

Infraestrutura de Hardware

Memória Virtual



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

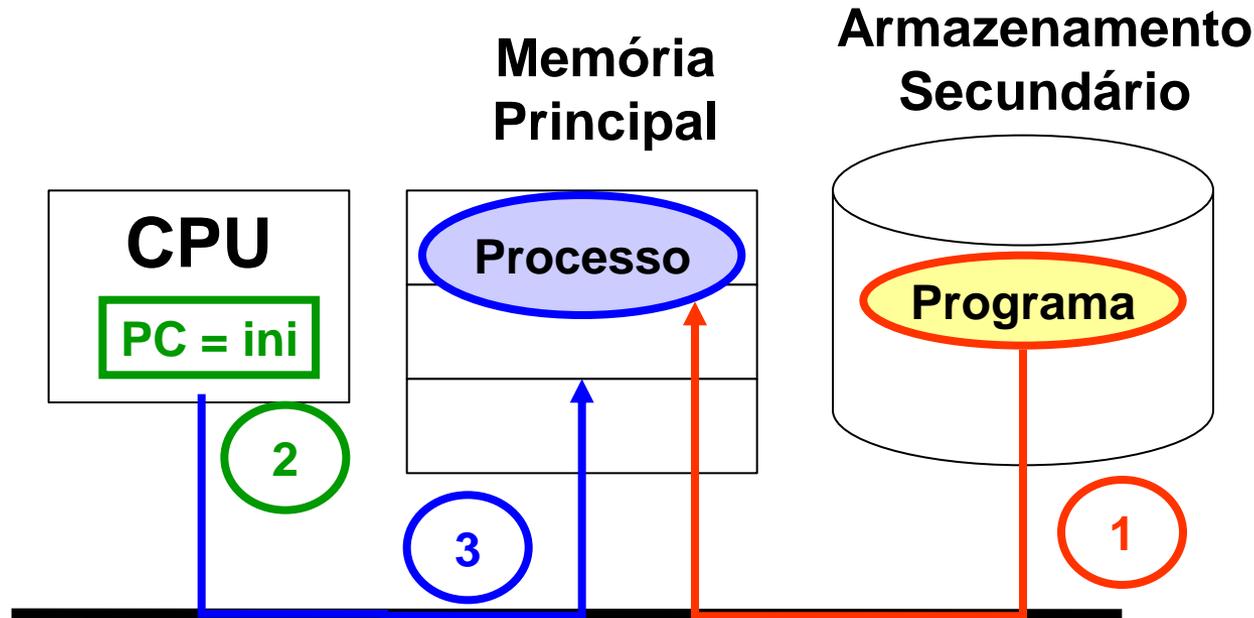
Perguntas que Devem ser Respondidas ao Final do Curso

- Como um programa escrito em uma linguagem de alto nível é entendido e executado pelo HW?
- Qual é a interface entre SW e HW e como o SW instrui o HW a executar o que foi planejado?
- O que determina o desempenho de um programa e como ele pode ser melhorado?
- **Que técnicas um projetista de HW pode utilizar para melhorar o desempenho?**

Memória

- Programas devem ser carregados na memória para poderem ser executados
- Memória é finita → não comporta todos os programas de um computador
- Hierarquia de memórias
 - Cache – rápida e cara
 - Memória principal – velocidade e custo médios
 - Disco – velocidade e custo baixo

Execução de um Programa



1. Programa vira processo e seus dados e código objeto são carregados na memória
2. PC aponta para endereço de memória do início do programa
3. CPU acessa memória para executar cada instrução e carregar dados e/ou escrever dados

Memória Virtual

- Técnica utilizada para criar ilusão de que memória é maior do que ela é
- Separação entre espaço de endereçamento visto pelo processo (e CPU) e o endereçamento real da memória

Memória Virtual (Lógica)

- Gerado pela CPU
- Pode ser maior do que memória principal, neste caso considera outros recursos de armazenamento como se fosse uma única grande memória

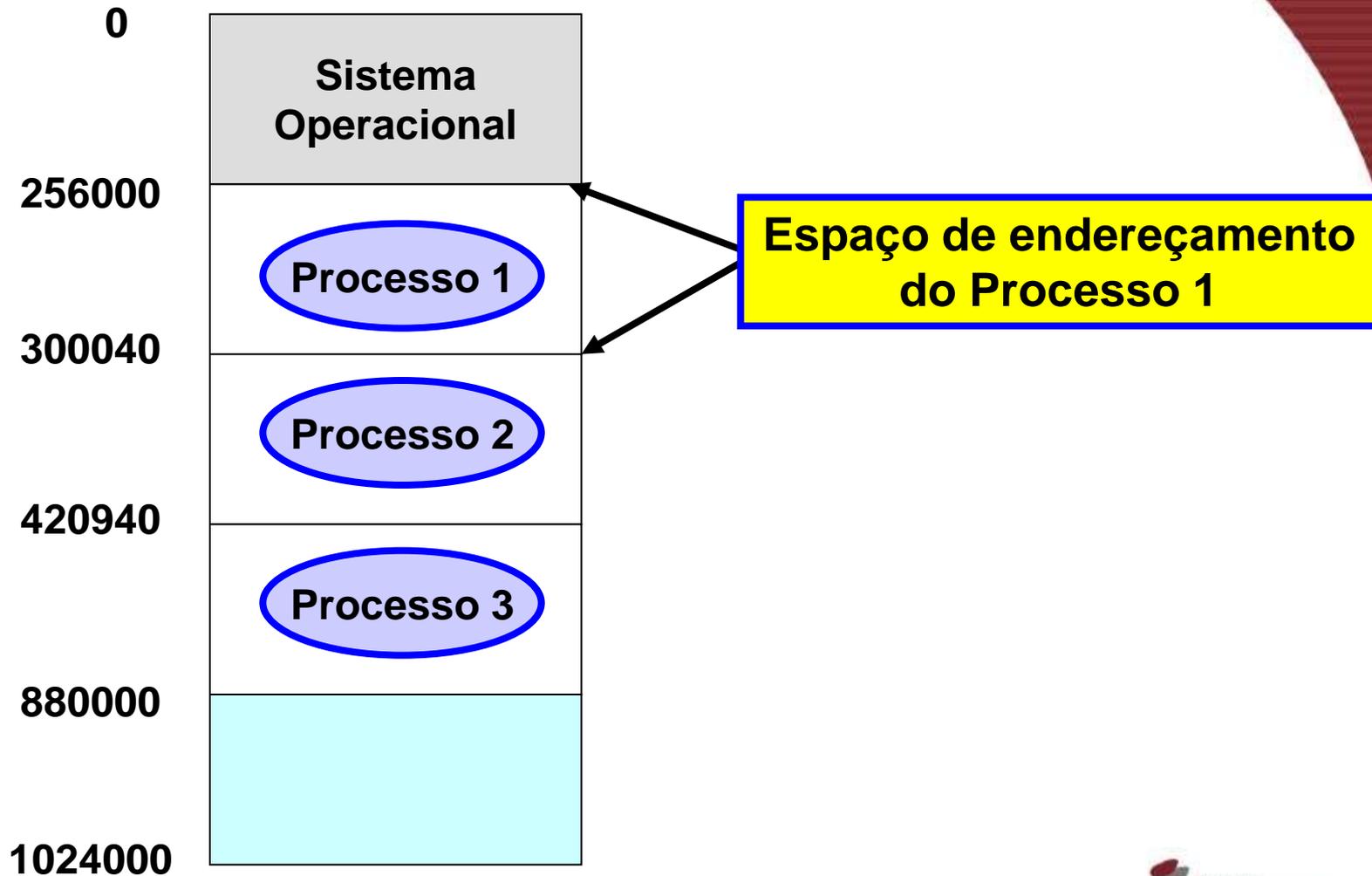
Memória Física

- Suportado pela memória principal

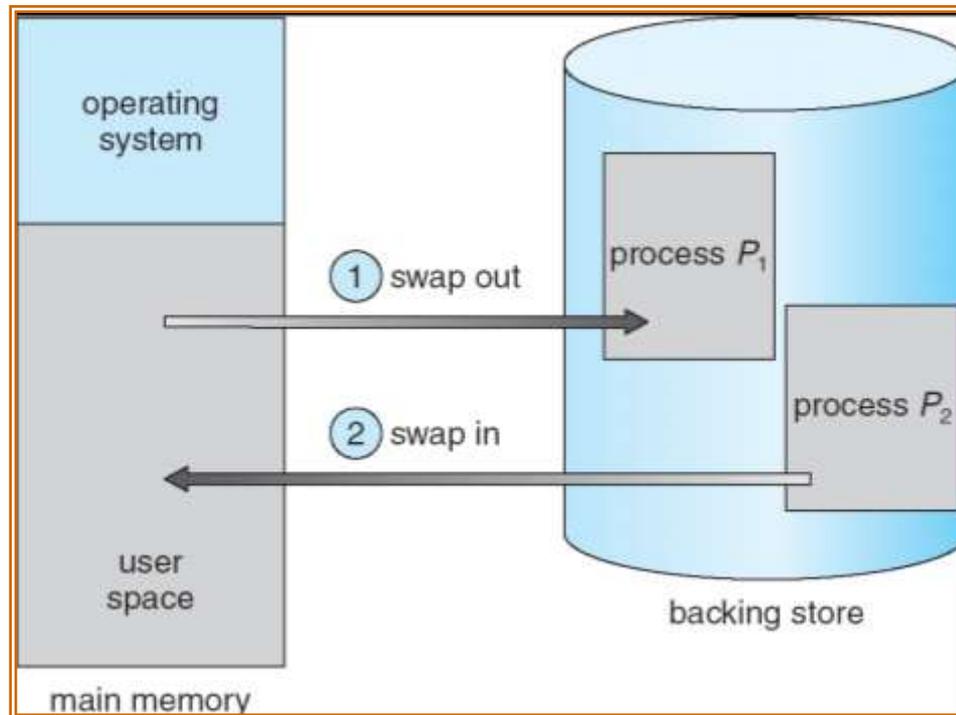
Mais Sobre Memória Virtual...

