Universidade Federal do Rio Grande do Sul
Instituto de Informática

Organização de Computadores

Aula 19

Memória virtual
primeira parte

Memória virtual
primeira parte

1. Introdução

2. Paginação

3. TLB - Translation Lookaside Buffer

4. Mecanismos de translação de endereços

1. Introdução

- **Memória principal semicondutora**
  - **capacidade limitada**
  - **tempo de acesso entre 10 e 20 ns**

- **Memória secundária em disco**
  - **capacidade muito maior**
  - **tempo de latência entre 10 e 30 ms**

O problema

- **Nosso computador tem 32 Kbytes de memória principal**
- **Como podemos?**

a) Rodar programas que usam mais do que 32 Kbytes?

Dividir o programa em pedaços <= 32 Kbytes

Deixar que o programador se preocupe como trazer cada pedaço para a memória no momento certo

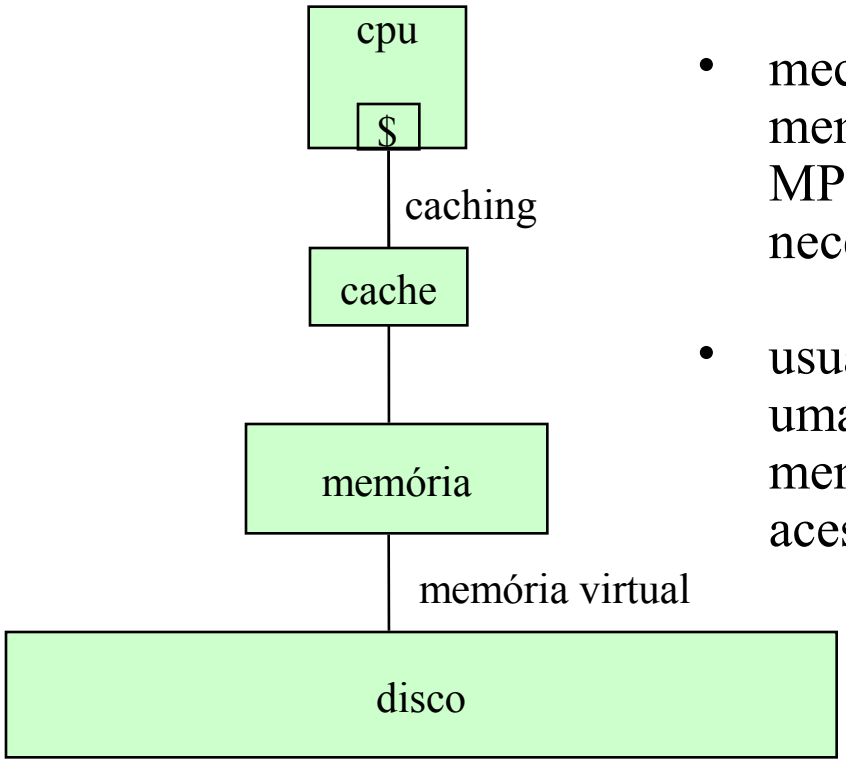
b) Permitir que vários usuários usem o computador?

Paga-se alguém para verificar cada programa e realizar as tarefas acima, considerando os diversos programas como um único programa imenso!

c) Executar vários programas ao mesmo tempo?

Memória Virtual: a solução!

- ***Memória Virtual* : técnica que nos permite ver a memória principal como uma cache de grande capacidade de armazenamento**
- **É apenas mais um nível na hierarquia de memórias**



- mecanismo automático de gerência de memória, que traz automaticamente para a MP os blocos de informação (do disco) necessários
- usuário tem a impressão de trabalhar com uma memória única, do tamanho da memória secundária, mas com tempo de acesso próximo do tempo da MP

Tempo de acesso

Tempo médio de acesso Tma é dado por

Tma = Tm + (1 – h) Ts

onde Tm = tempo de acesso à MP
Ts = tempo de acesso ao disco
h = hit ratio

p.ex.
se Tm = 20 ns, Ts = 20 ms, h = 0.9999
então Tma = 2,02 µs (100 x maior do que Tm)

Por que MV é diferente das caches?

- **Miss penalty é MUITO maior (milhões de ciclos)!**
  Se a informação não está na memória, está no disco!

- **Logo:**
  - ***miss ratio* precisa ser bem menor do que em cache**
  - **alta penalidade do *miss* => necessário buscar blocos maiores em disco**
  - **princípio de localidade opera sobre blocos maiores de dados ou instruções e leva a *hit ratios* bem mais elevados**
  - **mapeamento associativo das páginas**
  - ***misses* são tratados por software (há tempo disponivel)**
  - **técnica de escrita *write-through* não é uma opção. Usa-se *write-back*.**

Terminologia

- **Mesma idéia da cache, mas com terminologia diferente**

Cache	**MV**
bloco	**página (ou segmento)**
cache miss	***page fault***
endereço	**endereço virtual (ou lógico)**
índice	**endereço real (ou físico)**

- **endereço *virtual (lógico)*: gerado pelo programa**
  - **deve endereçar todo espaço em disco**
  - **maior número de bits**

- **endereço *real (físico)*: endereço na memória principal**
  - **menor número de bits**

Unidade de Gerenciamento de Memória

- **MMU (*Memory Management Unit*)**
  - **gerência da hierarquia de memória**
  - **proteção de memória**
  - **usualmente integrada dentro do microprocessador**

