Gerência de Memória Virtual

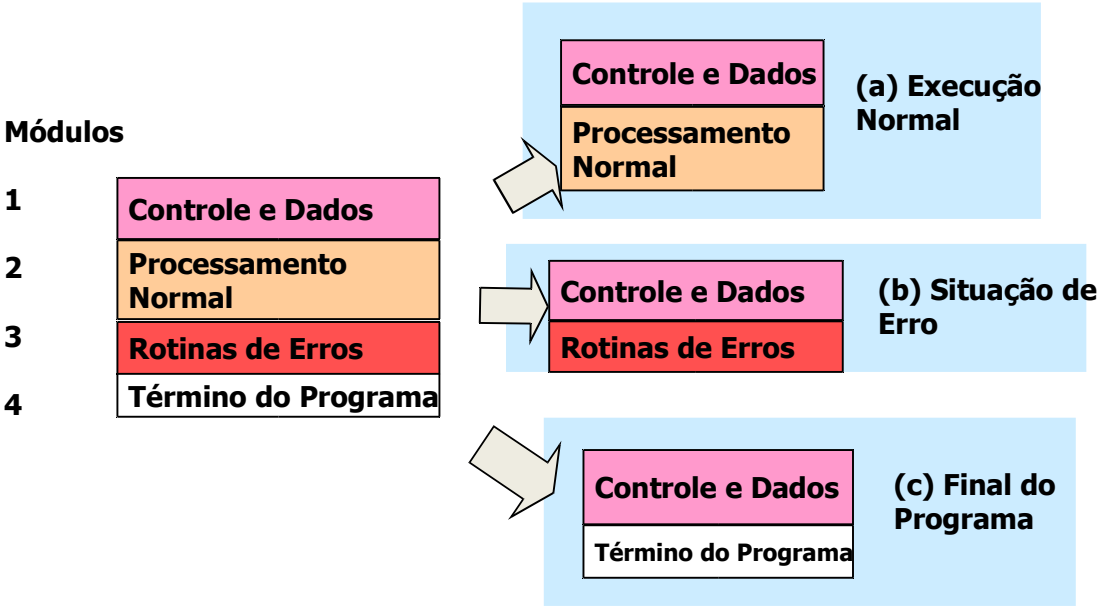
Prof. Dr. Antonio Carlos Sementille

Gerência de Memória

3. PRINCÍPIOS DE MEMÓRIA VIRTUAL

- **OVERLAYS**
  - **Conceito nascido no tempo em que os computadores possuíam pouca memória, e não comportavam programas grandes.**
  - **Idéia Principal: dividir o programa em partes (módulos), de modo que cada um caiba na memória.**
  - **Neste tipo de técnica, o programador é responsável por:**
    - **dividir o programa;**
    - **decidir onde armazenar cada parte; e**
    - **transportar cada parte entre a memória principal e secundária.**

Gerência de Memória



Gerência de Memória

- **Com o objetivo de automatizar o mecanismo de overlay, um grupo de pesquisa em Manchester (1961), criou um método, agora chamado de**

Memória Virtual.

