

# SISTEMAS OPERACIONAIS II

Prof. Renato Jensen

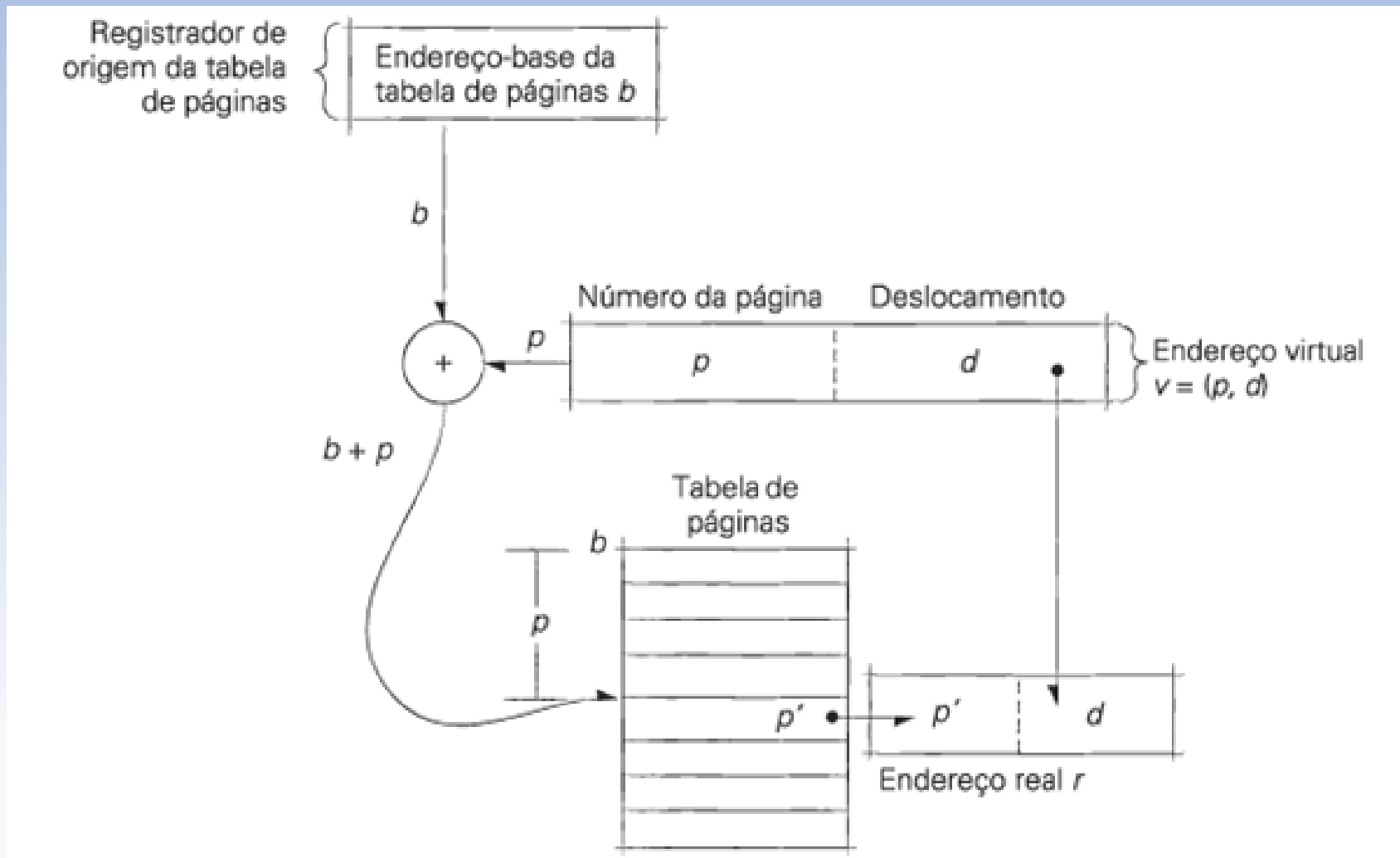
- **Memória virtual**

- Estratégias para o tratamento de **Tabela de Páginas**:

- **Mapeamento direto**
    - **Multiníveis**
    - **Páginas invertidas**

- **Memória virtual**
- **Tradução de endereço de paginação por mapeamento direto**
  - A tabela de páginas contém uma entrada para cada página do espaço de endereçamento virtual do processo armazenada contiguamente, parecido com um arranjo de um *array*. Qualquer entrada pode ser referenciada diretamente por um indexador.
  - Normalmente o sistema mantém a tabela de páginas na memória principal podendo consumir uma quantidade significativa desta memória.