- Memória principal funciona como uma “ cache” para o disco
- Gerenciamento de memória virtual é feita pela CPU e Sistema Operacional (S.O)
 - Tradução de endereços virtuais em reais
 - Busca de dados do disco para a memória
 - Proteção
- Programas compartilham memória
 - Cada programa tem seu espaço de endereçamento
 - Protegidos de acesso por parte de outros programas

Espaço de Endereçamento



Swapping – Troca de Processos



- Processos residentes na memória podem ser movidos para disco e vice-versa
- Tempo de mudança de contexto é alto

Maior parte gasto na transferência do disco!

Partições de Memória

- Multiprogramação → vários processos na memória
- Memória pode ser dividida em partições (blocos **contíguos** de memória)
- S.O aloca para cada processo um espaço contíguo de memória (partição)
- Nem sempre espaço contíguo existe

Solução:
Paginação

Paginação

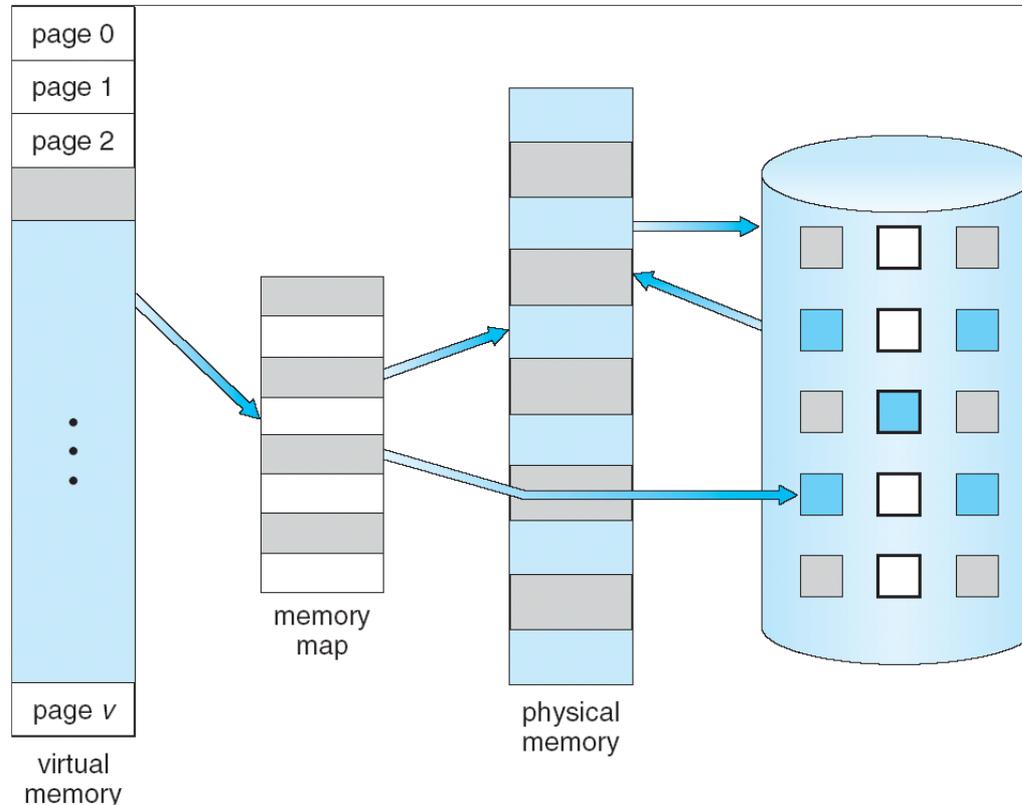
- Paginação é uma técnica de gerenciamento de memória que permite que espaço de endereçamento de processo seja não contíguo

Espaço de endereçamento é dividido em páginas

Cache: blocos, Memória Virtual: páginas

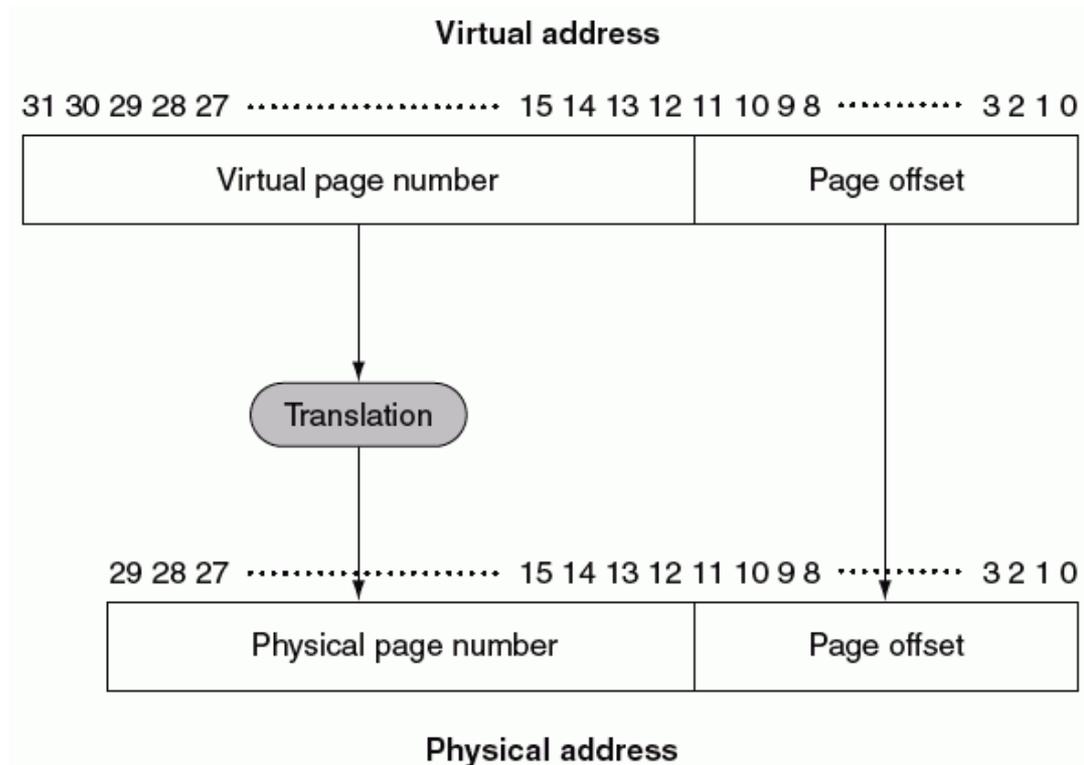
- Página pode corresponder mais de um bloco
 - Tamanho típico: 4KB – 64KB
-
- Vantagens:
 - Diferentes partes de um processo podem estar espalhados na memória
 - Algumas partes não utilizadas podem estar em disco

Paginação e Memória Virtual



- Espaço de endereçamento virtual é maior do que o oferecido pela memória física → algumas páginas podem estar no disco

