“Arquitetura e Organização de Computadores I – Aula_10 – Hierarquia de Memória - Continuação”

Prof. Dr. Emerson Carlos Pedrino

DC/UFSCar

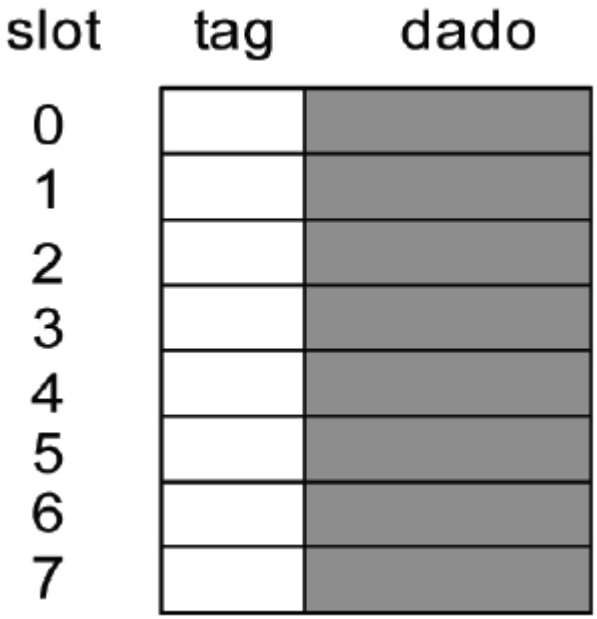
São Carlos

Sumário

- *Cache* em mapeamento associativo
- Memória virtual

Cache em mapeamento associativo

- Mapeamento associativo por conjunto de 1 via (mapeamento direto):



1 *slot* por bloco-> mapeamento inflexível

Alternativa

- Um mapeamento com associatividade maior do que 1 via permite diminuir a taxa de falta.
- Mapeamento de um bloco num conjunto de *slots*, da *cache*.

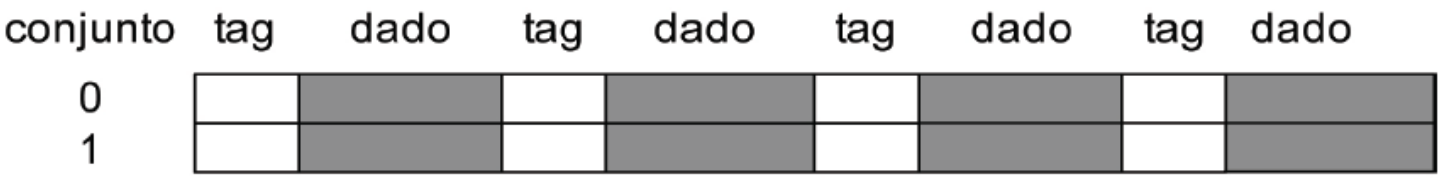
Mapeamento associativo por conjunto de 2 vias

- Um índice da *cache* aponta para uma linha de 2 *slots*.
- Um determinado bloco de memória tem um índice específico, como no mapeamento direto, assim, o bloco deve ser mapeado numa única linha.
- A associatividade de 2 vias significa que, naquela linha, a cópia do bloco pode estar em qualquer um dos 2 *slots*.

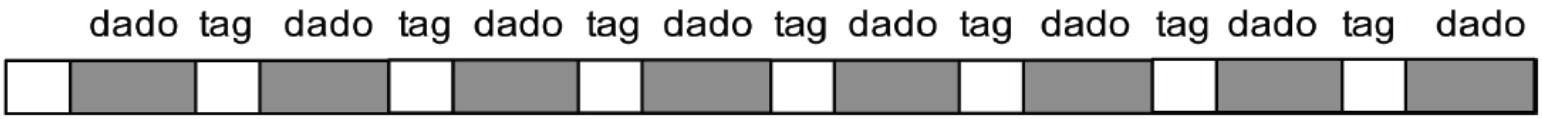
Mapeamento associativo por conjunto de 2 vias

- Portanto, 2 blocos diferentes com mesmo índice podem compartilhar uma mesma linha. No mapeamento direto, uma referência subsequente a esses dois blocos implica na substituição do primeiro para poder carregar o segundo.

