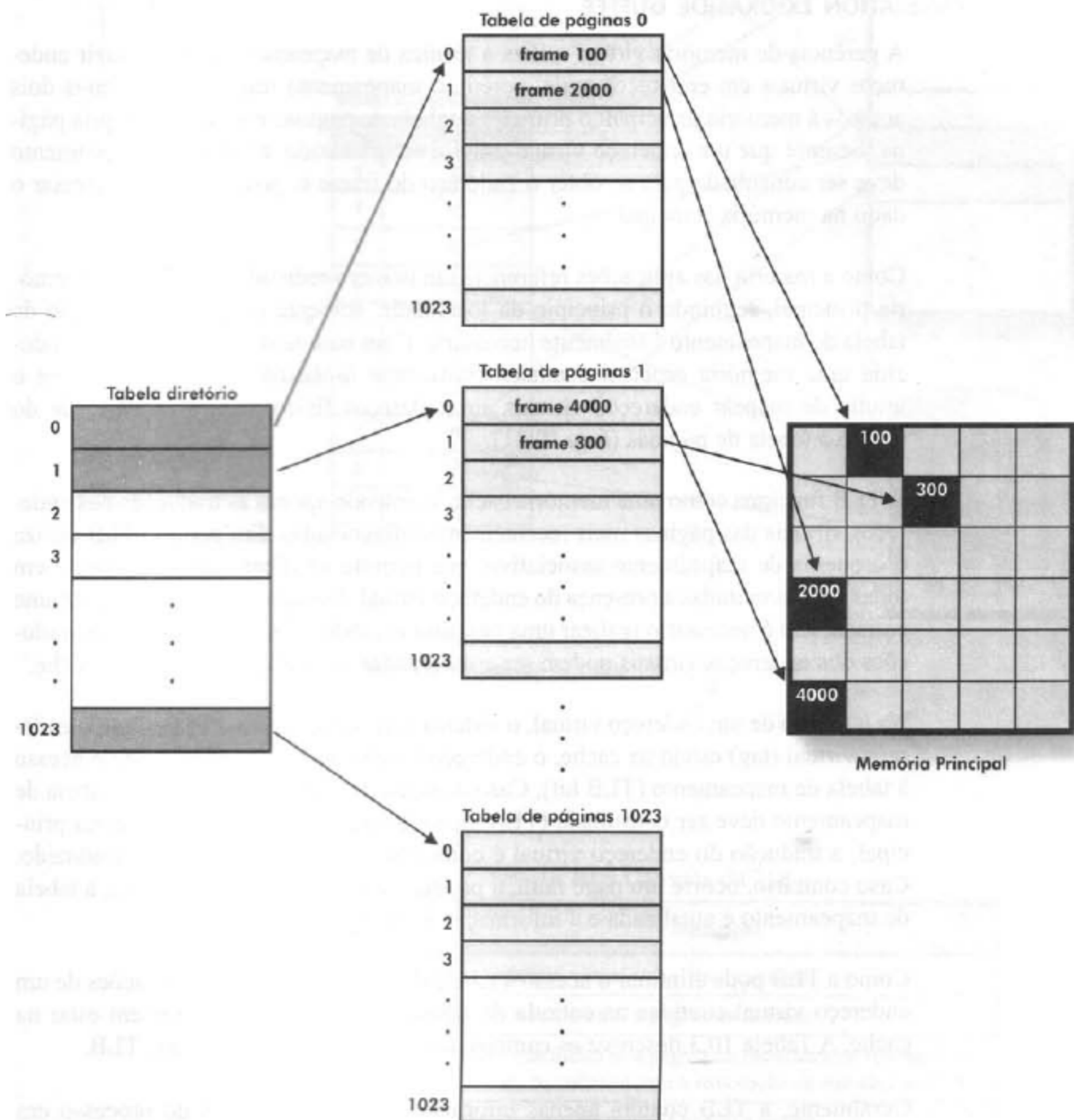


Fig. 10.20 Paginação em dois níveis.