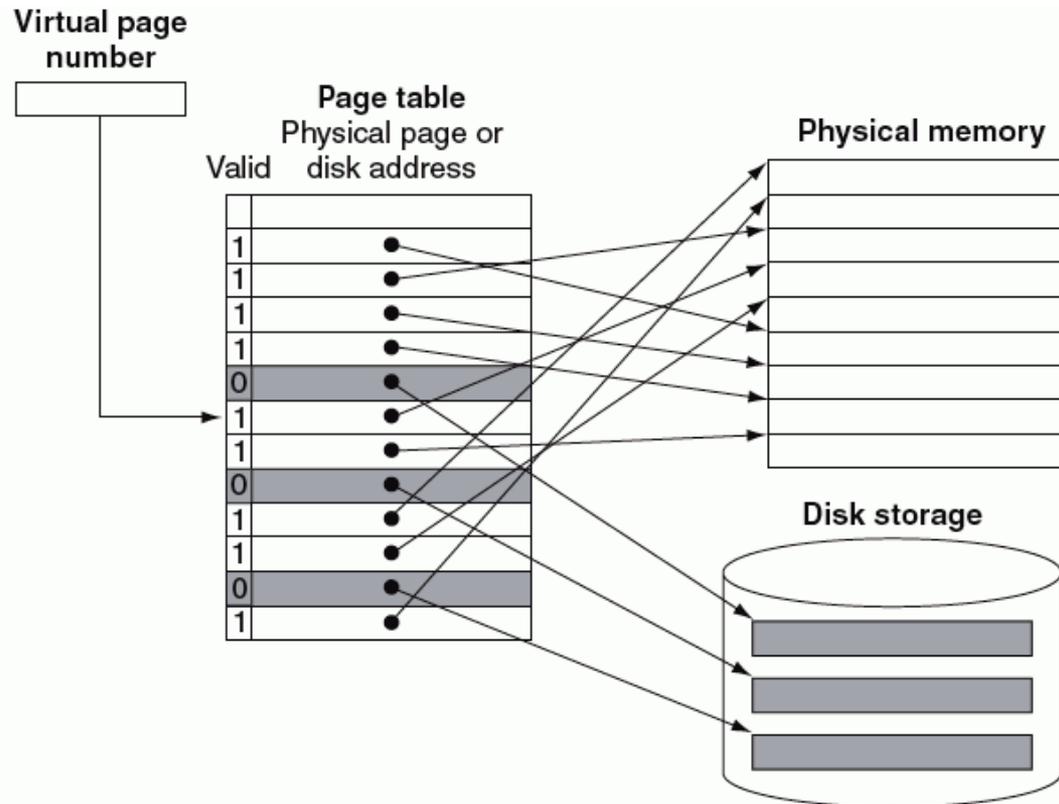
Tradução de Endereços



- Endereços virtuais precisam ser traduzidos para endereços reais

Geralmente bits mais significativos indicam número de página

Tabela de Páginas

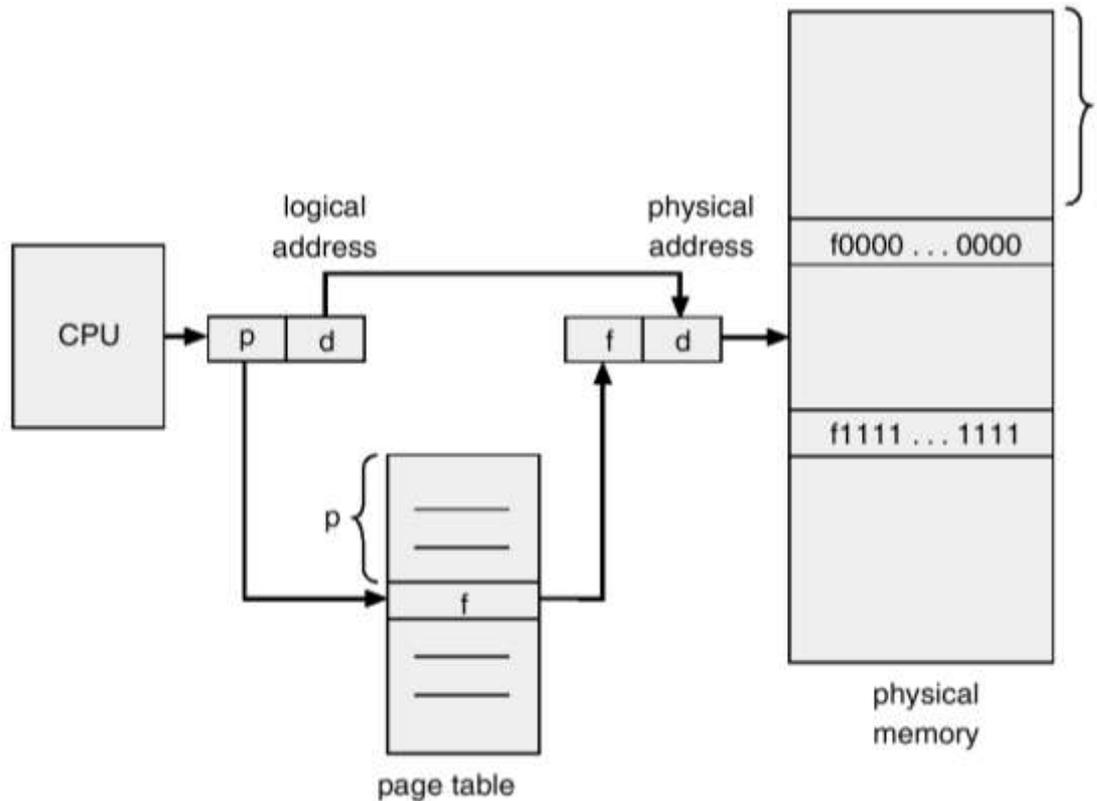


- Tabela de páginas faz a correspondência entre as páginas da memória virtual e os endereços das páginas da memória física

Implementação de Tabela de Páginas

- Tabela de páginas é armazenada na memória
- Indexada pelo número de página virtual
- Geralmente para cada processo existe uma tabela de páginas associada
- Comumente a CPU possui registradores que armazenam o endereço da tabela de **páginas**
Registrador **Context** no MIPS

Método de Tradução de Endereços Virtuais



- Se espaço de endereçamento virtual é 2^m e tamanho de página é 2^n , os $m - n$ bits de mais alta ordem de um endereço virtual dão o número da página e os n bits restantes dão o deslocamento dentro da página

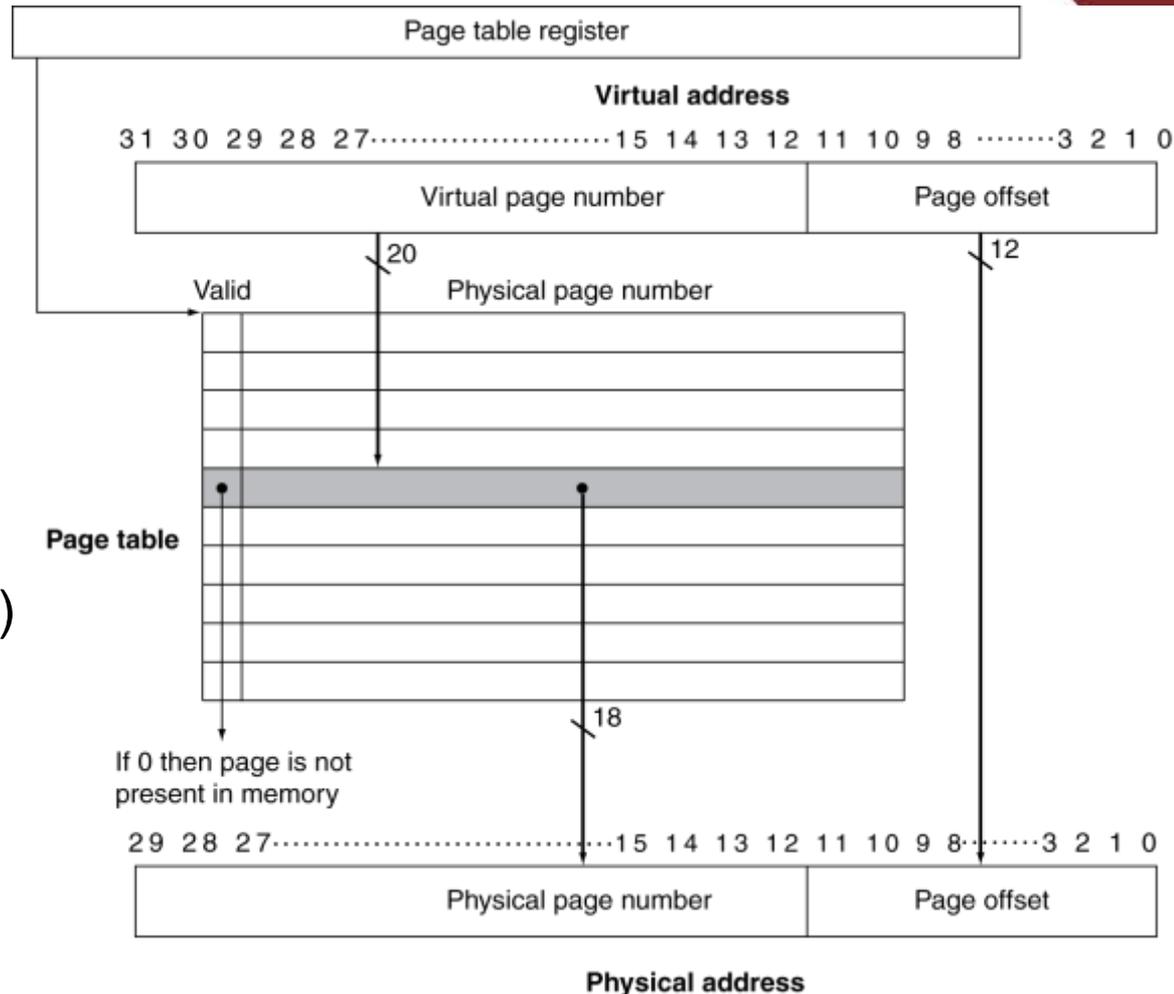
p – número da página virtual

d - deslocamento dentro da página

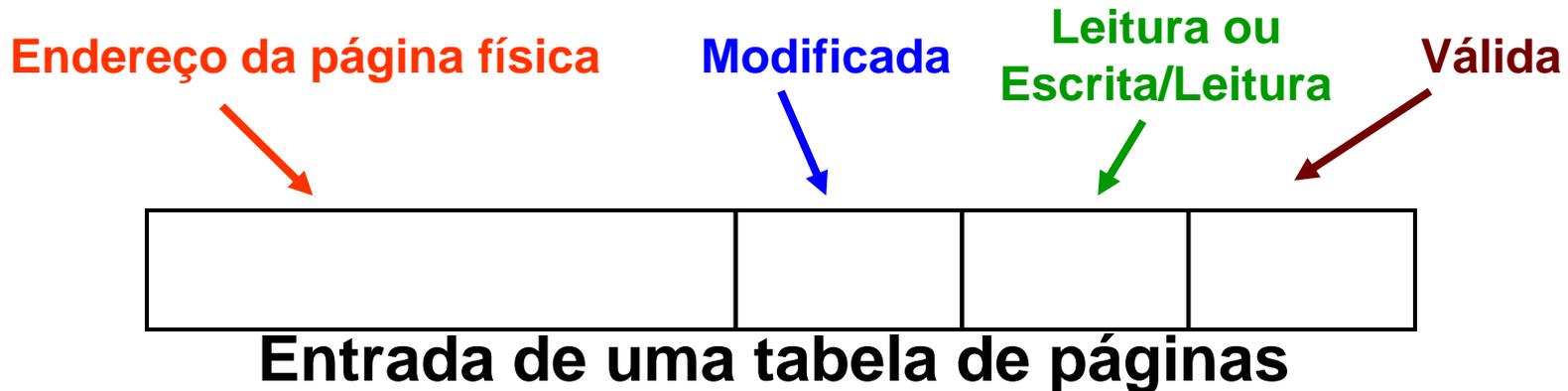
f – página física

Exemplo de Tradução de Endereços Virtuais

- Endereços de 32 bits
- Tamanho de página: 4KB (2^{12} bytes)
- Número de entradas na tabela de páginas: 2^{20}
- Espaço de endereçamento virtual: 4GB (2^{32} bytes)
- Espaço de endereçamento físico: 1GB (2^{30} bytes)



Informações de uma Tabela de Páginas



- Uma entrada em uma tabela de páginas pode conter outros dados além da localização da página:
 - Bit informando se a página é somente leitura ou leitura-escrita
 - Bit informando se a página é válida ou não (on - está na memória, off – está no disco)
 - Bit informando se a página foi modificada na memória (dirty bit)

Minimizando Uso de Memória na Paginação

- Se espaço de endereçamento virtual for grande, tabela de páginas será grande

Espaço de endereçamento virtual = 2^{32}

Tamanho de página = 4KB (2^{12})

Tabela de página tem 1 milhão ($2^{32} / 2^{12}$) entradas!

- Alocar tabela de páginas contiguamente na memória é impraticável

Solução:

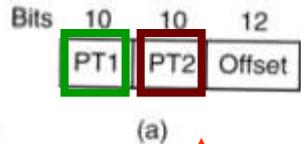
Tabelas de Páginas Multi-Níveis!

