



**Sistemas  
Operacionais**



**MEMÓRIA  
VIRTUAL**

# Sistemas Operacionais

## Memória Virtual

- Conceitos básicos
- Endereçamento Virtual
- Mapeamento
- Memória Virtual por Paginação
  - Políticas de substituição de páginas
  - Políticas de Alocação de Página
  - Algoritmos de Substituição de Página
- Memória Virtual por Segmentação

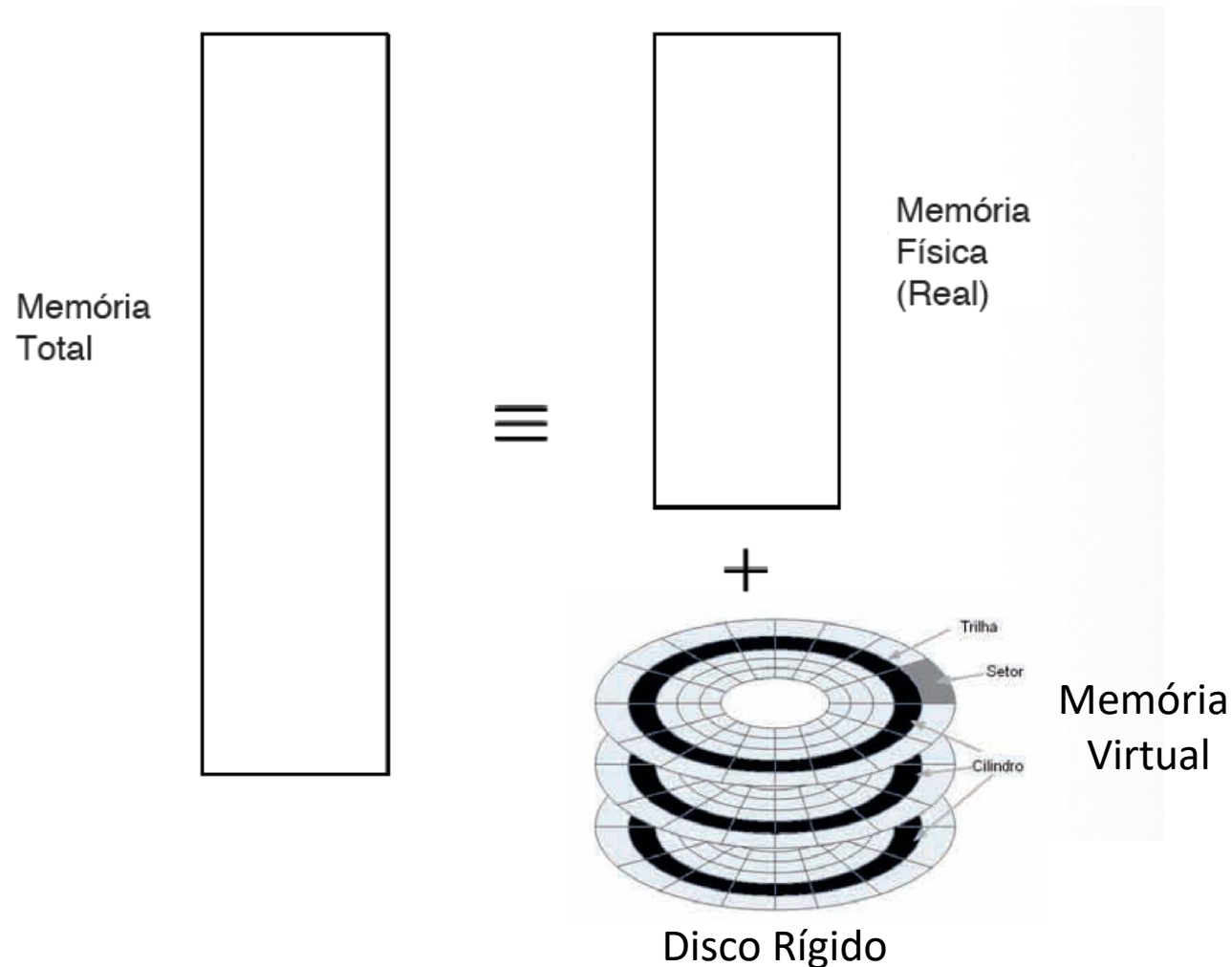
# Memória Virtual

## Conceitos básicos

- O conceito de memória virtual está fundamentado em desvincular o endereçamento feito pelo programa dos endereços físicos da memória principal (RAM). Assim, os programas e suas estruturas de dados deixam de estar limitados ao tamanho da memória física disponível.
- Para tal é criado um **espaço de endereçamento virtual**, linear e contínuo (semelhante a um vetor), para atender os requisitos de memória de um dado programa abstraindo-se questões de implementação física.
- Os objetivos principais desta técnica são o de maximizar o número de processos em memória, reduzir a fragmentação e permitir estruturas de dados maiores que a memória física.

# Memória Virtual

## Conceitos básicos



- A memória virtual de um sistema é, de fato, o(s) arquivo(s) de troca ou swap file(s) gravado(s) no disco rígido.
- Portanto, a memória total de um sistema, que possui memória virtual, é a soma de sua memória física, de tamanho fixo, com a memória virtual.
- O tamanho da memória virtual é denominado como **arquivo de paginação** no S.O Windows.

# Memória Virtual

## Endereçamento Virtual

- O conceito de memória virtual se aproxima muito da ideia de um vetor, existente nas linguagens de alto nível
- Quando um programa faz referência a um elemento do vetor, não há preocupação em saber a posição de memória daquele dado
- Um programa no ambiente de memória virtual não faz referência a endereços físicos de memória (endereços reais), mas apenas a endereços virtuais.
- No momento da execução de uma instrução, o endereço virtual é traduzido para um endereço físico, pois o processador acessa apenas posições da memória principal.

