

ARQUITETURA DE COMPUTADORES

Memória Virtual

1

Introdução

Nosso computador tem 32 Kbytes de memória principal.
Como podemos:

- ❑ 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 (*overlay*)
- ❑ 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!
- ❑ Executar vários programas ao mesmo tempo?

2

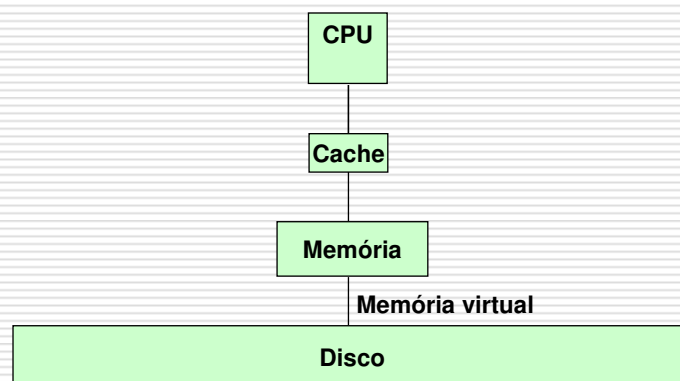
Memória Virtual

- ❑ Memória principal (MP) semicondutora
 - capacidade limitada
 - tempo de acesso entre 10 e 20 ns
- ❑ Memória secundária (MS) em disco
 - capacidade muito maior
 - tempo de latência entre 10 e 30 ms
- Se o programa não cabe inteiramente na MP, deve ser trazido da MS (disco) em partes
- Alternativa: mecanismo automático de gerência de memória, que traz automaticamente para a MP os blocos de memória (do disco) necessários
- **Memória virtual**: 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

3

Memória Virtual - conceito

- **Memória Virtual**: técnica que nos permite ver a memória principal como uma cache de grande capacidade de armazenamento



4

Por que MV é diferente de caches?

- ❑ *Miss penalty* é MUITO maior (milhões de ciclos)! Se não está na memória está no disco!
- ❑ logo:
 - *Miss ratio* precisa ser bem menor do que em cache
 - Necessário buscar blocos maiores em disco => alta penalidade do *miss*
 - 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 disponível e algoritmos mais complexos)
 - técnica de escrita *write-through* não é uma opção. Usa-se *write-back*

5

Terminologia

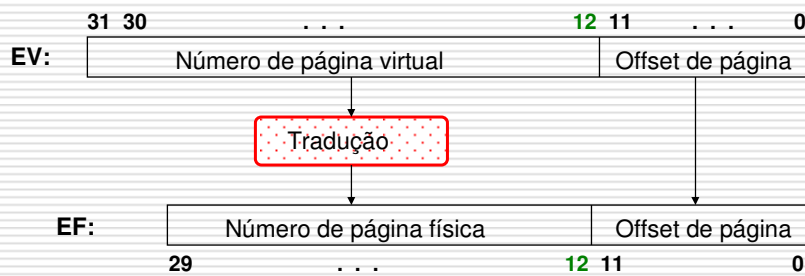
- ❑ Mesma idéia da cache, mas com terminologia diferente

■ <u>cache</u>	<u>MV</u>
■ Bloco (linha)	página
■ <i>cache miss</i>	<i>page fault</i>
■ endereço	endereço virtual (lógico)
■ índice	endereço real (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

6

Tradução de endereço

Um **endereço virtual (EV)** é traduzido para um **endereço físico (EF)** pela combinação de hardware e software



O tamanho de página é $2^{12} = 4\text{KB}$.

O número de páginas físicas permitido na memória é 2^{18} .

A memória principal pode ter no máximo 1GB.

O espaço de mem virtual possui 4GB.

