

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 27: Memória virtual

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br



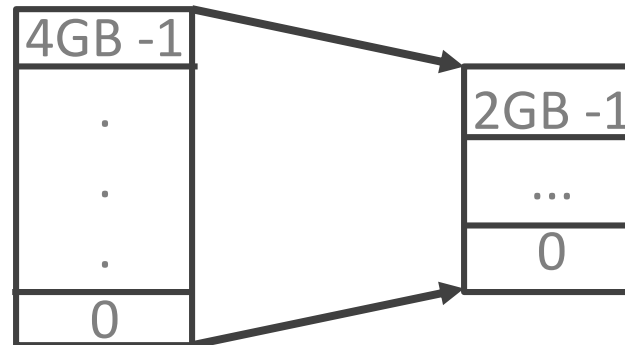
Sumário

- Princípios da memória primária
- Memória virtual
- Tabela de páginas
- TLB
- Considerações finais
- Exercícios

PRINCÍPIOS DA MEMÓRIA PRIMÁRIA

Princípios da memória primária

- Problemas a serem tratados na memória principal
 - O que fazer se eu **posso endereçar 4 GB mas tenho apenas 2 GB de memória** no meu computador?
 - Escrevo programas que usam apenas 2 GB?
 - E se eu trocar de computador?



Princípios da memória primária

- Problemas a serem tratados na memória principal
 - Como fazer para **rodar programas diferentes ao mesmo tempo?**
 - **Dados não devem se misturar/sobrescrever!**
 - Programas de **usuários diferentes?**
 - **Dividir a memória** para cada programa/usuário?



Princípios da memória primária

- Soluções
 - Como ver mais memória principal do que existe?
 - **Memória principal como cache da memória secundária!**
 - Princípio da localidade também se aplica
 - Paralelos entre cache e memória principal

Memória Cache	Memória principal
Blocos	Páginas/segmentos
Falha de cache	Falta de página

Princípios da memória primária

- Soluções
 - Como rodar programas diferentes ao mesmo tempo?
 - Cada programa enxerga endereços “virtuais” na memória
 - Endereços virtuais são mapeados para endereços reais
 - Assim como endereços de blocos são mapeados para posições na cache

MEMÓRIA VIRTUAL

Memória virtual

- Memória virtual
 - Técnica de usar a memória principal como cache da memória secundária
 - Programas enxergam endereços virtuais que são mapeados para endereços físicos
 - **Tradução e mapeamento feitos de forma automática**
 - Anteriormente cabia ao programador o controle do que trazer para a memória

Memória virtual

- Requisitos

- **Eficiência**

- Degradação de desempenho aceitável
 - Busca de dados em disco pode levar 100.000x mais tempo do que em memória

- **Segurança**

- Garantia de não interferência
 - Um processo não enxerga e não altera dados de outro processo

Memória virtual

- Propriedades
 - **Cada processo tem seu próprio espaço de endereçamento (virtual)**
 - Proteção dos espaços de endereçamento
 - Espaço de endereçamento pode ser maior do que a memória física

Memória virtual

- Propriedades

- **Localidade**

- Somente porções em uso precisam estar na memória
 - Porções inativas ficam em disco

- **Relocação de programas e dados**

- Simplifica carregar programas para execução

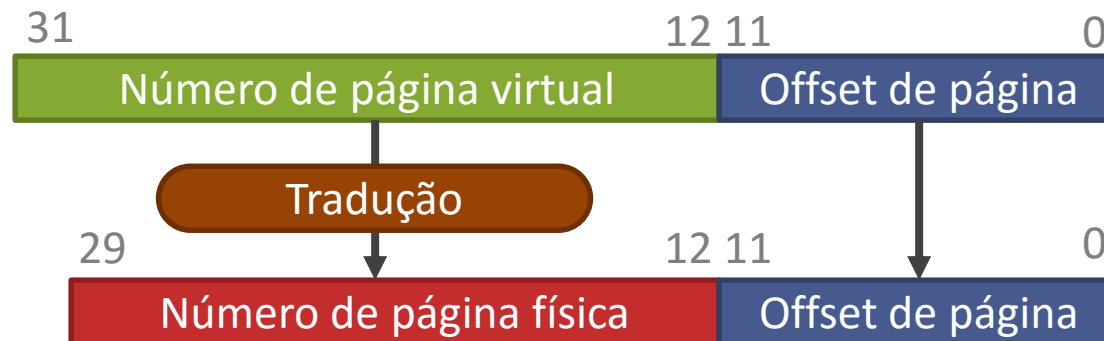
Memória virtual

- **Tradução de endereço**

- Transformação de endereço virtual para endereço físico

- Exemplo

- Páginas de 4KB (2^{12})
 - 4GB endereçáveis (2^{32})
 - 1GB de memória física (2^{30})