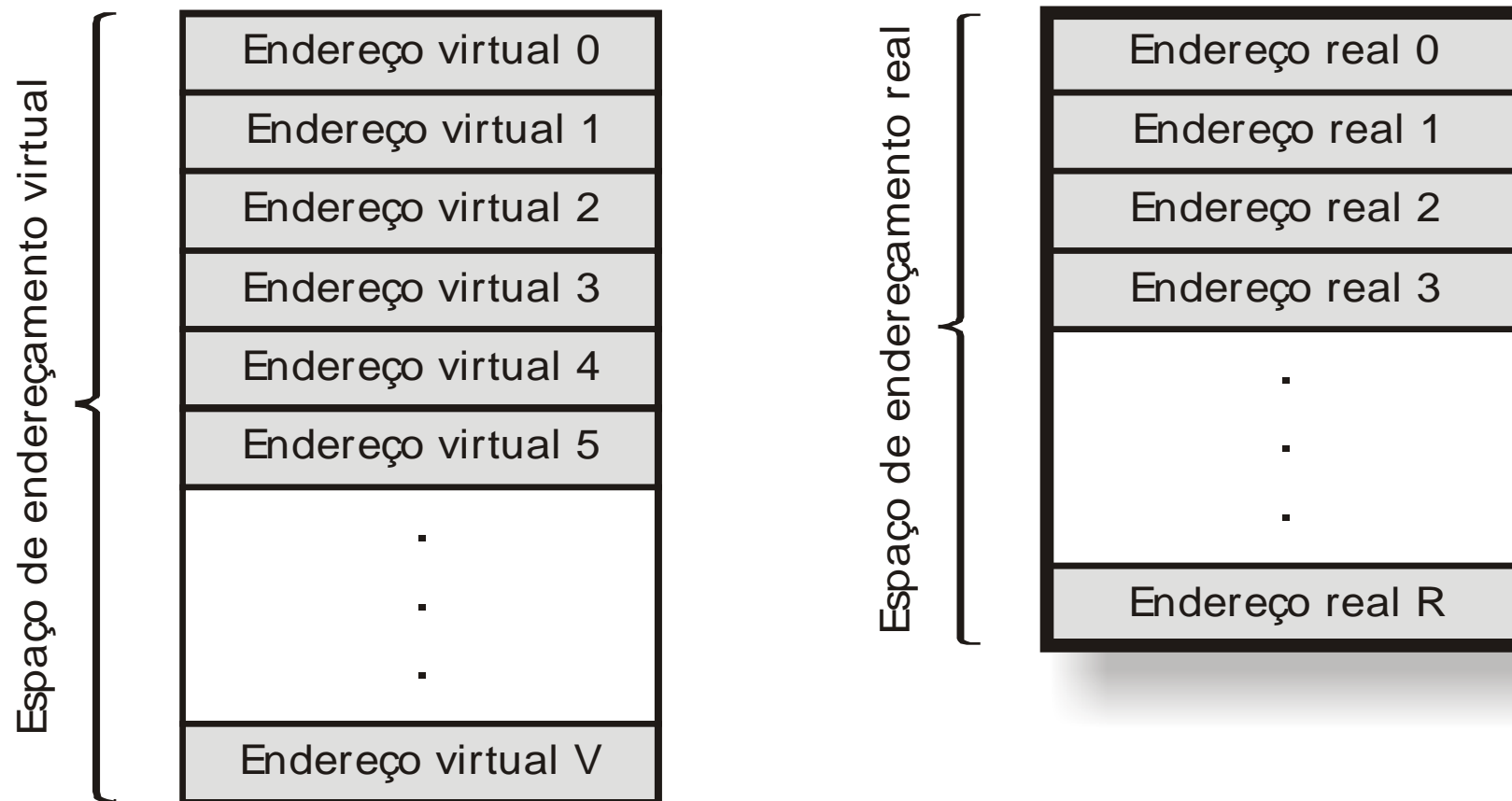
### Endereço Físico

500		VET [1]
501		VET [2]
502		VET [3]
503		VET [4]
504		VET [5]
.	.	.
.	.	.
.	.	.
599		VET [100]

# Memória Virtual

## Endereçamento Virtual

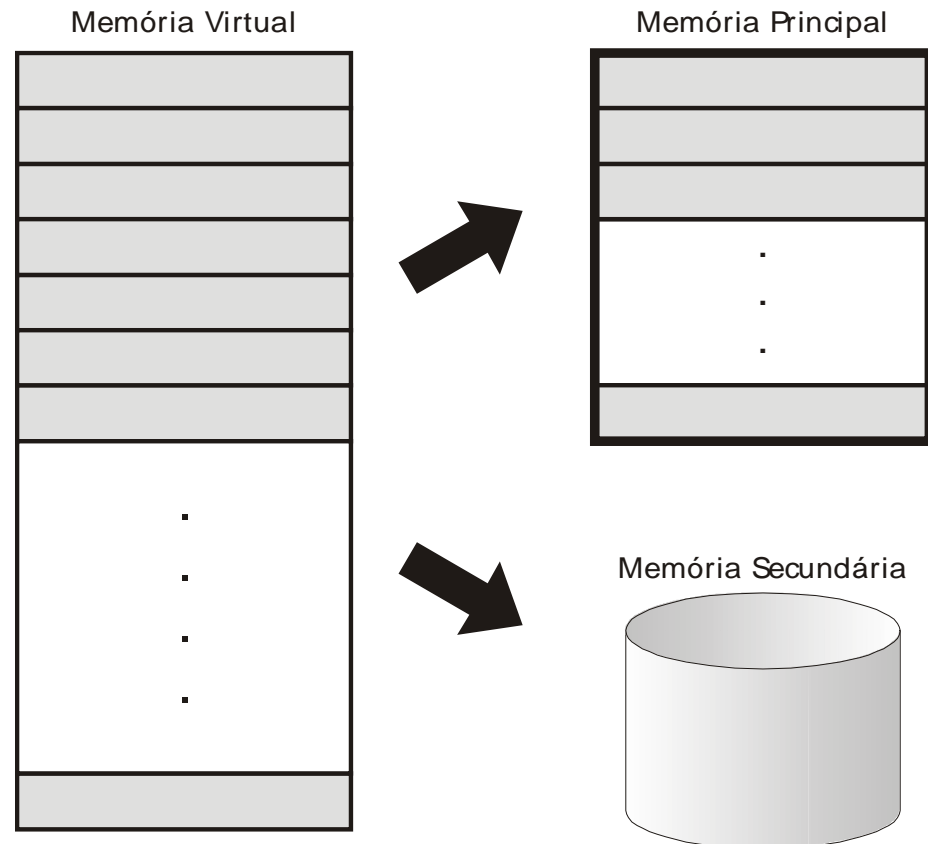
- O espaço de endereços virtuais, arranjados na forma de vetor, corresponde à **memória virtual** e pode ser maior que o espaço de memória real.



