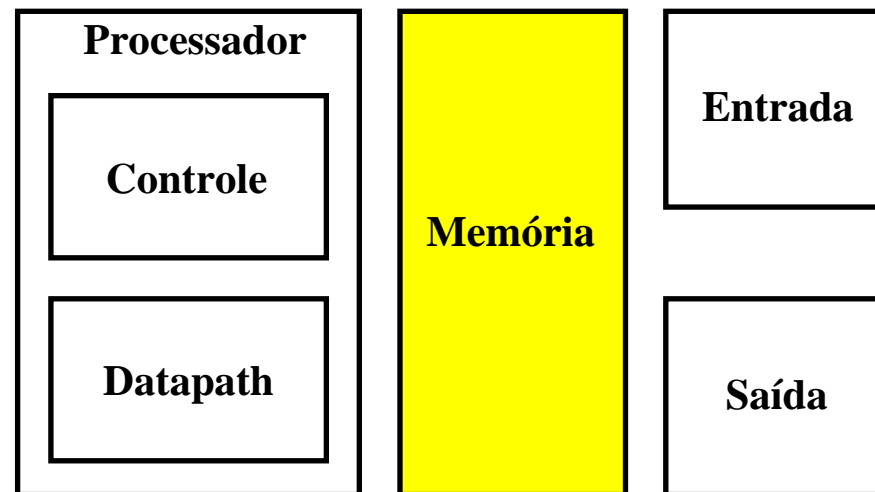


Memória virtual (um estudo introdutório)



A noção de memória virtual

MP como “cache” do HD

- **Execução simultânea**
 - Múltiplos programas
 - » Memória requerida > capacidade da MP
- **Requisito: compartilhamento de memória**
 - Eficiente: degradação aceitável do desempenho
 - Seguro: garantia de não-interferência

Idéia-chave

**“Nem todas as porções de cada programa
estão simultaneamente ativas”**

(localidade)

- **MP deve conter só porções ativas**
 - De cada programa em execução
- **Porções inativas mantidas em HD**
 - Até tornarem-se ativas

Propriedades da memória virtual

- **Extensão do espaço de endereçamento**
 - Além dos limites da MP
 - Requer controle da transferência: MP \leftrightarrow HD
- **Relocação de programas e dados**
 - Simplifica carga de programas em MP
- **Compartilhamento de memória**
 - Requer proteção

Proteção do sistema

- **Mecanismos para garantir não-interferência de múltiplos processos ao compartilhar:**
 - Processador
 - **Memória**
 - Dispositivos de E/S
- **Não-interferência**
 - Um processo não deve ler/escrever dados de outro processo
- **Proteção de memória:**
 - Um programa só deve ler/escrever as porções da MP que lhe foram atribuídas

Proteção da memória (Solução 1)

- **Faixas de endereços exclusivas**
 - Atribuir posições de MP acessíveis apenas por um determinado programa
- **Dificuldade:**
 - Como determinar que programas?
 - » Impossível em tempo de compilação
 - » Programas em execução mudam dinamicamente