Tabelas de Páginas Multi-Níveis

- Consiste em **paginar** a tabela de páginas
- Quebra-se espaço de endereçamento virtual em múltiplas tabelas de páginas
- Existe uma tabela de tabela de páginas (a de maior nível) que contém os índices (localização) para as demais tabelas de páginas
- Tabelas de páginas são carregadas na memória de acordo com a necessidade

Tabela de Páginas de 2 Níveis

Tabela de tabelas de páginas (2^{10} entradas)



Endereço de 32 bits com 3 campos (PT1 e PT2 -10 bits e Offset – 12 bits)



PT1 indica a tabela de páginas desejada

Uma vez carregada a tabela de páginas desejada, PT2 indica qual é a página que deve ser carregada

Falta de Página (Page Fault)

- Execução de programa não implica necessariamente no carregamento de todo ele do disco para a memória
- Geralmente se carrega algumas páginas do programa e quando necessário outras páginas são trazidas
- Quando a execução do programa requisita uma página que não está na memória, dizemos que ocorreu uma **falta de página ou page fault**

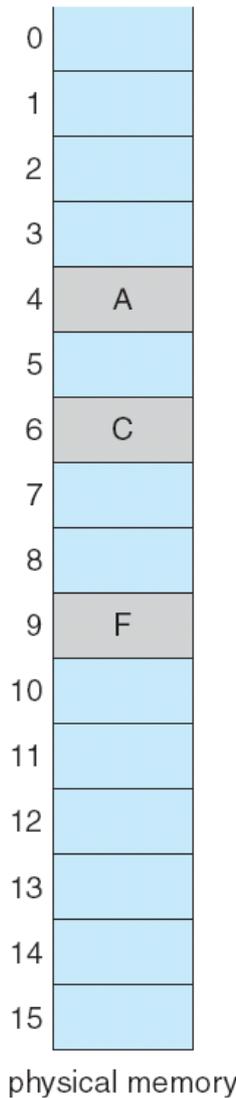
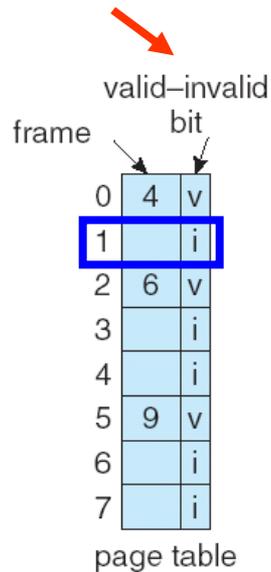
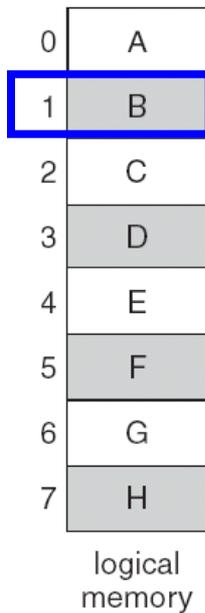
Preço de uma Falta de Página

- Preço da penalidade

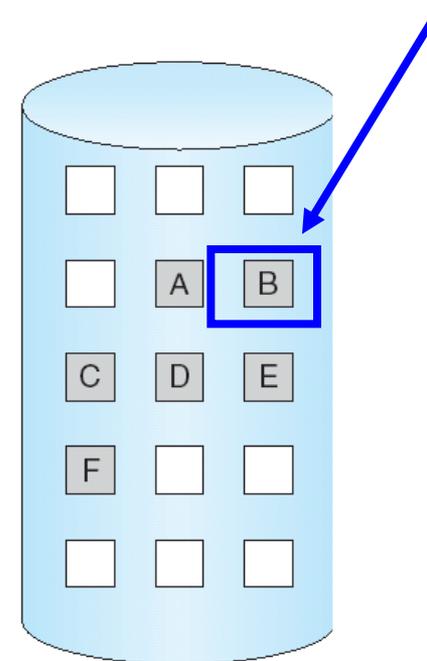
Tecnologia	Tempo de acesso
SRAM	5-25 ns
DRAM	60-120 ns
Disco	10-20 million ns

Tabela de Páginas e Falta de Página

Bit válido/inválido informa se página está na memória

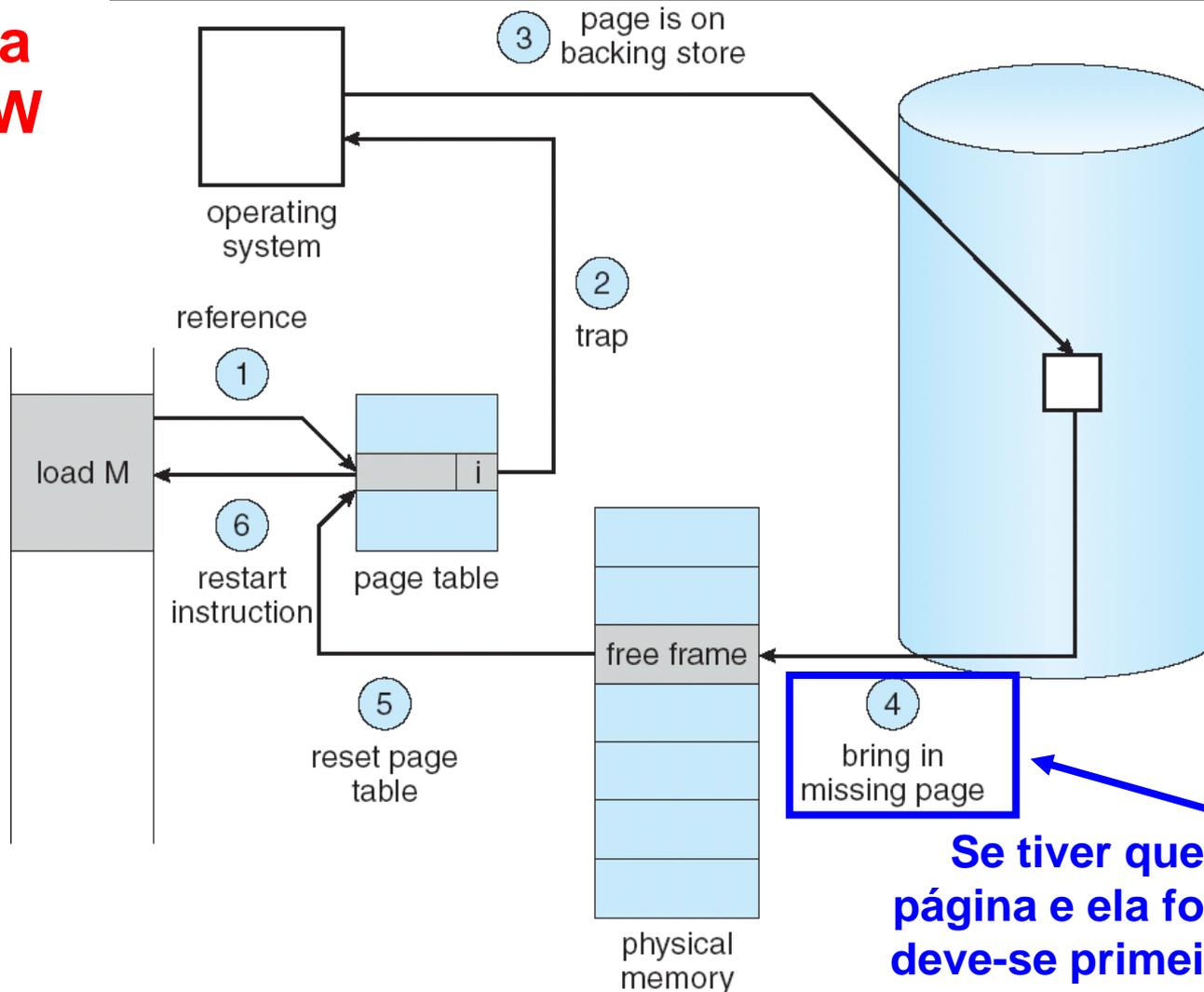


Ocorre falta de página - Página 1 está no disco



Tratando a Falta de Página

Tratada pelo SW (S.O)



Se tiver que substituir página e ela foi modificada, deve-se primeiro escrevê-la no disco

MMU e Memória Virtual

- Memory Management Unit (MMU) é um dispositivo de HW que mapeia endereços virtuais em endereços físicos

Requisição de CPU passa pela MMU

MMU decodifica endereço e utiliza tabela de páginas

Avisa falta de páginas a CPU

- MMU auxilia na proteção de processos, detecção de falhas, realocação de endereços, etc (**veremos em breve**)

Acelerando a Paginação

- Mapeamento de endereços virtuais para endereços físicos deve ser rápido
- Tabela de páginas na memória → acesso ao espaço de endereçamento do processo requer no mínimo dois acessos à memória
 - Um para acessar a tabela
 - Outro para acessar a página

Solução:

Transaction Look-aside Buffer (TLB)!