- **MMU deve fazer mapeamento do endereço virtual para endereço real**

- **SO usa a MMU**

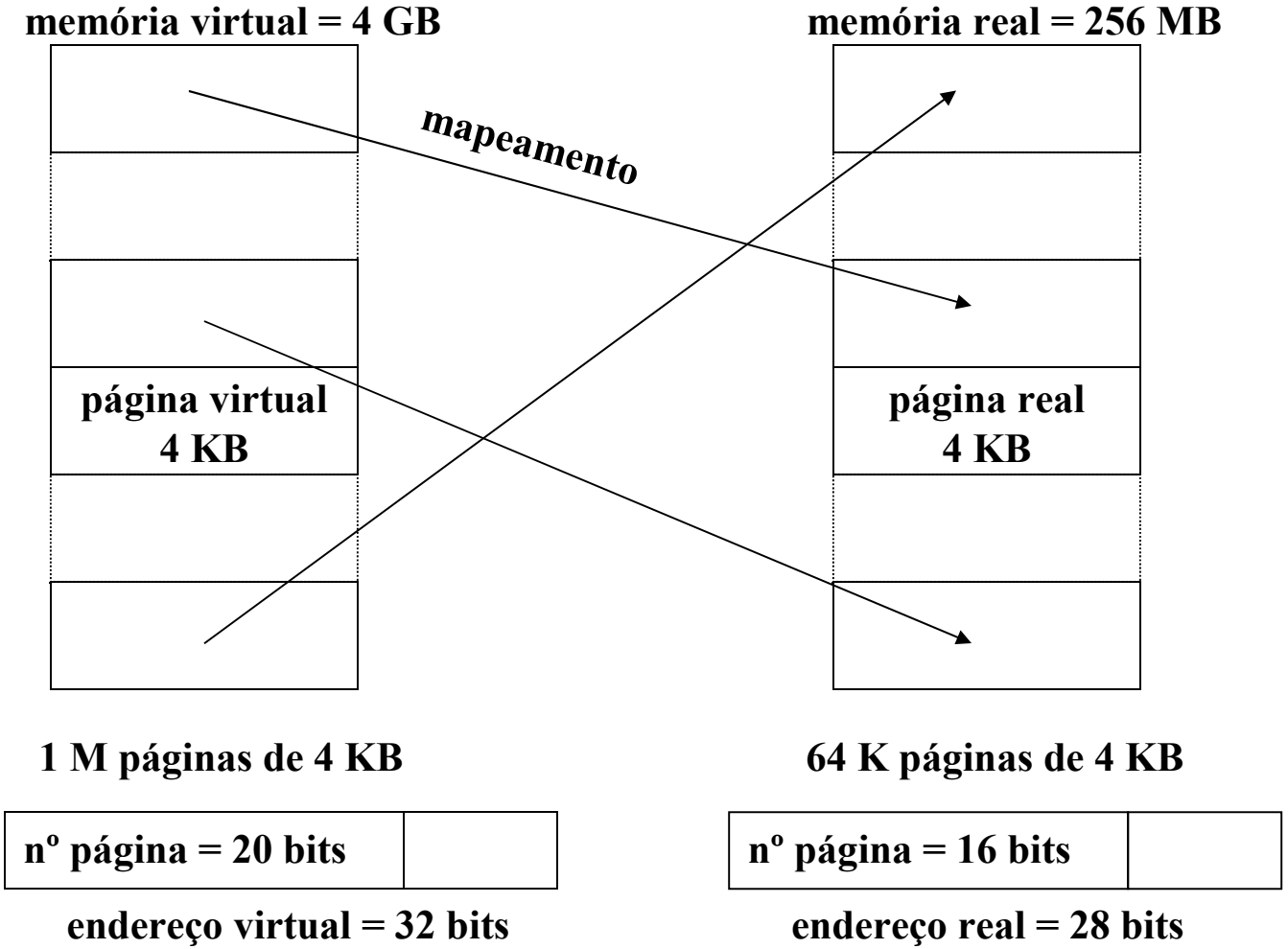
2. Paginação

- **Por que paginação? Resposta: mecanismo simples para tradução de endereços virtuais em reais e para gerenciamento do espaço de memória**

- **Espaços de memória real e virtual divididos em blocos chamados de *páginas***
  - **páginas tem tipicamente de 64 bytes a 4 Kbytes**

- **Endereços virtuais e reais divididos em 2 campos**
  - **endereço da página**
  - **endereço da linha (ou palavra), dentro da página**