Mapeamento associativo por conjunto de 4 vias



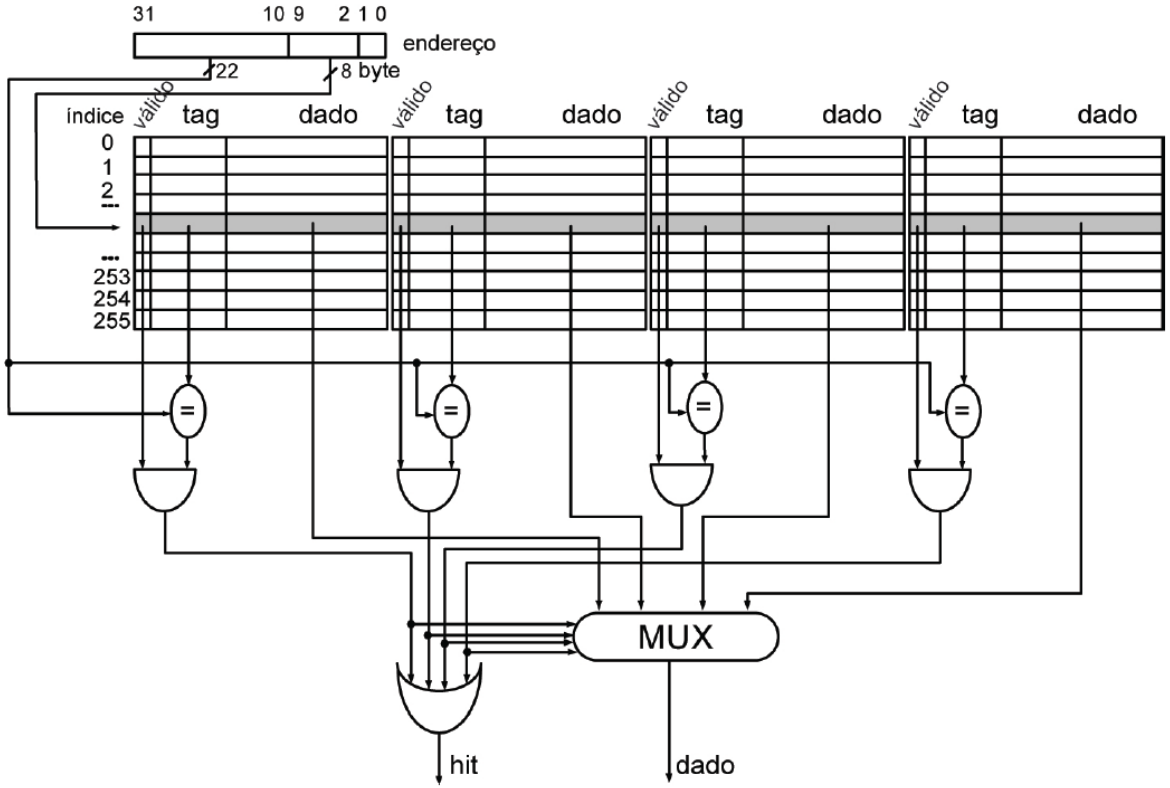
Cache de 1 índice – Totalmente associativo



- Quanto maior a associatividade, mais complexo o circuito da *cache*, pois todos as *tags* devem ser comparadas com a *tag* do endereço da CPU, uma vez que a cópia do bloco pode estar em qualquer *slot*.

Organização da *cache* associativa por conjunto de 4 vias.