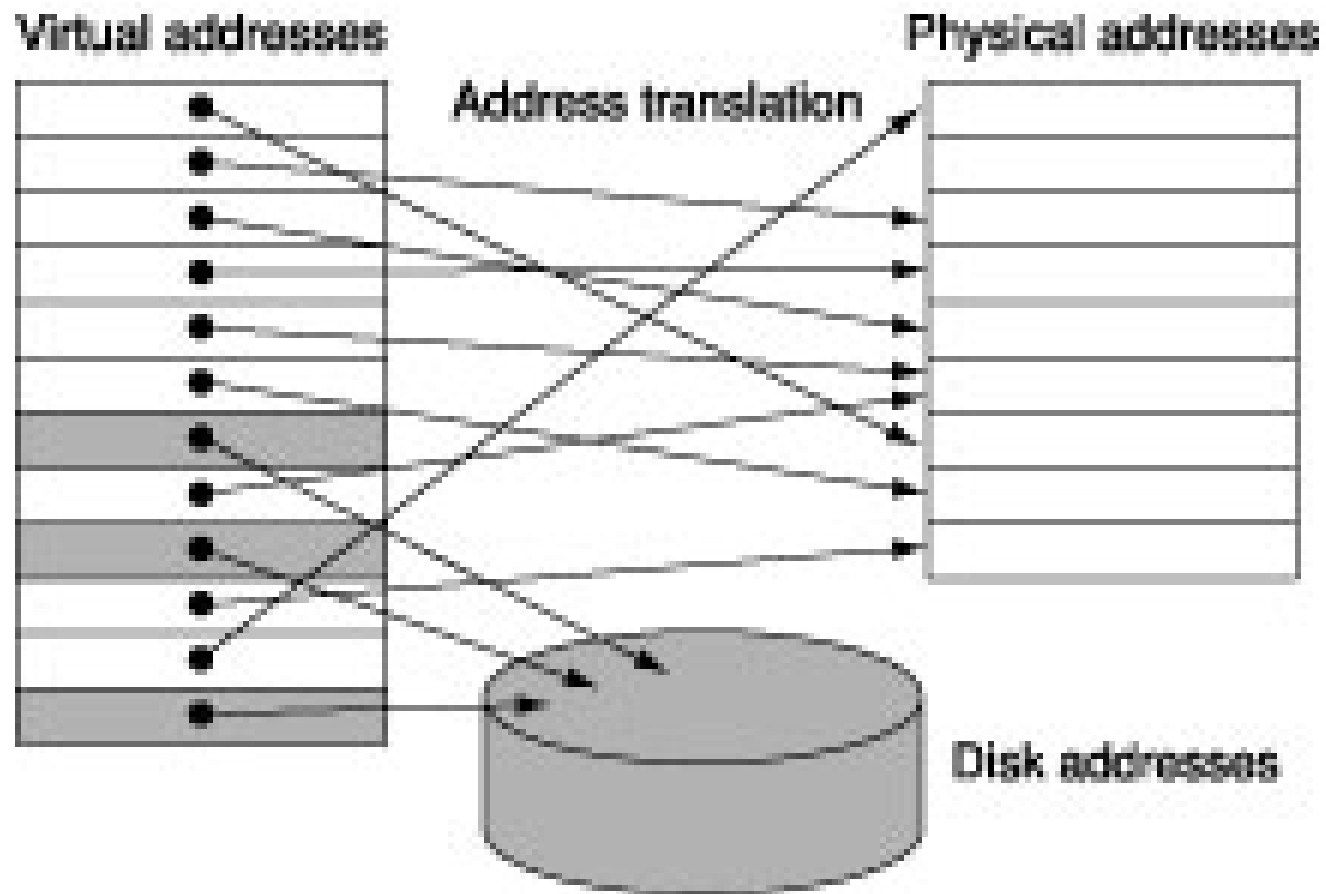
Proteção da memória (Solução 2)

- **Espaços de endereçamento separados**
 - Cada programa compilado em seu próprio
 - » Espaço virtual de endereçamento
- **Dificuldade:**
 - Como harmonizar espaços separados na MP?
- **Idéias-chave:**
 - Endereço virtual \neq endereço físico
 - Tradução ao carregar na MP
 - » Dinâmica