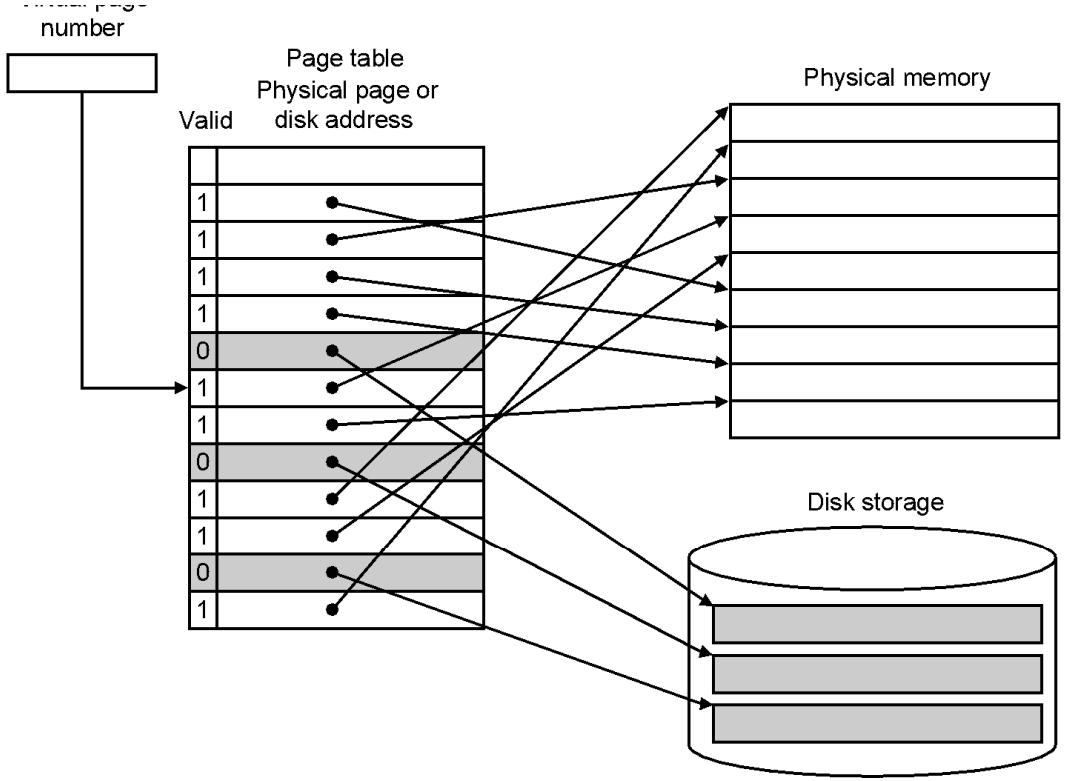
Paginação


memória virtual = 4 GB
memória real = 256 MB
mapeamento
página virtual
4 KB
página real
4 KB
1 M páginas de 4 KB
64 K páginas de 4 KB
nº página = 20 bits
nº página = 16 bits
endereço virtual = 32 bits
endereço real = 28 bits

Paginação

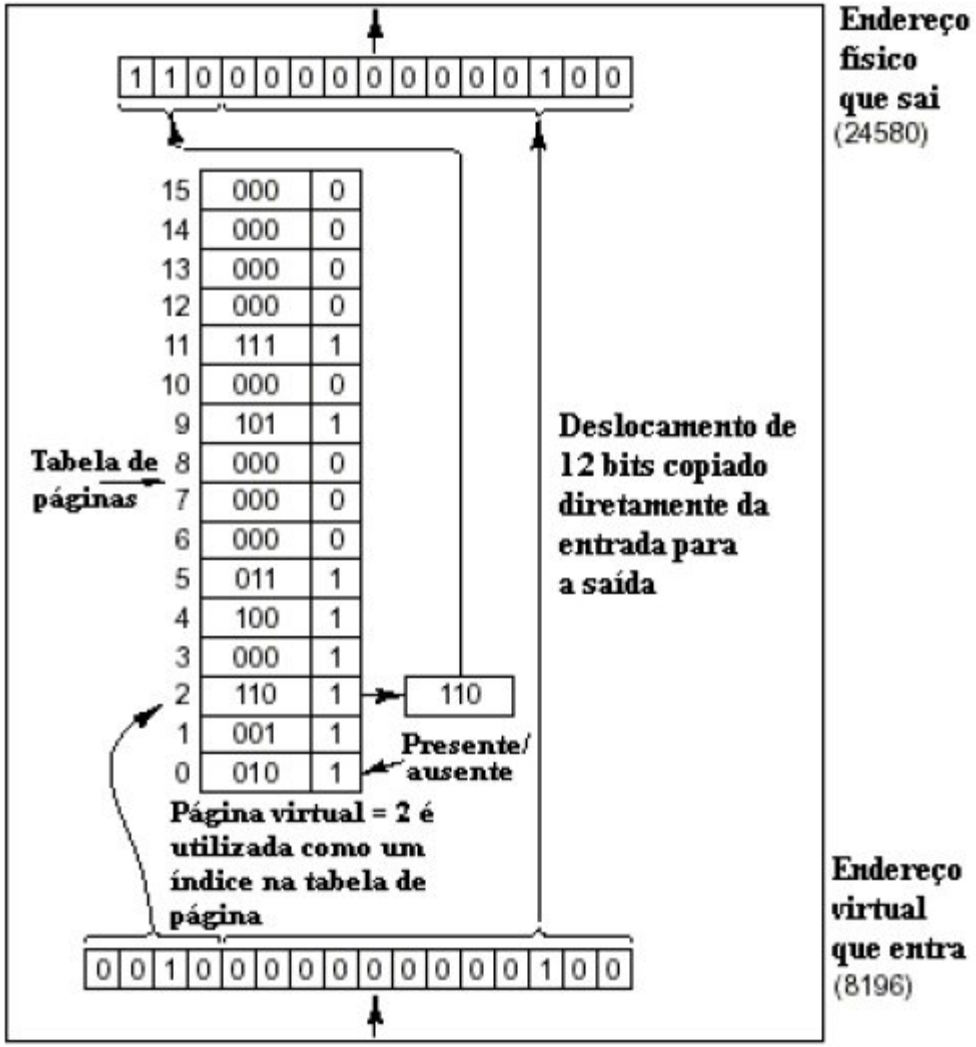
- ***Page fault*** **ocorre quando a página virtual não está na memória principal**
- **Mapeamento completamente associativo, mais eficiente, ajuda a diminuir alta penalidade dos *page faults***
- ***Como transformar endereçamento original do programa no endereçamento real?***
- ***Page tables***
  - **guardam a correspondência entre páginas virtuais e páginas reais**
  - **permitem a translação de endereços**