- **Memória virtual**
- **Tradução de endereço de paginação por mapeamento direto**
  - O processo referencia o endereço virtual  $v = (p, d)$ .
    - O endereço da memória principal da tabela de páginas de um processo é carregado no registrador de origem da tabela de páginas.
    - O mecanismo DAT adiciona o endereço-base da tabela de páginas do processo,  $b$ , ao número da página referenciada,  $p$ .
    - $b + p$  forma o endereço de memória principal da PTE para a página  $p$ .
    - Essa PTE indica que a página virtual  $p$  corresponde à moldura de página  $p'$ .
    - O sistema concatena  $p'$  com o deslocamento,  $d$ , para formar o endereço real,  $r$ .



## Tradução de endereço de paginação por mapeamento direto

- **Memória virtual**

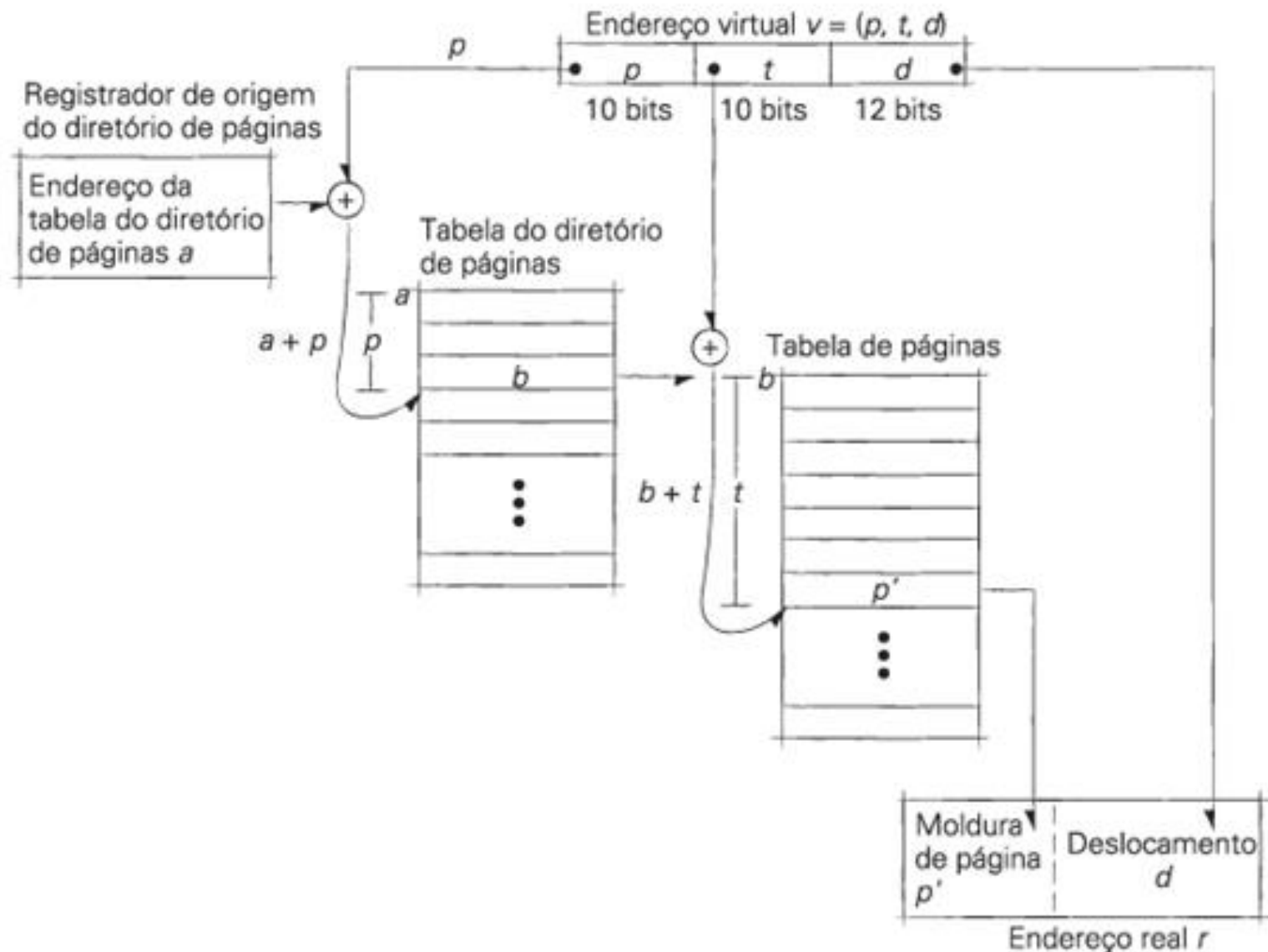
- **Tabelas de páginas multiníveis**

- Pode diminuir a sobrecarga de memória em comparação ao sistema de mapeamento direto.
- Esse sistema pode armazenar em localizações não contíguas da memória principal apenas as porções da tabela de páginas que o processo está usando.
- As demais porções da tabela de páginas podem ser criadas da primeira vez que forem usadas e transferidas para o armazenamento secundário quando não estiverem mais sendo usadas.
- Cada nível contém uma tabela que armazena ponteiros para tabelas do nível que está abaixo.
- O nível mais inferior compreende as tabelas que contêm os mapeamentos de molduras.

- **Memória virtual**