# Memória Virtual

## Espaço de endereçamento virtual

- A porção de memória virtual ativa do programa fica situada na memória principal, o restante fica armazenado em disco.

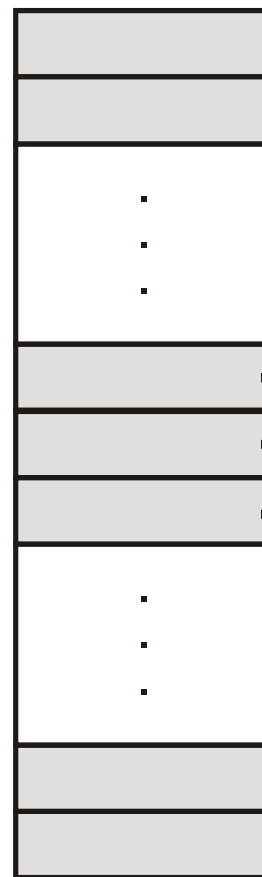


# Memória Virtual

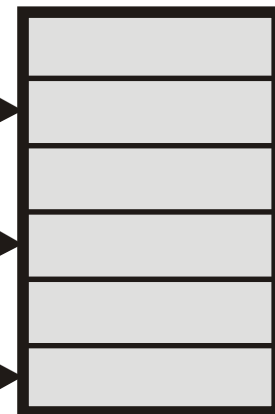
## Mapeamento

- A unidade de gerenciamento de memória mapeia os endereços virtuais em endereços físicos.
- Permite traduzir um endereço localizado no espaço virtual para um associado no espaço real.
- OBS: um programa não precisa estar em endereços contíguos da memória principal para ser executado.

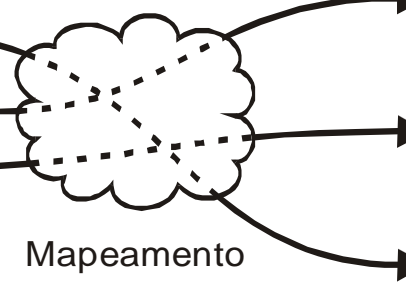
Memória Virtual



Memória Principal



Mapeamento

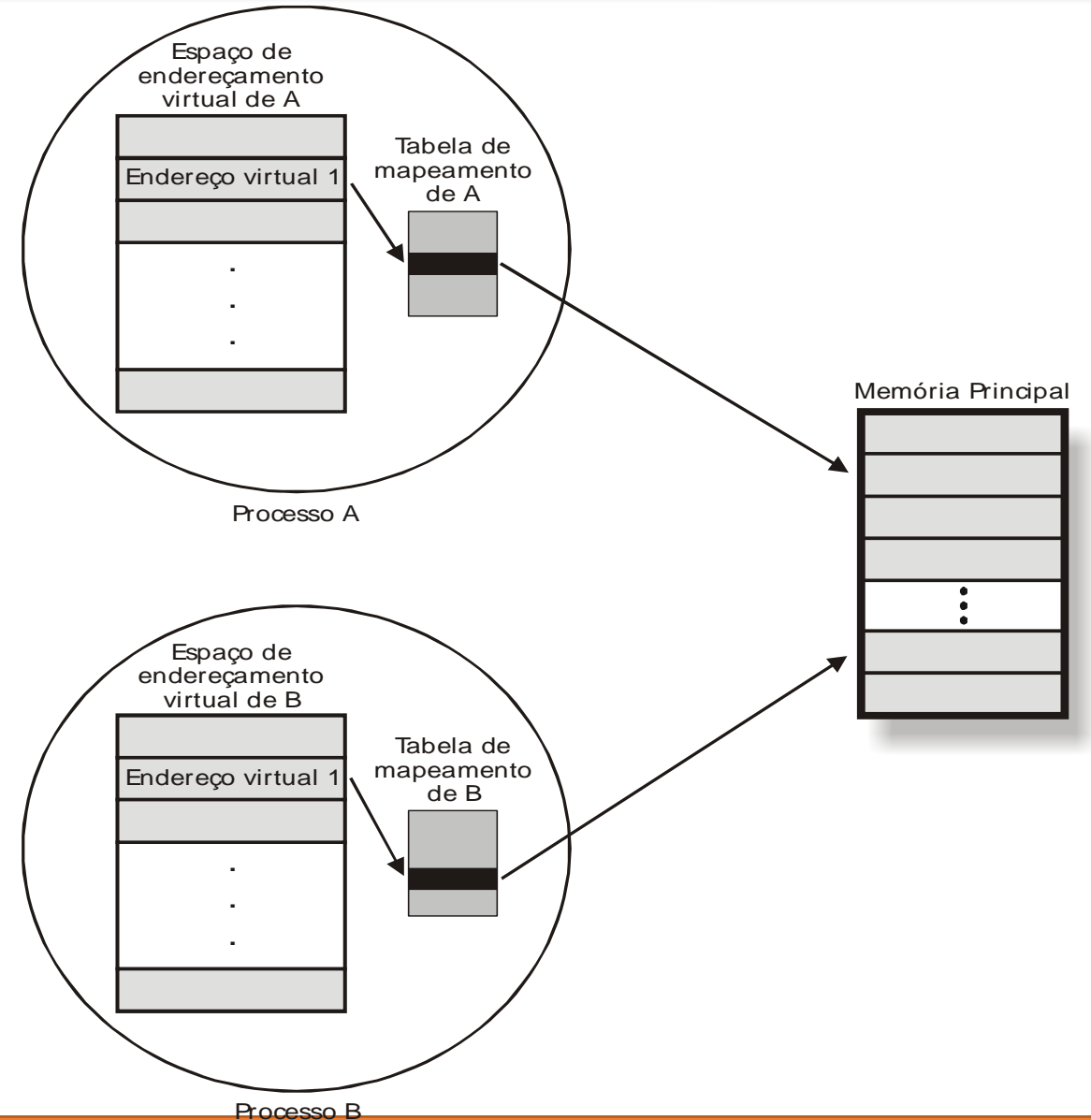




# Memória Virtual

## Mapeamento

- O mecanismo de tradução se encarrega de manter tabelas de mapeamento exclusivas para cada processo