Paginação



- **É apenas uma função de mapeamento dos endereços virtuais (do disco) para endereços reais (físicos) na memória principal**

Paginação


Endereço físico que sai (24580)
1 1 0 0 0 0 0 0 0 0 0 0 1 0 0
15 000 0
14 000 0
13 000 0
12 000 0
11 111 1
10 000 0
9 101 1
Tabela de páginas
8 000 0
7 000 0
6 000 0
5 011 1
4 100 1
3 000 1
2 110 1
1 001 1
0 010 1
110
Presente/ ausente
Deslocamento de 12 bits copiado diretamente da entrada para a saída
Página virtual = 2 é utilizada como um índice na tabela de página
0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0
Endereço virtual que entra (8196)

Gerência de processos

- **Cada processo tem sua própria tabela de páginas**
  - **processos são compilados para espaços de endereçamento virtuais**
  - **tabela de páginas define toda a utilização do espaço de endereçamento pelo processo**

- **Sistema operacional é responsável pela alocação de espaço físico para o espaço virtual de cada processo**
  - **SO carrega tabela de páginas de cada processo**

- **Hardware possui registrador que aponta para início da tabela de páginas do processo atual**

- **Quando novo processo passa a ser ativo, sistema operacional só precisa atualizar valor deste registrador**

Paginação

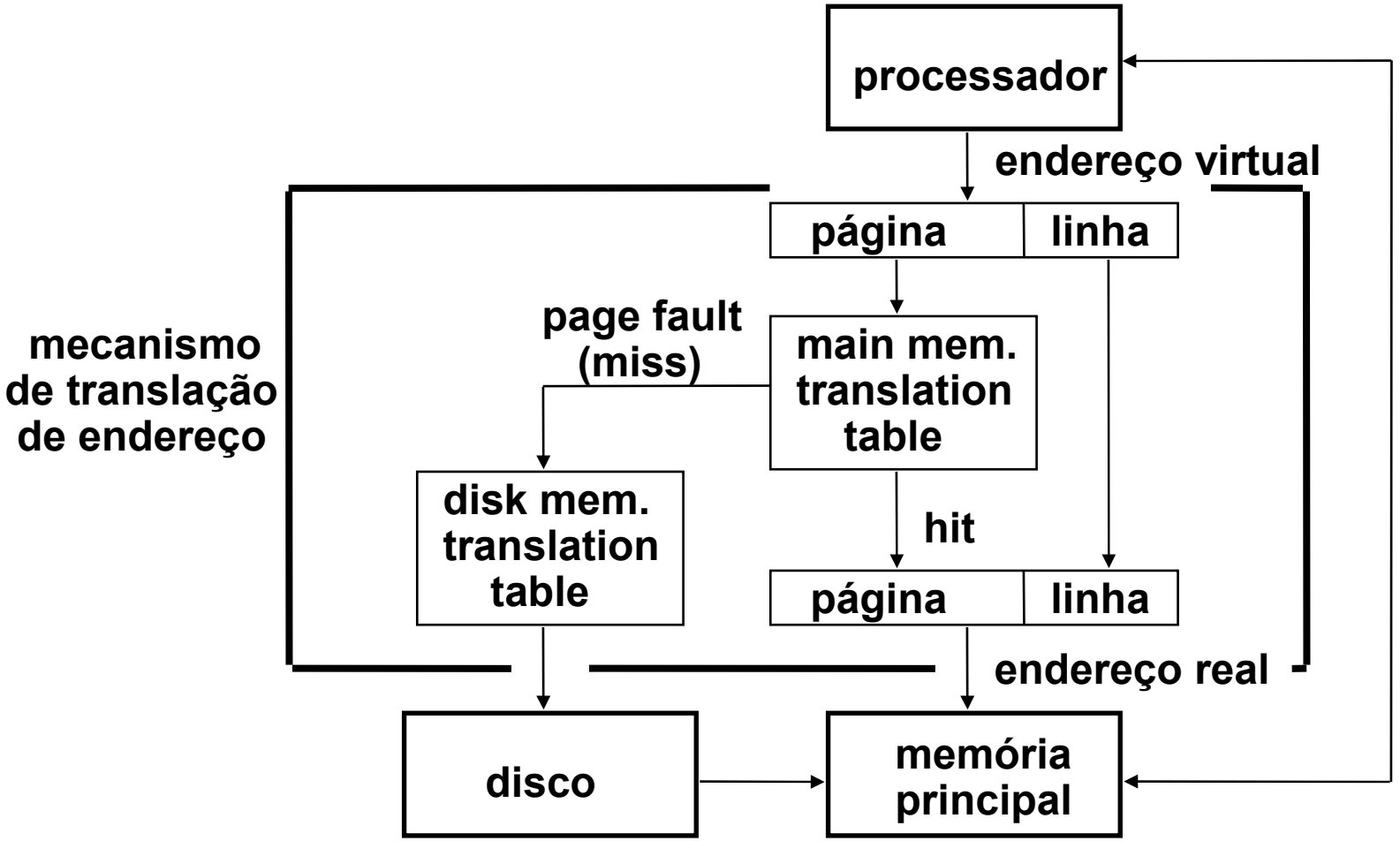
- ***Main memory translation table (MMTT)***
  - **implementada em hardware**
  - **tamanho = nº de páginas na memória principal**

- ***Disk memory translation table (DMTT)***
  - **implementada em software, armazenada na memória principal**
  - **tamanho = nº de páginas em disco**

- **Algoritmo de substituição, em software, para selecionar página da memória principal a ser substituída em caso de *page fault***