- **Tabelas de páginas multiníveis**

- Por exemplo, um sistema que utilize dois níveis de tabelas de páginas:
  - O endereço virtual é a tripla ordenada  $\mathbf{v} = (\mathbf{p}, \mathbf{t}, \mathbf{d})$  onde  $(\mathbf{p}, \mathbf{t})$  indica o número da página.
  - O endereço da memória principal do diretório de páginas ( $\mathbf{a}$ ) é carregado no registrador de origem do diretório de páginas e é adicionado valor de  $\mathbf{p}$ .
  - A entrada  $\mathbf{a} + \mathbf{p}$  contém o endereço de início da tabela de páginas correspondente,  $\mathbf{b}$ .
  - O sistema adiciona  $\mathbf{t}$  à  $\mathbf{b}$  para localizar a entrada da tabela de páginas que armazena um número de moldura de página,  $\mathbf{p}'$ .
  - O sistema concatena  $\mathbf{p}'$  com o deslocamento,  $\mathbf{d}$ , para formar o endereço real,  $\mathbf{r}$ .



## Tradução de endereço de páginas multiníveis



- **Memória virtual**

- **Tabelas de páginas invertidas**

- Armazenam exatamente uma PTE na memória para cada ***moldura*** de página no sistema.
- O número de PTEs é proporcional ao tamanho da memória física e não ao tamanho do espaço de endereçamento virtual do processo.
- Usam **funções de hash** para mapear páginas virtuais para entradas de tabela de páginas invertidas.
- Uma **função de hash** é uma função matemática que toma um número como entrada e dá um número de saída denominado valor de hash, dentro de uma faixa finita.

