

Sistemas Operativos

Licenciatura Engenharia Informática
Licenciatura Engenharia Computacional

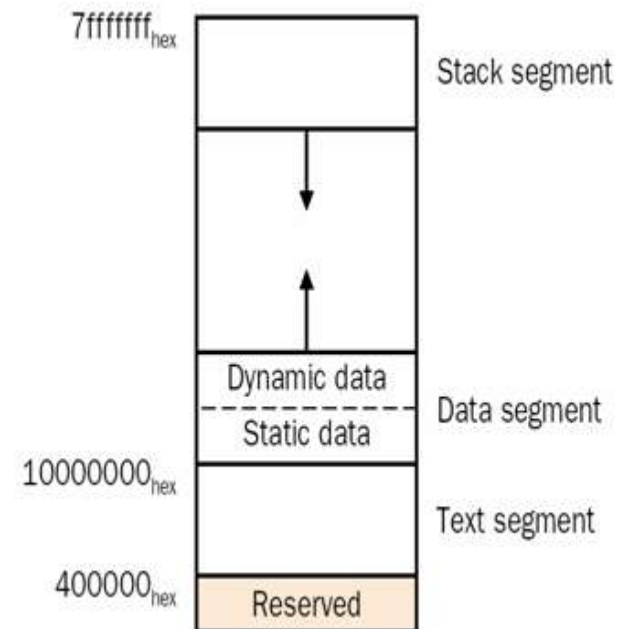
Ano letivo 2024/2025

Nuno Lau (nunolau@ua.pt)

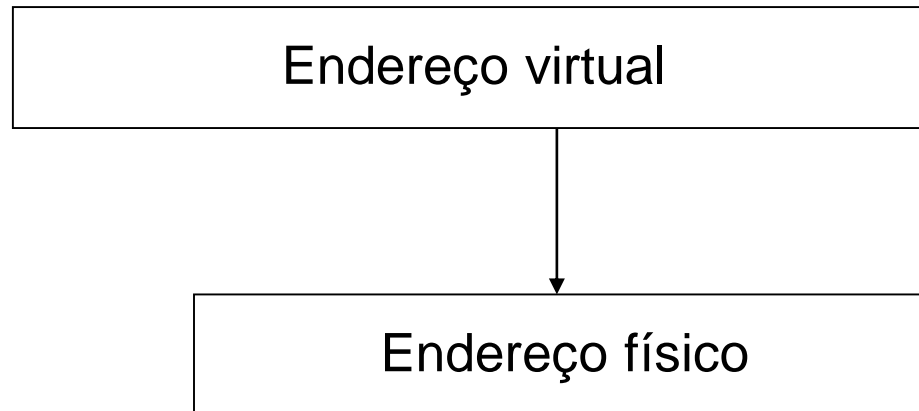
- Eficiência da utilização da memória
 - Memória deve ser partilhada pelos processos
 - Manter em memória apenas o necessário
 - Endereços usados pelos processos não são endereços da memória física
- Segurança
 - Mecanismos de segurança que impeçam que um processo altere as zonas de memória dos outros processos.
- Transparência
 - Processo tem acesso a muita memória (eventualmente mais do que a memória física)
 - Processo corre como se toda a memória lhe pertencesse
- Partilha de memória
 - Vários processos acedem à mesma zona de memória (de forma controlada)

Memória virtual

- Cada processo corre num espaço de endereçamento virtual (igual para todos)
- Os endereços usados pelos processos e os endereços físicos que lhes correspondem podem ser distintos
- Os endereços que o processo usa são virtuais, o mesmo endereço virtual de 2 processos pode corresponder a endereços físicos distintos
- Os endereços da memória virtual têm de ser convertidos em endereços de memória física
- Alguns endereços de memória virtual podem estar armazenados em disco