- **Bom desempenho é garantido pelo princípio de localidade**

Paginação


processador
endereço virtual
página
linha
mecanismo
de translação
de endereço
page fault
(miss)
main mem.
translation
table
disk mem.
translation
table
hit
página
linha
endereço real
disco
memória
principal

Tamanho de páginas

- **Tamanhos de páginas variam muito, de 64 bytes a 4 Mbytes**

- **Página de pequeno tamanho**
  - **tempo curto para transferência de página entre disco e memória**
  - **muitas páginas de diferentes programas podem estar residentes em memória**
  - **exige *page tables* muito grandes, que ocupam espaço em memória**
  - **mais adequada para instruções**

- **Página de grande tamanho**
  - ***page tables* pequenas**
  - **tempo longo para transferência de página entre disco e memória**
  - **mais adequada para dados (gráficos exigem páginas muito grandes)**

Tamanho de páginas

- **Solução de compromisso: permitir páginas de tamanhos diversos para código e dados**
- **Pentium permite selecionar página de 4 K ou 4 Mbytes**
- **Motorola MC88200**
  - **páginas de 4 Kbytes para programas de usuário**
  - **páginas de 512 Kbytes para programas do sistema, que devem residir sempre em memória**

3. Translation Lookaside Buffer

- **nº de páginas na memória secundária é muito grande**
  - **espaço virtual de 2^{32}bytes, páginas de 4K bytes, 4 bytes por entrada na tabela**
  - **4 MBytes apenas para a tabela de páginas!!!**
  - **tamanho excessivo da *main memory translation table***

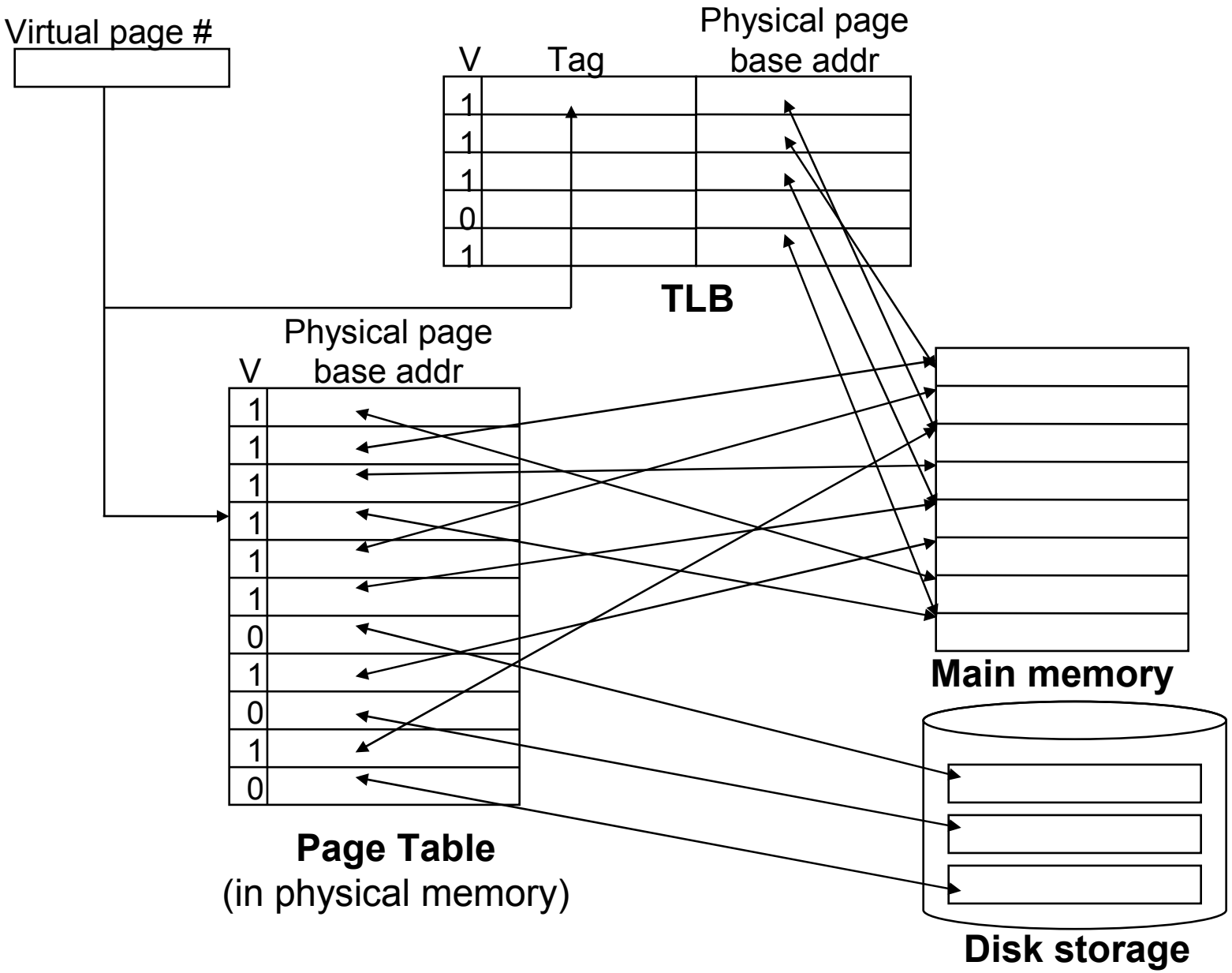
- **Se tabela ficar na memória principal => dois acessos à memória a cada *cache miss***