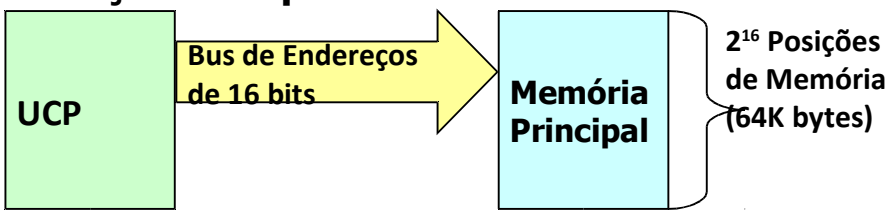
Gerência de Memória

PAGINAÇÃO

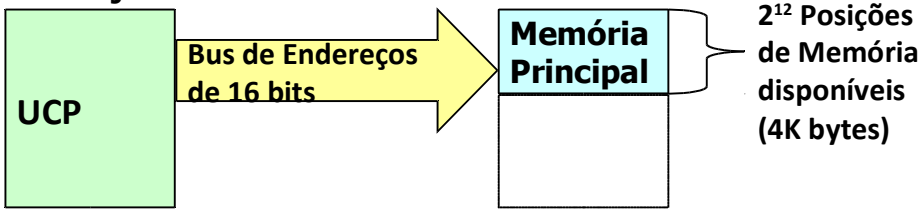
- **Idéia da Paginação: separar os conceitos de espaço de endereçamento e posições de memória.**
- **EXEMPLO: Suponha um computador com um campo de endereço de 16 bits e uma memória principal de 4096 bytes (4K).**
- **Qualquer programa neste computador pode, então, gerar endereços que variam de 0 à 64 kbytes (2^{16}), apesar das posições de endereços acima de 4096 não existirem fisicamente.**
- **O hardware forçava uma correspondência de um para um entre espaços de endereçamento e endereços reais.**

Gerência de Memória

- **Endereçamento possível**

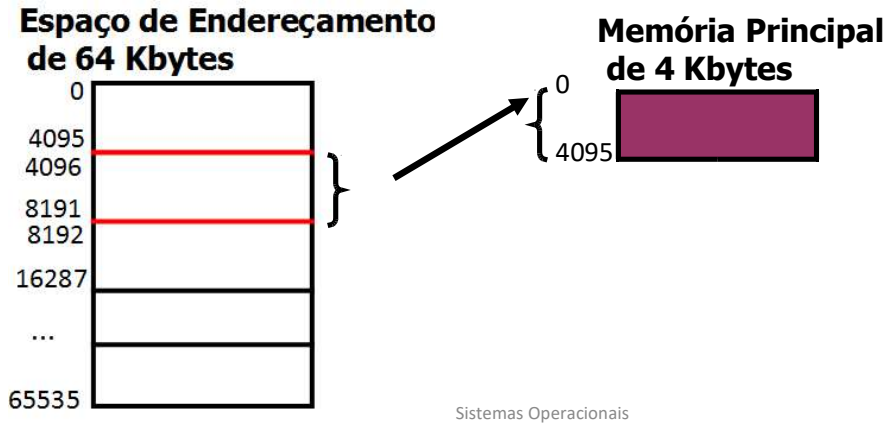


- **Endereçamento Real**



Gerência de Memória

- **A idéia de separar o espaço de endereçamento e os endereços de memória é a seguinte: em qualquer instante, 4096 bytes podem ser acessados, mas não precisam corresponder aos endereços 0 à 4096.**



Gerência de Memória

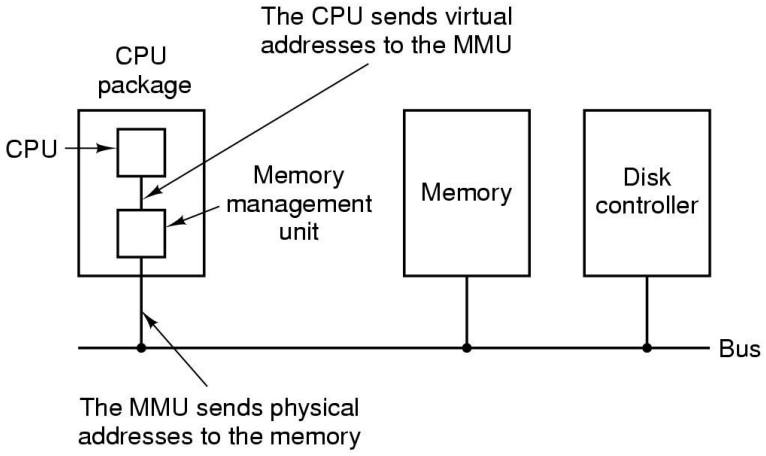
- **Pergunta interessante: o que acontece se um programa salta para um endereço entre 8192 e 12287?**
  - Ocorreriam os seguintes passos:

1. O conteúdo da memória principal seria salvo na memória secundária;
2. As posições entre 8192 e 12287 seriam localizadas na memória secundária;
3. Seriam trazidas para a memória principal;
4. O mapa de endereços seria trocado para mapear os endereços de 8192 a 12287, nas posições 0 a 4095;
5. A execução continuaria.