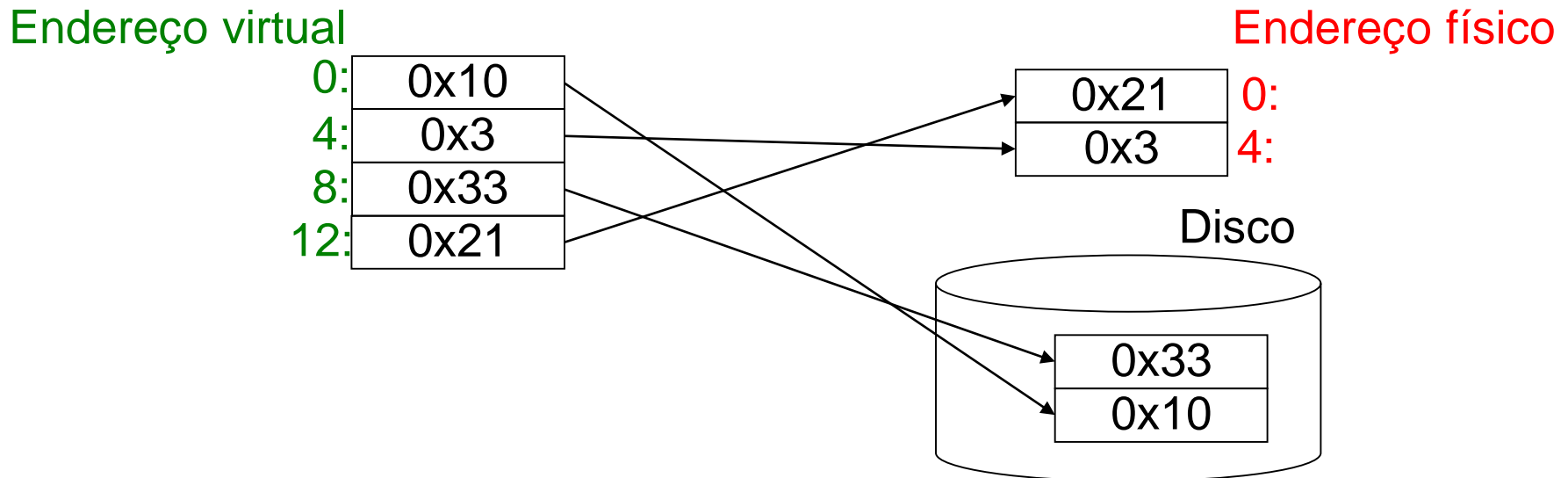


- É necessária a conversão (rápida) do endereço virtual para o endereço físico



Memória física como cache

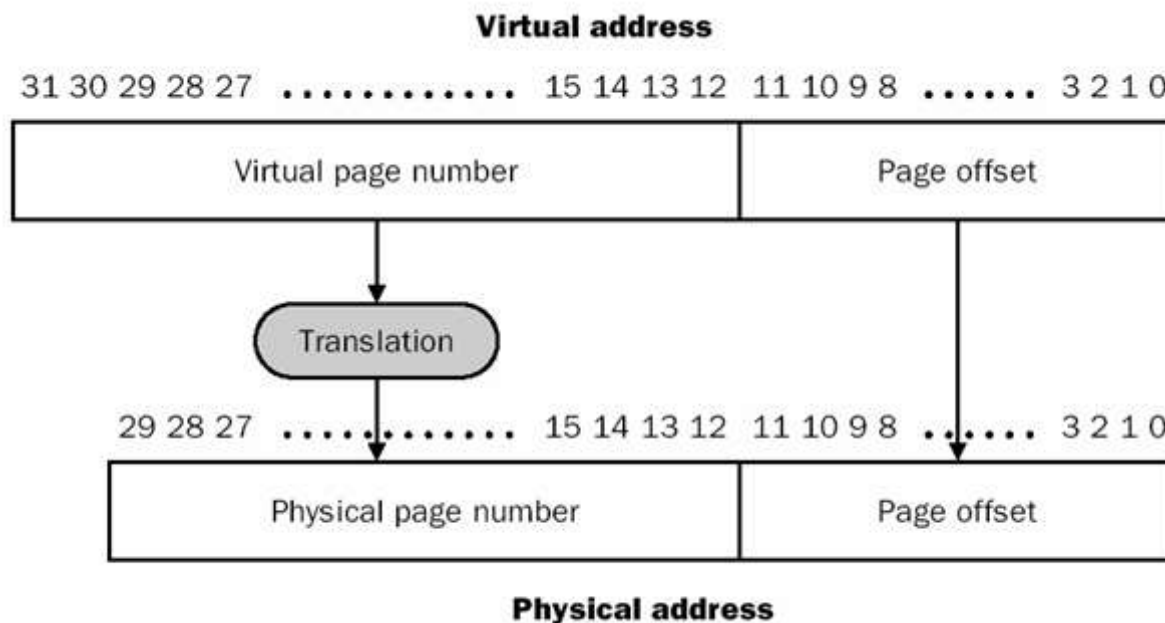
- O conteúdo dos endereços virtuais pode ser armazenado em disco



Endereço virtual paginado

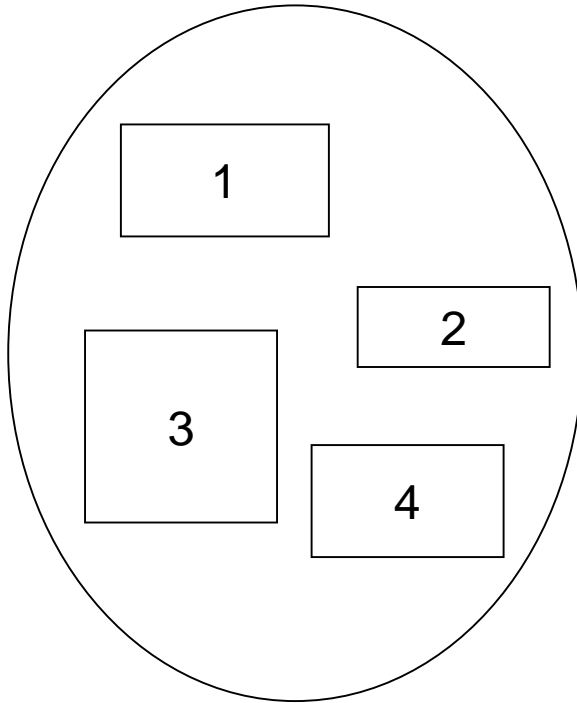
- N^o página virtual
- *Offset* no interior da página

Tamanho da página?

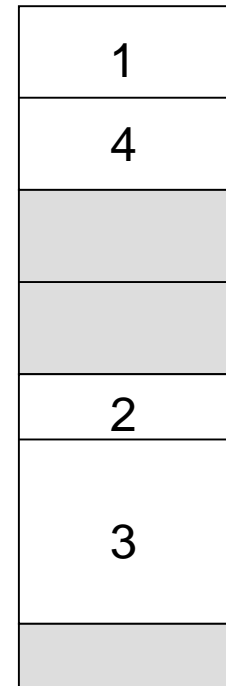


- **Paginação**
 - Memória dividida em páginas
 - Páginas têm tamanho fixo
 - Fragmentação interna
 - Um processo necessita de um certo número de páginas livres
- **Segmentação**
 - Memória dividida em segmentos
 - Segmentos podem ter tamanho variável
 - Fragmentação externa
 - Um processo necessita de uma zona contígua de memória livre para cada segmento

Segmentação



user space



physical memory space

Paginação e Segmentação

	Página	Segmento
Palavras por endereço	1	2 (<i>seg. e offset</i>)
Visível ao programador	Não	Depende
Substituir um bloco	Trivial	Difícil
Utilização memória	Fragmentação interna	Fragmentação externa
Acesso ao disco	Eficiente	Menos eficiente

- O custo de um *page fault* é muito elevado
- Páginas são relativamente grandes para amortizar o tempo de acesso
 - 4KB a 64KB
- Usar uma organização completamente associativa
- *Page faults* são tratados por software pois o atraso principal é o acesso ao disco.
 - Algoritmos mais sofisticados
- *Write-back*