7

Unidade de gerenciamento de memória

- ☐ Quem faz a tradução?
- ☐ 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
- ☐ sistema operacional usa a MMU

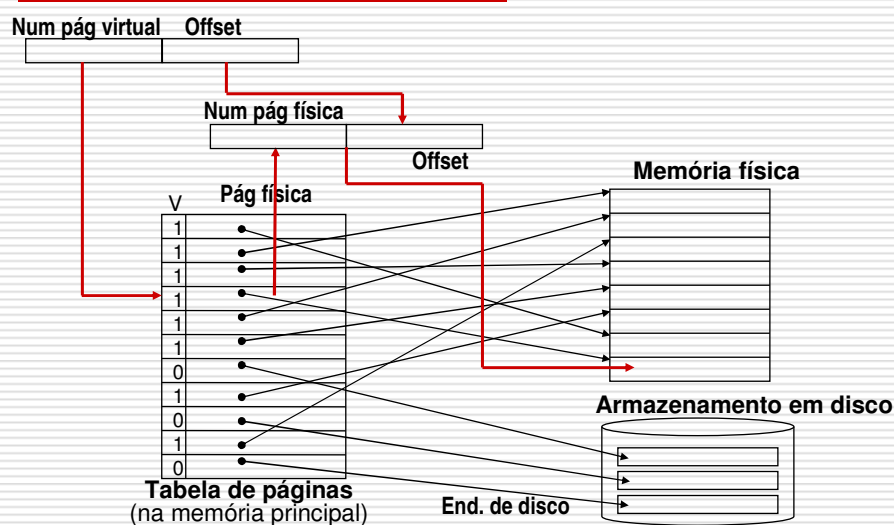
8

Propriedades da MV

- Espaços de memória real e virtual divididos em blocos chamados de *páginas*
 - páginas tem tipicamente de 4 Kbytes a 64Kbytes
- *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 (tradução) de endereços

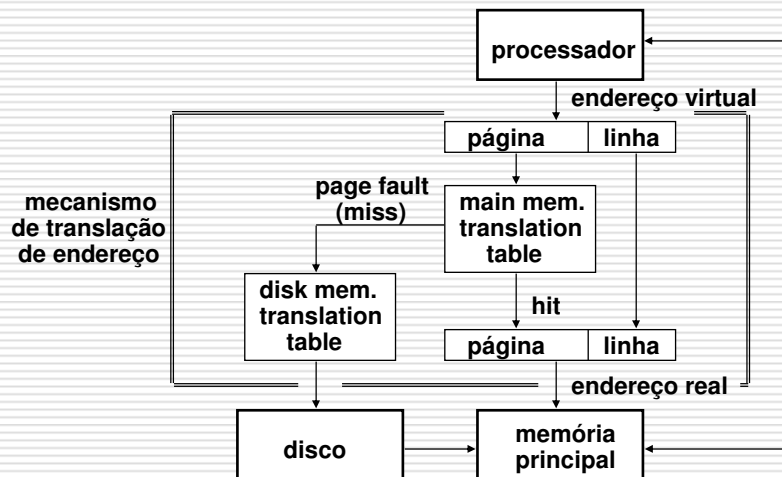
9

Mecanismo de tradução de endereço



10

Mecanismo de tradução de endereço – outra visão



11

Mecanismo de tradução de endereço - conceitos

- ❑ *Main memory translation table (MMTT)*
 - implementada em hardware (em tese)
 - 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 seleccionar página da memória principal a ser substituída em caso de *page fault*
- ❑ Bom desempenho é garantido pelo princípio de localidade

12

Tamanho de páginas varia muito

- 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)
- Solução de compromisso: permitir páginas de tamanhos diversos para código e dados (Pentium permite selecionar página de 4KB ou 4 MB)

13

Literatura

- *Patterson – cap 7*

14

O único lugar onde o sucesso
vem antes do trabalho é no
dicionário.

Albert Einstein

15

Procure ser um homem de
valor, em vez de ser um
homem de sucesso.

Albert Einstein

16

Questões adicionais...

- Investigue o conceito de TLB (Translation Lookaside Buffer).
 - Por que TLBs são utilizadas em hierarquia de memória?
 - Como funciona?