Gerência de Memória

- **Suposição: há sempre espaço suficiente na memória secundária para armazenar um programa completo e seus dados.**
- **A paginação dá ao programador a ilusão de uma memória principal grande, linear e contínua, do mesmo tamanho do espaço de endereçamento, quando, de fato, a memória principal disponível pode ser menor que o espaço de endereçamento.**
- **A simulação desta memória ampla não pode ser detectada por programa: é um mecanismo transparente ao usuário.**

A MMU (*Memory Management Unit*), presente na UCP, é responsável pela conversão do endereço virtual em endereço físico



A posição e função da MMU

Gerência de Memória

- **Termos Importantes:**
  - **Espaço de endereçamento virtual: são os endereços que o programa pode referenciar.**
  - **Espaço de endereçamento físico: são os endereços reais de memória.**
  - **Tabela de Páginas: relaciona os endereços virtuais com endereços físicos.**

Gerência de Memória

- **Implementação da Paginação**

- **Suposições Básicas**
  - **Existência de uma memória secundária para manter um programa completo.**
  - **A cópia de um programa na memória secundária pode ser considerado o original.**
  - **As partes trazidas da memória principal, de vez em quando, podem ser consideradas cópias.**
  - **Quando modificações são feitas na cópia da memória principal, elas devem ser refletidas no original.**

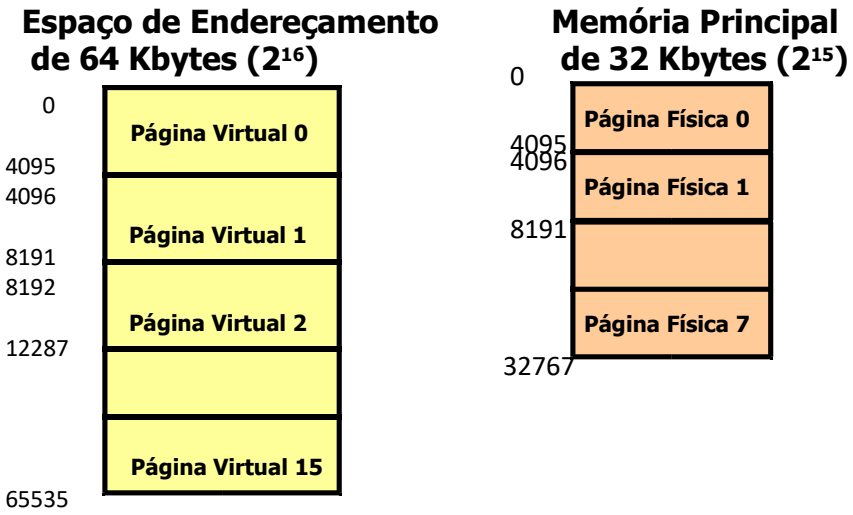
Gerência de Memória

Características da Implementação

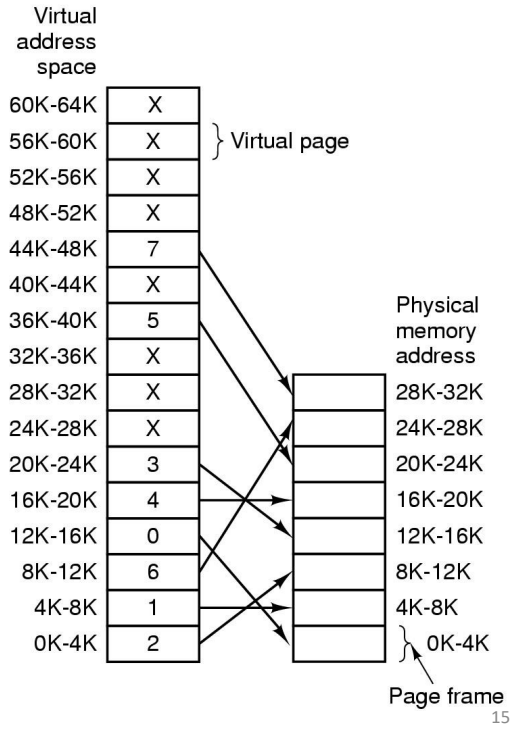
- O espaço de endereçamento virtual é dividido em páginas de tamanhos iguais (geralmente de 512 a 32K endereços por página).
- O espaço de endereçamento físico tem tamanho de página idêntico ao do endereçamento virtual (estas páginas são chamadas páginas físicas, ou molduras de páginas – *page frames*).

