

SISTEMAS OPERACIONAIS 1

21270 A

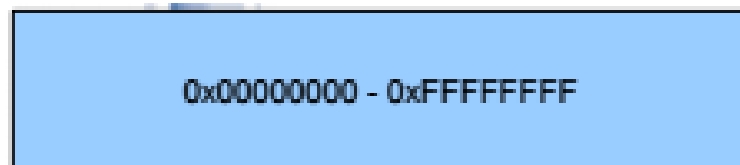


Departamento de Computação
Prof. Kelen Cristiane Teixeira Vivaldini

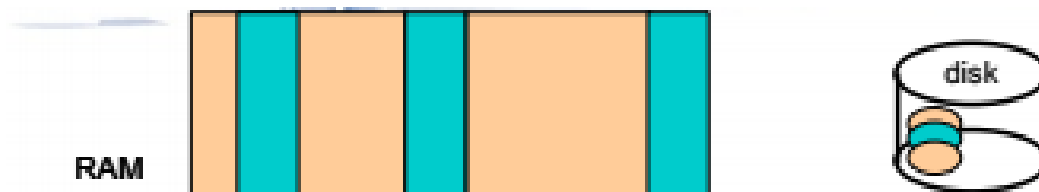
Gerenciamento de Memória

Memória Virtual (MV)

- A ideia básica para a memória virtual é que cada programa tem seu próprio espaço de endereçamento, que é dividido em blocos chamados páginas. Cada página é uma série contíguo de endereços.



- De fragmentos de RAM física e disco



Gerenciamento de Memória

Memória Virtual (MV)

virtual → physical

- Software lidam com virtual addresses
- Hardware precisam de physical addresses

0x12345678 → ???



Gerenciamento de Memória

Memória Virtual (MV)

- Programas maiores que a memória eram divididos em pedaços menores chamados *overlays*;
 - Programador define áreas de *overlay*;
 - Vantagem: expansão da memória principal;
 - Desvantagem: custo muito alto;

Gerenciamento de Memória

Memória Virtual (MV)

- Sistema operacional é responsável por dividir o programa em *overlays*;
- Sistema operacional realiza o chaveamento desses pedaços entre a memória principal e o disco;
- Década de 60: ATLAS → primeiro sistema com MV (Universidade Manchester - Reino Unido);
- 1972: sistema comercial: IBM System/370;

Gerenciamento de Memória

Memória Virtual (MV)

- Com MV existe a sensação de se ter mais memória principal do que realmente se tem;
- O *hardware* muitas vezes implementa funções da gerência de memória virtual:
 - SO deve considerar características da arquitetura;

Gerenciamento de Memória

Memória Virtual

- Espaço de Endereçamento Virtual de um processo é formado por todos os endereços virtuais que esse processo pode gerar;
- Espaço de Endereçamento Físico de um processo é formado por todos os endereços físicos/reaís aceitos pela memória principal (RAM);

Gerenciamento de Memória

Memória Virtual

- Um processo em Memória Virtual faz referência a endereços virtuais e não a endereço reais de memória RAM;
- No momento da execução de uma instrução, o endereço virtual é traduzido para um endereço real, pois a CPU manipula apenas endereços reais da memória RAM → MAPEAMENTO;

Gerenciamento de Memória

Mapeamento MV

- MMU: Realiza mapeamento dos endereços lógicos (usados pelos processos) para endereços físicos;

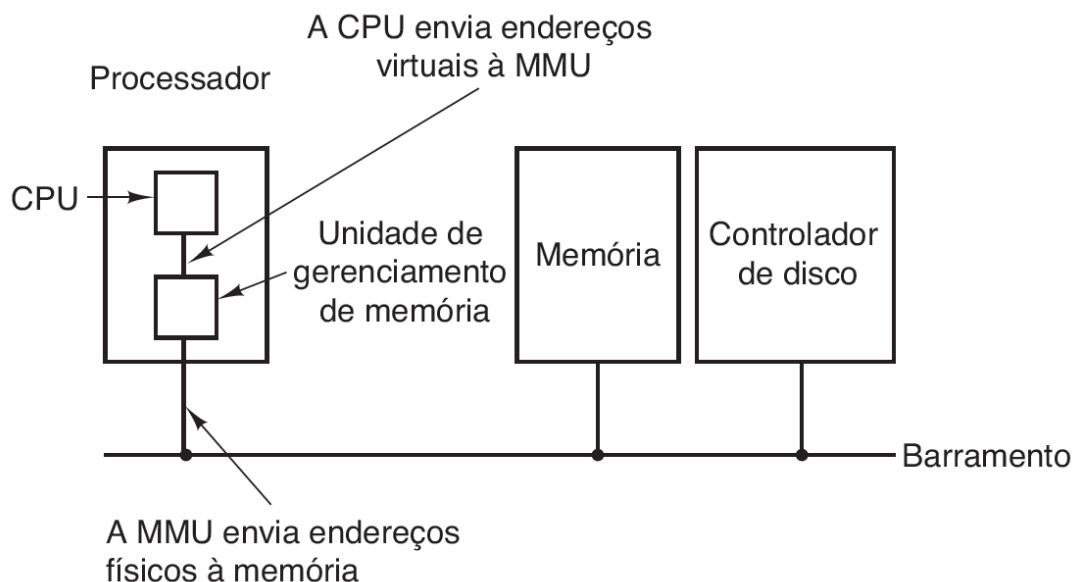


Figura 3.8 A posição e a função da MMU. Aqui a MMU é mostrada como parte do chip da CPU (processador) porque isso é comum atualmente. Contudo, em termos lógicos, poderia ser um chip separado, como ocorria no passado.