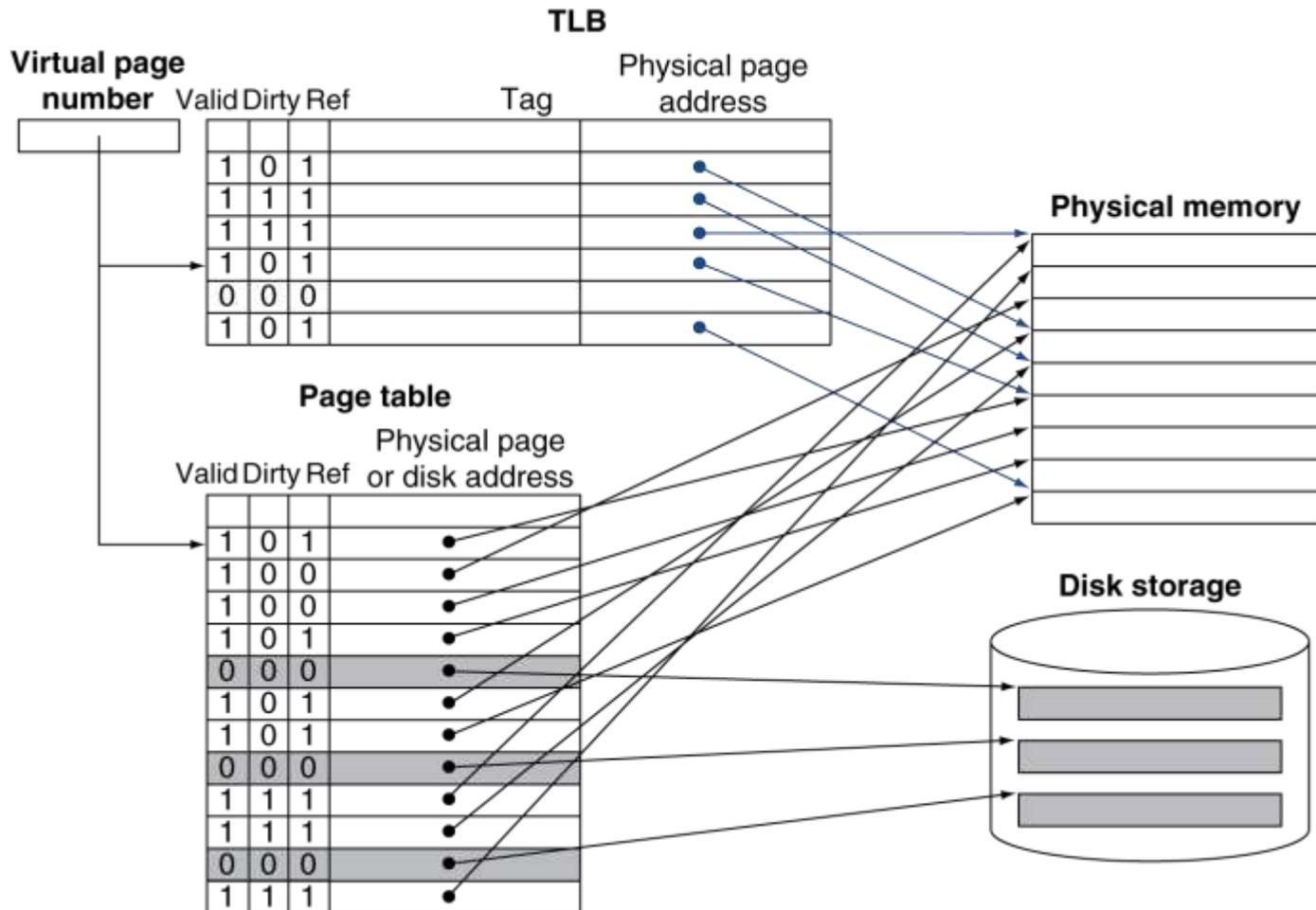
Transaction Look-aside Buffer (TLB)

- Memória (cache) associativa pequena e rápida
Geralmente localizada na MMU
Política de substituição de entradas: Randômica
- Contém algumas entradas da tabela de páginas
Entradas da tabela usadas mais recentemente
 - Aproveita-se de localidade temporalFunciona como uma cache
- CPU gera um n° de página virtual, que é comparado simultaneamente com todas as entradas da TLB

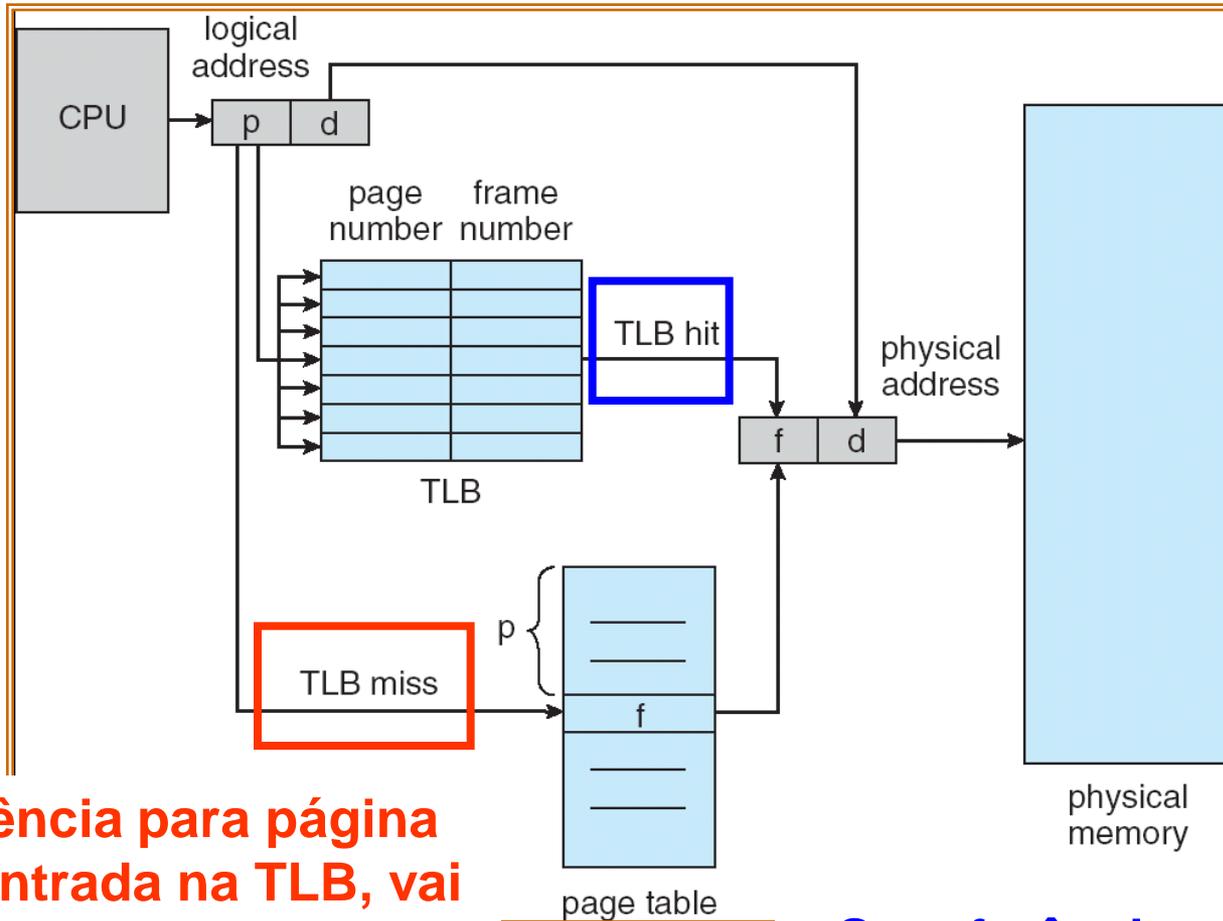
Valores Típicos de TLBs

- Tamanho : 16 – 512 entradas
- Tamanho do bloco: 1-2 entradas da tabela de páginas
Entrada da tabela de páginas: 4- 8 bytes
- Hit time : 0,5 – 1 ciclo
- Miss penalty: 10 – 100 ciclos
- Miss rate: 0,01% - 1%

Tradução Rápida Usando uma TLB



Miss (Falta) em TLB



Se referência para página não encontrada na TLB, vai para a tabela de páginas

Se referência para página encontrada na TLB, acessa página

Tratando um Miss em TLB

- **Se página estiver na memória**

Carrega a entrada da tabela de páginas da memória e tenta novamente

Pode ser feita pelo HW (MMU)

Ou em SW

- Levanta exceção, S.O pode tratá-la

- **Se página não estiver na memória (page fault)**

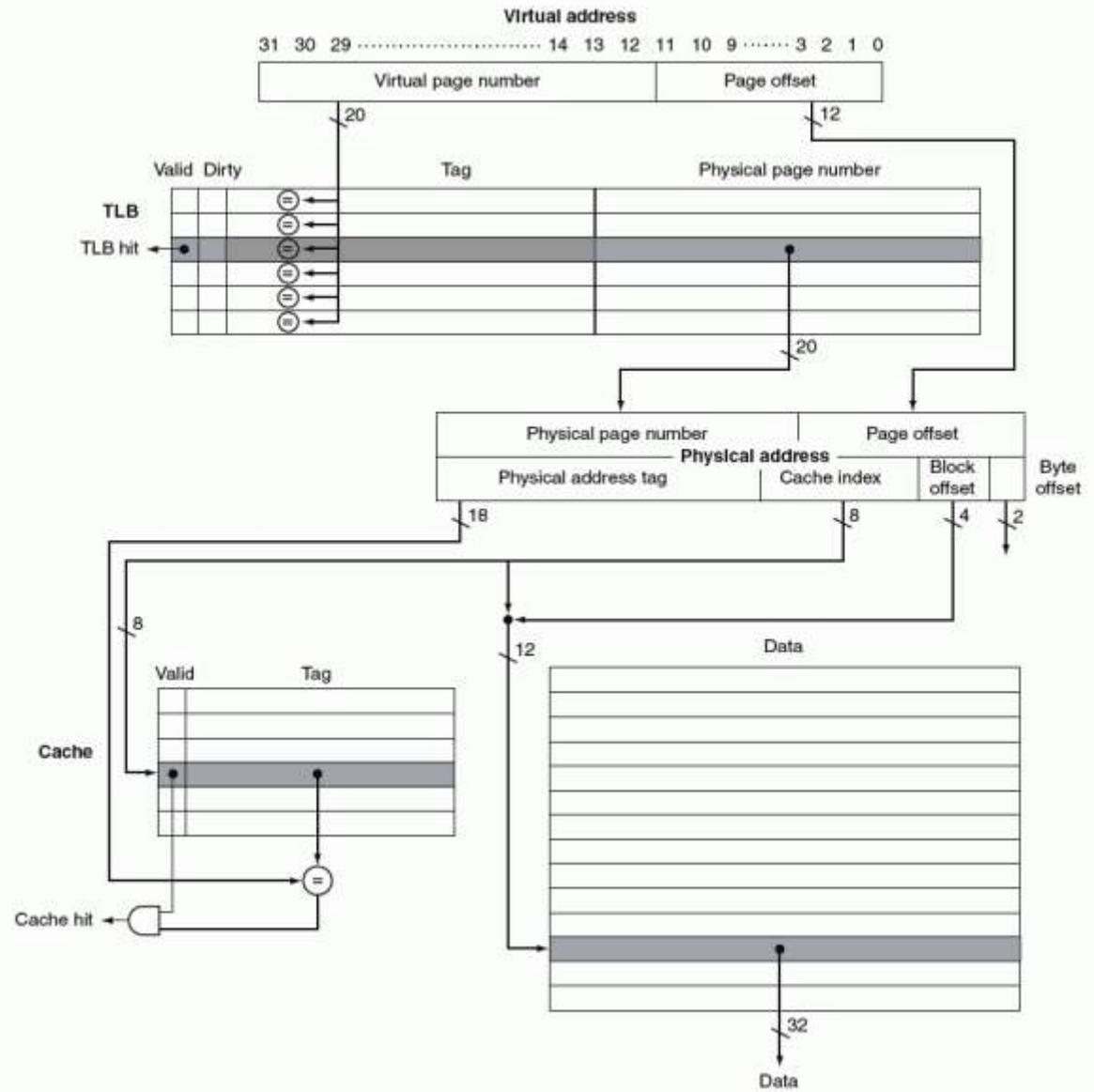
S.O trata, buscando a página e atualizando a tabela de páginas