Em uma arquitetura de 64 bits, a estrutura em dois níveis já não é mais adequada devido ao espaço de endereçamento de 2^{64} . Considerando páginas de 4 K endereços, NPV2 com 10 bits e NPV1 com 42 bits, teríamos tabelas com 2^{42} entradas. Novamente a solução passa por dividir a tabela diretório, criando uma estrutura em três níveis. Por exemplo, NPV1 com 32 bits, NPV2 com 10 bits e NPV3 com 10 bits.

Translation Lookaside Buffer (TLB)

A gerência de memória virtual utiliza a técnica de mapeamento para traduzir endereços virtuais em endereços reais, porém, o mapeamento implica pelo menos dois acessos à memória principal: o primeiro à tabela de páginas e o outro à própria página. Sempre que um endereço virtual precisa ser traduzido, a tabela de mapeamento deve ser consultada para se obter o endereço do frame e, posteriormente, acessar o dado na memória principal.

Como a maioria das aplicações referencia um número reduzido de frames na memória principal, seguindo o princípio da localidade, somente uma pequena fração da tabela de mapeamento é realmente necessária. Com base neste princípio, foi introduzida uma memória especial chamada *translation lookaside buffer* (TLB), com o intuito de mapear endereços virtuais em endereços físicos sem a necessidade do acesso à tabela de páginas (Fig. 10.21).