Gerenciamento de Memória

Memória Virtual

- Técnicas de MV:
 - **Paginação:**
 - Blocos de tamanho fixo chamados de páginas;
 - SO mantém uma fila de todas as páginas;
 - Endereços Virtuais formam o espaço de endereçamento virtual;
 - O espaço de endereçamento virtual é dividido em páginas virtuais;
 - Mapeamento entre endereços reais e virtuais (MMU);

Gerenciamento de Memória

Memória Virtual

- Processo é dividido em Páginas
- A Memória é dividida em Molduras (ou Frames) de mesmo tamanho
 - Tamanho das Páginas = tamanho das Molduras
- Páginas/Molduras são de pequeno tamanho (e.g., 1K):
 - fragmentação interna pequena
- Processo não precisa ocupar área contígua em memória
 - Elimina fragmentação externa
- Processo não precisa estar completamente na MP;
- SO mantém uma tabela de páginas por processo ;
- Endereços são gerados dinamicamente em tempo de execução;

Gerenciamento de Memória

Memória Virtual

- Técnicas de MV:
 - **Segmentação:**
 - Blocos de tamanho arbitrário chamados segmentos;
- Arquitetura (hardware) tem que possibilitar a implementação tanto da paginação quanto da segmentação;

Gerenciamento de Memória

Memória Virtual - *Swapping*

- ***Swapping***: chaveamento de processos inteiros entre a memória principal e o disco;
 - *Swap-out*;
 - *Swap-in*;
 - Pode ser utilizado tanto com partições fixas quanto com partições variáveis;
 - *Swapper*;