Encontrar uma página

- Organização completamente associativa
 - Página virtual pode ser mapeada em qualquer página física
 - Reduzir *page faults*
- Não é possível usar procura associativa
- Tabela relaciona página virtual com a sua posição
 - *Page table*
 - Indexada por n^o página virtual
 - Entrada indica posição real da página virtual
 - **Cada processo tem a sua tabela de página**

Tabela de página

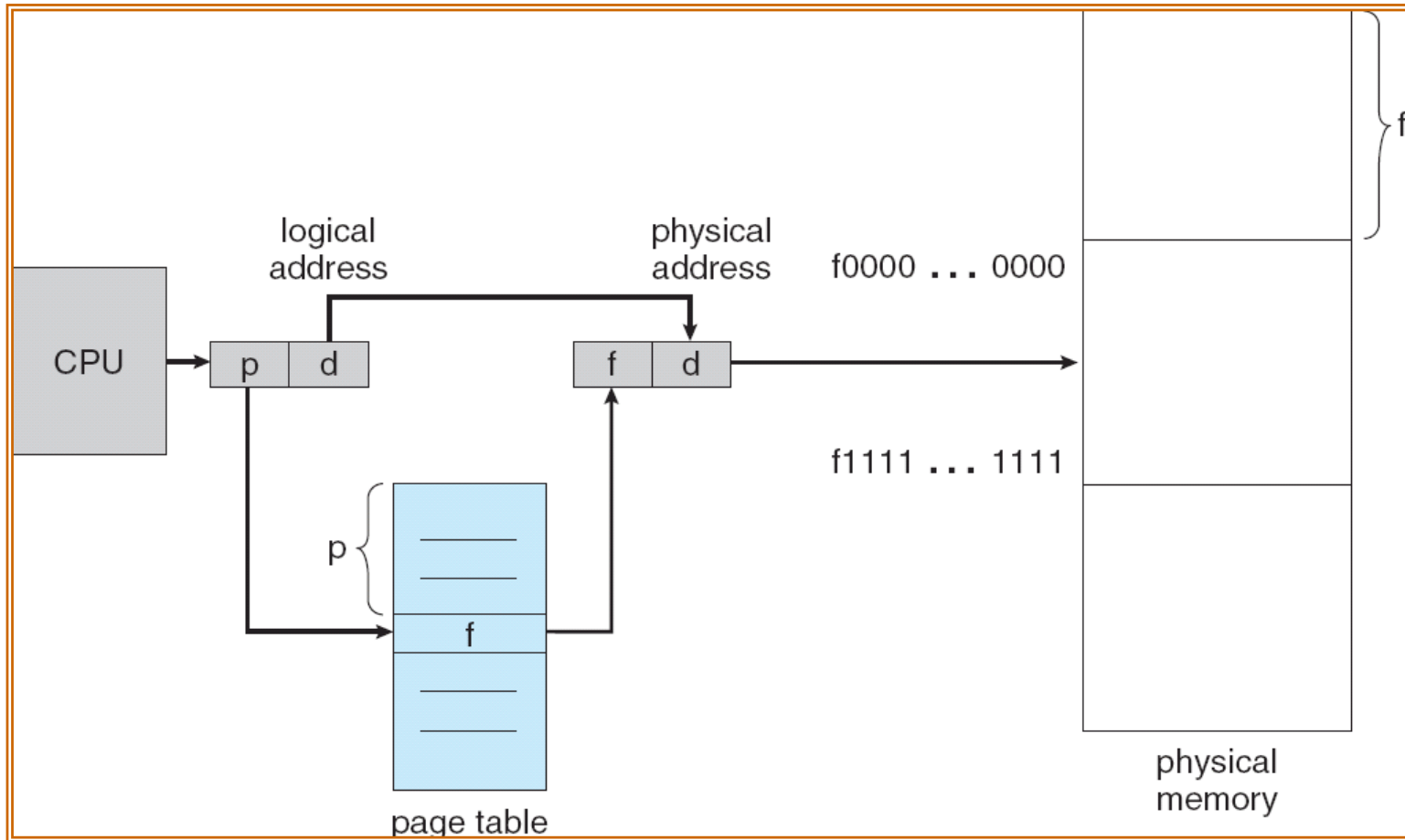
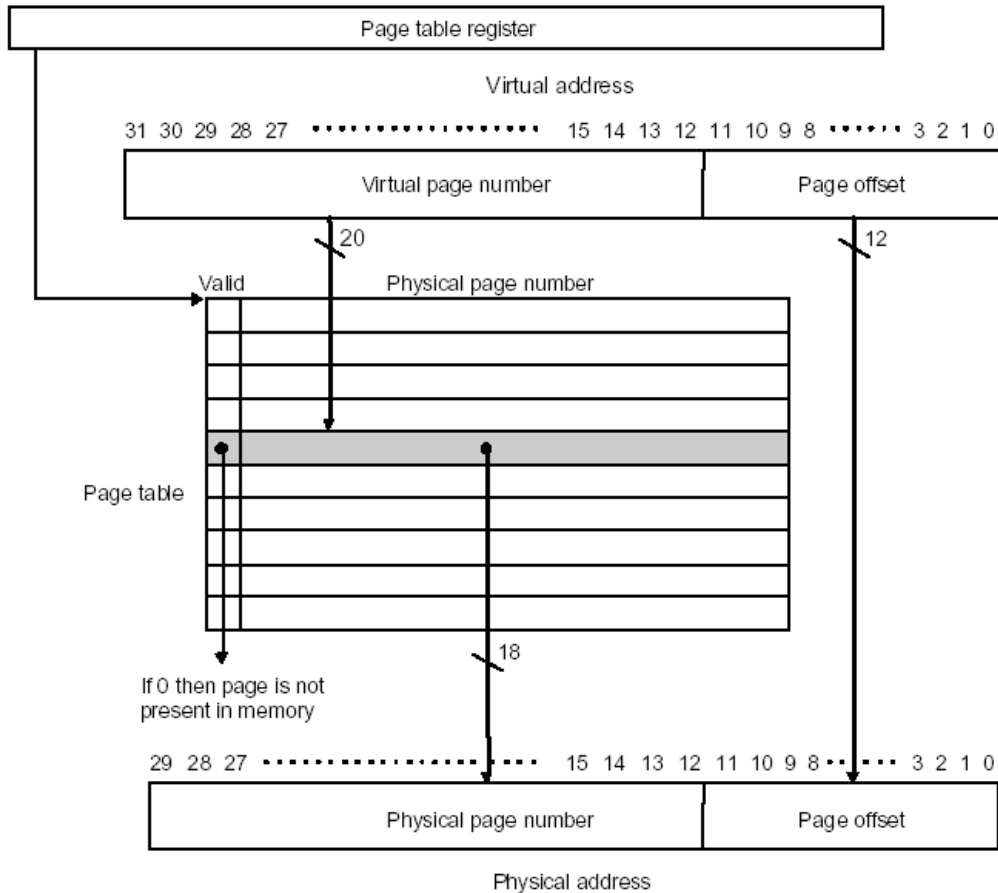
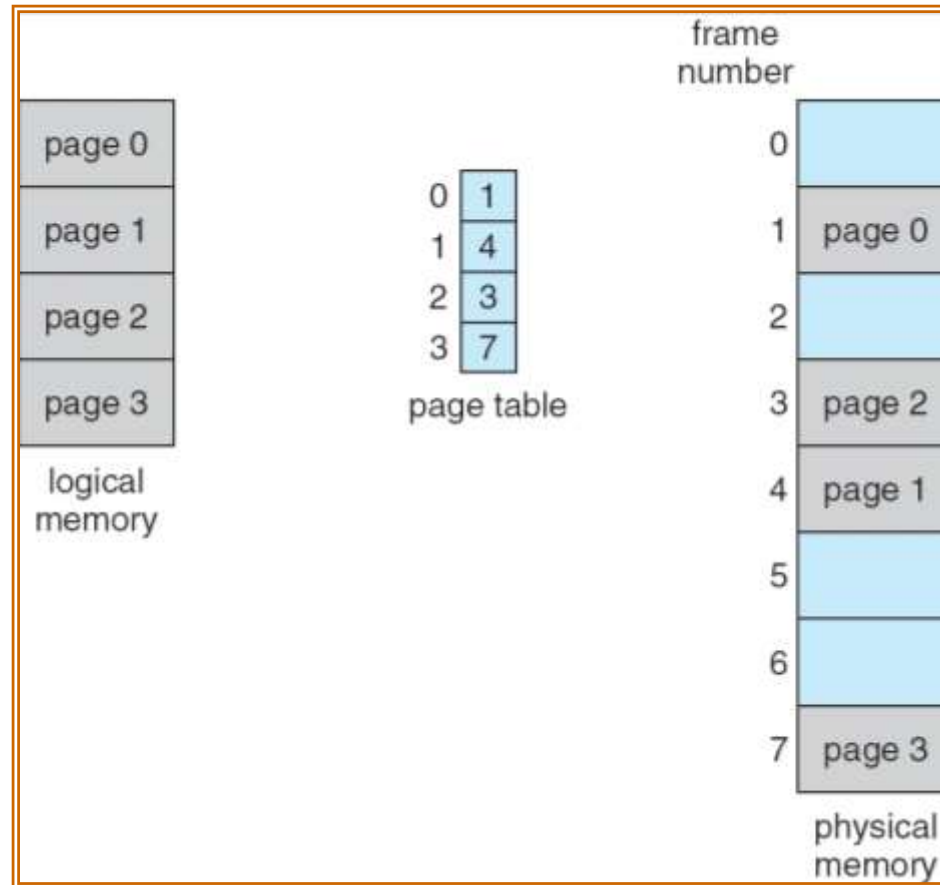