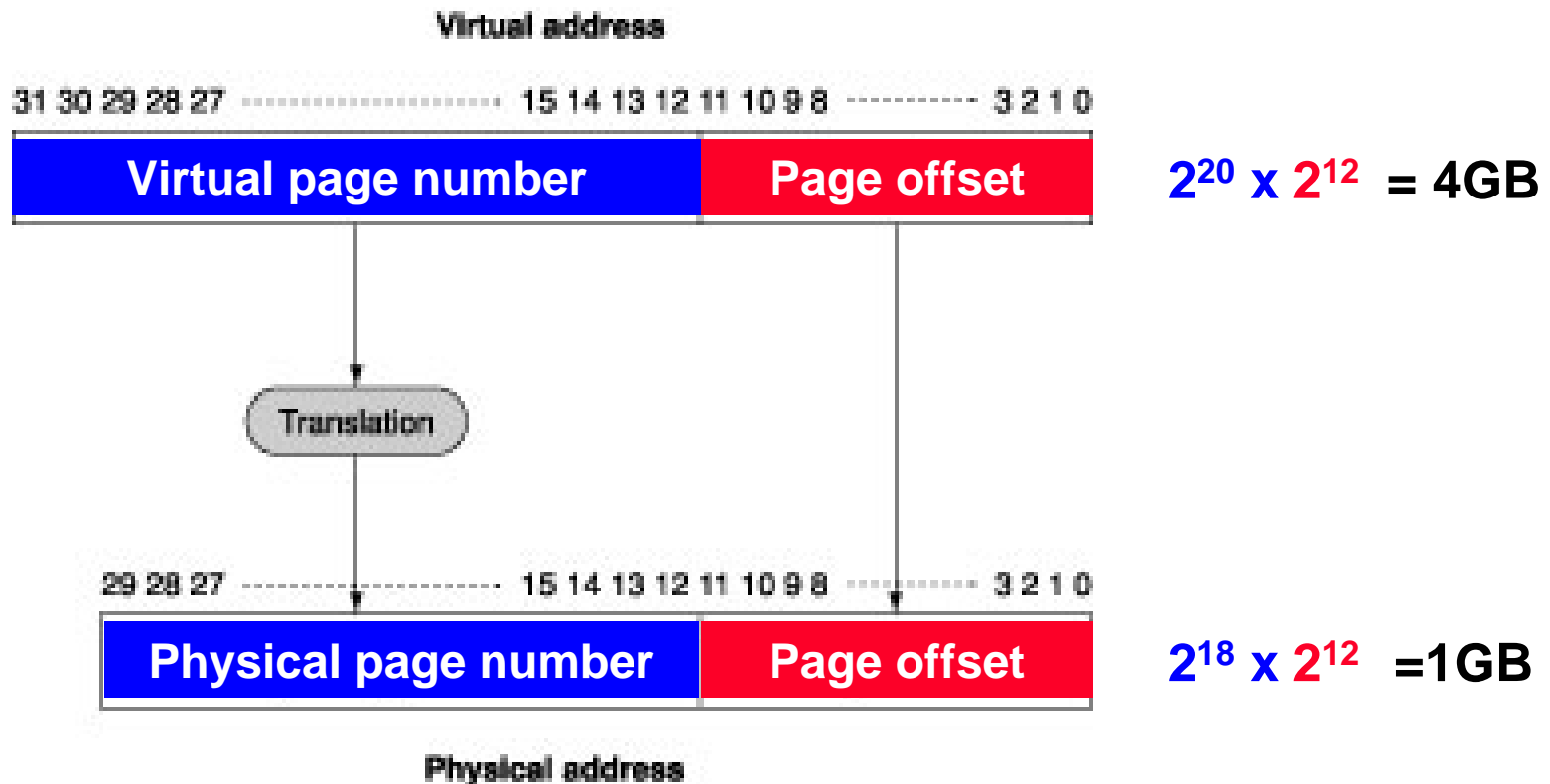
Tradução de endereços

- **Mapeamento:**
 - Endereço no espaço virtual (HD)
 - Endereço no espaço real (MP)
- **Memória virtual e física**
 - Divididas em **páginas**
 - » Blocos de tamanho fixo
- **Análoga ao mapeamento de caches**
 - » Bloco → página
 - » Miss → page fault
 - » Endereço de bloco → endereço de página
 - » Índice → endereço físico (real)

Tradução de endereços



Mapeamento



Como é implementada a tradução ?

Relocação

- **Antes de acessar uma posição de memória**
 - Ocorre mapeamento: endereço virtual → físico
- **Conseqüências:**
 - Dados e instruções automaticamente relocados
 - Podem ser carregados em qualquer lugar da MP
- **Relocação feita pelo ligador**
 - Feita no espaço de endereçamento virtual
 - » Viabiliza modularidade dos programas
- **Relocação decorrente de memória virtual**
 - Feita no espaço de endereçamento real
 - » Flexibiliza a carga de programas e dados em MP

Falta de página: características

- **Pode levar milhões de ciclos de relógio**
 - **HD $\approx 100.000\times$ MP; MP $\approx 10\text{-}100\times$ CPU**
 - » **Conseqüência de diferentes tecnologias**
- **Penalidade de falta**
 - **Dominada pelo acesso à primeira palavra**
 - » **Conseqüência de como são construídos os HDs**
- **Como dimensionar o sistema?**

Princípios de projeto

- **Páginas grandes (4KB a 16KB)**
 - Amortizar o grande tempo de acesso
- **Posicionamento totalmente associativo**
 - Reduzir a taxa de faltas
- **Falta de página resolvida em SW**
 - Degradação pequena face ao acesso ao HD
- **“Write-back”**
 - Escritas lentas demais para viabilizar “write-through”