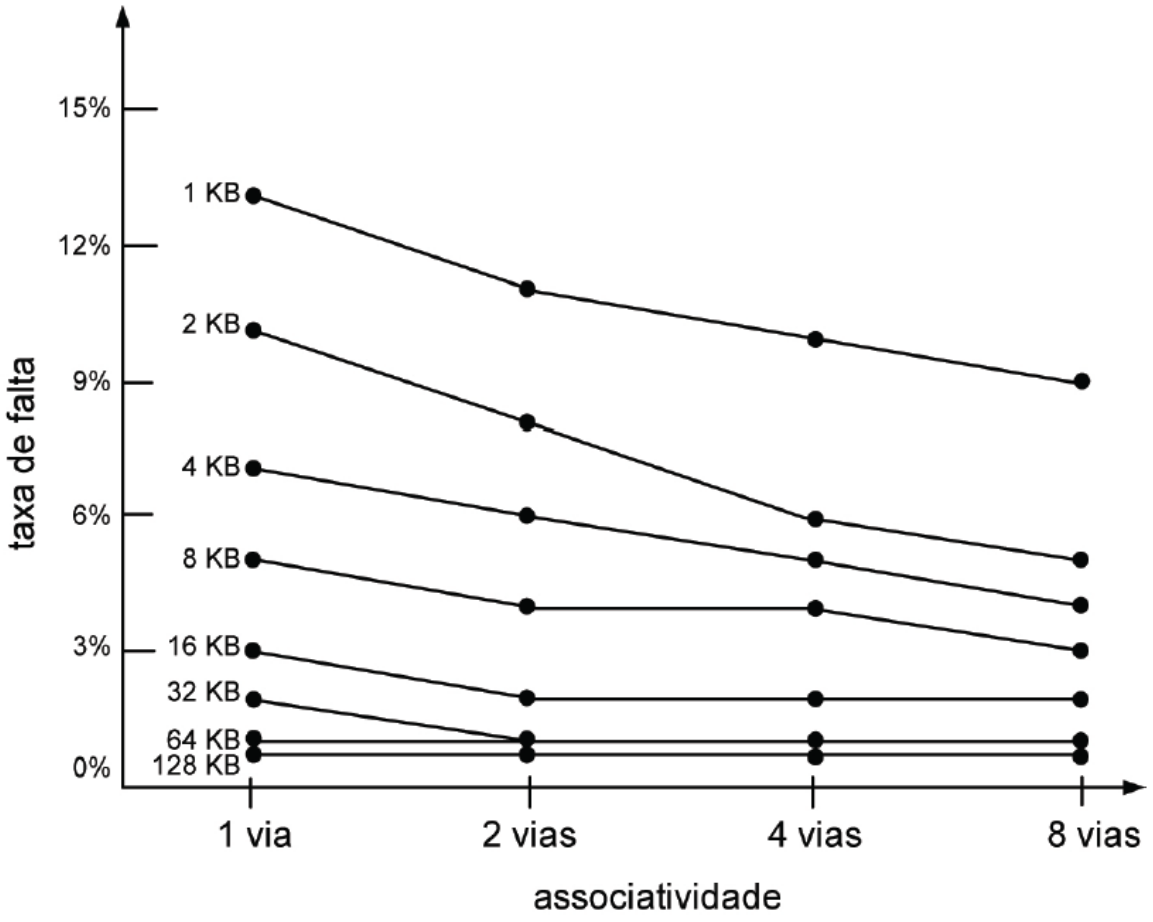
- Memória organizada em *bytes*. CPU referenciando palavras. 4 circuitos de comparação das *tags*.



Taxa de falta x associatividade x tamanho

- *Caches* pequenas: >associatividade -> taxa de falta diminui. Um número maior de blocos pode compartilhar uma mesma linha, evitando-se a substituição.
- Tamanho > de *cache* -> menos coincidência dos blocos num mesmo índice, logo, o efeito da associatividade diminui (ex: *cache* de 128 *Kbytes*).

Taxa de falta x associatividade x tamanho



Observações

- Para melhorar o desempenho do computador, existem duas formas:
  - diminuir a taxa de falta;
  - diminuir a penalidade.
- Diminuir a penalidade implica em diminuir a latência de acesso à memória. Uma possibilidade é adicionar um segundo nível de *cache*. *Cache* nível 1: dentro do processador. Também é possível adicionar outra *cache* acima da MP (nível 2).