Tabela de página

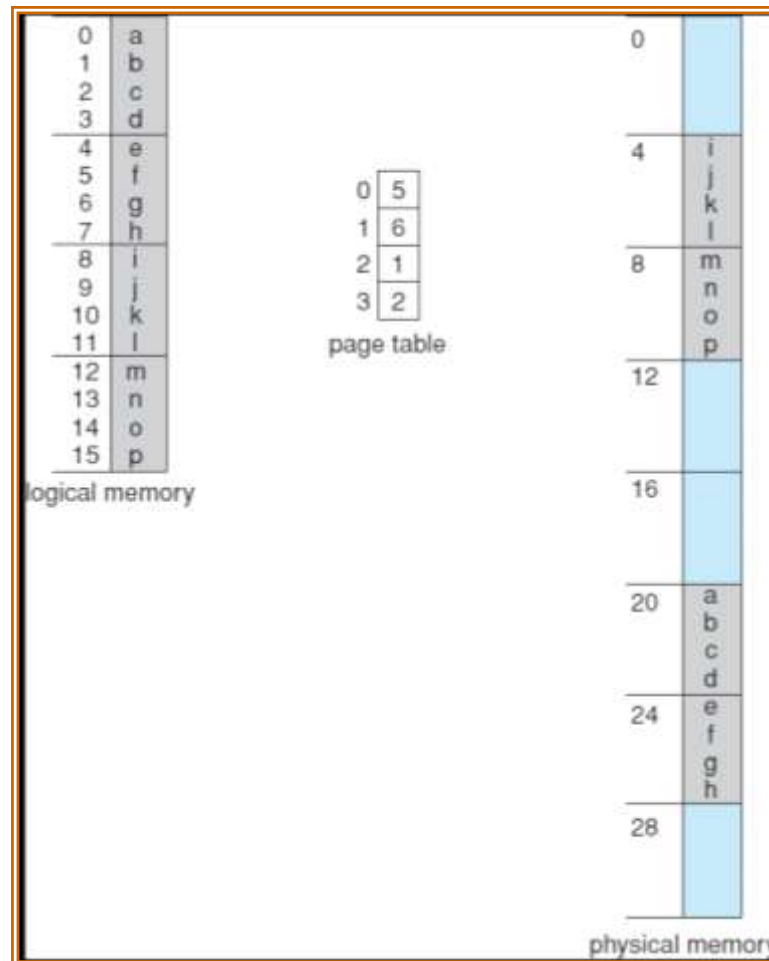


- Registo de tabela de página
- *Valid bit*
- Permissões

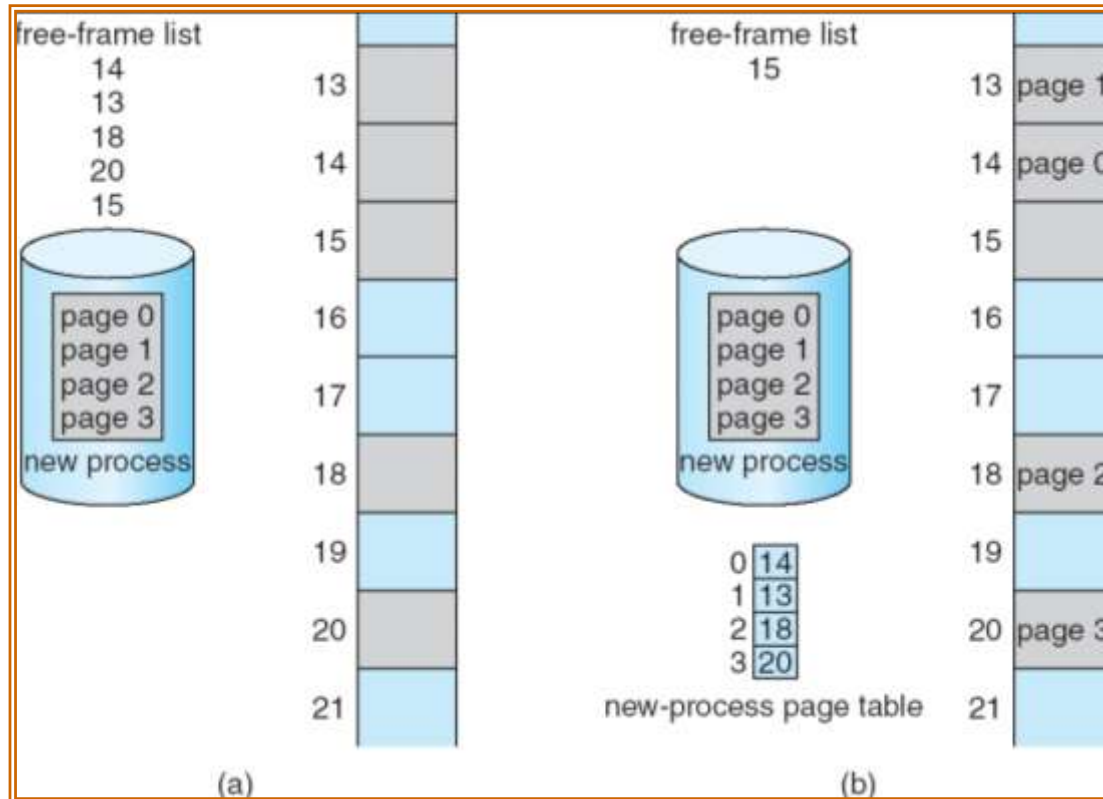
Endereços lógicos e físicos



Endereços lógicos e físicos



Endereços lógicos e físicos

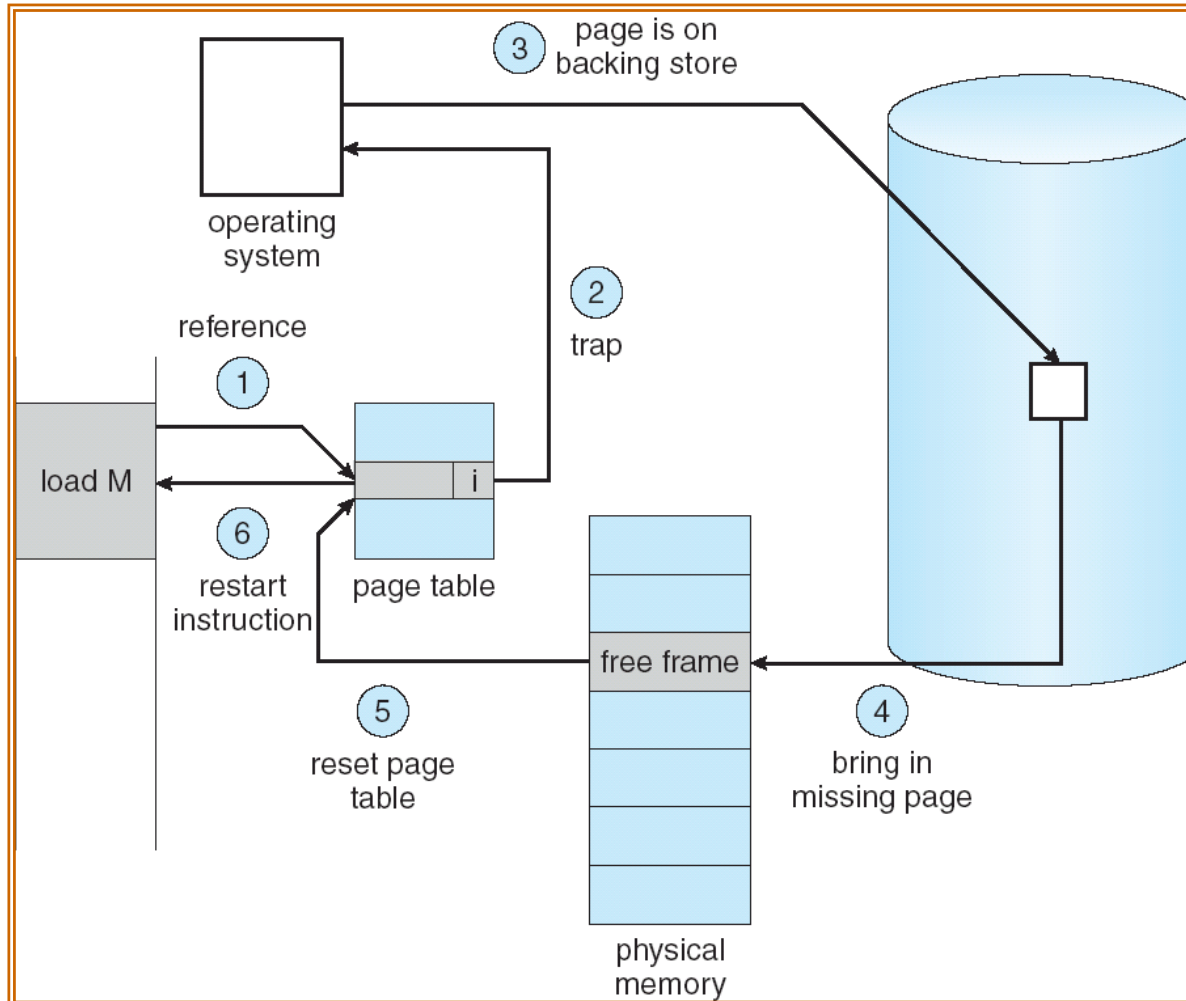


Before allocation

After allocation

- *Valid bit* da tabela de página com 0
- Tratado pelo sistema operativo
- Estrutura mantém posição das página virtuais em disco
- Política de substituição tipo LRU aproximado
 - Tabela de páginas contém *reference bit*
 - Periodicamente *reference bits* colocados a zero
 - Se página é acedida (*touched*) *reference bit* a 1
 - Substituir páginas com *reference bit* a 0

Page fault



Tamanho da tabela de página