Gerenciamento de Memória

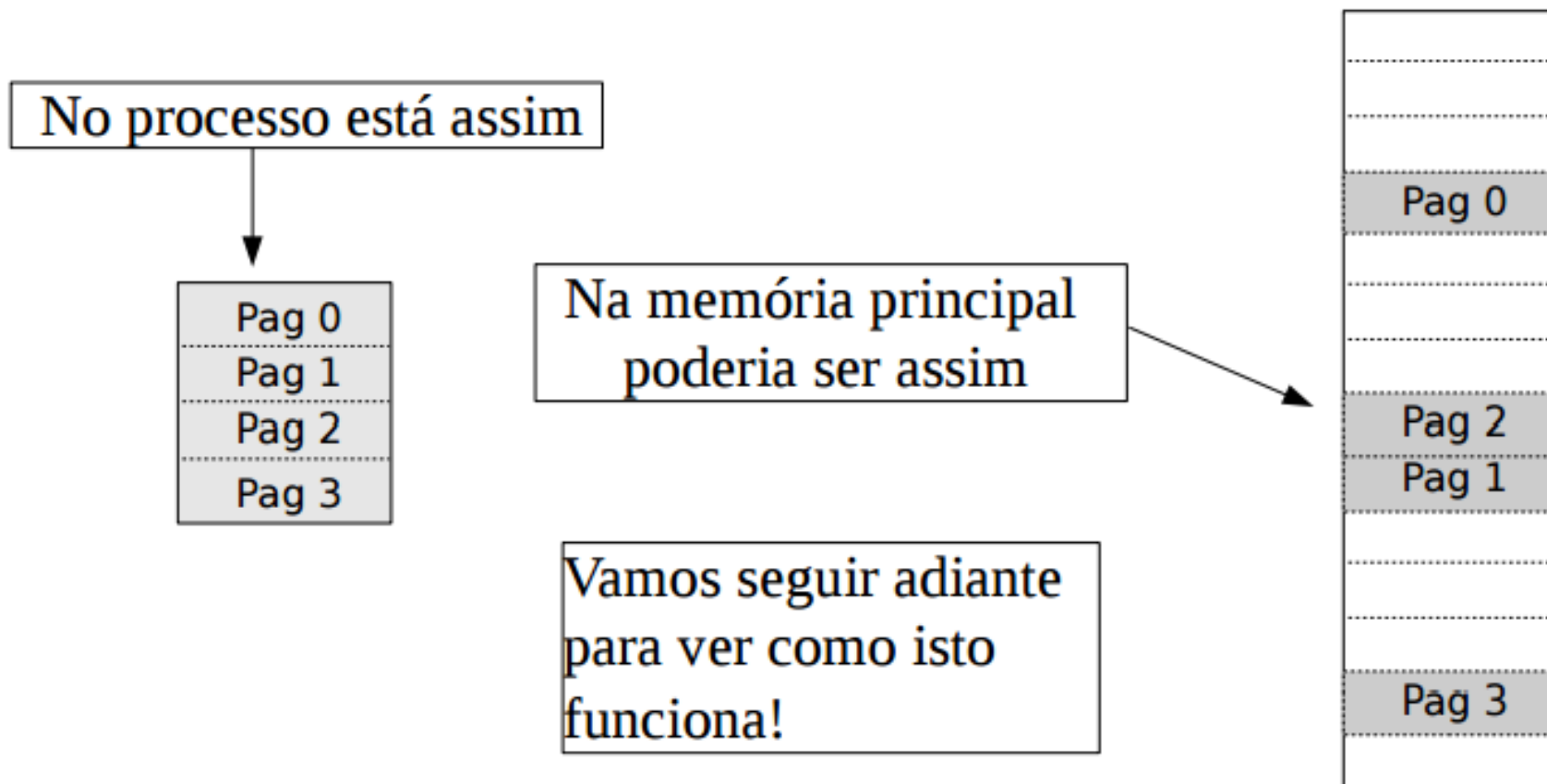
Memória Virtual - Paginação

- Memória Principal e Memória Secundária são organizadas em páginas de mesmo tamanho;
- Página é a unidade básica para transferência de informação;
- Tabela de páginas: responsável por armazenar informações sobre as páginas virtuais:
 - argumento de entrada → número da página virtual;
 - argumento de saída (resultado) → número da página real (ou moldura de página - *page frame*);