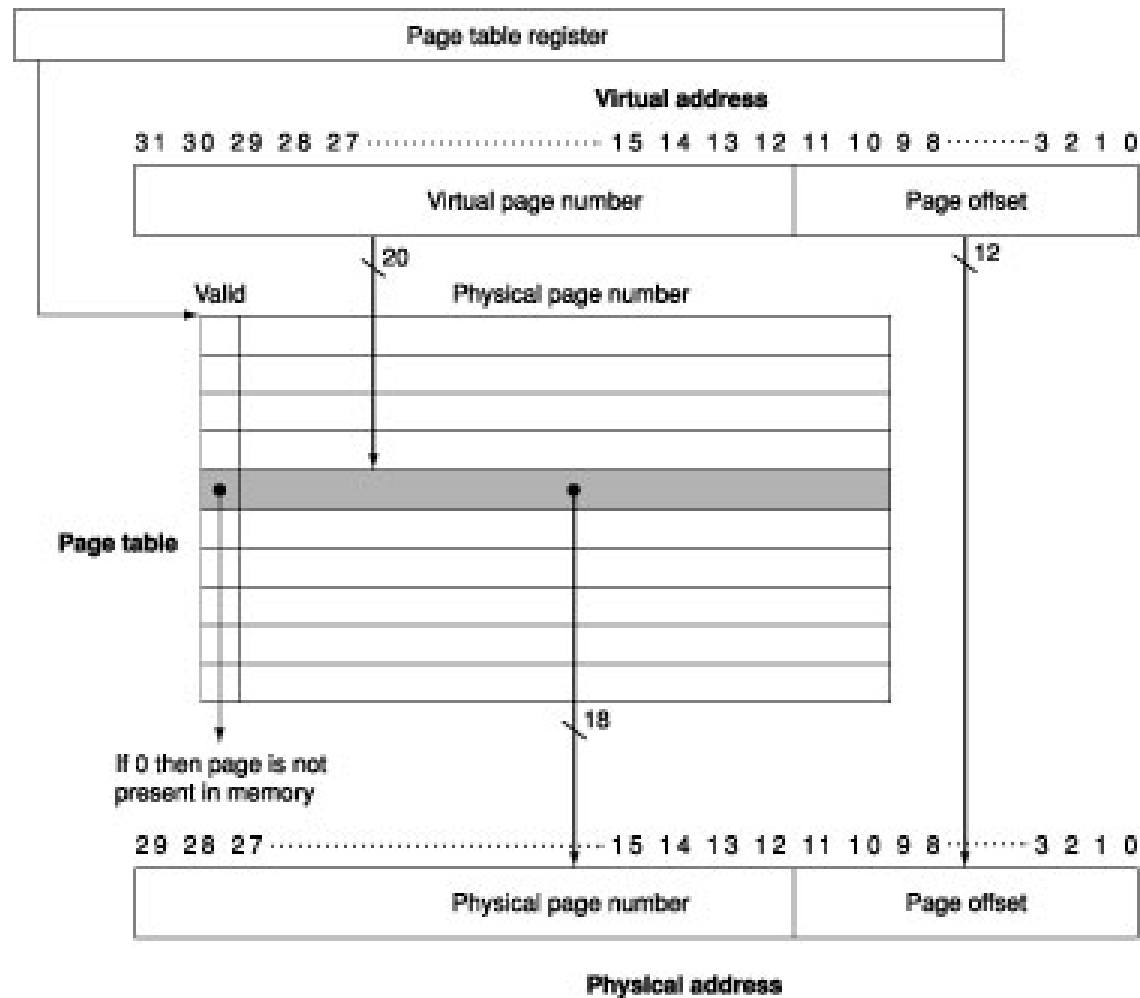
Posicionamento de páginas na MP

- **Posicionamento totalmente associativo**
 - Página virtual mapeada p/ qualquer página física
 - » Pois penalidade de falta é muito alta
- **SO usa algoritmo sofisticado**
 - Para monitorar uso de páginas
 - Tentar substituir páginas que não serão necessárias por um longo período
- **Mas busca não pode ser exaustiva**
 - Páginas são localizadas através de uma **tabela**

Tabela de páginas (TP)

- **Tradução = consulta à TP**
 - Entrada: número da página virtual (índice)
 - Saída: número da página física
- **Onde é mantida?**
 - Residente na MP
 - Cada programa tem a sua
- **Suporte de HW**
 - Registrador que aponta para TP (RTP)
 - » “Page table register”