Exemplo

- Numa máquina de *500 MHz* com 5% de taxa de falta e acesso à DRAM de *200 ns*, adicionando-se *cache* nível 2 de tempo de acesso de *20 ns*, a taxa de falta cai para 2%.

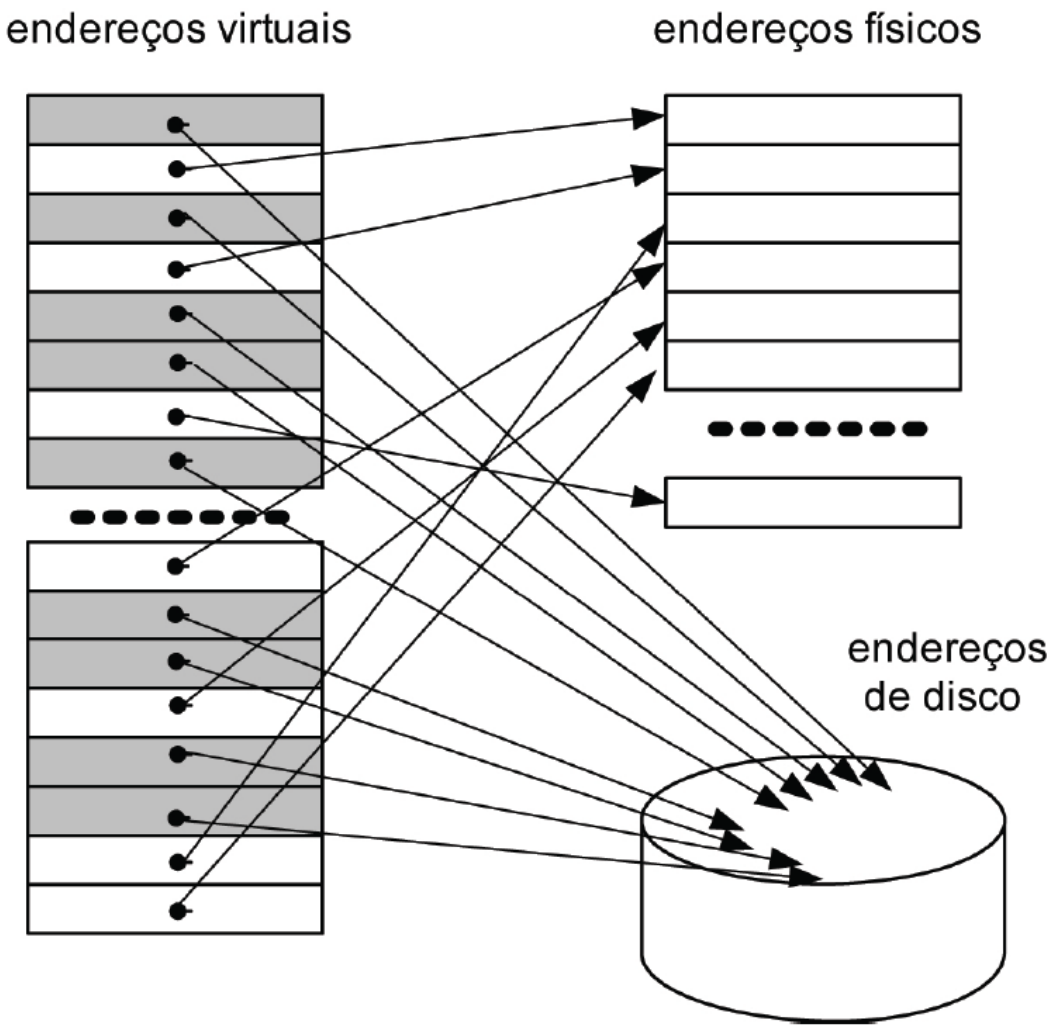
Memória Virtual

- Sistema de memória virtual: disco magnético como nível de memória inferior à MP (memória principal) –> Memória física.
- Cópias de porções de memória do disco são carregadas na memória DRAM, e quando referenciadas num sistema de memória virtual como cópias de blocos da memória DRAM, são carregadas na *cache*, num sistema de *cache*.
- Esse tipo de sistema apresenta a ilusão de se ter uma memória física enorme, possibilitando a relocação e a proteção de memória.