Gerenciamento de Memória

Memória Virtual - Paginação

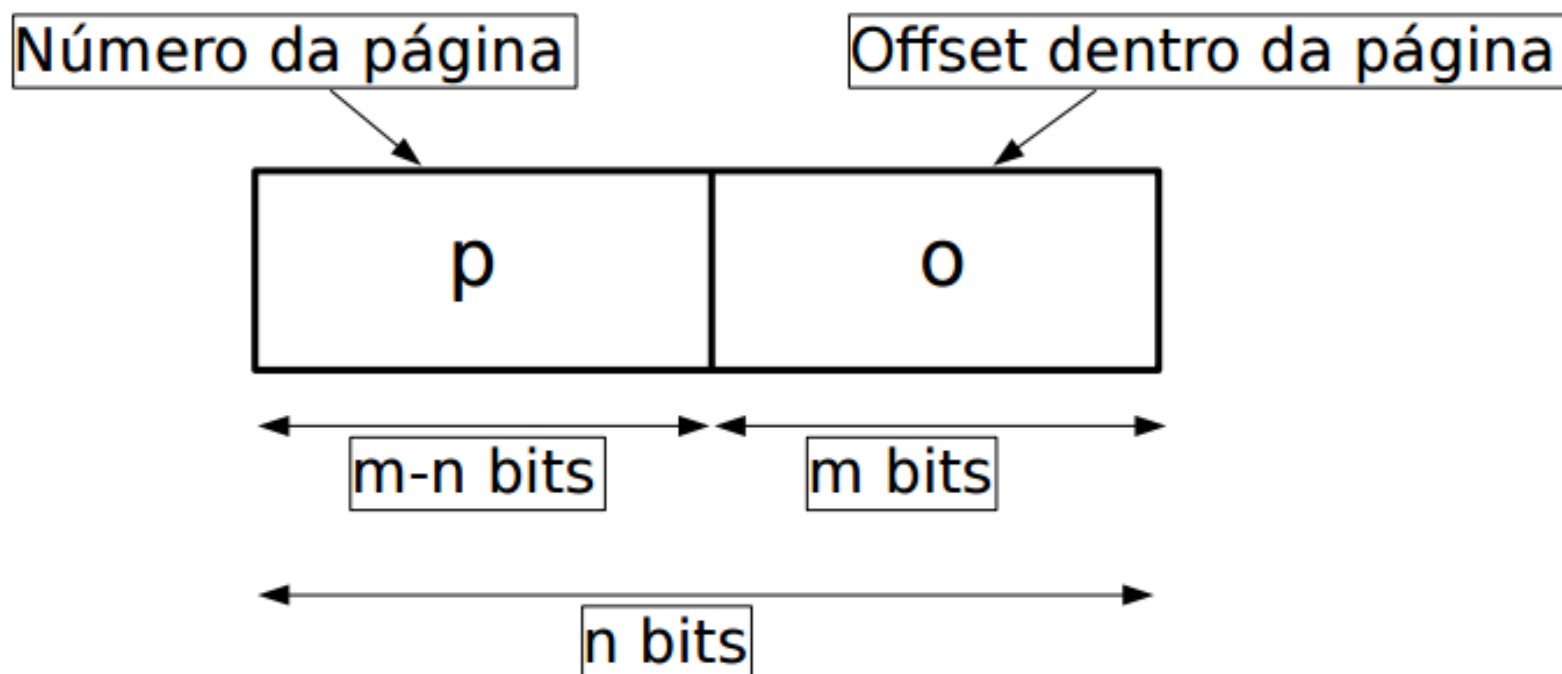
- Visão Simplificada



Gerenciamento de Memória

Memória Virtual - Paginação

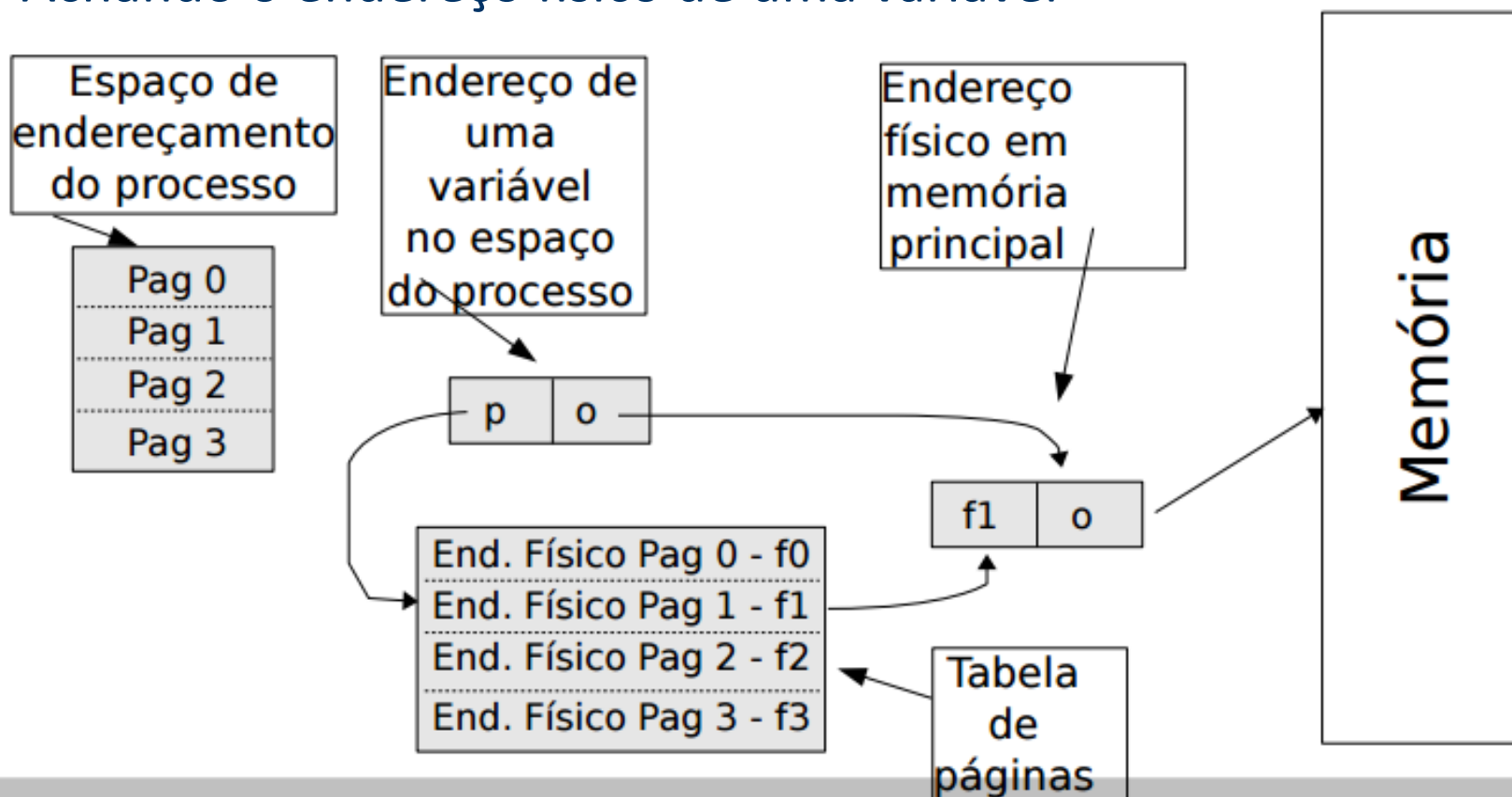
- Endereçamento
 - Todo endereço é dividido em duas partes