Gerência de Memória

Exemplo de uma Implementação

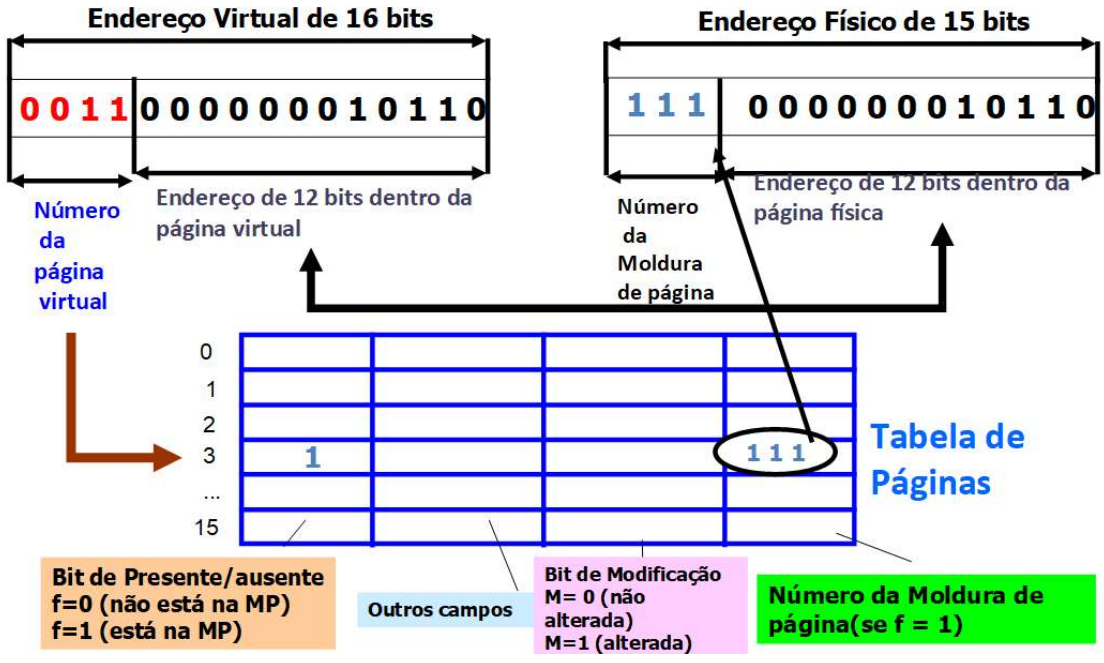


A relação entre endereços virtuais e endereços físicos dada pela tabela de páginas