# Memória Virtual

## Mapeamento

- **Problema:** se cada entrada na tabela de mapeamento representar uma célula da memória virtual, o espaço ocupado pela tabela seria da ordem de tamanho da memória virtual, inviabilizando sua implementação.
- **Solução:** particiona-se a memória virtual em blocos.

Cada bloco da memória virtual estará representado por uma entrada na tabela de mapeamento.

- Tamanho do bloco determina o número de entradas na tabela de mapeamento.
- 3 técnicas são usadas para o particionamento em blocos: paginação, segmentação e segmentação com paginação.
- Abordagens também se apoiam no princípio da localidade.

# Memória Virtual

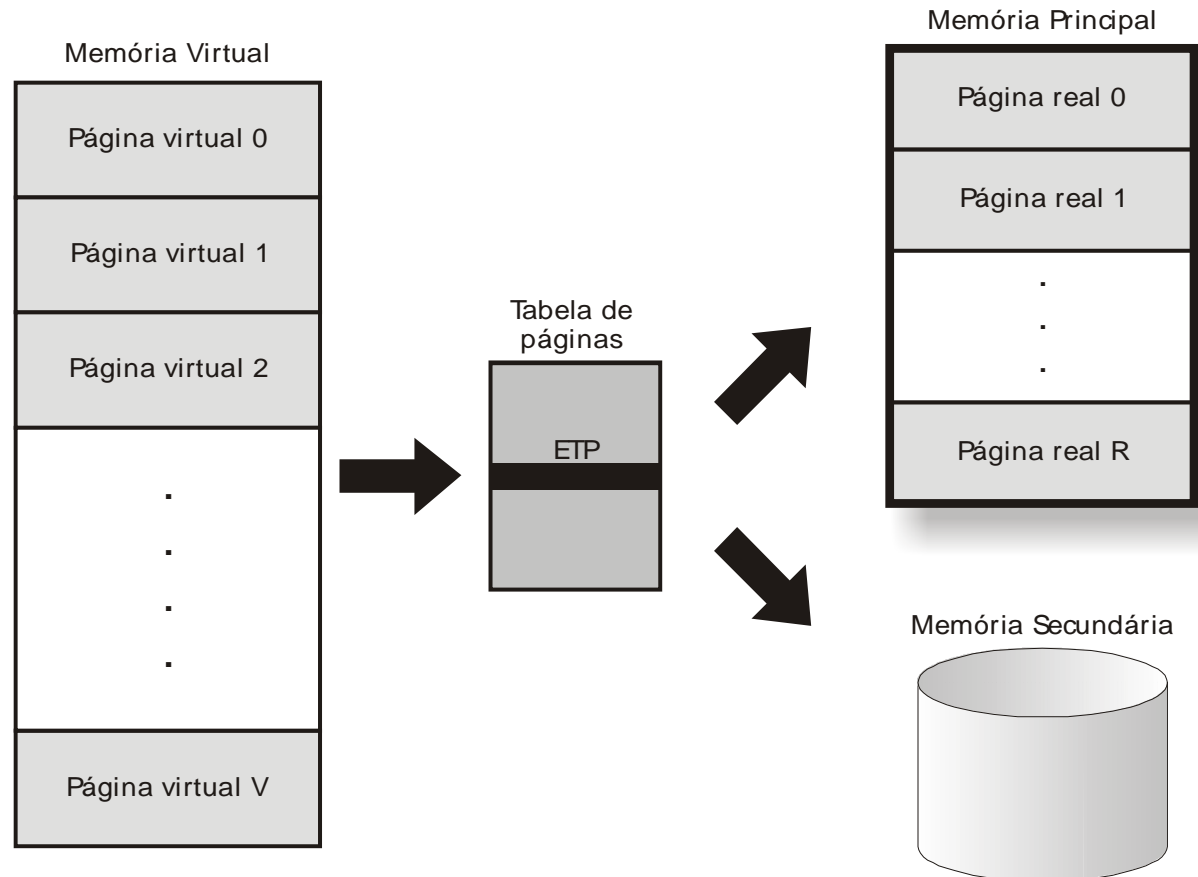
## Memória Virtual por Paginação

- A técnica mais comum utilizada para implementar memória virtual é a paginação.
- Nesta técnica, tanto memória virtual como memória principal são divididas em blocos de igual tamanho chamados **páginas**.
- Tamanho do bloco determina o nível de fragmentação.
- Quando uma página referenciada não é encontrada na RAM, diz-se que houve um ***page fault***.
- Na ocorrência de um *page fault*, realiza-se uma operação de paginação, ou seja, de troca de páginas entre a memória principal e a memória virtual em disco.