Gerenciamento de Memória

Memória Virtual - Paginação

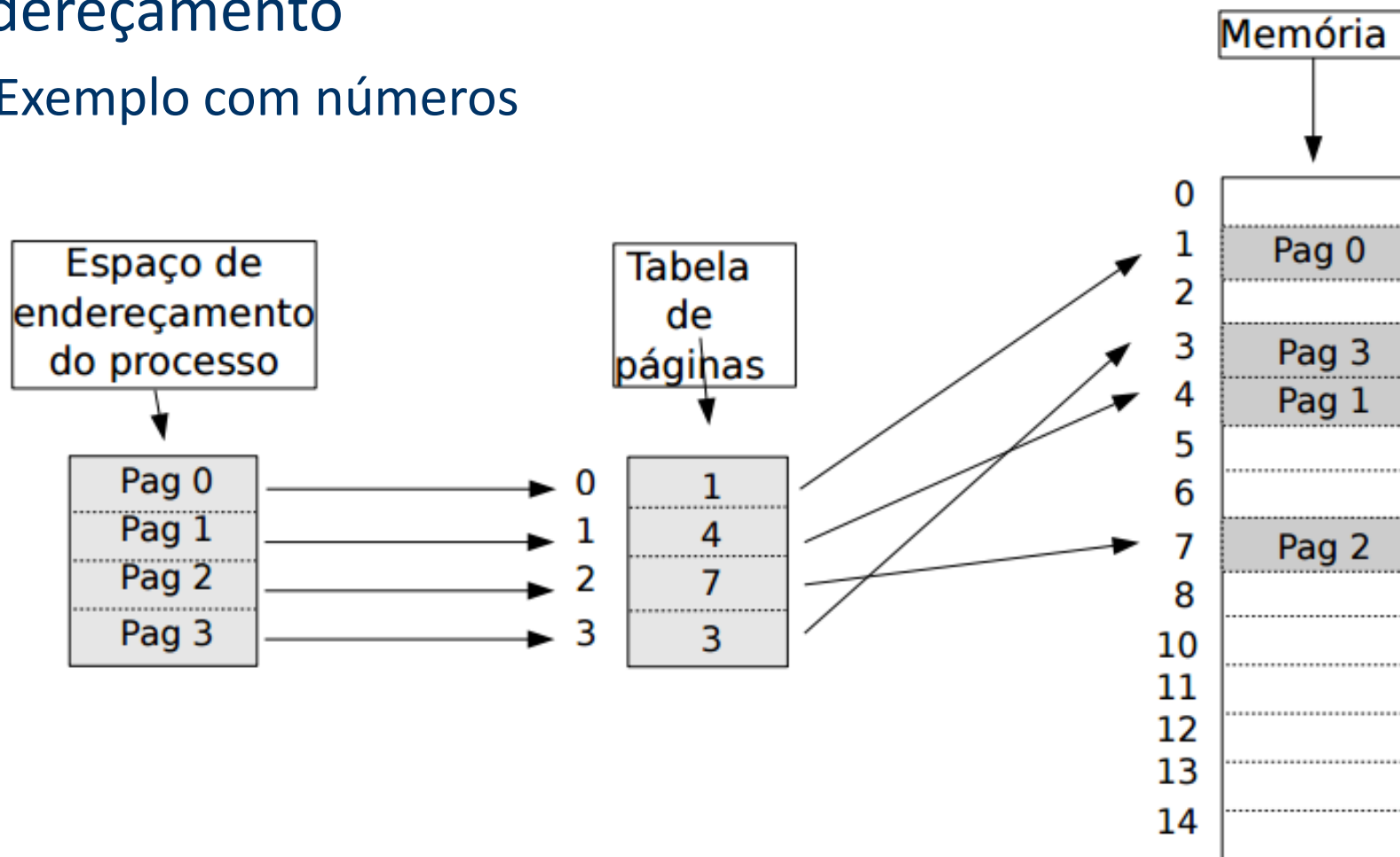
- Endereçamento
 - Achando o endereço físico de uma variável