Então reinicia a instrução que causou a falta

Interação TLB e Cache

- Se tag da cache utiliza endereço físico

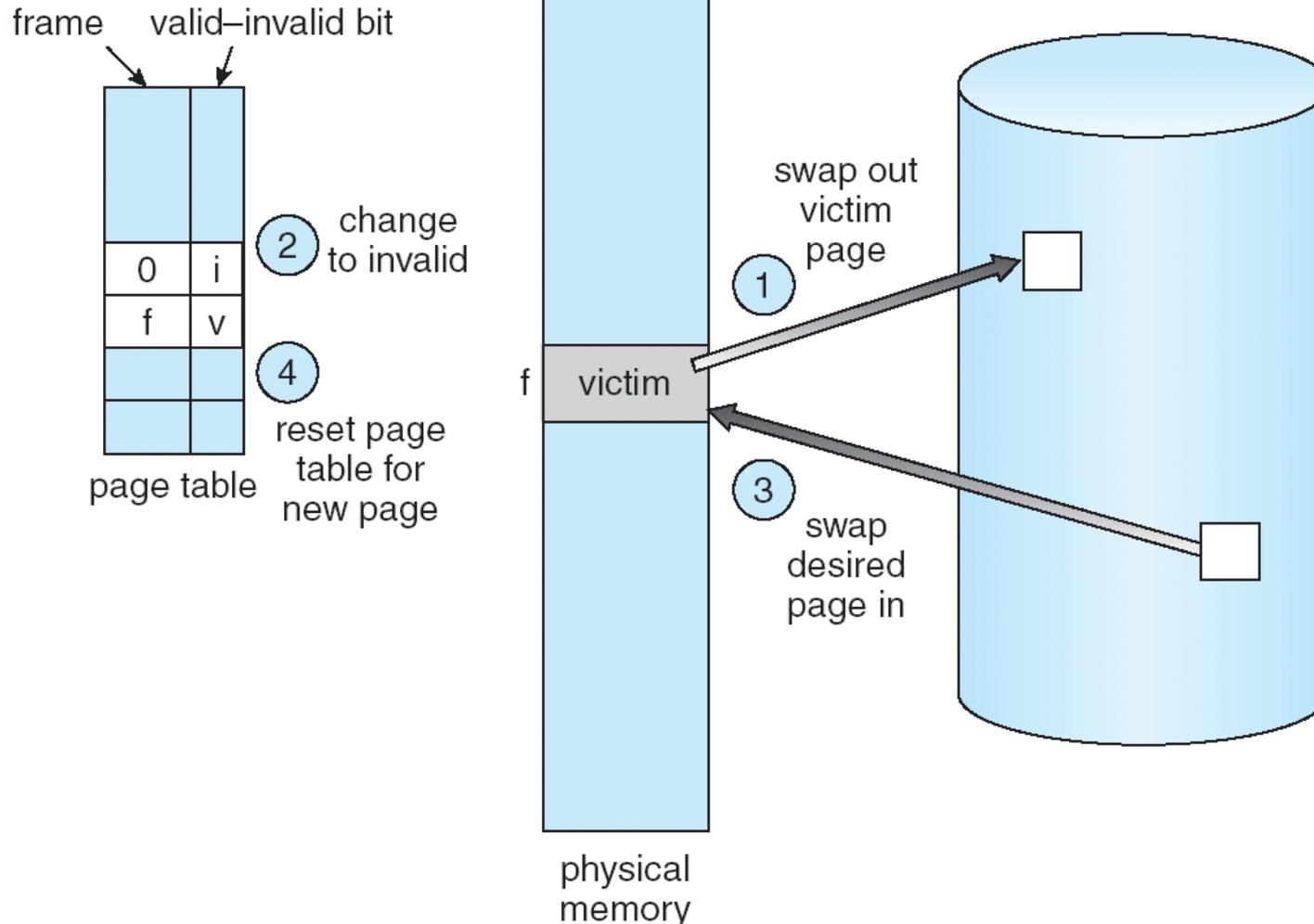
Necessidade de traduzir endereço antes de procurar na cache



Substituição de Página

- Quando ocorre falta de página e não há espaço disponível para carregar página, uma página na memória é removida e substituída por página requisitada
- Página removida, se alterada, é salva em disco
- Escolha das páginas a serem substituídas deve ser feita com cuidado pelo S.O para não provocar uma repetidas faltas de páginas
 - Grande impacto no desempenho do sistema

Esquema Geral de Substituição de Página



Algoritmos de Substituição de Página

- S.O se encarrega de substituição de páginas
- Diferentes algoritmos podem ser utilizados para a substituição de páginas:
 - First In First Out (FIFO)**
 - Least Recently Used (LRU) – Muito Utilizado**
 - Segunda Chance
- Objetivo é reduzir ao máximo o número médio de falta de páginas

Resumindo Paginação...