# Memória Virtual

## Paginação

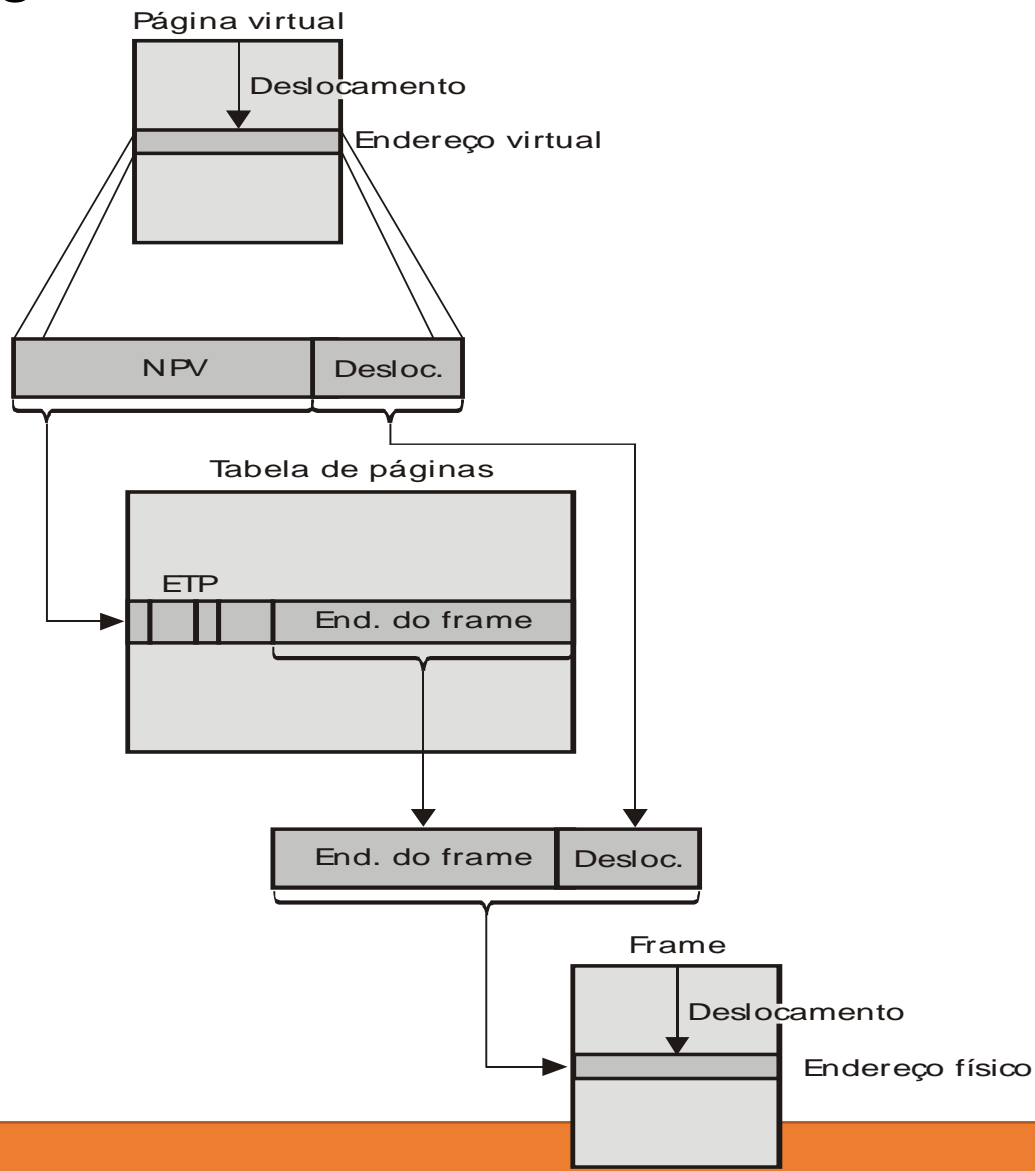
- Cada processo possui sua própria tabela de páginas.
- Uma página virtual contém um bloco de endereços virtuais



- **ETP: Entrada da Tabela de Páginas**, associada a cada página virtual de um processo.
- Os blocos da memória principal que abrigam as páginas reais são chamados **frames**.

# Memória Virtual

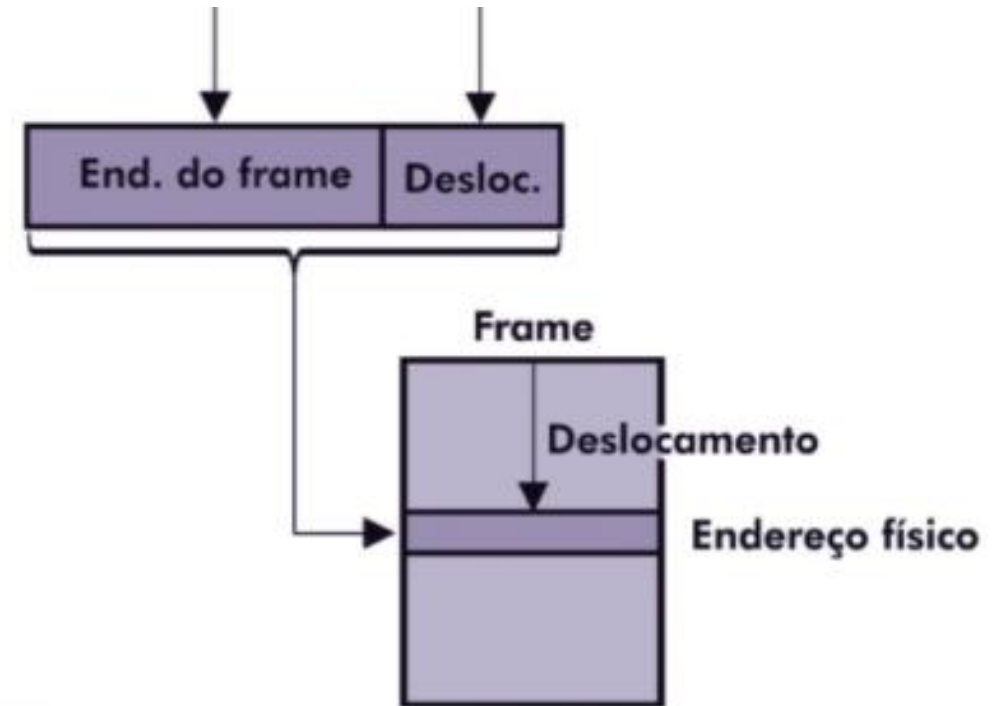
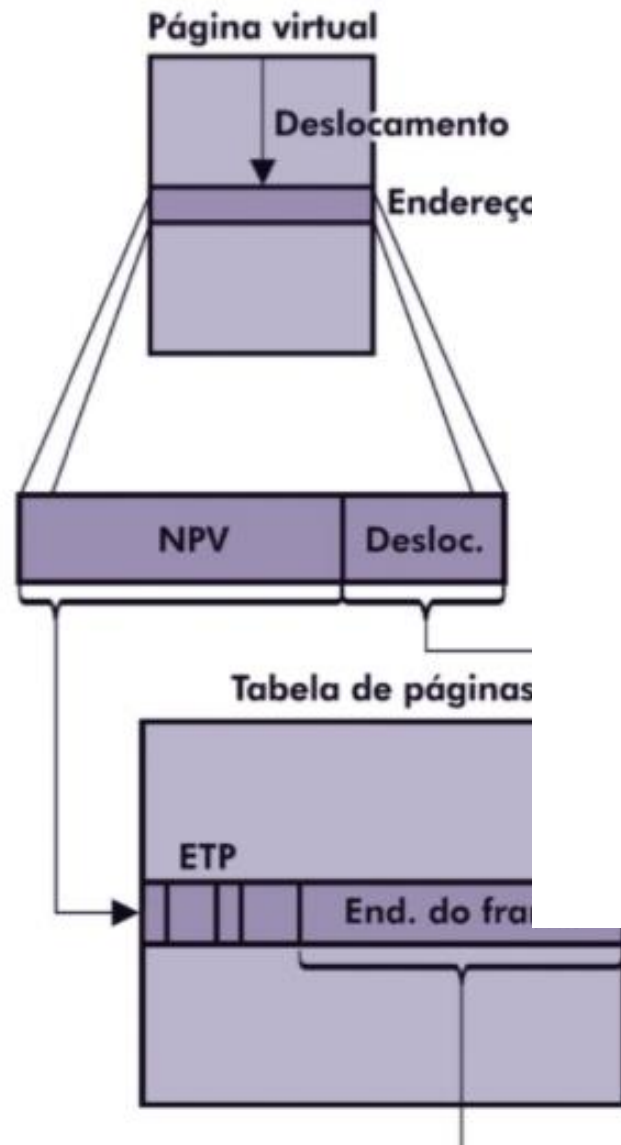
## Paginação