Gerenciamento de Memória

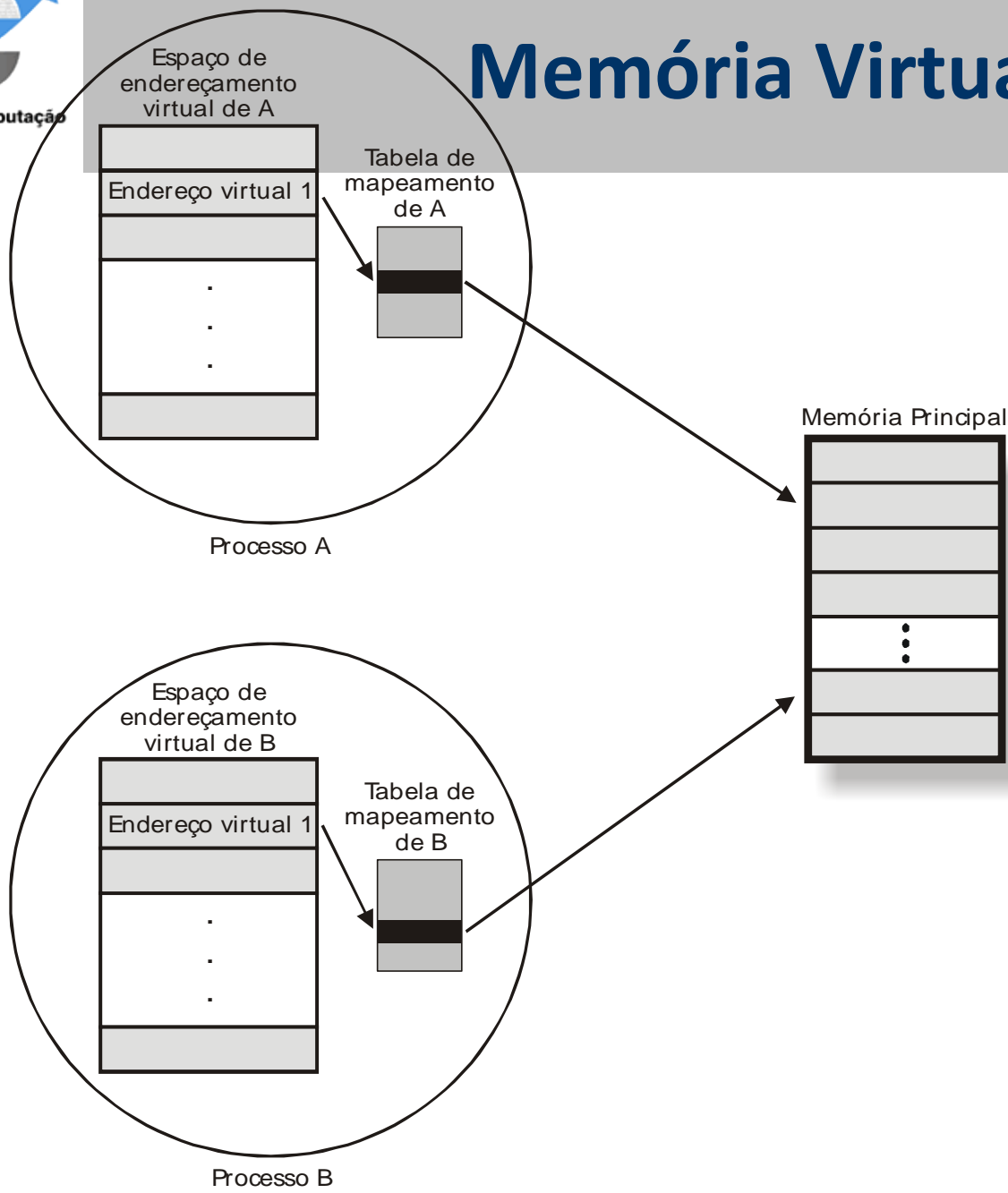
Memória Virtual - Paginação

- Endereçamento
 - Exemplo com números



Gerenciamento de Memória

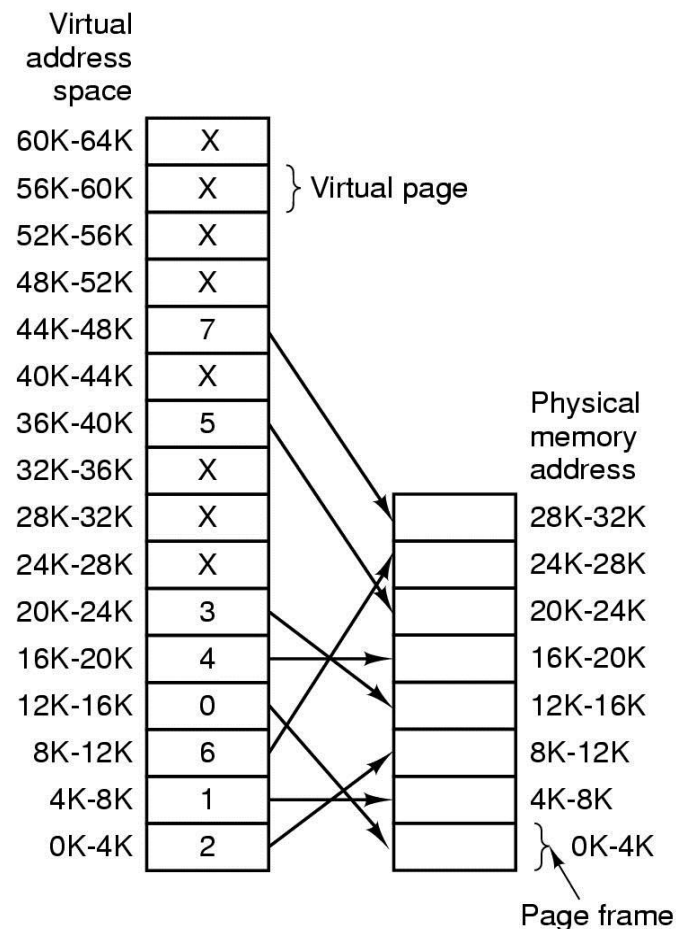
Memória Virtual - Paginação



Gerenciamento de Memória

Memória Virtual

- Exemplo:
 - Páginas de 4Kb
 - 4096 bytes/endereços (0-4095);
 - 64Kb de espaço virtual;
 - 32Kb de espaço real;
 - Temos:
 - 16 páginas virtuais;
 - 8 páginas reais;



Gerenciamento de Memória

Memória Virtual

Espaço Virtual X Tamanho da Página

Espaço de Endereçamento Virtual	Tamanho da página	Número de páginas	Número de entradas nas tabela de páginas
2^{32} endereços	512 bytes	2^{23}	2^{23}
2^{32} endereços	4 kbytes	2^{20}	2^{20}
2^{64} endereços	4 kbytes	2^{52}	2^{52}
2^{64} endereços	64 kbytes	2^{48}	2^{48}