- **Localização da página**

Associativa

- **Política de escrita**

Write-back

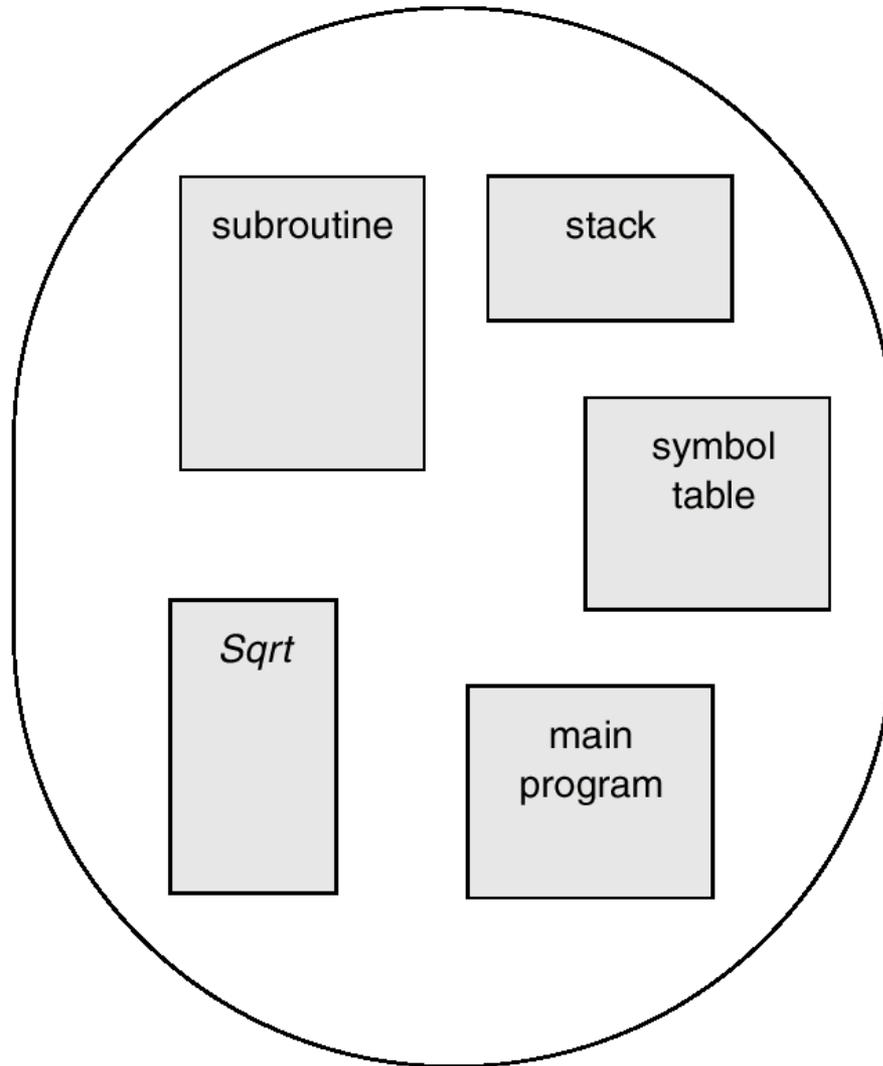
- **Falta de páginas**

tratamento por software

Segmentação

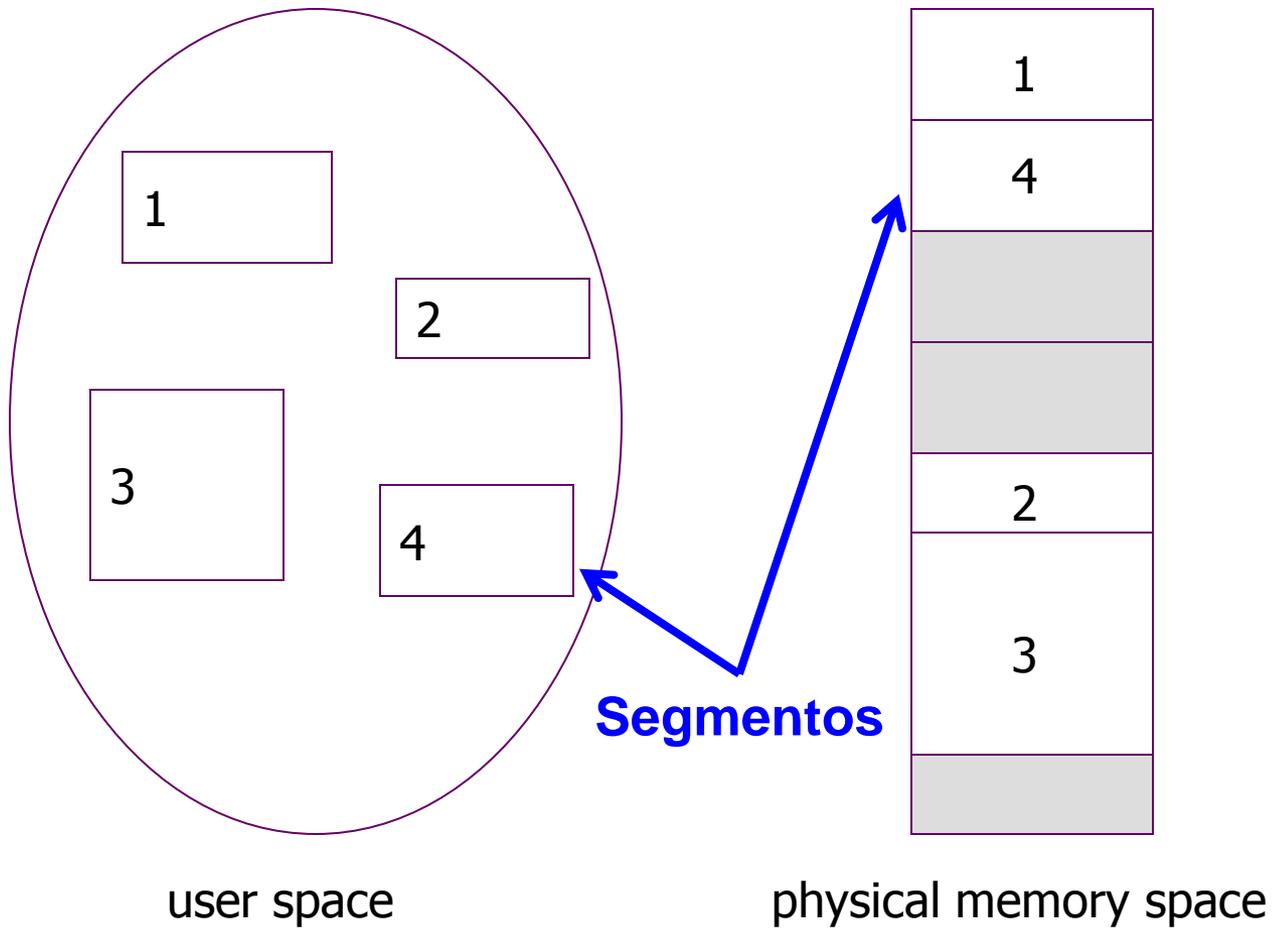
- Segmentação é uma técnica de gerenciamento de memória que permite que programa seja dividido em blocos de tamanho variável chamados de segmentos
 - Paginação → blocos de tamanho fixo (páginas)
 - Segmentação → blocos de tamanho variável (segmentos)
- Idéia é dividir programas em segmentos lógicos
 - Programa principal, funções (métodos), variáveis globais, variáveis locais, classes, etc
- Divisão é feita baseada em como o programador enxerga um programa
- Utilizado em processadores Intel Pentium

Visão do Programa pelo Usuário



logical address space

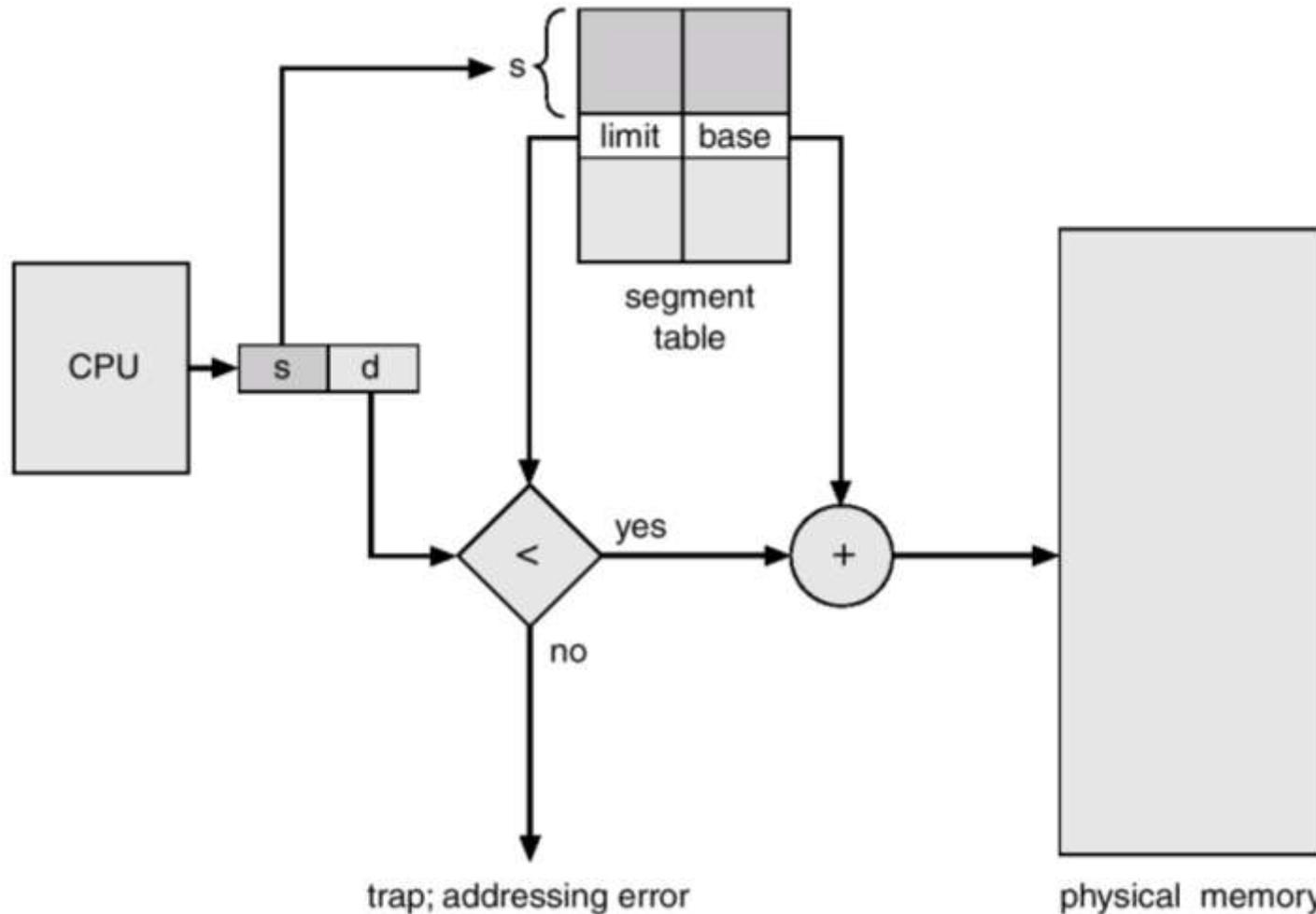
Idéia Geral de Segmentação



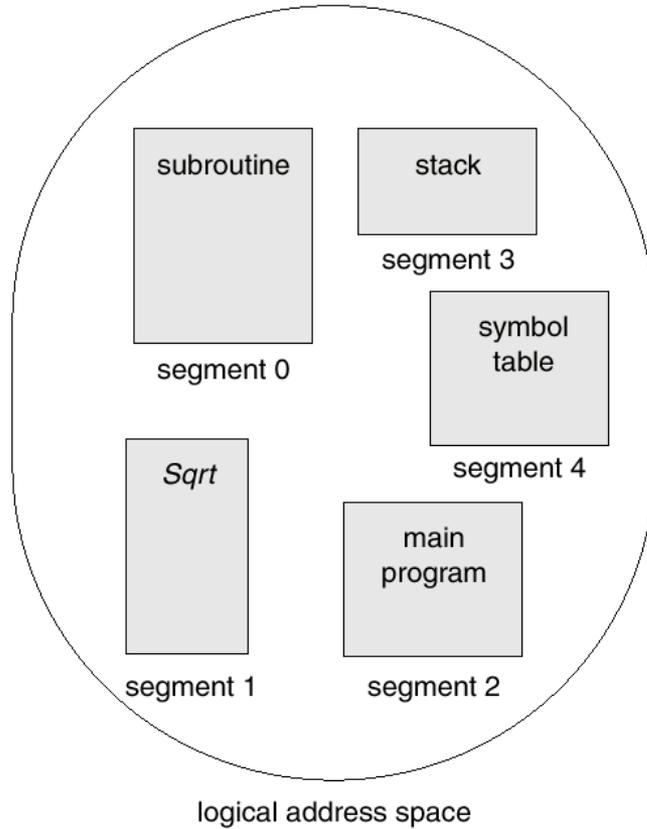
Implementando Segmentação

- Endereços virtuais consistem de um número de segmento (bits mais significativos) e deslocamento dentro do segmento (bits menos significativos)
- Tabela de segmentos mapeia endereços virtuais em físicos
 - Indexada pelo número do segmento
 - Cada entrada possui endereço físico do começo do segmento e o limite do segmento
 - Entradas possuem bits de controle
 - Leitura/escrita
 - Válido
 - Modificado (dirty)

Implementação de Segmentação

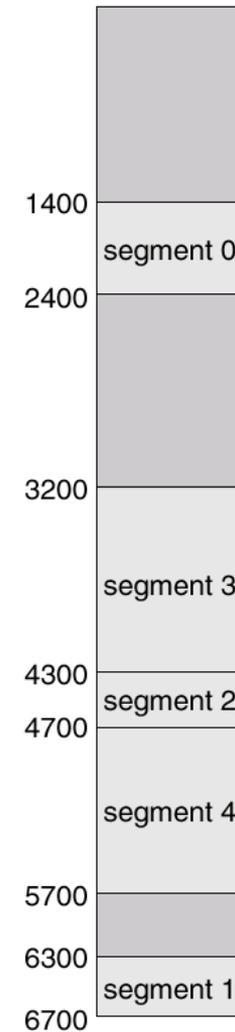


Exemplo de Segmentação



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table

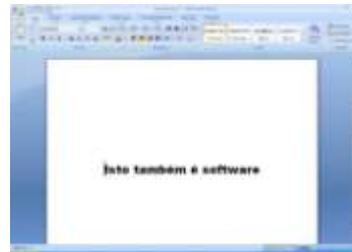


Proteção de Memória

- Processo pode ser carregado em praticamente qualquer endereço de memória
- Diferentes processos podem ter espaços de endereçamento diferentes
- Ou diferentes processos podem compartilhar espaços de endereçamento desde que autorizados
- HW pode verificar se processo tenta acessar espaços de endereçamento inválidos