- O **endereço virtual** é composto pelo *número da página virtual* (NPV) mais seu deslocamento na página.
- O NPV serve para indexar a tabela de páginas e obter o endereço do **frame**.
- O **endereço físico** correspondente ao endereço virtual é obtido a partir do endereço físico do **frame** encontrado na tabela de páginas mais o deslocamento.

# Memória Virtual

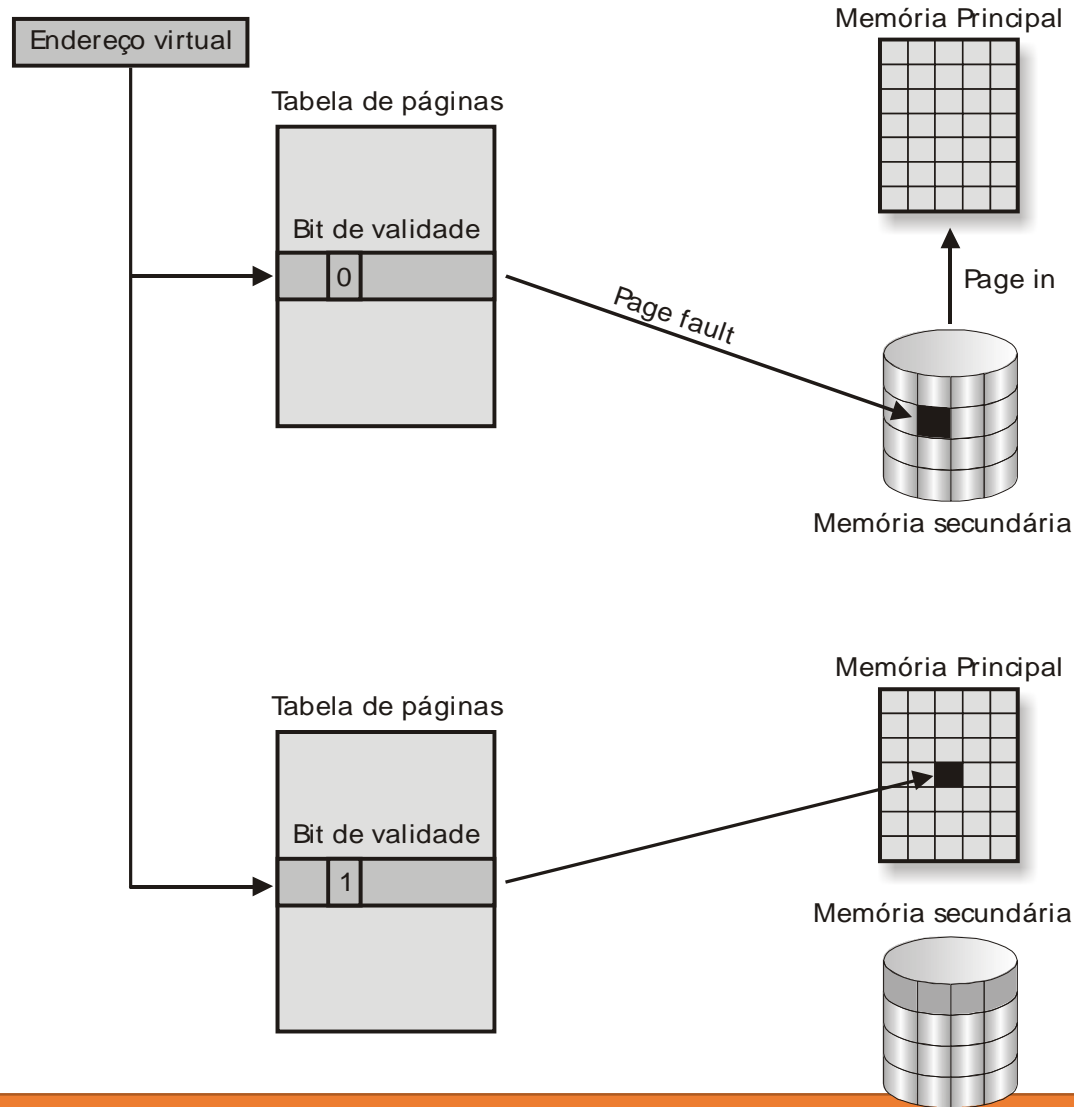
## Paginação



**Fig. 10.7** Tradução do endereço virtual.

# Memória Virtual

## Paginação

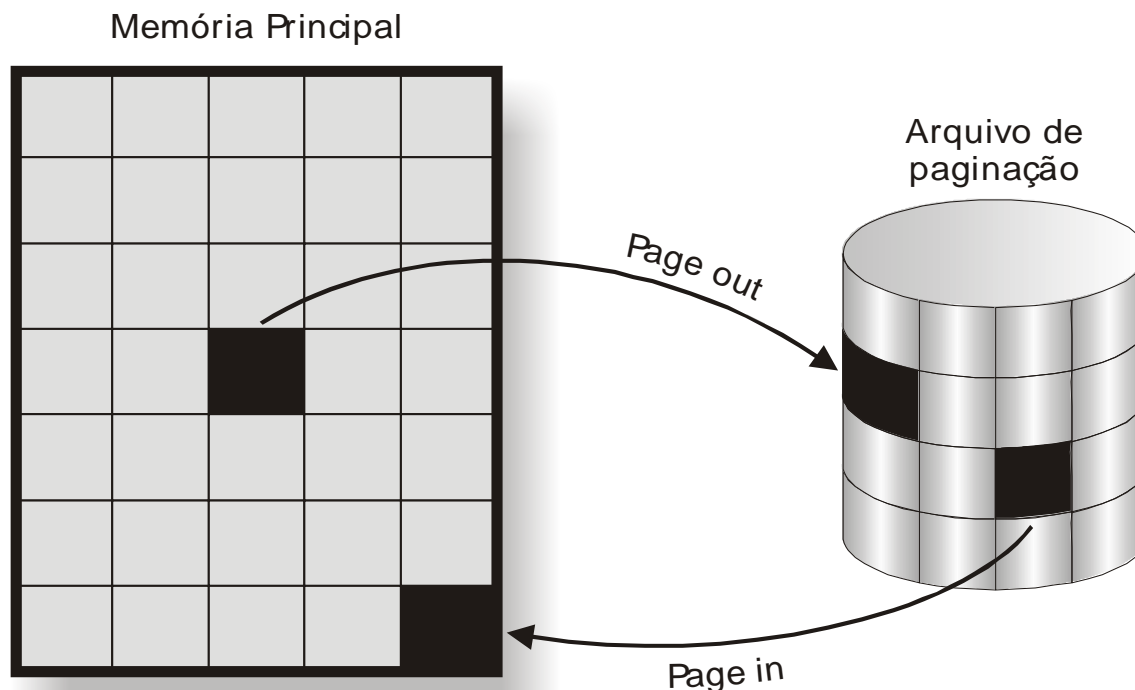


- O **bit de validade** numa ETP indica se a página em questão encontra-se na memória principal ou não.
- Se a ETP da página que contém o endereço virtual desejado tem bit de validade 0, ou seja, não está na memória principal, ocorre um **page fault**

# Memória Virtual

## Paginação

- Quando ocorre um **page fault**, a página referenciada é carregada para a memória principal (**page in**) no local de uma página que já se encontrava na RAM (**page out**, supondo memória RAM cheia). Esta página a ser substituída é escolhida segundo algum algoritmo de substituição





# Memória Virtual

## Paginação

- **Políticas de Substituição de Página:**

- Política de Substituição Local: apenas páginas referentes ao processo corrente podem ser substituídas.

- Política de Substituição Global: páginas de qualquer processo podem ser substituídas.

- **Políticas de Busca de Páginas:**

- Paginação por Demanda: páginas são transferidas para a memória principal apenas quando referenciadas.

- Paginação Antecipada: além da página referenciada, o sistema carrega um conjunto de outras páginas para a memória esperando que elas venham a ser referenciadas mais tarde.

# Memória Virtual

## Paginação

**Políticas de Alocação de Página:** determina quantos *frames* cada processo pode manter na memória principal.

- Política de Alocação Fixa: cada processo tem um número máximo de *frames* que podem ser usados durante sua execução.