Gerenciamento de Memória

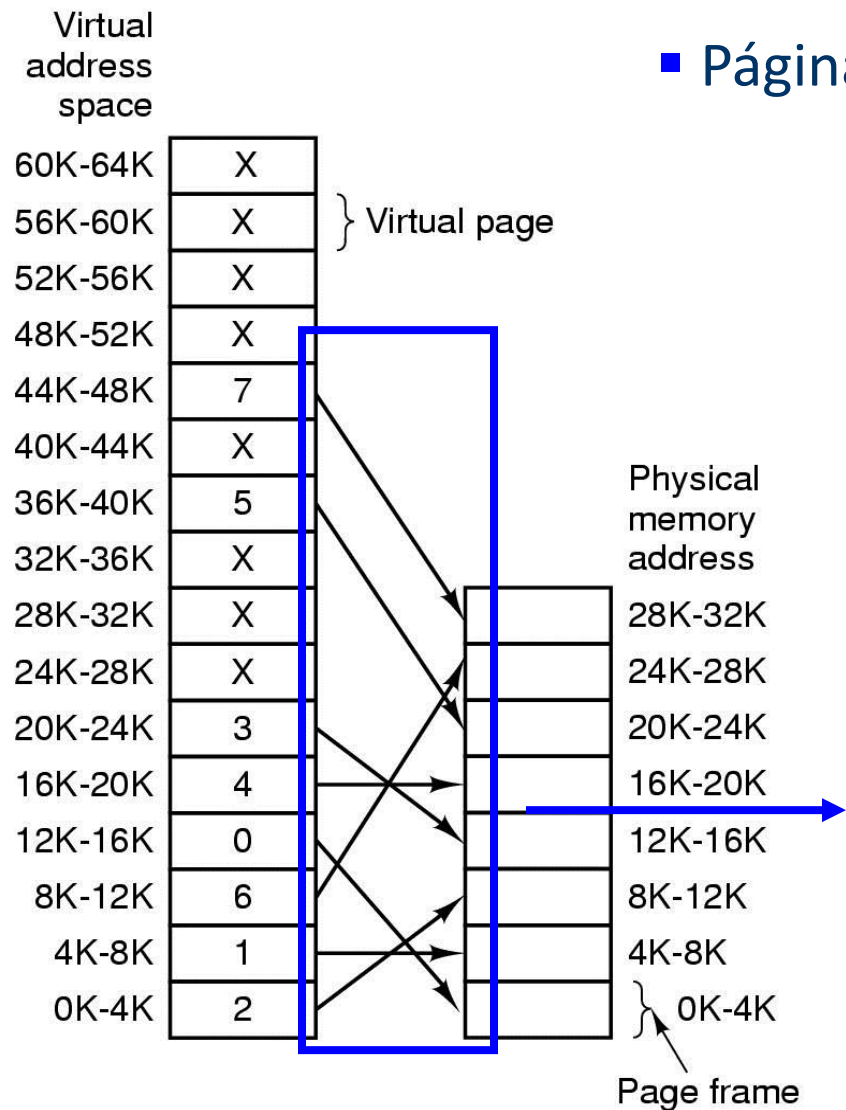
Memória Virtual - Paginação

- Problemas:
 - Fragmentação interna;
 - Definição do tamanho das páginas;
 - Geralmente a MMU que define e não o SO;
 - Páginas maiores: leitura mais eficiente, tabela menor, mas maior fragmentação interna;
 - Páginas menores: leitura menos eficiente, tabela maior, mas menor fragmentação interna;
 - Sugestão: 1k a 8k;
- Mapa de bits ou uma lista encadeada com as páginas livres;

Gerenciamento de Memória

Endereço Virtual → Endereço Real

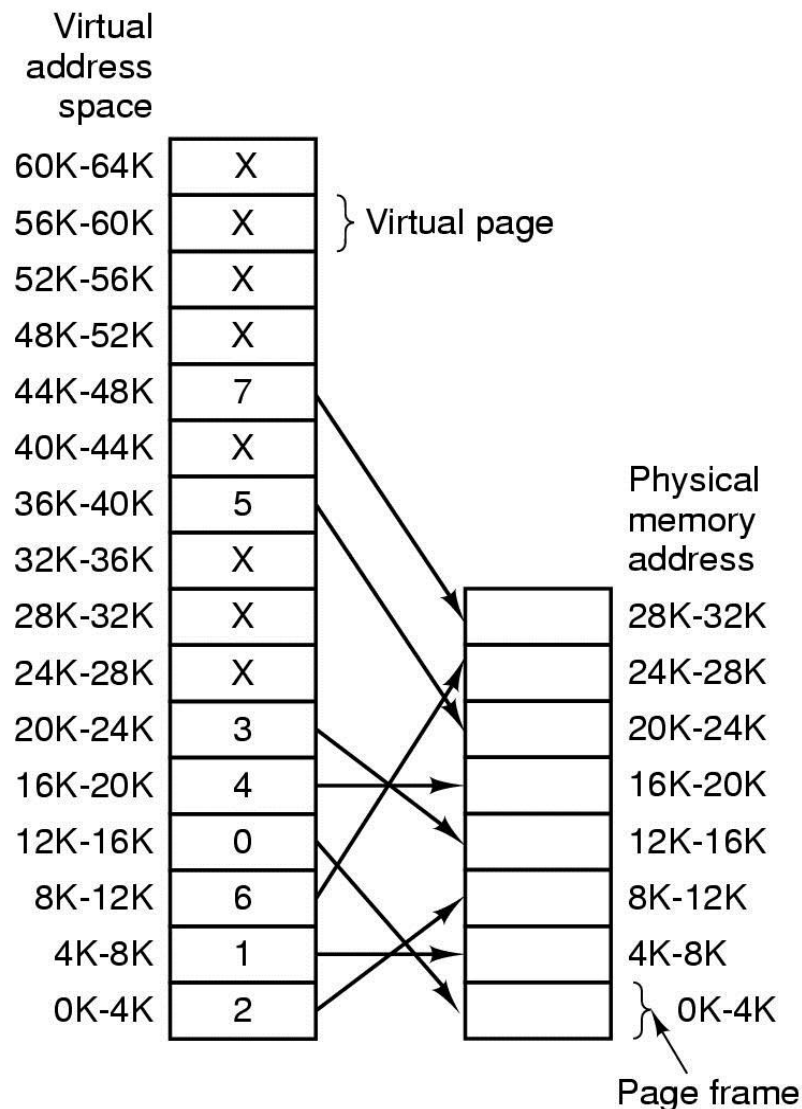
- Página virtual **mapeada** para página real;