- **Memória virtual**

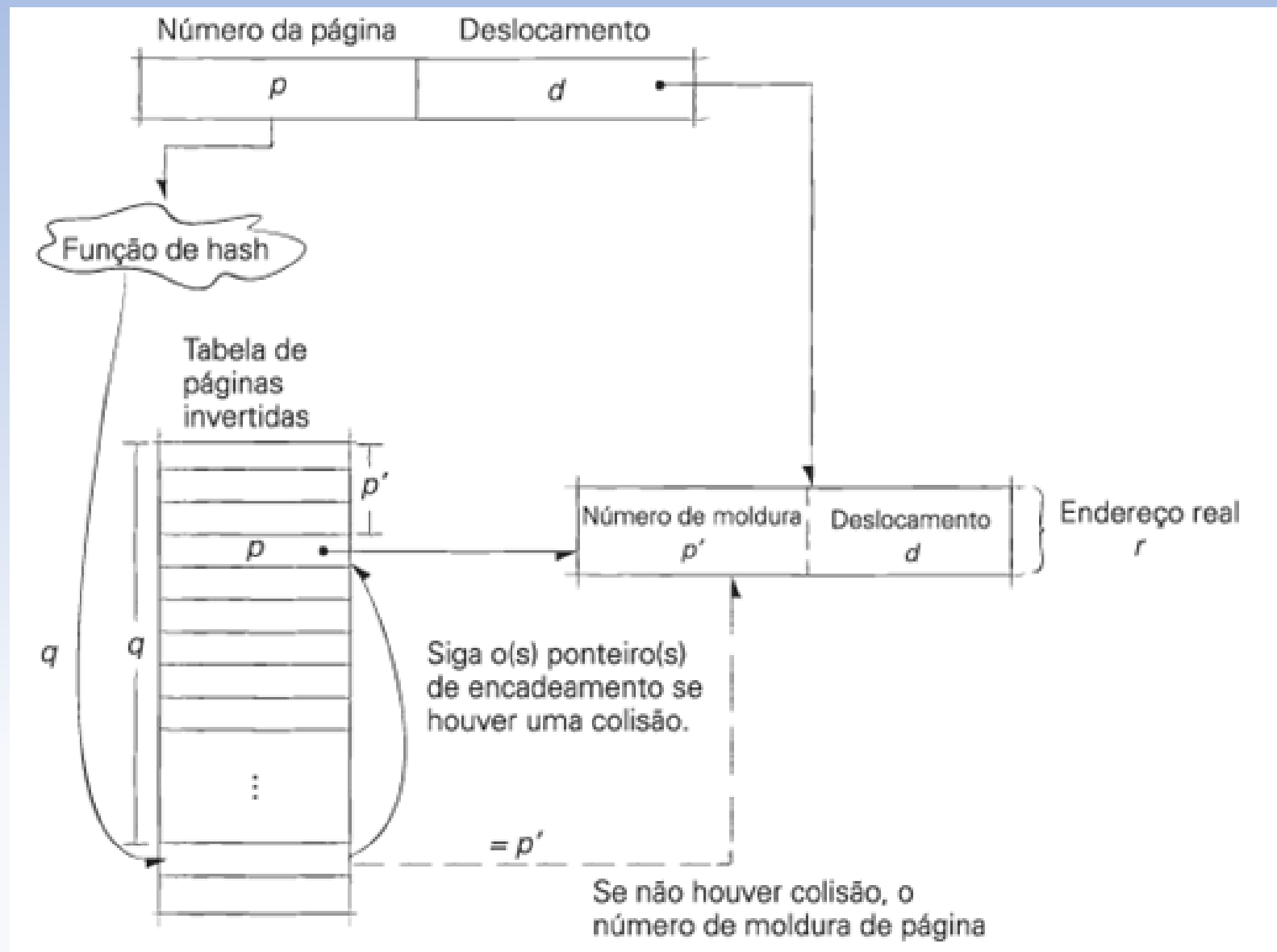
- **Tabelas de páginas invertidas**

- A função de hash eventualmente pode gerar uma saída repetida, já utilizada por outra entrada, que é denominada como **colisão**.
- Quando um valor de hash mapeia um item para uma localização que já está ocupada, uma nova função de hash é aplicada e a entrada é colocada na nova localização. Um ponteiro para essa nova localização é anexado à entrada do valor do hash original para garantir que seja encontrado. Normalmente são utilizadas listas encadeadas para essa finalidade.

- **Memória virtual**

- **Tabelas de páginas invertidas**

- O processo referencia o endereço virtual  $v = (p, d)$ .
  - Para localizar a entrada de tabela de hash da página virtual  $p$ , é aplicada a função de hash a  $p$  que gera um valor  $q$ .
  - Se a posição  $q$  da tabela de páginas invertidas contiver  $p$ , o endereço virtual requisitado estará na moldura de página  $q$ .
  - Se a posição  $q$  da tabela de páginas invertidas tiver outro conteúdo que não  $p$ , será verificado o ponteiro de encadeamento e se for diferente de *nulo* acontecerá uma colisão. O sistema seguirá os ponteiros da cadeia até encontrar uma entrada que contenha  $p$  ou até a cadeia terminar, quando será emitida uma requisição de falta de página residente.
  - Então  $p$  está localizado na moldura de página  $p'$  e o sistema concatena  $p'$  com o deslocamento,  $d$ , para formar o endereço real,  $r$ .



## Tradução de endereço de página usando tabelas de páginas invertidas