  - Definido no momento da criação do processo de acordo com o tipo de aplicação que será executada.
- Política de Alocação Variável: número máximo de *frames* pode variar durante sua execução de acordo com a taxa de paginação e a ocupação da memória principal.
  - Mais flexível, mas exige **overhead** para o monitoramento dos processos.

# Memória Virtual

## Paginação

- **Algoritmos de Substituição de Páginas:** o algoritmo ótimo seria (utopicamente) aquele que selecionasse uma página que não será mais referenciada ou que será a menos referenciada.
- Exemplos de algoritmos:
  - Aleatório;
  - FIFO (escolhe a página mais antiga);
  - LFU (*Least Frequetly Used*);
  - LRU (*Least Recently Used*);
  - NRU (*Not Recently Used*, similar ao LRU porém mais simples);
  - FIFO circular (se a página mais antiga foi referenciada recentemente, ela vai para o final da fila).

# Memória Virtual

## Paginação

- Proteção: cada página conta com dois bits de proteção nas ETPs, um permitindo ou não acesso de leitura, outro permitindo ou não acesso de escrita.



  
Bits de proteção

LG	Descrição
00	Sem acesso
10	Acesso de leitura
11	Acesso para leitura/gravação

# Memória Virtual

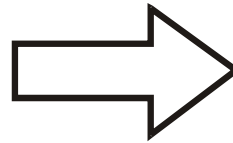
## Memória virtual por segmentação

- A segmentação é a técnica de gerenciamento de memória em que os programas são divididos logicamente em sub-rotinas e estrutura de dados e colocados em blocos de informação na memória
- Espaço de endereços é dividido em blocos de diferentes tamanhos chamados **segmentos**.
- Mantém uma relação lógica com a estrutura do programa e sua alocação na memória principal.
- A definição dos segmentos costuma ser realizada pelo compilador e cada segmento pode representar um procedimento, uma função, um vetor ou uma pilha.
- É atribuído a cada processo um número máximo de segmentos e um tamanho máximo para cada segmento.