A tabela de páginas



A interface HW/SW

- **Estado de um programa (processo)**
 - PC, registradores e TP
 - Preservado antes de permitir que outro programa use o processador
 - Restaurado para continuar execução
- **Processo**
 - Ativo: tem posse do processador
 - Inativo: em caso contrário

A interface HW/SW

- **Cada processo tem sua TP em MP**
 - **Antes de ser desativado**
 - » RTP é preservado (ao invés de toda a tabela)
 - **Antes de ser ativado**
 - » RTP é carregado com endereço de sua TP
- **SO aloca memória física**
 - **Páginas livres ou liberadas**
- **SO mantém e atualiza TPs**
 - **Toda vez que uma página é carregada em MP**
 - **Isso garante não-colisão de espaços de endereçamento**

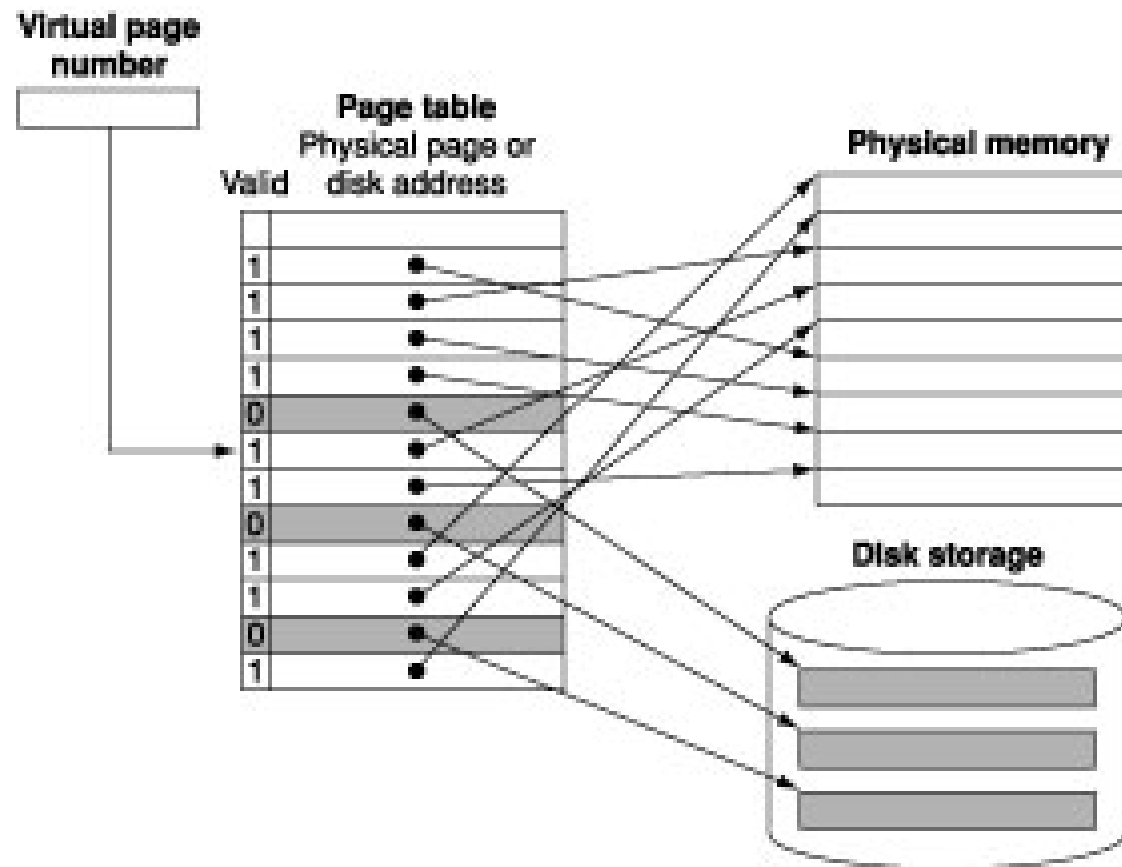
Falta de página

- Detectada quando bit de validade é nulo
- Uma exceção é disparada
 - Transferindo o controle para o SO
- SO busca a página no HD
- SO decide onde carregá-la na MP

Como achar a página no HD?

- Quando SO cria um processo...
- Cria um espaço no HD
 - Para armazenar todas as sua páginas
 - » “**swap space**”
- Cria também uma estrutura de dados
 - Para registrar onde no HD está cada página virtual
 - » Parte da TP ou estrutura auxiliar similar

Organização da memória virtual



E quando não há mais páginas livres?