Memória virtual

- **Tradução de endereço**

- Transformação de endereço virtual para endereço físico

- Exemplo

- Páginas de 4KB (2^{12})

- 4GB endereçáveis (2^{32})

- 1GB de memória física (2^{30})

Virtual: 0001 0010 0100 1000 0000 1110 1101 1011

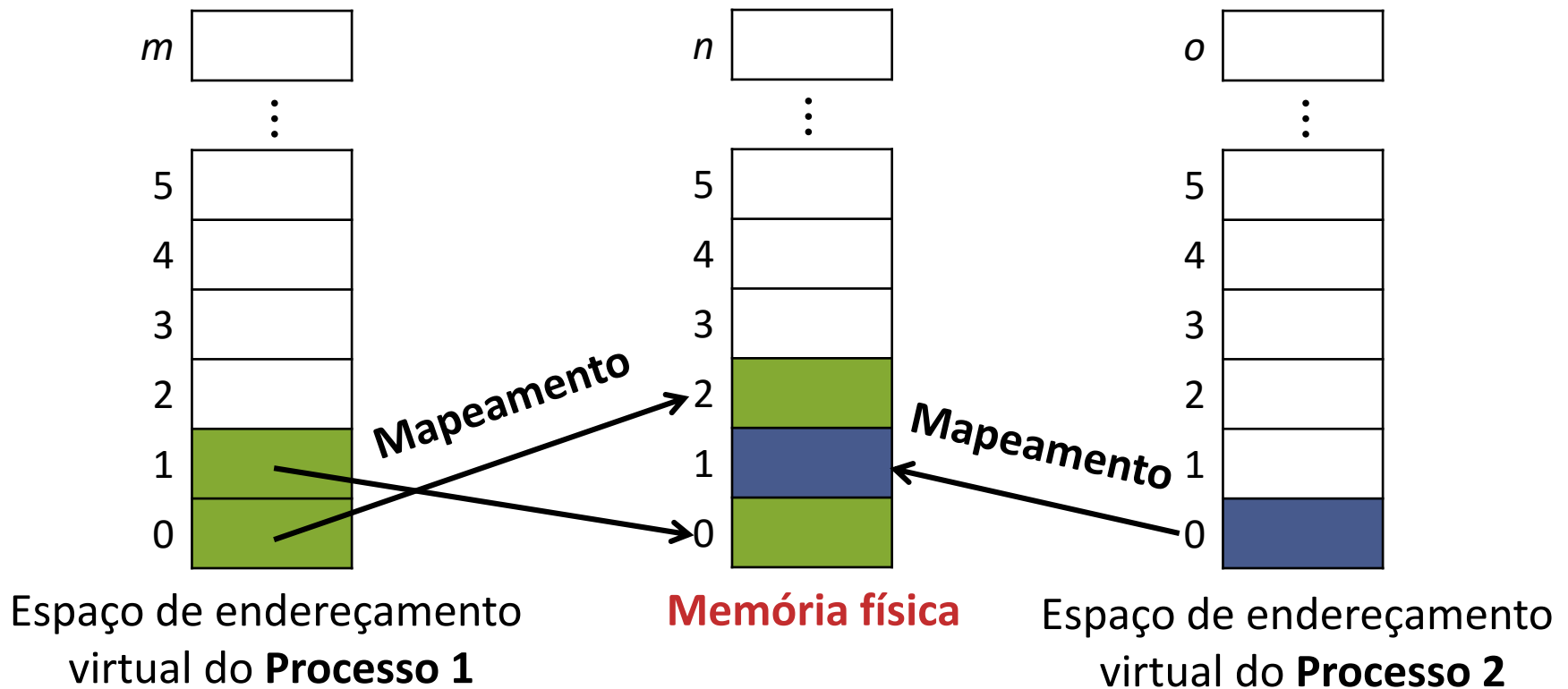
Físico: 11 0011 1111 0000 0101 1110 1101 1011