Proteção de Memória

Compartilhamento de Recursos

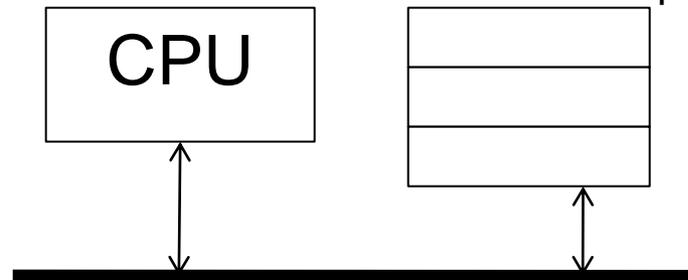


Erro

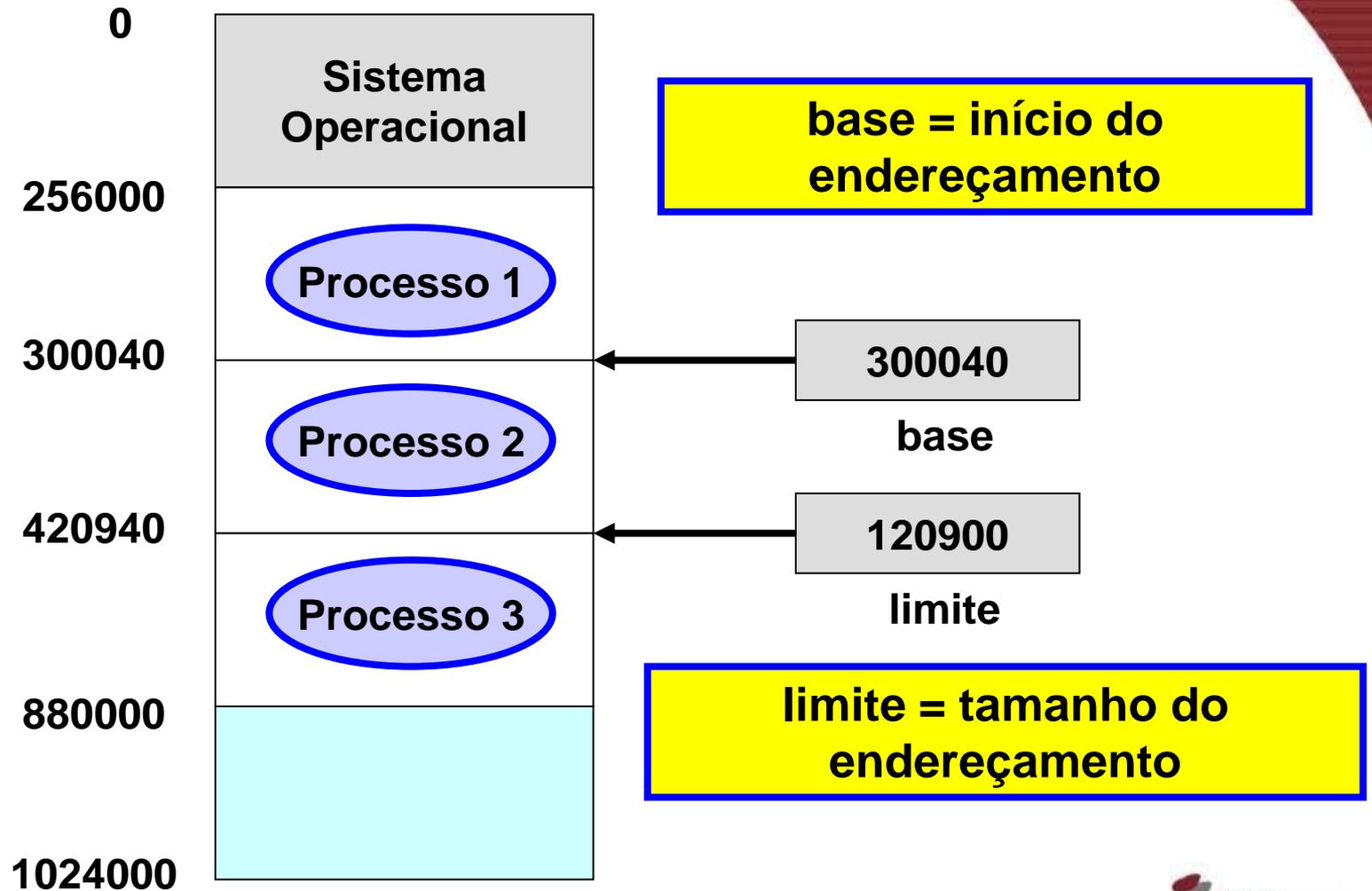
Programa defeituoso não deve interferir em outro

CPU

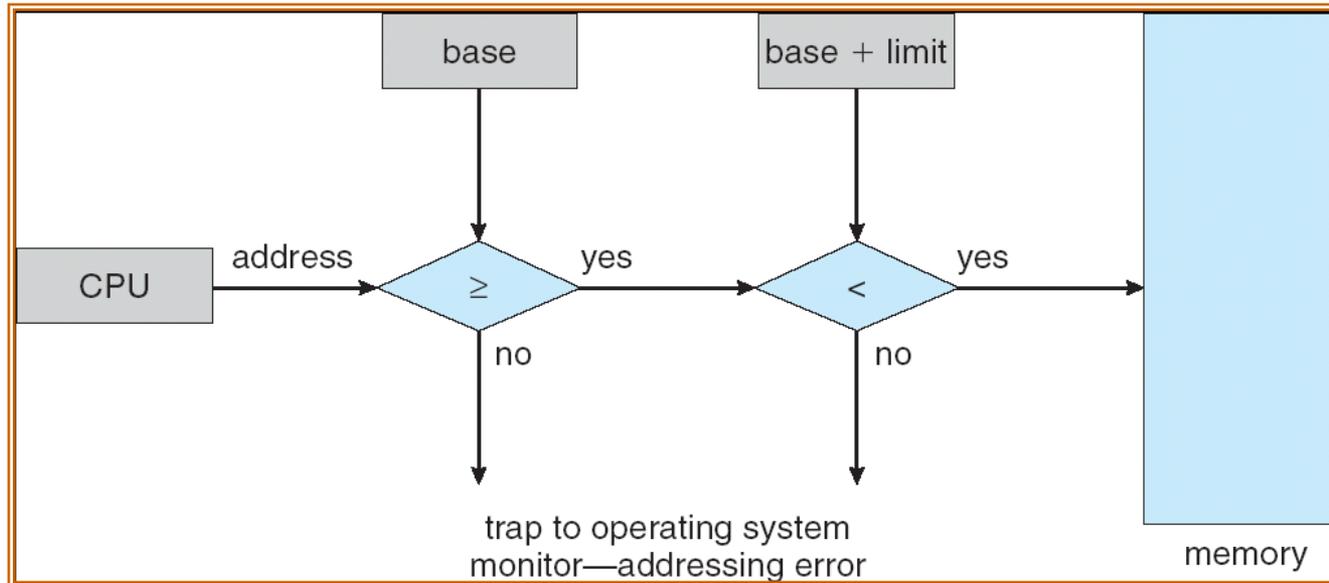
Memória Principal



Registradores de Base e Limite



Proteção com Registradores de Base e Limite



- MMU compara endereço gerado por processo com registradores
 - Se há violação, avisa ao sistema operacional**
- Somente S.O pode carregar os registradores

Sistemas Operacionais e Operação em Dual-Mode

- Compartilhamento de recursos requer um SO para garantir que um programa defeituoso não cause erros em outros programas
- Computadores atuais dão suporte em hardware para executar instruções em dois modos de operação

User mode – execução de instrução relativa a um aplicativo do usuário

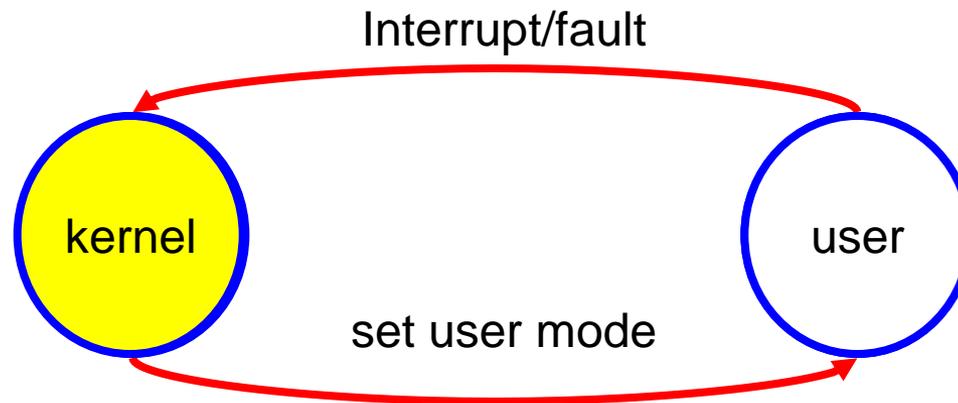
Kernel mode (ou system mode) – execução de uma instrução vinda de um SO

Sistemas Operacionais e Operação em Dual-Mode

- Hardware possui um bit para indicar o modo em um determinado instante
 - (0) kernel mode
 - (1) user mode
- Instruções de aplicativos não podem ser executados quando o HW está em kernel mode
- Instruções que só podem ser executadas em kernel mode são chamadas de instruções privilegiadas (privileged instructions)
 - Ex: instruções de E/S, acesso a tabela de páginas
 - S.O usa instruções privilegiadas

Sistemas Operacionais e Operação em Dual-Mode

- Quando um erro ou uma interrupção ocorre, hardware muda para kernel mode



Sistema Hierárquico

- TLB, Memória e Cache

Cache	TLB	Tabela pag.	Possível? Como?
miss	hit	Hit	
hit	miss	Hit	
miss	miss	Hit	
miss	miss	Miss	
miss	hit	Miss	
hit	hit	Miss	
hit	miss	Miss	

Sistema Hierárquico

■ TLB, Memória e Cache

Cache	TLB	Tabela pag.	Possível? Como?
miss	hit	Hit	Sim – falta de cache
hit	miss	Hit	Sim – TLB substituída
miss	miss	Hit	Sim, TLB subst. e falta de cache
miss	miss	Miss	Sim
miss	hit	Miss	impossível
hit	hit	Miss	Impossível
hit	miss	Miss	impossível