# Memória Virtual

## Segmentação

```
PROGRAM Segmento;  
  VAR A: ARRAY...  
  C: ...  
  
  PROCEDURE X;  
  END;  
  
  FUNCTION Y;  
  
  END;  
  
  BEGIN  
  
  END.
```

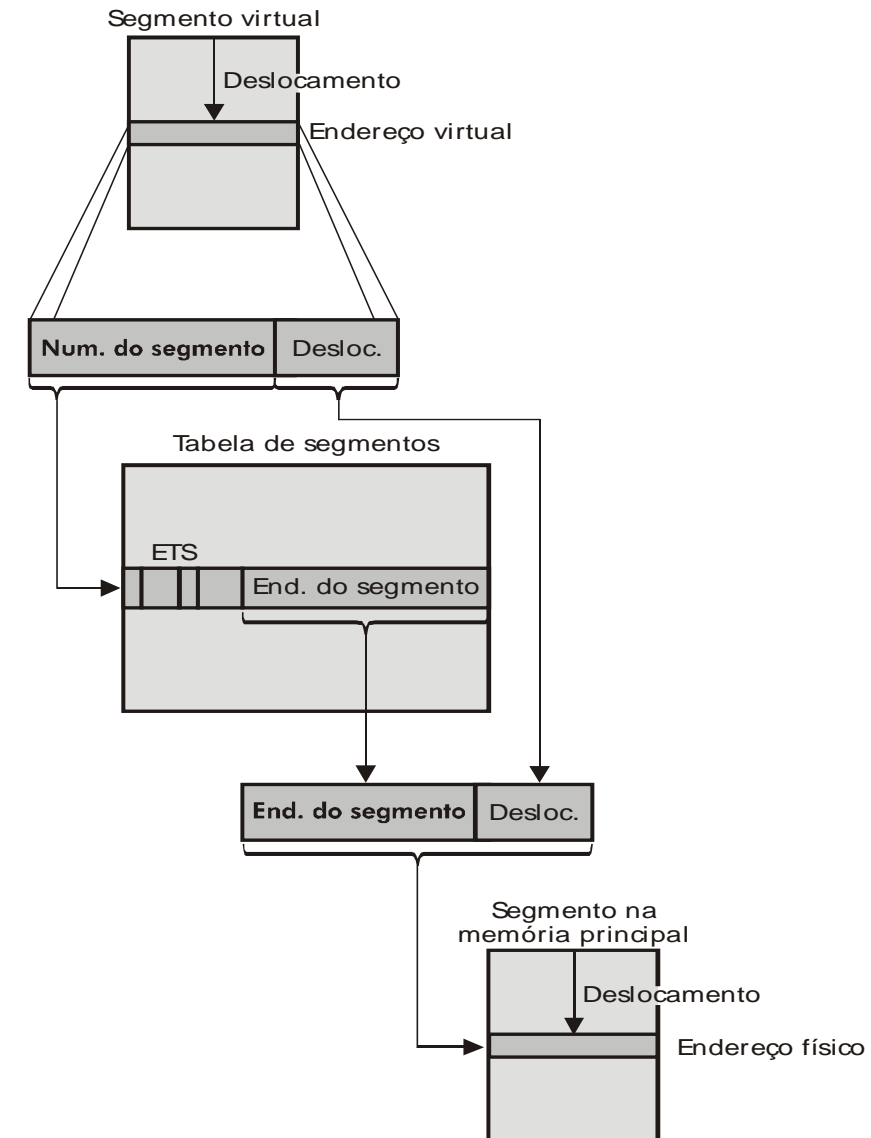


# Memória Virtual

## Segmentação

- O mecanismo de mapeamento é semelhante ao da paginação.
- 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.



# Memória Virtual

## Segmentação

- A grande diferença entre a paginação e a segmentação é que, enquanto a primeira divide o programa em partes de tamanho fixo, sem qualquer ligação com a estrutura do programa, a segmentação permite uma relação entre a lógica do programa e sua divisão na memória.
- O seu mapeamento é muito semelhante ao da paginação. Além do endereço do segmento na memória física, cada entrada na tabela de segmentos possui informações sobre o tamanho do segmento e se ele está ou não na memória. Se as aplicações não estiverem divididas em módulos, grandes pedaços de código estarão na memória desnecessariamente, não permitindo que outros usuários também utilizem a memória.
- Pode ter problema de fragmentação externa, pois as áreas livres são tão pequenas que não acomodam nenhum dado que necessite ser carregado. Logo é preciso conviver com problemas de fragmentação durante a alocação, ou ainda, ter programas que auxiliem na desfragmentação da memória.



# Memória Virtual

## Resumo

- Sistemas atuais trabalham com múltiplos processos disputando recursos e compartilhando memória, daí a necessidade de um **gerenciamento de memória**.
- Cada processo utiliza uma determinada área de memória, demanda total de memória costuma ser maior que a memória RAM disponível.
- Solução é utilizar o disco para armazenamento temporário. Duas técnicas são empregadas: **Troca de Processos (*Swapping*)** e **Memória Virtual**.

# Memória Virtual

## Resumo

- é criado um **espaço de endereçamento virtual linear**, semelhante a um vetor, para atender aos requisitos de memória do programa;
- a memória virtual pode ser maior que a memória principal, usa o disco para armazenar os endereços que não estão ativos;
- a unidade de gerenciamento mapeia os endereços virtuais em endereços físicos, cada processo tem sua própria tabela de mapeamento;
- vantagens da **Memória Virtual**: maximização do número de processos em memória, redução da fragmentação e permitir estruturas de dados maiores que a RAM;
- técnica mais usada é a **paginação**, que divide a memória virtual em blocos de igual tamanho chamados páginas;
- quando ocorre um **page fault**, é realizada uma operação de paginação (**page out, page in**);