- **SO escolhe uma página para substituir**
 - **Critério: LRU (aproximadamente)**
 - **Para monitorar uso:**
 - » **“Reference bit” na TP**
- **A página escolhida é escrita no swap space**
 - **Atualização: “Write-back” (“copy-back”)**
 - **Para aumentar eficiência:**
 - » **“Dirty bit” na TP**

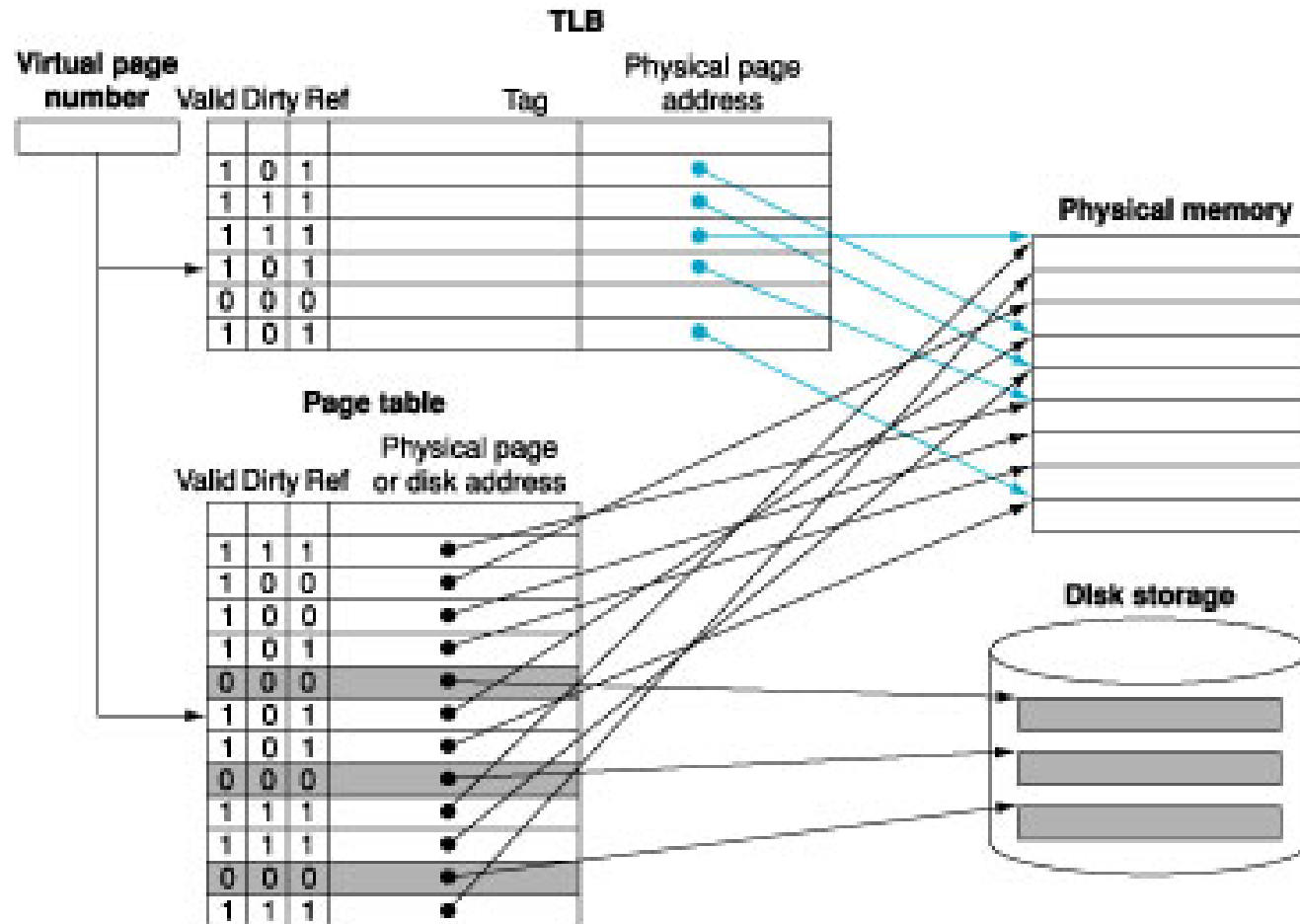
Aceleração da tradução

- **Cada acesso à memória virtual**
 - **Leva no mínimo o tempo de 2 acessos à MP**
 - » **Um para obter o endereço físico na TP**
 - » **Outro para obter o dado**
- **Capturar localidade de referência à TP**
 - **Se um número de página virtual foi traduzido**
 - **Então será provavelmente reusado**
 - » **Localidade temporal e espacial das palavras na página**