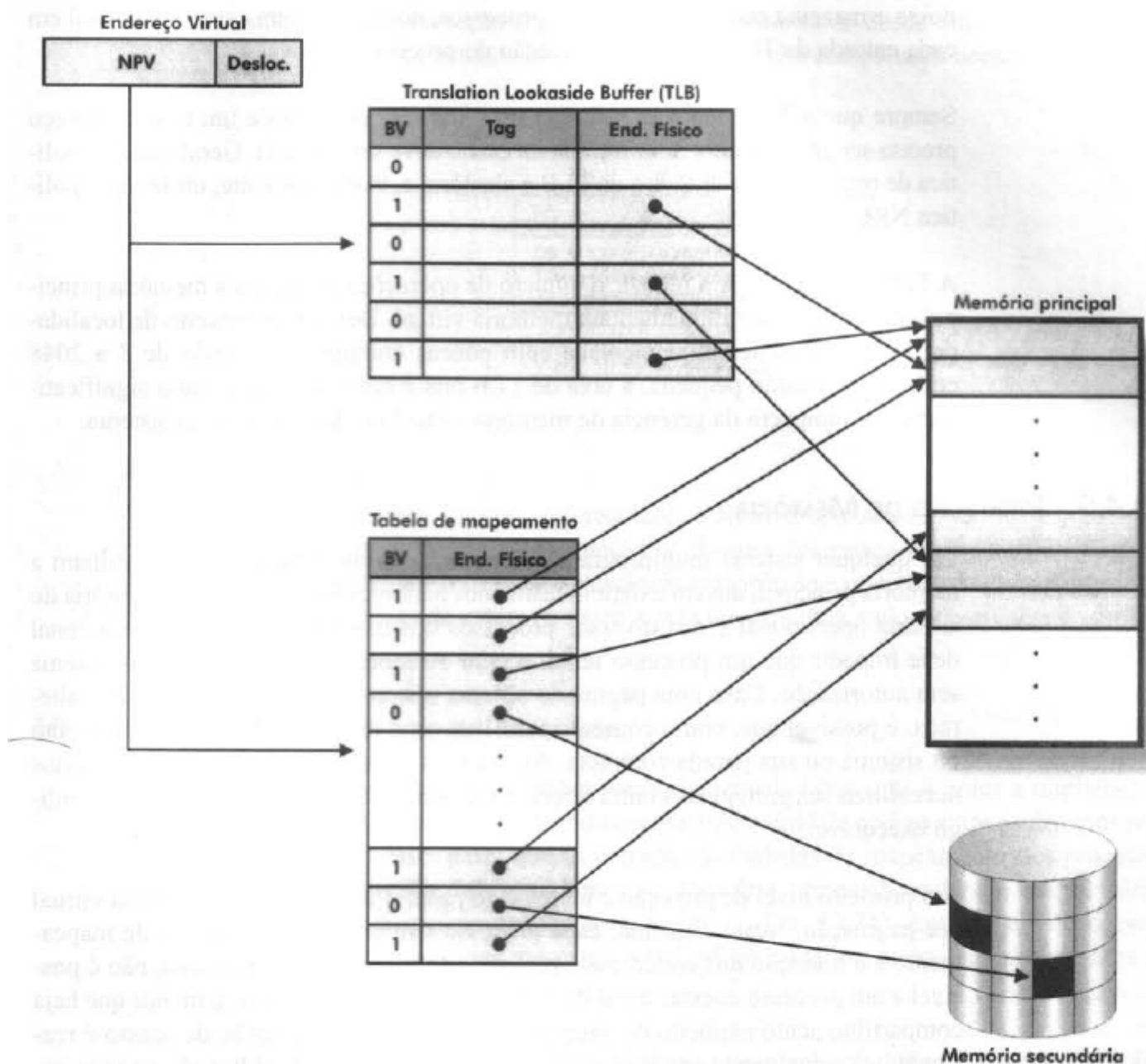


Fig. 10.21 Translation lookaside buffer (TLB).

Tabela 10.3 Campos da TLB

Campo	Descrição
Tag	Endereço virtual sem o deslocamento.
Modificação	Bit que indica se a página foi alterada.
Referência	Bit que indica se a página foi recentemente referenciada, sendo utilizada para a realocação de entradas na TLB.
Proteção	Define a permissão de acesso à página.
Endereço físico	Posição do frame na memória principal.

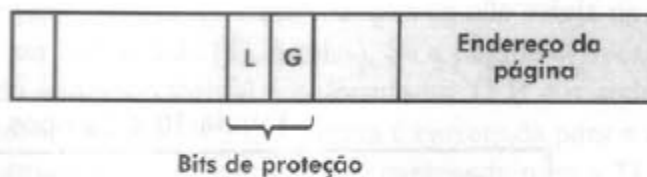


Fig. 10.22 Proteção para páginas.

Endereçamento Virtual por Segmentação

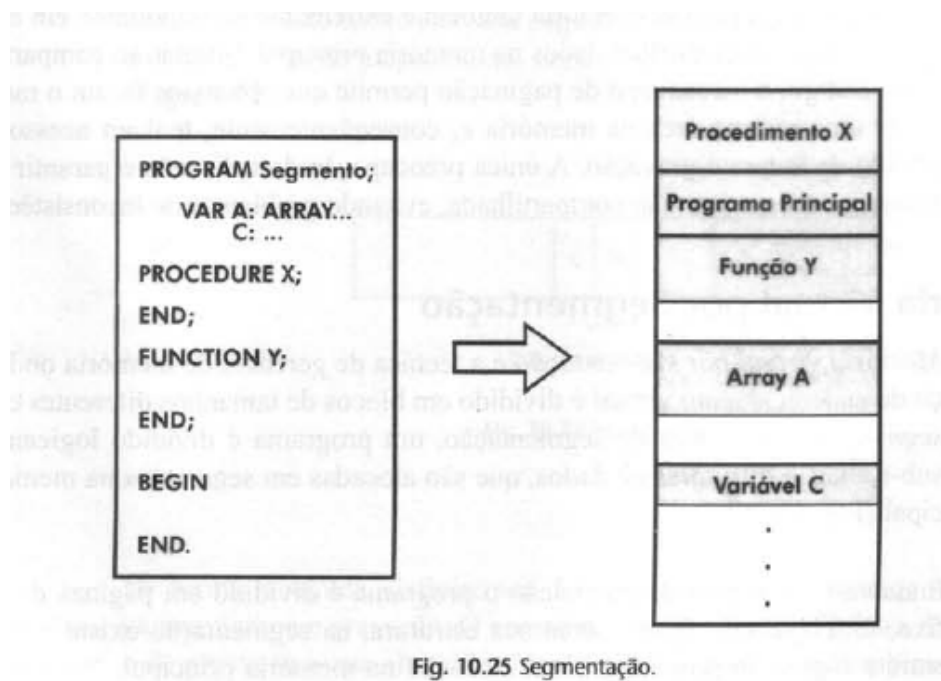


Fig. 10.25 Segmentação.