- Considere um sistema com endereços de 32 bits, páginas de 4KB e 4 bytes por entrada na tabela de página. Qual o espaço ocupado pela tabela de página?

$$\text{N}^\circ \text{ de entradas} = 2^{32} / 2^{12} = 2^{20}$$

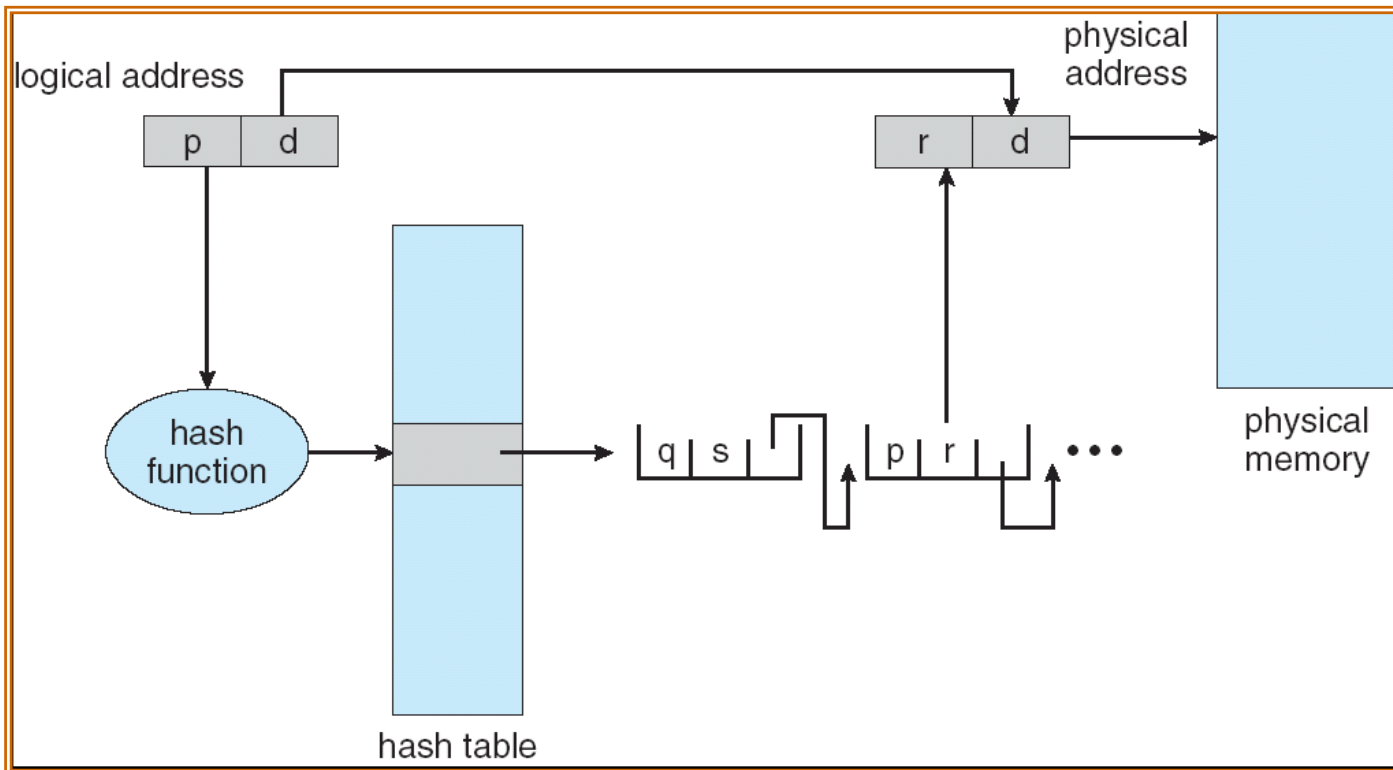
$$\text{Tamanho da tabela} = 2^{20} * 4 = 4\text{MB}$$

100 processos implicaria 400MB de memória usados em tabelas de página!

- Técnicas para reduzir tabela de página
 - Registo limita o tamanho da tabela
 - Memória associada ao processo cresce num só sentido
 - 2 tabelas de página por processo com 2 registos limite
 - 2 segmentos (*stack* e *heap*) crescem em direcções opostas
 - *Hashing*
 - Tabela de página depende do tamanho da memória física
 - Tabelas de página com vários níveis
 - Tabelas de página em memória virtual!