- ***Working set* = conjunto de páginas mais prováveis de serem acessadas num dado momento, devido ao princípio de localidade**

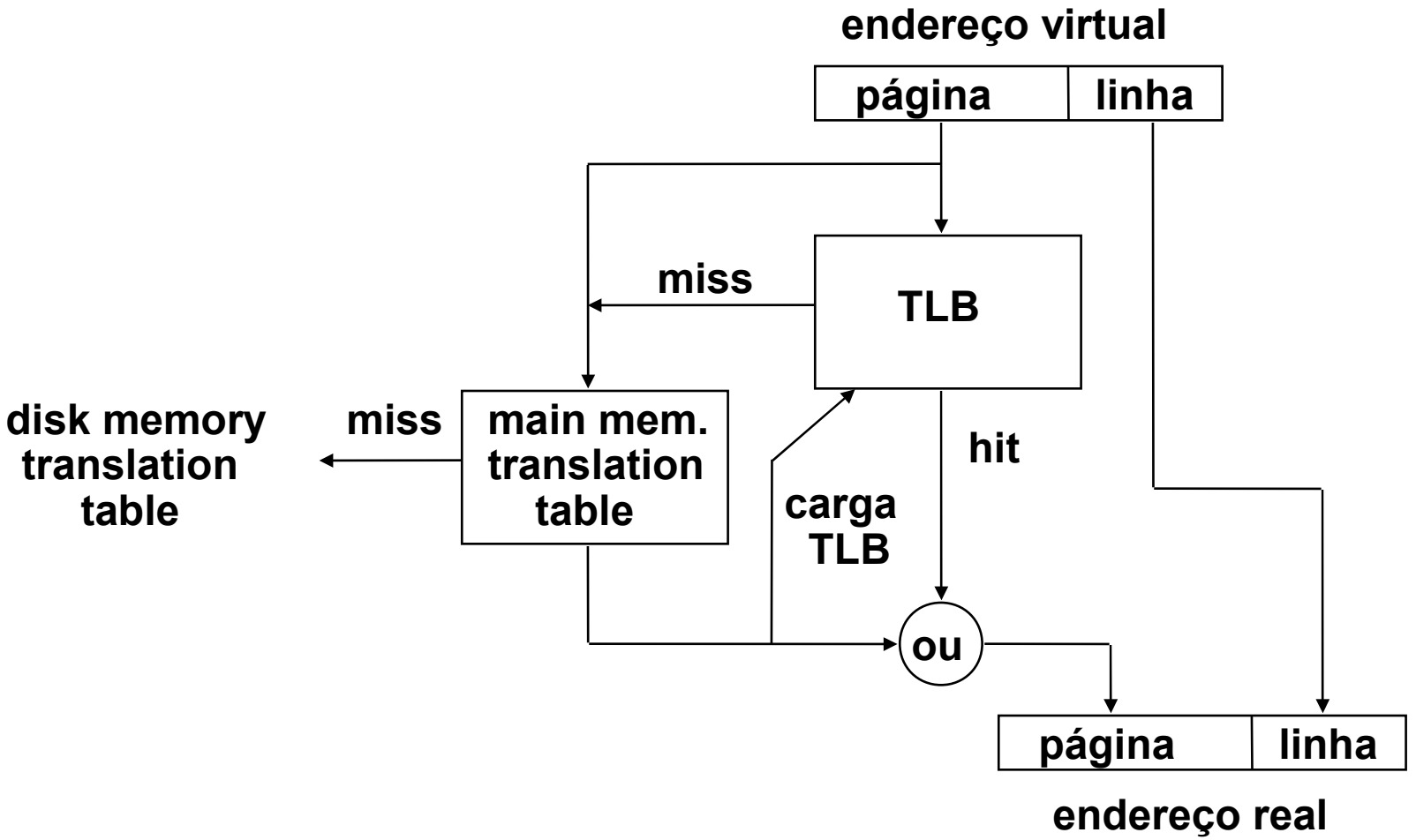
- ***Translation Lookaside Buffer* (TLB)**
  - **implementado em hardware**
  - **traduz endereços virtuais para endereços reais**
  - **só inclui páginas do *working set***
  - **pode ser considerado como uma "cache" da MMTT**

- **Main Memory Translation Table (MMTT)**
  - **implementado o gerenciamento em software**

TLB


Virtual page #
V
Tag
Physical page base addr
1
1
1
0
1
TLB
Physical page base addr
V
1
1
1
1
1
1
0
1
0
1
0
Page Table
(in physical memory)
Main memory
Disk storage