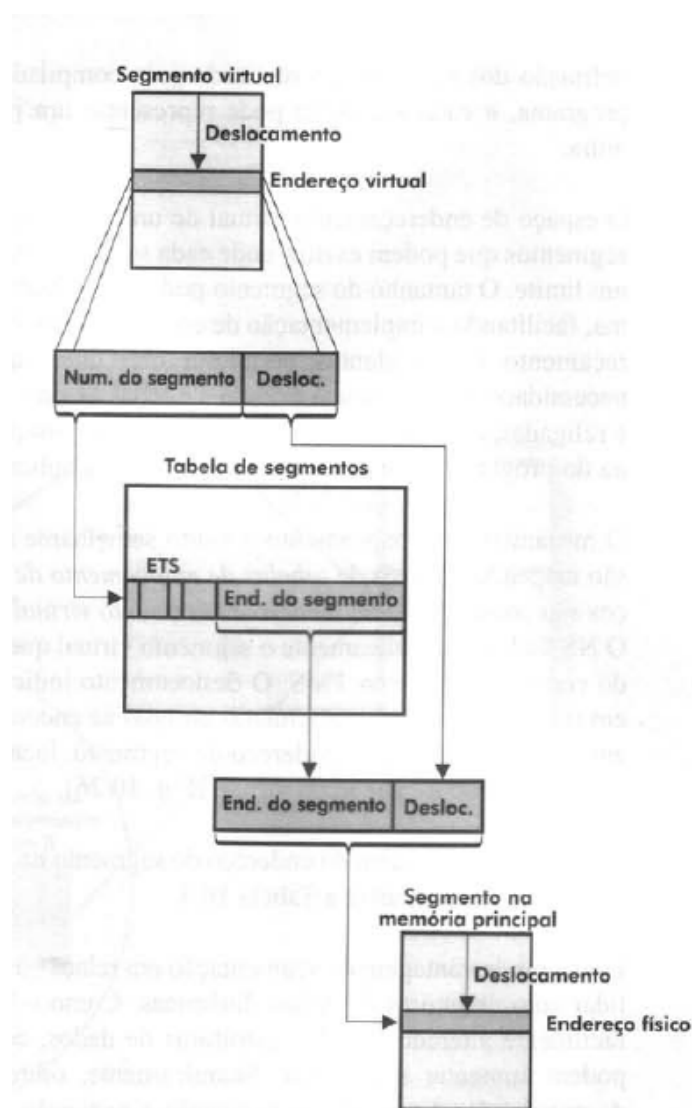


Fig. 10.26 Tradução do endereço virtual.

Tabela 10.4 Campos da ETS

Campo	Descrição
Tamanho	Especifica o tamanho do segmento.
Bit de validade	Indica se o segmento está na memória principal.
Bit de modificação	Indica se o segmento foi alterado.
Bit de referência	Indica se o segmento foi recentemente referenciado, sendo utilizado pelo algoritmo de substituição.
Proteção	Indica a proteção do segmento.

Tabela 10.5 Paginação × segmentação

Característica	Paginação	Segmentação
Tamanho dos blocos de memória	Iguais	Diferentes
Proteção	Complexa	Mais simples
Compartilhamento	Complexo	Mais simples
Estruturas de dados dinâmicas	Complexo	Mais simples
Fragmentação interna	Pode existir	Não existe
Fragmentação externa	Não existe	Pode existir
Programação modular	Dispensável	Indispensável
Alteração do programa	Mais trabalhosa	Mais simples

Endereçamento Virtual por Segmentação com Paginação

Memória virtual por segmentação com paginação é a técnica de gerência de memória onde o espaço de endereçamento é dividido em segmentos e, por sua vez, cada segmento dividido em páginas. Esse esquema de gerência de memória tem o objetivo de oferecer as vantagens tanto da técnica de paginação quanto da técnica de segmentação.