Aceleração da tradução

- **Idéia-chave:**
 - Armazenamento das traduções mais recentes
- **“Translation-lookaside buffer” (TLB)**
 - Cache especial para traduções de endereço

Organização com TLB

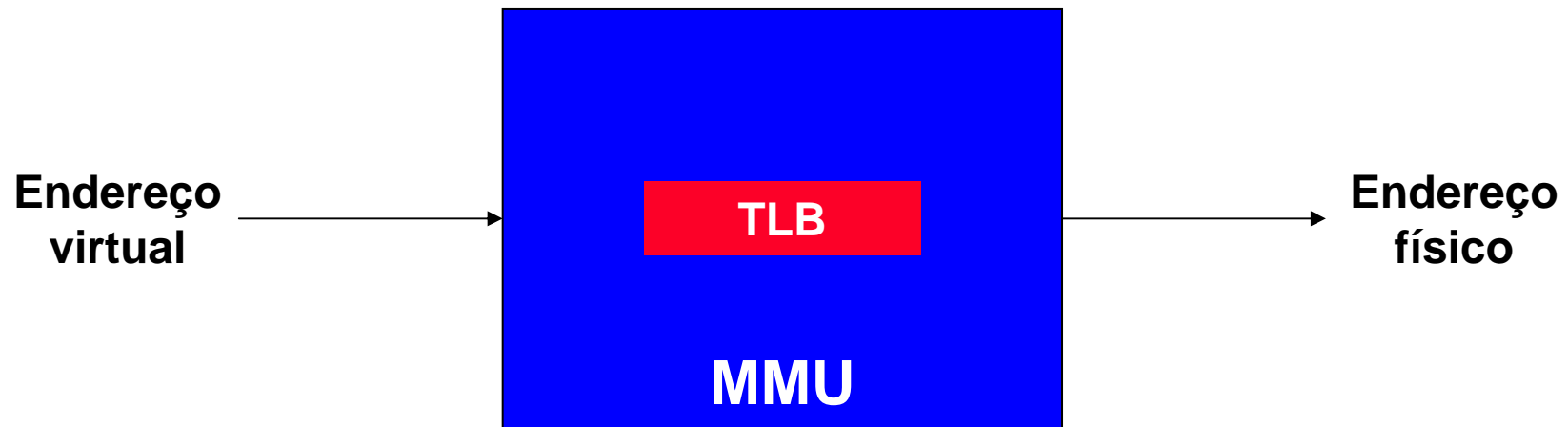


TLB

- **Associatividade da cache**
 - Totalmente associativa, se pequena
 - Associativa por conjunto, se maior
- **Valores típicos**
 - Tamanho total: 16 a 512 entradas
 - Tamanho do bloco: 1 a 2 entradas
 - » 4 a 8 bytes cada
 - Hit time: 0,5 a 1 ciclo
 - Penalidade de falta: 10-100 ciclos
 - Taxa de faltas: 0,01% a 1%

Unidade de gerenciamento de memória

- “Memory management unit” (**MMU**)



Conclusões

- **Hierarquia de memória**
 - Memória virtual é nível mais baixo
 - » Tempo de acesso: milhões de ciclos de relógio
- **Mapeamento MP \leftrightarrow cache em HW**
 - Acesso MP/acesso cache ≈ 10 a 100
 - Mecanismo: índices e “tags”
- **Mapeamento HD \leftrightarrow MP em SW**
 - Acesso HD/acesso MP = 100.000
 - Mecanismo: atualização da TP pelo SO

Conclusões

- **Mecanismo de memória virtual permite:**
 - Compartilhamento de memória
 - Extensão do espaço de endereçamento em MP
 - Proteção
 - Relocação
- **Chaves do mecanismo**
 - Divisão da memória em páginas
 - Tradução de endereços

Conclusões

- **Tradução de endereços**
 - Via tabela de páginas
 - Acelerada com TLB
- **Estudo aprofundado de memória virtual**
 - **INE 54?? - Sistemas Operacionais I**
 - » “Memória: Alocação, Gerência e Memória Virtual”