Translation Lookaside Buffer



Translation Lookaside Buffer

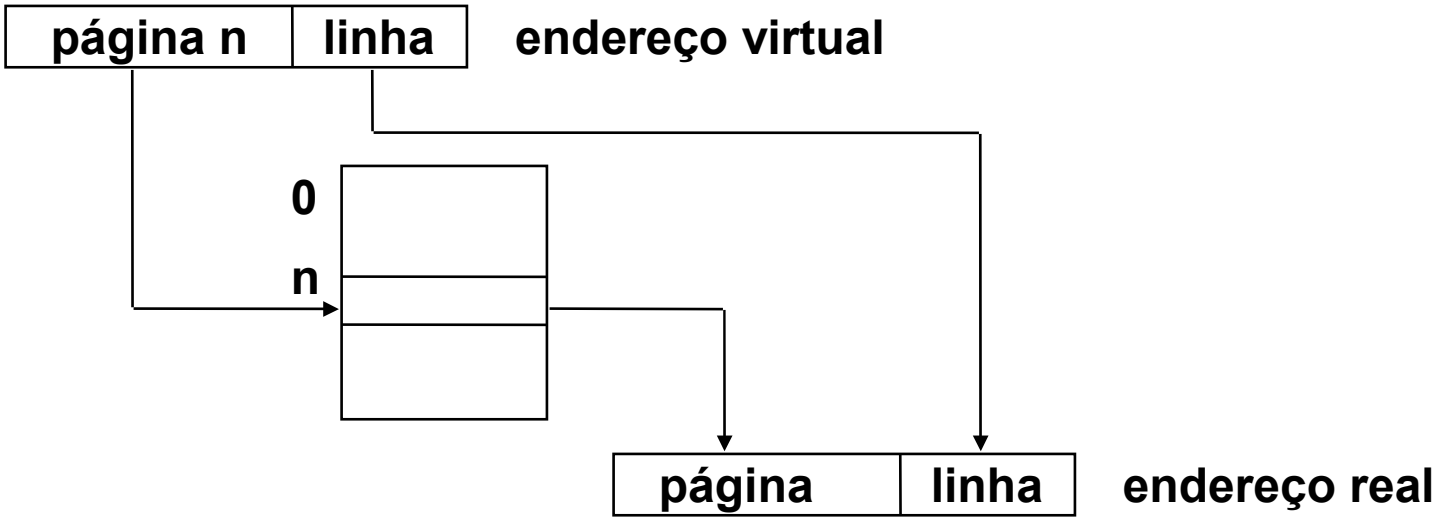
- **Algoritmo de substituição entre TLB e MMTT**
  - **usualmente em hardware**

- **Diversos microprocessadores recentes processam *misses* no TLB em software**
  - **menor desempenho**
  - **maior flexibilidade no algoritmo de substituição**

- **Exemplos**
  - **MIPS**
  - **Alpha**
  - **HP PA**
  - **UltraSparc**

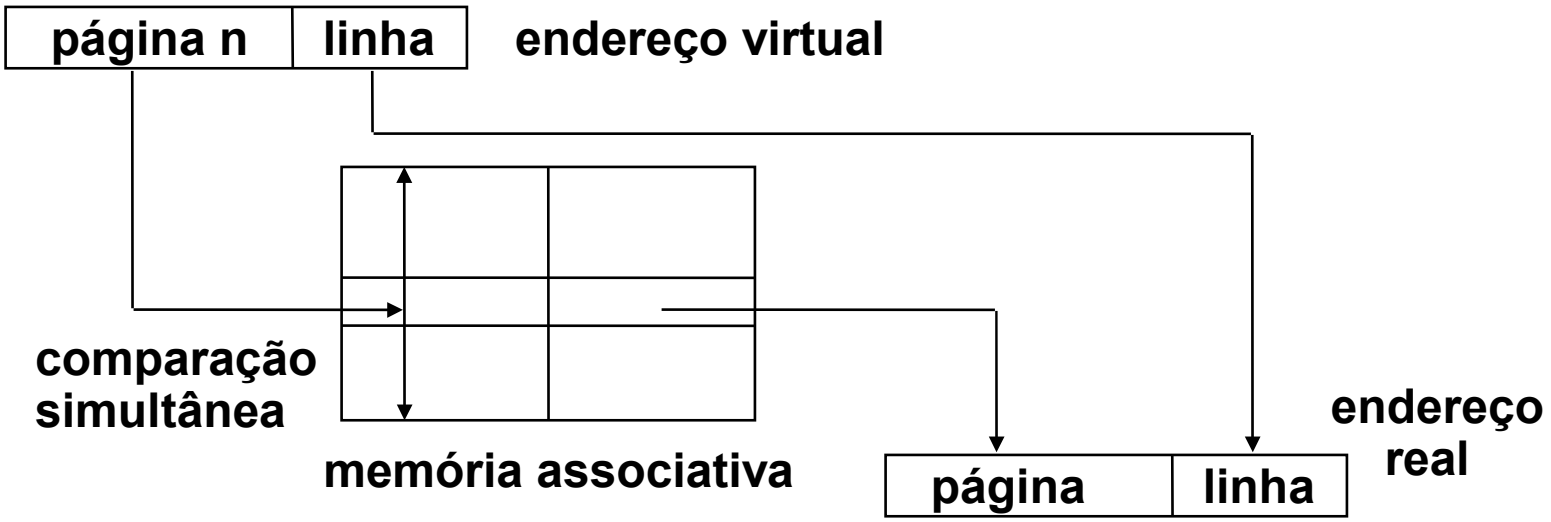
4. Mecanismos de translação de endereços
mapeamento direto

- **Endereço de página virtual é utilizado como endereço de uma memória cujo conteúdo é o endereço de página real procurado**
- **Tamanho = nº de páginas na memória virtual**
- **Utilizado na MMTT (em software), mas não na TLB**



Mecanismos de translação de endereços
mapeamento completamente associativo

- **Memória associativa contém endereços virtual e real**
- **Comparação simultânea com todos os endereços virtuais**