Memória virtual

- Tradução de endereço
 - Memória virtual de 2^v bytes
 - Memória física de 2^f bytes
 - Páginas de 2^p bytes
 - Bits do offset de página = p
 - Bits do número de página virtual = $v-p$
 - Bits do número de página física = $f-p$

Memória virtual

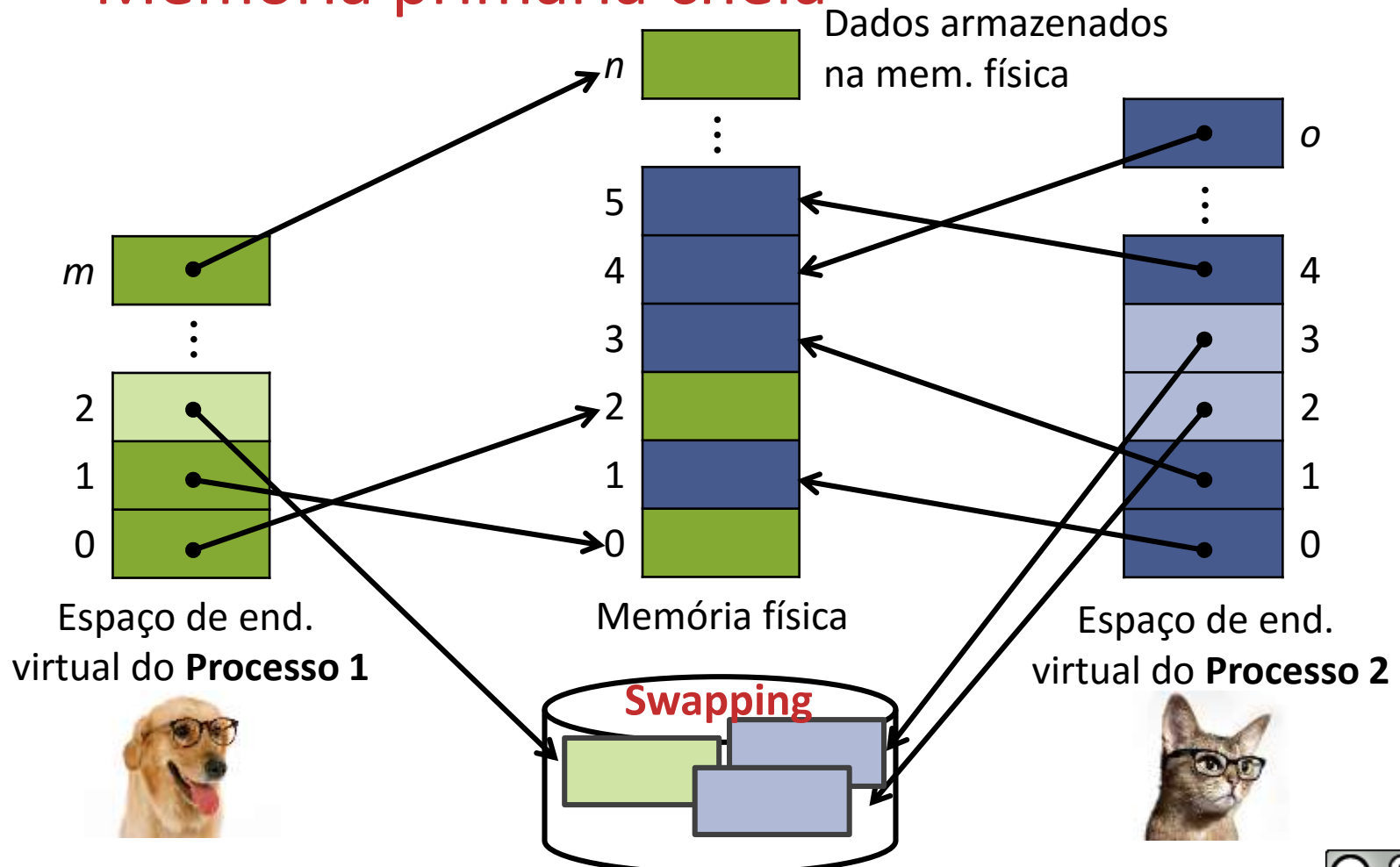
- Mapeamento virtual para físico



Memória virtual

- Mapeamento virtual para físico

- Memória primária cheia



Memória virtual

- Características
 - **Mapeamento completamente associativo**
 - Remover faltas de páginas por conflitos
 - **Mapeamento gerenciado em software**
 - Sobrecusto é pequeno comparado ao custo de ter uma página na memória secundária