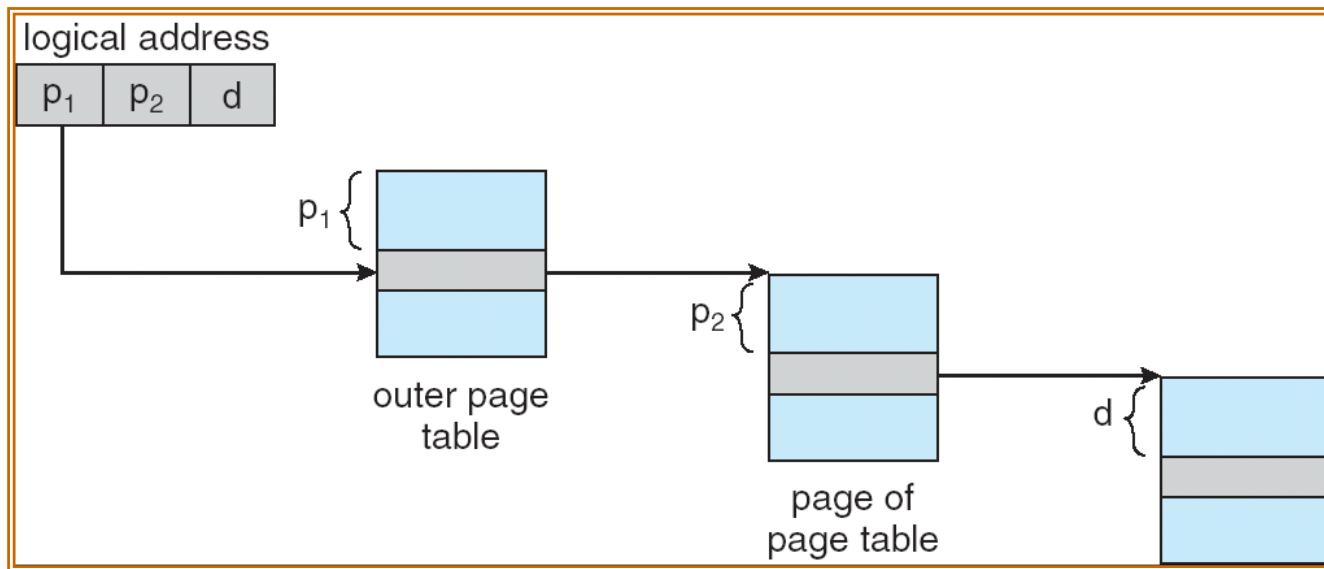
Tabelas de página com *hashing*

- Função de *hash* determina entrada a ser usada
- Na mesma entrada podem co-existir várias traduções



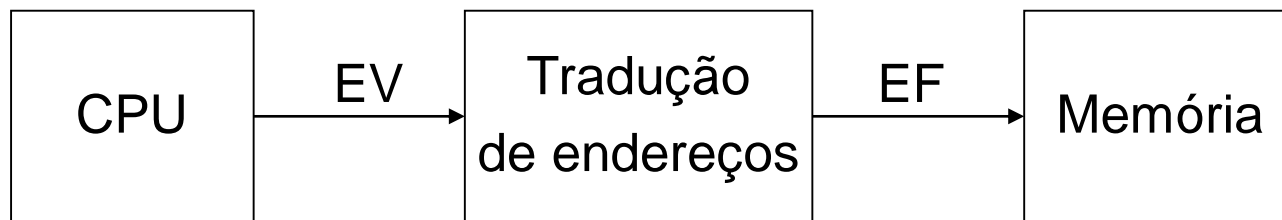
Tabelas de página com vários níveis

- 2 níveis



- *Write-through* não é eficiente
- *Write-back*
 - Escrita em disco página a página
 - Página só é escrita em disco quando necessita de ser retirada de memória física...
- *Dirty bit*
 - e só se foi alterada desde que foi lida do disco
- *Write allocate*
 - Escrita numa página que não reside em memória física, carrega essa página para a memória física e escreve

- Memória virtual obriga a 2 referências à memória para aceder a um endereço virtual!
 - Tabela de página reside em memória
 - Acesso à tabela de página
 - Acesso à memória física



- Os acessos à memória virtual contêm localidade espacial e temporal
- Cache de endereços traduzidos recentemente
- TLB – *Translation-lookaside buffer*
- Entrada do TLB pode incluir *tag*, página física e *reference, dirty, access bits*
- Em cada acesso o TLB é verificado
 - *Hit* - continua
 - *Miss* - verificar a tabela de página
 - Podem originar (ou não) *page faults*
 - Podem ser tratados por hardware ou software