Swapping de Memória

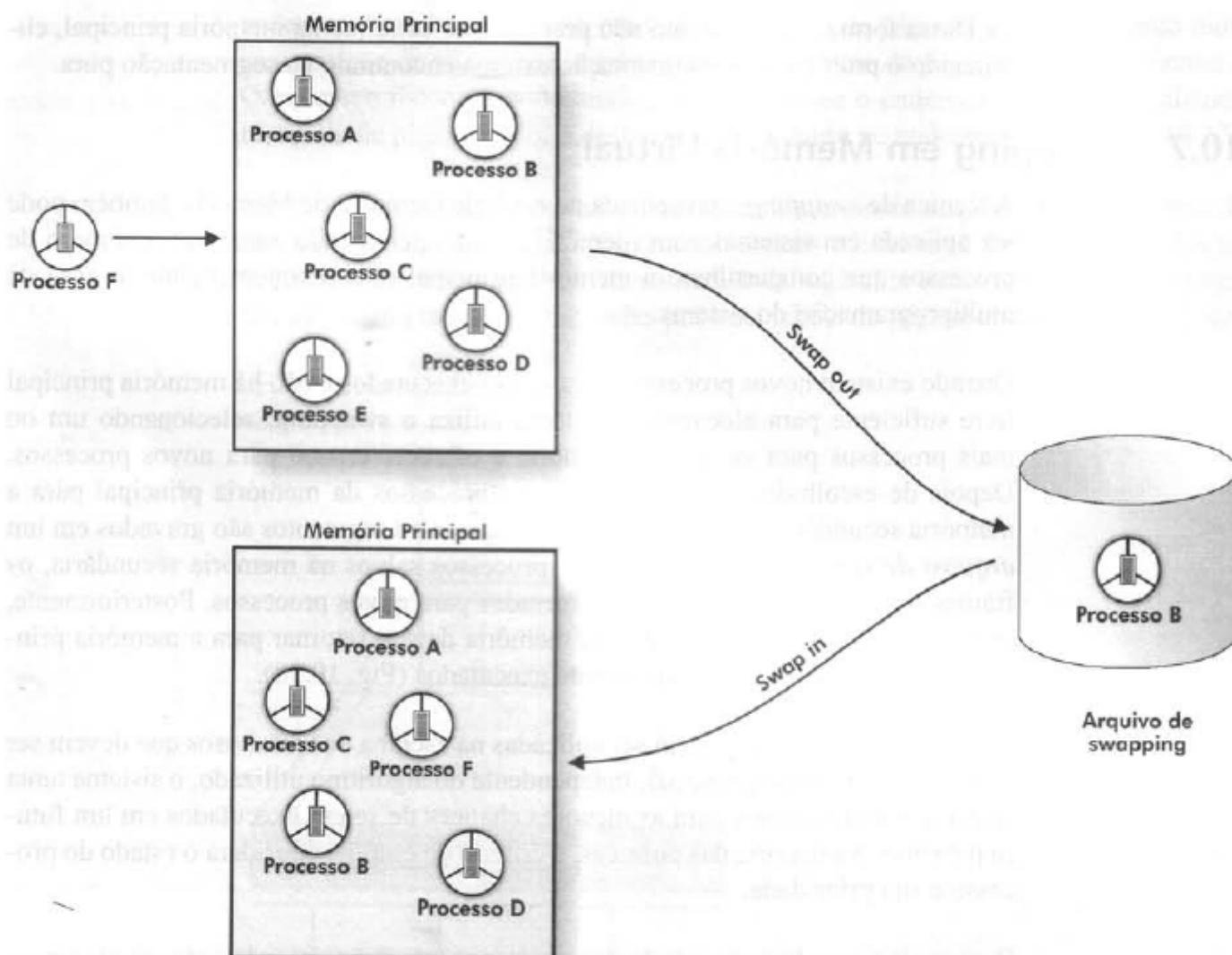


Fig. 10.28 Swapping em memória virtual.