Memória virtual

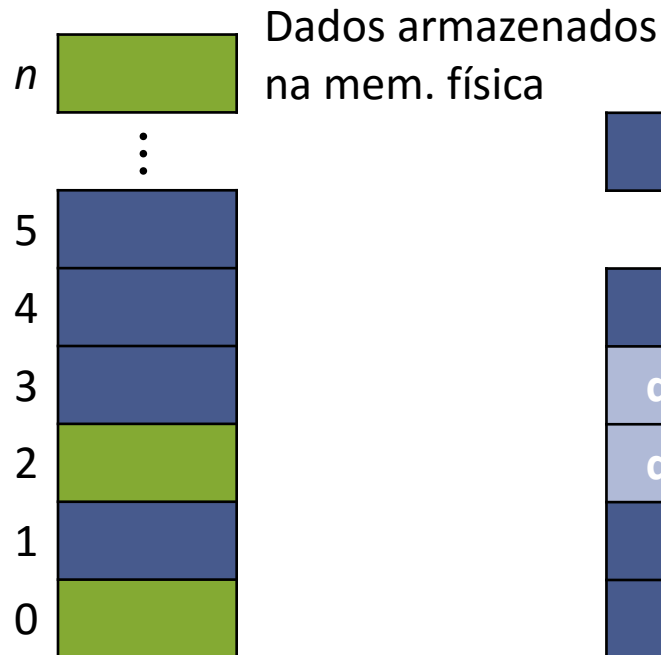
- Como saber o mapeamento de cada página?
 - Tentar bater todas as posições da memória com o endereço virtual base?
 - **Usar uma tabela de tradução**
 - Uma por processo
 - Gerenciada pelo SO

Memória virtual

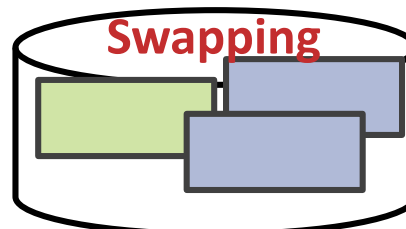
- Tabela de tradução

m	n
\vdots	
2	disco
1	0
0	2

Tabela do **Processo 1**



Memória física



4	o
\vdots	
5	4
disco	3
disco	2
3	1
1	0

Tabela do **Processo 2**



TABELA DE PÁGINAS

Tabela de páginas

- **Tabela de páginas**

- Estrutura de dados usada para a tradução de endereços
- Indexada pelo número de página virtual
- Contém o número de página física
- Uma por processo
- **Não é uma cache!**
 - Todas posições são mapeadas por ela

Tabela de páginas

- Exemplo
 - Páginas de 1KB
 - Memória virtual de 8KB
 - Memória física de 4KB

Tabela de páginas



Índice	End. físico
0	-
1	3
2	-
3	2
4	0
5	-
6	-
7	-

Endereço virtual 0 0100 1001 1000
Endereço físico?

Memória principal

Índice	Dados
0	***
1	-
2	***
3	***

Tabela de páginas

- Componentes da tabela de páginas
 - Bit de validade (1)
 - Indica se a página está na memória principal ou memória secundária
 - Bits de proteção (2)
 - Indica se página pode ser lida, escrita, ou ambos
 - Bit de modificado (1)
 - Indica se dados foram escritos na página
 - Bit de referenciado (1)
 - Indica se a página foi acessada recentemente
 - Número de página física (f-p)

Tabela de páginas

- Exemplo
 - Páginas de 1KB
 - Memória virtual de 8KB
 - Memória física de 4KB

**Tabela de
páginas**



Índice	Validade	Proteção (r/w)	Modificado	Referenciado	End. físico
0	0				-
1	1	11	1	1	3
2	0				-
3	1	10	0	1	2
4	1	11	0	0	0
5	0				-
6	0				-
7	0				-

Tabela de páginas

- Tamanho da tabela de páginas
 - 2^v endereços virtuais, 2^f físicos
 - 2^p bytes em uma página
 - b bits extras (validade, proteção, etc.)
 - 2^{v-p} linhas na tabela
 - Linhas de $b+f-p$ bits
 - Tamanho total = $2^{v-p} * (b+f-p)$ bits