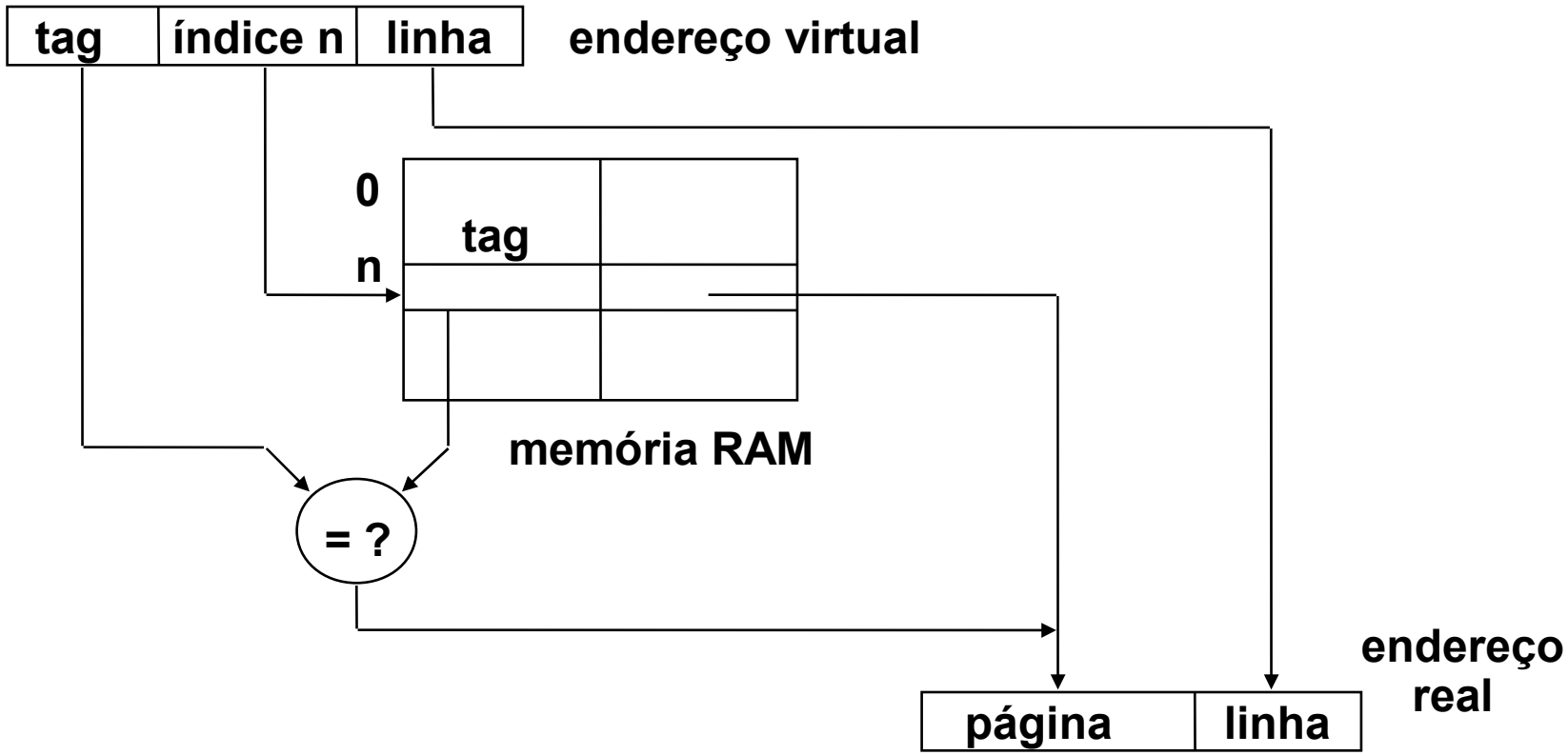


Mecanismos de translação de endereços
mapeamento completamente associativo

- **Utilizado em diversos microprocessadores:**
  - **MIPS R2000 / R3000 – TLB com 64 posições**
  - **Motorola RISC MC88100 – TLB com 56 posições**
  - **Alpha 21164**
    - **TLB de instruções – 48 posições**
    - **TLB de dados – 64 posições**

Mecanismos de translação de endereços

mapeamento conjunto – associativo (*1-way*)



Mecanismos de translação de endereços
mapeamento conjunto – associativo

- **Endereços divididos em 3 campos: tag, índice, linha**
- **Endereço da página = tag e índice**
- **1–way associativo: cada posição da tabela contém um par < end. página virtual, end. página real >**
  - **apenas um comparador**
  - **endereços de páginas virtuais armazenados na tabela têm índices diferentes**
- **N–way associativo: cada posição da tabela contém n pares de endereços de página**
  - **n comparadores**
  - **n endereços de páginas virtuais armazenados na tabela têm mesmo índice**

Mecanismos de translação de endereços
mapeamento conjunto – associativo

- **Motorola 68040**
  - **2 TLBs 4-way associativos, com 64 posições**
- **Intel 486**
  - **TLB 4-way associativo, com 32 posições**
- **Intel i860**
  - **TLB 4-way associativo, com 64 posições**
- **PowerPC 604**
  - **TLB 8-way associativo, com 64 posições**
- **Pentium II**
  - **TLB de instruções. 4-way associativo, com 32 posições**
  - **TLB de dados, 4-way associativo, com 64 posições**

Combinações de eventos na TLB

TLB	Page Table	Cache	Possível? Quais circunstâncias?
Hit	Hit	Hit	Sim – situação desejada!
Hit	Hit	Miss	Sim –embora a Page Table não é verificada se existe acerto da TLB (hits)
Miss	Hit	Hit	Sim – TLB miss, PA esta na page table
Miss	Hit	Miss	Sim – TLB miss, PA esta na page table, mas dado não na cache
Miss	Miss	Miss	Sim – page fault
Hit	Miss	Miss/ Hit	Impossível – a translação da TLB não é possível se a página não está presente na memória
Miss	Miss	Hit	Impossível – dados não permitidos na cache se a página não está na memória

FIM