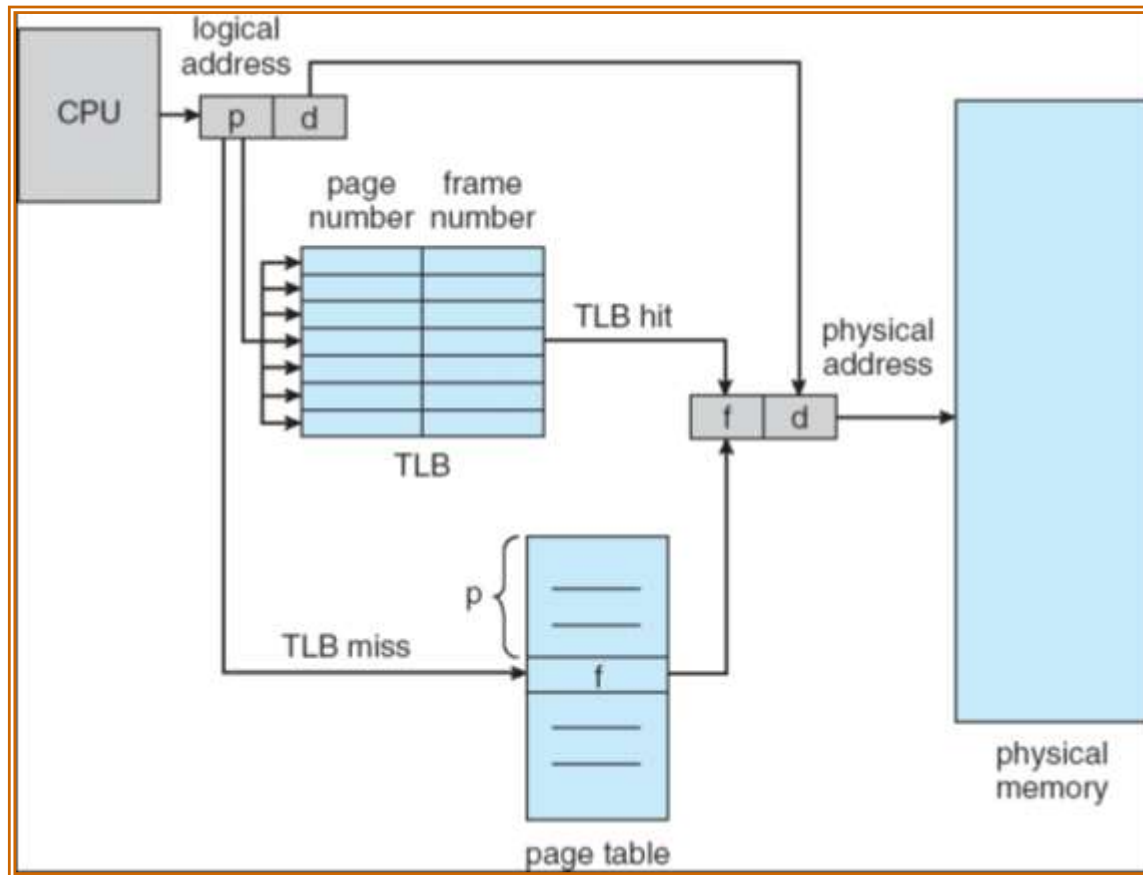
- TLB é uma memória associativa com procura paralela

Page #	Frame #

Tradução de endereços

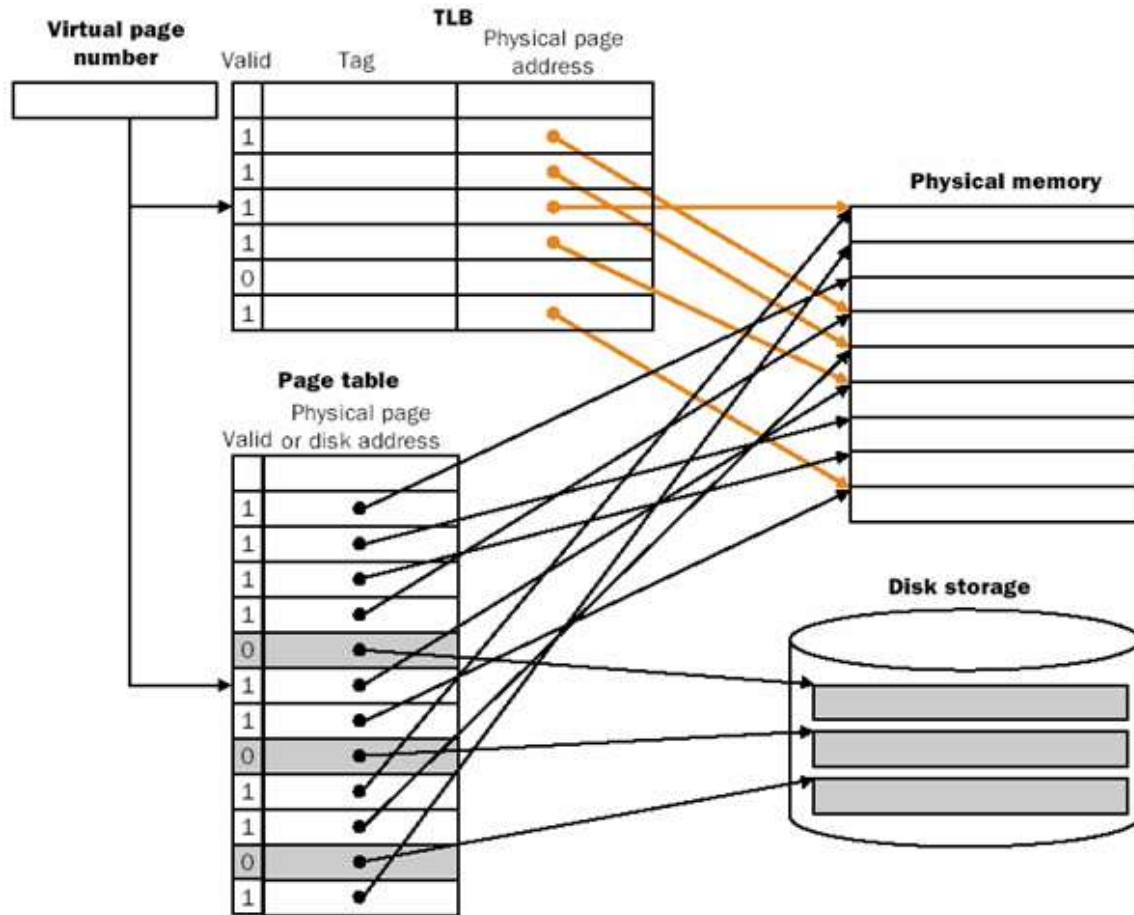
- Se p está na tabela associativa, determina # frame no TLB
- Senão procura # frame na tabela de página

Memória virtual com TLB



- TLB funciona como cache das entradas da tabela de página
- Páginas residentes em disco não são referenciadas no TLB
- *Tag, valid, etc*

Memória virtual com TLB



- TLB funciona como cache das entradas da tabela de página
- Páginas residentes em disco não são referenciadas no TLB
- *Tag, valid, etc*