Tabela de páginas

- Exemplo

- 512KB endereçáveis (2^{19}), 64KB físicos (2^{16})
- Páginas de 2KB (2^{11})
- 5 bits de controle
- N° de linhas = $2^{19-11} = 2^8 = 256$
- Tamanho das linhas = $5+16-11 = 10$
- Tamanho da tabela de página = $256 * 10 = 2560$ bits = 320 bytes

Tabela de páginas

- Processo de tradução
 - Programa tenta acessar dado em memória
 - Sistema obtém o número de página virtual a partir do endereço
 - Sistema verifica entrada na tabela de páginas
 - Se válida, retorna nº de página física
 - Se inválida, traz página do disco e retorna nº de página física
 - Sistema busca o dado na memória principal

Tabela de páginas

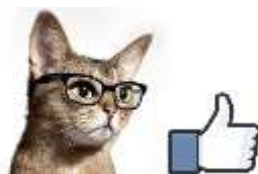
- Quem é esse sistema?
 - ***Memory Management Unit***
 - Unidade de gerenciamento de memória
 - Parte do processador
 - Desempenho!
 - Sistema responsável pela tradução de endereços
 - **Envolve o hardware e o sistema operacional**

Tabela de páginas

- Processo de tradução
 - Programa tenta acessar dado em memória
 - **MMU** obtém o número de página virtual a partir do endereço
 - **MMU** verifica entrada na tabela de páginas
 - Se válida, retorna nº de página física
 - Se inválida, traz página do disco e retorna nº de página física
 - **MMU** busca o dado na memória principal

Tabela de páginas

- Problema de desempenho
 - **Cada acesso à memória principal vira dois acessos!**
 - Um para obter o número de página física
 - Outro para obter o dado da memória principal
 - Como acelerar isso?
 - **E se tivéssemos uma cache para a tabela de páginas?**



TLB

TLB

- *Translation Look-aside Buffer*
 - Cache de tradução
 - **Cache da tabela de páginas de um programa**
 - Se o endereço para traduzir não está na TLB
 - Busca na tabela de páginas
 - Coloca endereço na TLB para casos futuros
 - Se o endereço está na TLB, traduz automaticamente

TLB

- Se o endereço que o programa usa é virtual, a cache usa endereços virtuais ou físicos?
 - **Cache fisicamente endereçada**
 - Todo acesso à cache passa primeiro pela TLB
 - TLB tem que ser supermegaextra rápida
 - **Cache virtualmente endereçada**
 - TLB só é necessária quando se sai da cache
 - Precisa identificar de qual thread é o endereço...

CONSIDERAÇÕES FINAIS

Considerações finais

- Memória virtual permite
 - Compartilhar a memória com outros processos e usuários



- Prover um espaço de endereçamento ao programador maior do que a memória física
- Proteção dos dados

Considerações finais

- Características principais da memória virtual
 - **Divisão da memória em páginas**
 - **Tradução de endereços virtuais para físicos**
 - Através da tabela de páginas
 - Tabela na memória principal
 - Acelerada através da TLB
 - Cache da tabela

EXERCÍCIOS

Exercícios

- Dada um sistema que endereça 16 GB de memória física com palavras de 64 bits e uma cache 4-associativa com 256 blocos de 16 bytes cada, apresente o tamanho efetivo da cache.

Exercícios

- Sabendo que a tabela de páginas ocupa 64KB em um sistema que endereça 64MB de memória virtual com páginas de 8KB, qual é o tamanho da memória principal se apenas 1 bit é usado para controle na tabela?

Exercícios

- Utilizando a tabela de páginas abaixo, traduza os seguintes endereços virtuais: 0xDCDC, 0x0220, 0xFAFA

**Tabela de
páginas**



Índice	Validade	Proteção (r/w)	Modificado	Referenciado	End. físico
0	1	11	0	0	0x00
1	1	11	1	1	0x13
2	1	10	1	0	0x26
3	1	10	0	1	0x39
4	1	11	0	0	0x52
5	1	11	1	0	0x65
6	1	10	1	0	0x78
7	1	10	0	1	0x91

Exercícios

- Calcule quantos ciclos a tradução de endereços adiciona ao CPI de um processador tendo em vista as informações abaixo.
 - TLB
 - Hit rate: 95%
 - Hit time: 1 ciclo
 - Miss penalty: 100 ciclos

INE5607 – Organização e Arquitetura de Computadores

Hierarquia e Gerência de Memória

Aula 27: Memória virtual

Prof. Laércio Lima Pilla

laercio.pilla@ufsc.br