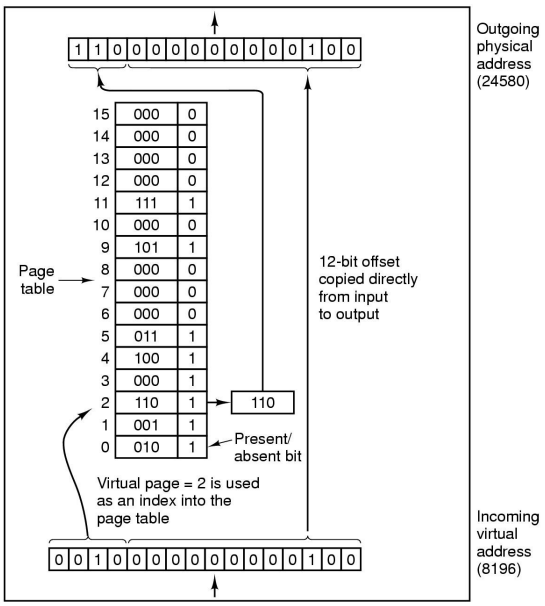


15

Gerência de Memória

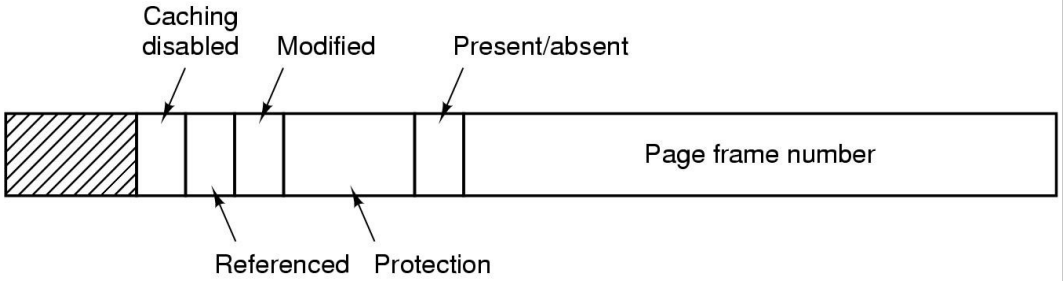


Tabelas de Páginas



Operação interna da MMU com 16 páginas de 4KB

Uma entrada típica da tabela de páginas



- Número da moldura de página: número da página física onde a PV está carregada
- Bit presente/ausente: se a PV está (bit 1) ou não está (bit 0) presente na memória
- Proteção: tipos de acesso permitidos, por exemplo, leitura, escrita e execução
- Modificado: se a PV foi modificada na memória
- Referenciada: se a PV foi referenciada (bit 1) ou não (bit 0), para leitura/escrita enquanto estava na memória
- Bit de cache desabilitado: desabilita o cache para a página em questão

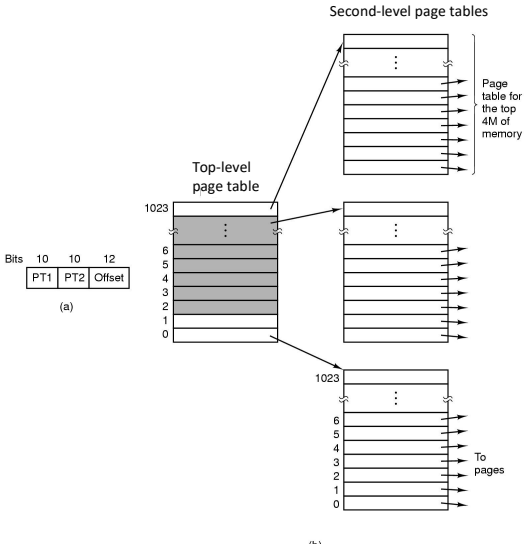
Tabelas de Páginas Multiníveis

- Muitos computadores: dividem a tabela de páginas em múltiplos niveis, para minimizar o espaço de memória necessário para seu armazenamento
- O segredo para o método multinível de tabelas de páginas é evitar que todas elas sejam mantidas na memória ao mesmo tempo
- Exemplo: considerando um endereço virtual de 32 bits

0000 0000 01	0000 0000 11	0000 0000 0100
Pt1	Pt2	deslocamento

= 000403004 hexa

Tabela de Páginas



- Endereço virtual de 32 bits com 2 campos para endereçamento de tabelas de páginas
- Tabelas de páginas com 2 níveis

Gerência de Memória

Considerações Importantes

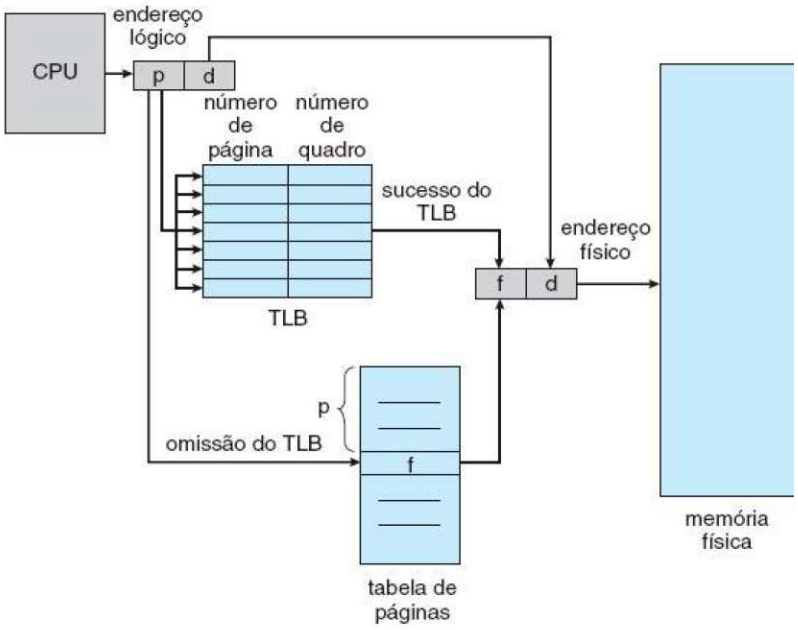
- **Se o S.O. fizesse a conversão de endereços, a paginação seria inviável.**
- **Para acelerar conversão de endereços, uma parte da tabela de páginas é mantida na UCP em registradores associativos ou TLBs – *Translation Lookaside Buffers***