Memória Virtual

- CPU -> manipula a memória com endereços virtuais.
- Memória virtual -> implementada no disco magnético.
- Quando a CPU faz referência à memória, por endereço virtual, o conteúdo pode estar na memória física, caso aquele endereço já tenha sido referenciado. Senão, o conteúdo deve ser lido do disco.

Memória Virtual



Página

- Porção de dados transferidos de um vez do disco para a memória física.
- Latência grande e transferência por setor.
- Uma página tem tamanho de múltiplos setores.
- Falta de página -> os dados não estão na memória física e devem ser recuperados do disco. Gerenciamento por *software*.
- Tal penalidade é grande, logo, usam-se páginas grandes (4 *Kb*, por exemplo). Também é importante reduzir as faltas de página.
- Escrita: feita pela abordagem de *write-back*. Assim, a escrita no disco é feita somente quando uma página carregada na memória física deve ser substituída.

Tradução de endereço virtual para físico

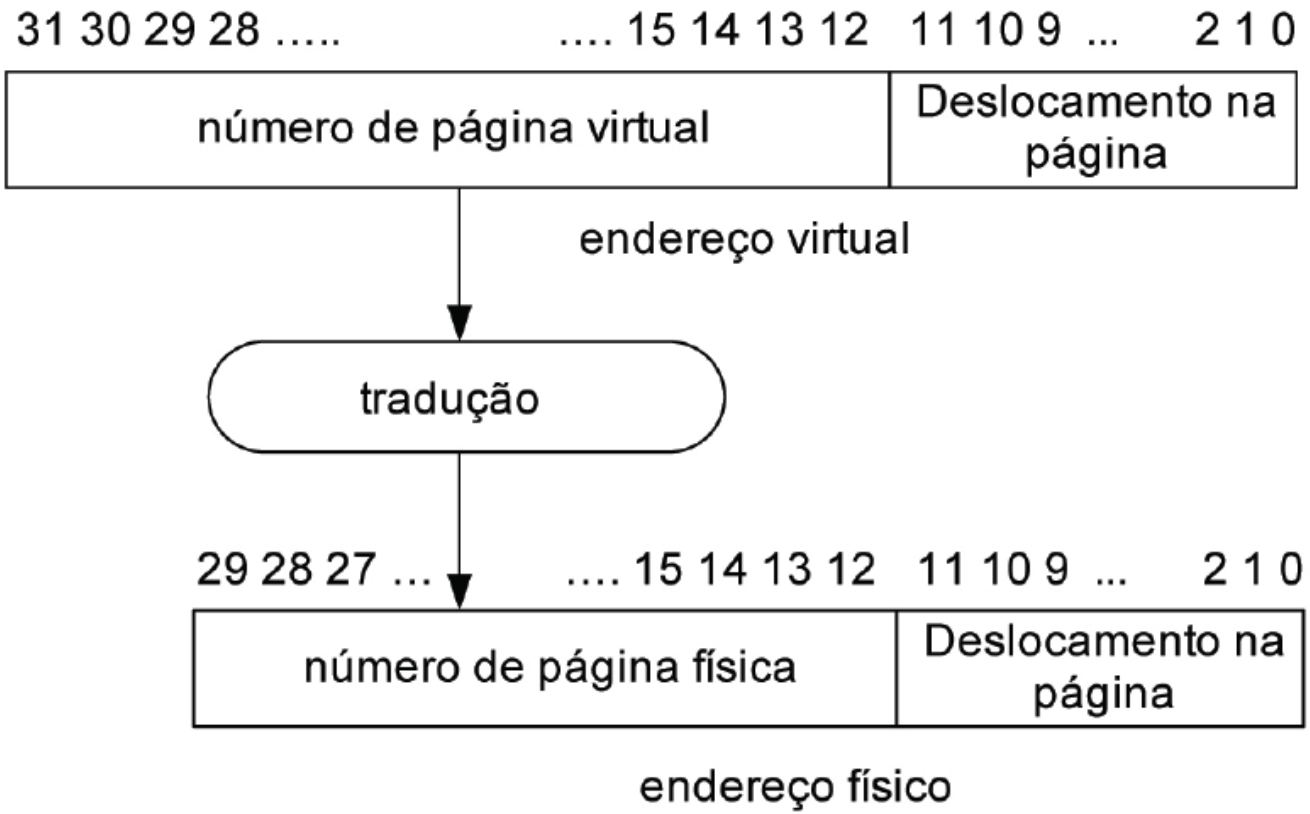


Tabela de página

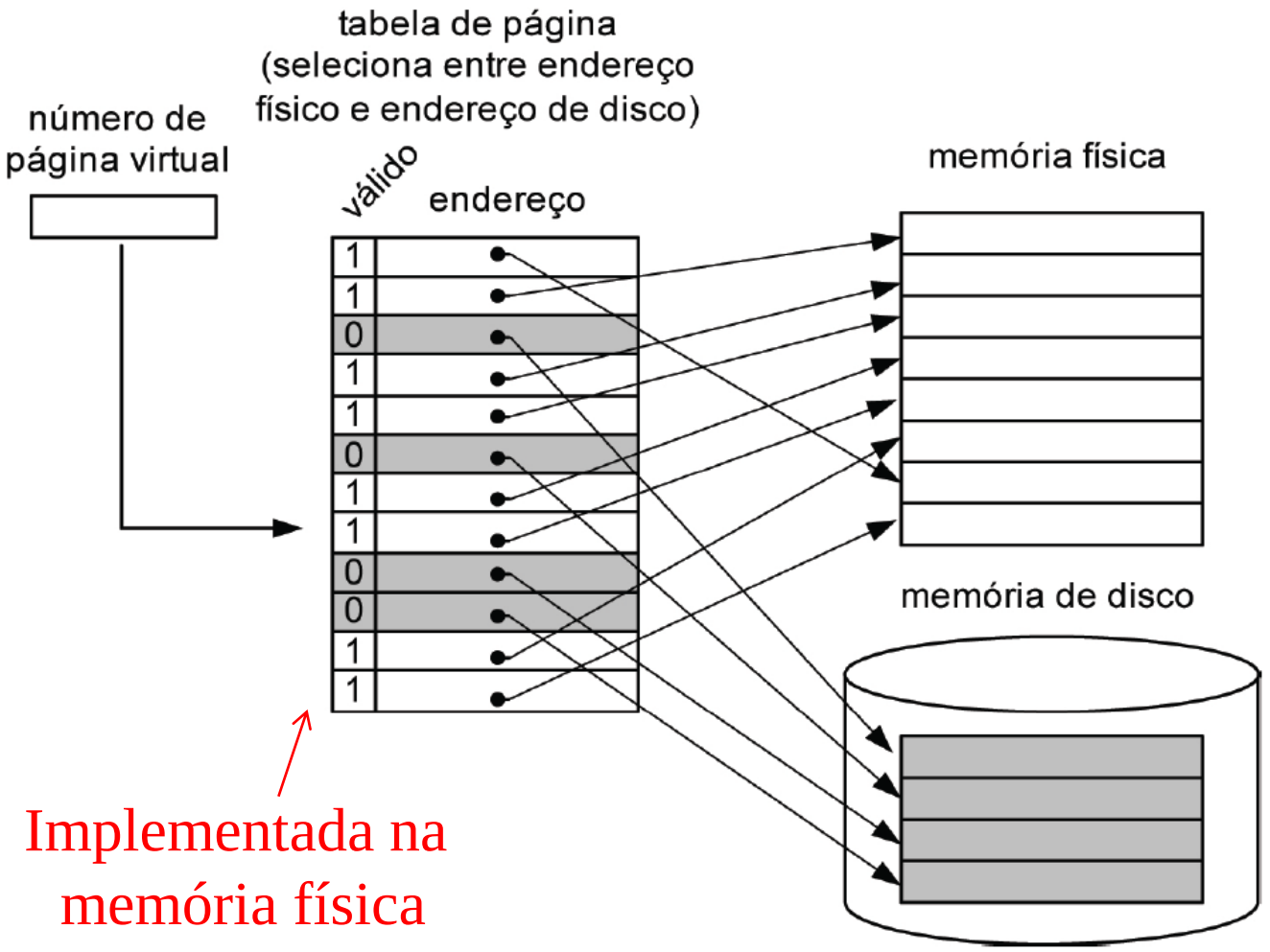
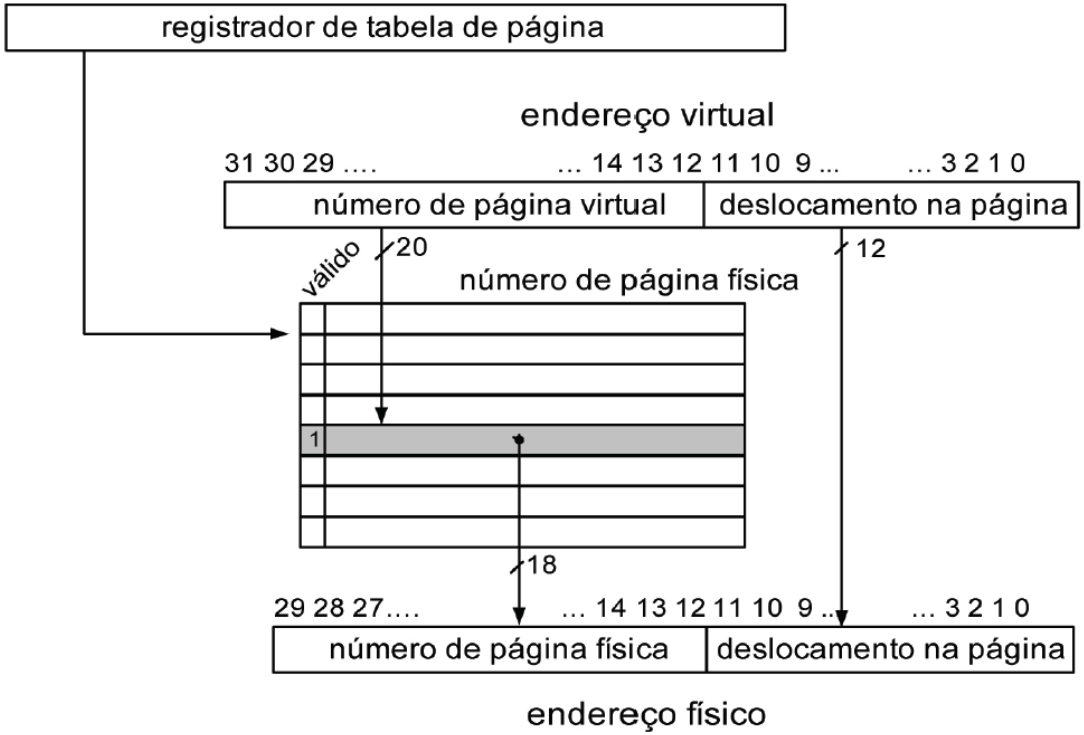