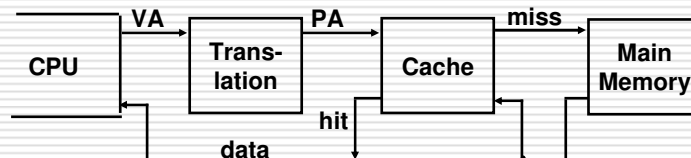
17

Translation Lookaside Buffer

- N° de páginas na memória secundária é muito grande
 - espaço virtual de 2^{32} bytes, páginas de 4 Kbytes $\Rightarrow 2^{20}$ páginas
 - Supondo 4 bytes por entrada na tabela, 4 MBytes apenas para a tabela de páginas!!!
 - tamanho excessivo da *main memory translation table*
- Dois acessos a memórias a cada cache miss
 - conversão de endereço na MMTT + dado na memória principal
- *Working set* = conjunto de páginas mais prováveis de serem acessadas num dado momento, devido ao princípio de localidade
- *Translation Look-Aside Buffer (TLB)*
 - implementado em hardware
 - traduz endereços virtuais para endereços reais
 - só inclui páginas do *working set*
- MAIN Memory Translation Table (MMTT)
 - implementada em software

18

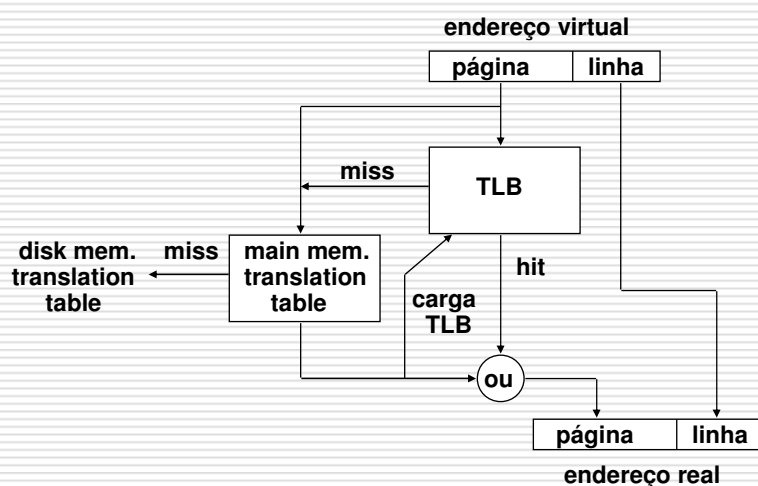
Translation Lookaside Buffer



Translation Lookaside Buffer (TLB) – uma pequena cache que mantém os mapeamentos de endereços usados recentemente, evitando fazer uma pesquisa na tabela de tradução (tabela de páginas)

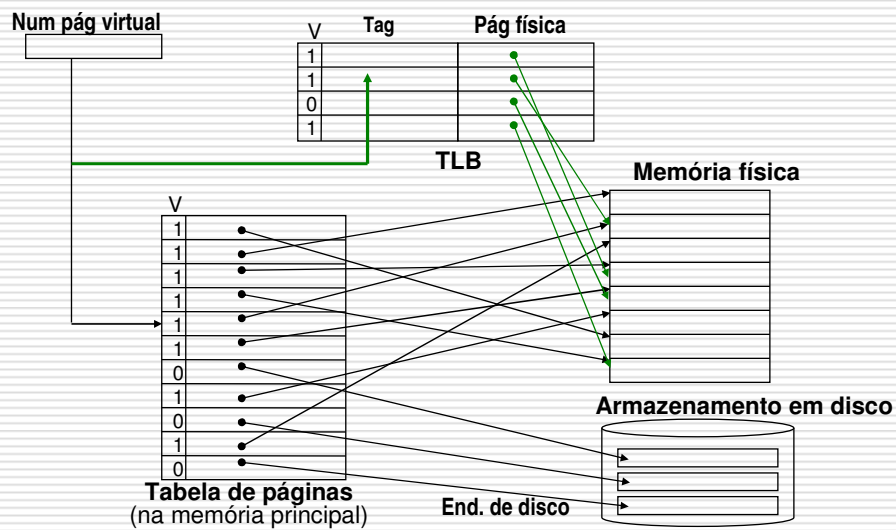
19

Translation Lookaside Buffer



20

Translation Lookaside Buffer – outra visão



21

Quem você é quando ninguém
está olhando?

22