TLBs – *Translation Lookaside Buffers* (Tabela de tradução de endereços)

Valid	Virtual page	Modified	Protection	Page frame
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Uma TLB para acelerar a paginação

TLBs – *Translation Lookaside Buffers* (Tabela de tradução de endereços)



Hardware de paginação com TLB.

23

Tabelas de Páginas Invertidas

- Computadores de 64 bits => tabela de páginas muito grande (nos moldes tradicionais)
- Exemplo: se o computador tem 64 bits de endereço virtual então seu endereçamento virtual é 2^{64} bytes, se adotarmos páginas de 4096 bytes, teremos uma tabela com 2^{52} entradas.
- Se cada entrada tiver 8 bytes, teremos uma tabela de 30.000.000 gigabytes de tamanho

24

Tabelas de Páginas Invertidas

- Uma solução: tabela de páginas invertidas
- Uma entrada por moldura de página na memória real
- Cada entrada informa que o par (processo, página virtual) está localizada na moldura de página
- Ex: computadores de 64 bits com páginas de 4KB e 256 MB de RAM, a tabela de páginas terá 65356 bytes

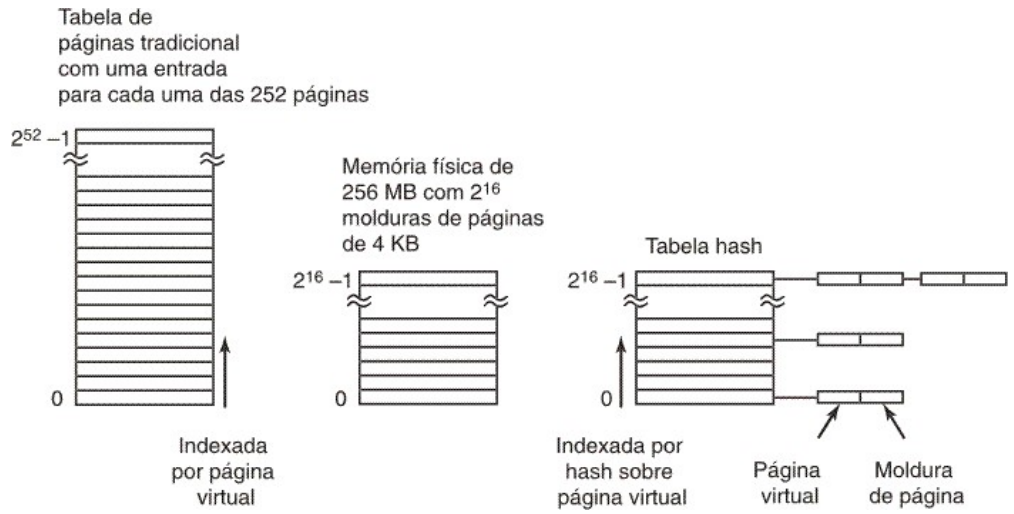
25

Tabelas de Páginas Invertidas

- A economia de espaço tem um preço: dificuldade na conversão do endereço virtual em físico
- Se processo *n* referencia a página virtual *p*, *p* não pode ser usado como índice na tabela
- Ao invés: o hardware deve pesquisar toda a tabela de páginas invertidas à procura do par (*n, p*)
- Este procedimento deve ser feito à cada referência à memória
- Para acelerar este processo pode-se manter as páginas mais usadas na TLB e manter uma tabela hash dos endereços virtuais

26

Tabelas de Páginas Invertidas



Comparação de uma tabela de páginas tradicional com uma tabela de páginas invertidas

27