Tabela de página apontada por um registrador de tabela de página

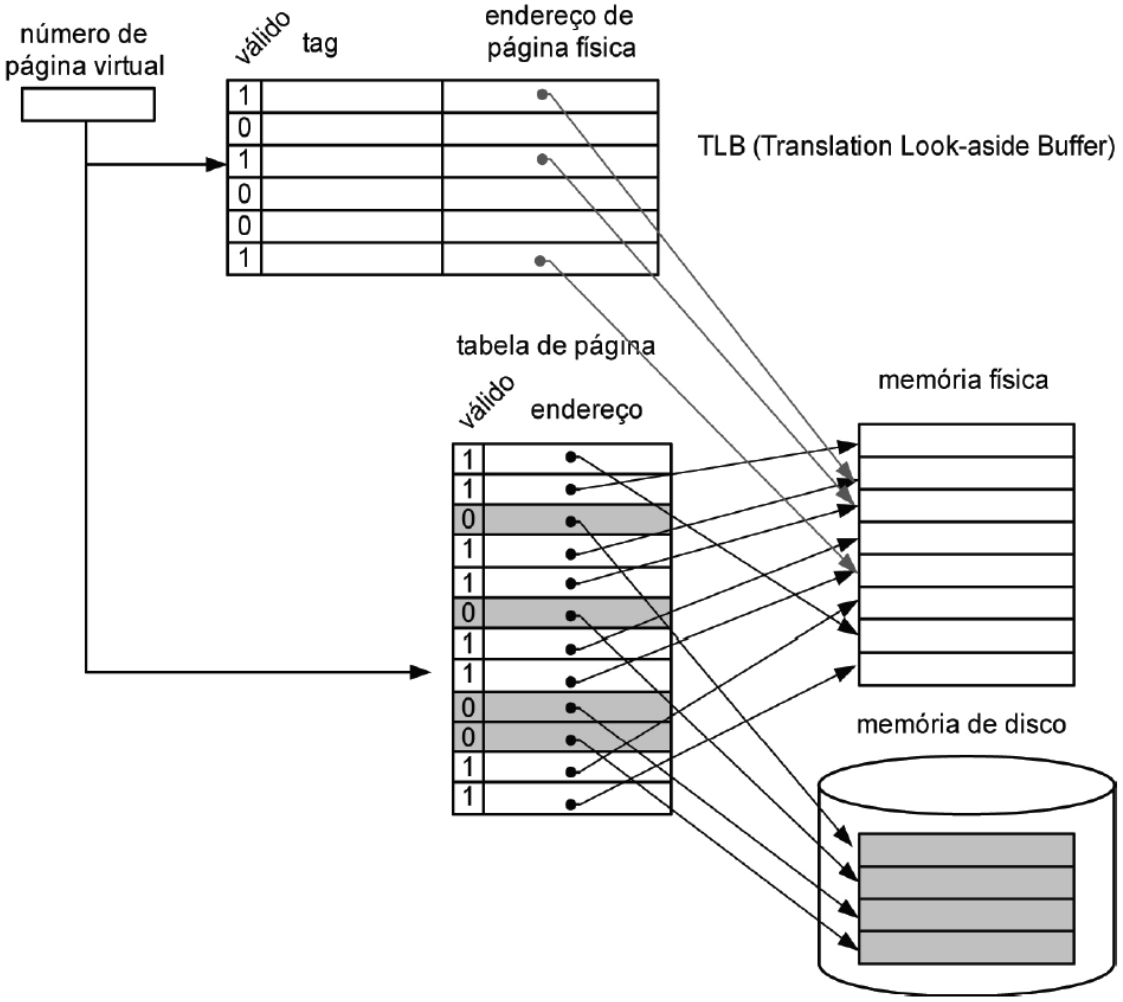
- O diagrama a seguir ilustra a possibilidade de existência de tabelas de página diferentes em ambientes multitarefas, ou seja, cada tabela seria usada por uma determinada tarefa.



Observações

- Uma tabela de página é grande para conter o número de todas as páginas virtuais e, portanto, deve ficar na memória física. Num sistema de memória virtual devemos consultar a tabela de página para verificar o número de página física, o que implica num acesso à memória. Após a consulta, a palavra referenciada deve ser lida ou escrita por meio de outro acesso à memória. Isso torna o sistema de memória lento. Uma forma de melhorar esse tempo é usar um cache específico para a tradução de página, TLB (*Translation Look-aside Buffer*). Assim, a maioria das consultas à tabela de página ocorre na TLB, diminuindo a latência de tradução.

TLB


número de página virtual
válido
tag
endereço de página física
1
0
1
0
0
1
TLB (Translation Look-aside Buffer)
tabela de página
válido
endereço
1
1
0
1
1
0
1
1
0
0
1
1
memória física
memória de disco

Fluxograma de acesso à memória virtual