- MMU realiza o mapeamento

Gerenciamento de Memória

Endereço Virtual → Endereço Real



■ **MOV REG, 20500**

■ **Qual é a página?**

Pag. 5, que contém os endereços de 20k (20480) até 24k-1 (24575)

■ **Esta página está em qual moldura?**

Na moldura 3, que contém end. físicos de 12k (12288) a 16k-1 (16384)

■ **Qual o deslocamento do endereço 20500 dentro da página?**

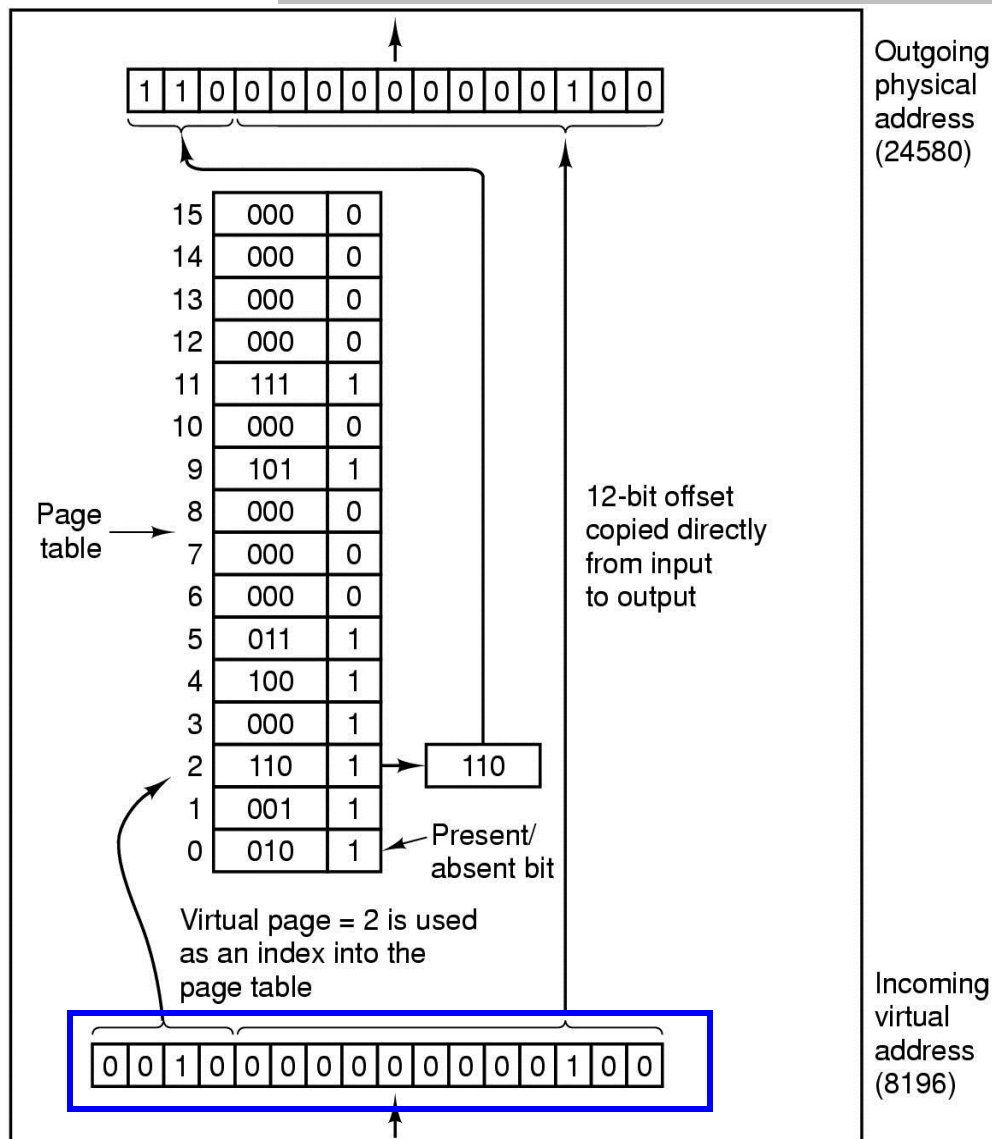
Desl. = End. virtual - End. virtual do
1º byte da página
= 20500 - 20480 = 20

■ **Qual será o endereço físico correspondendo ao end. virt. 20500?**

= End. do 1º byte da moldura + desloca.
= 12288 + 20 = 12308

Gerenciamento de Memória

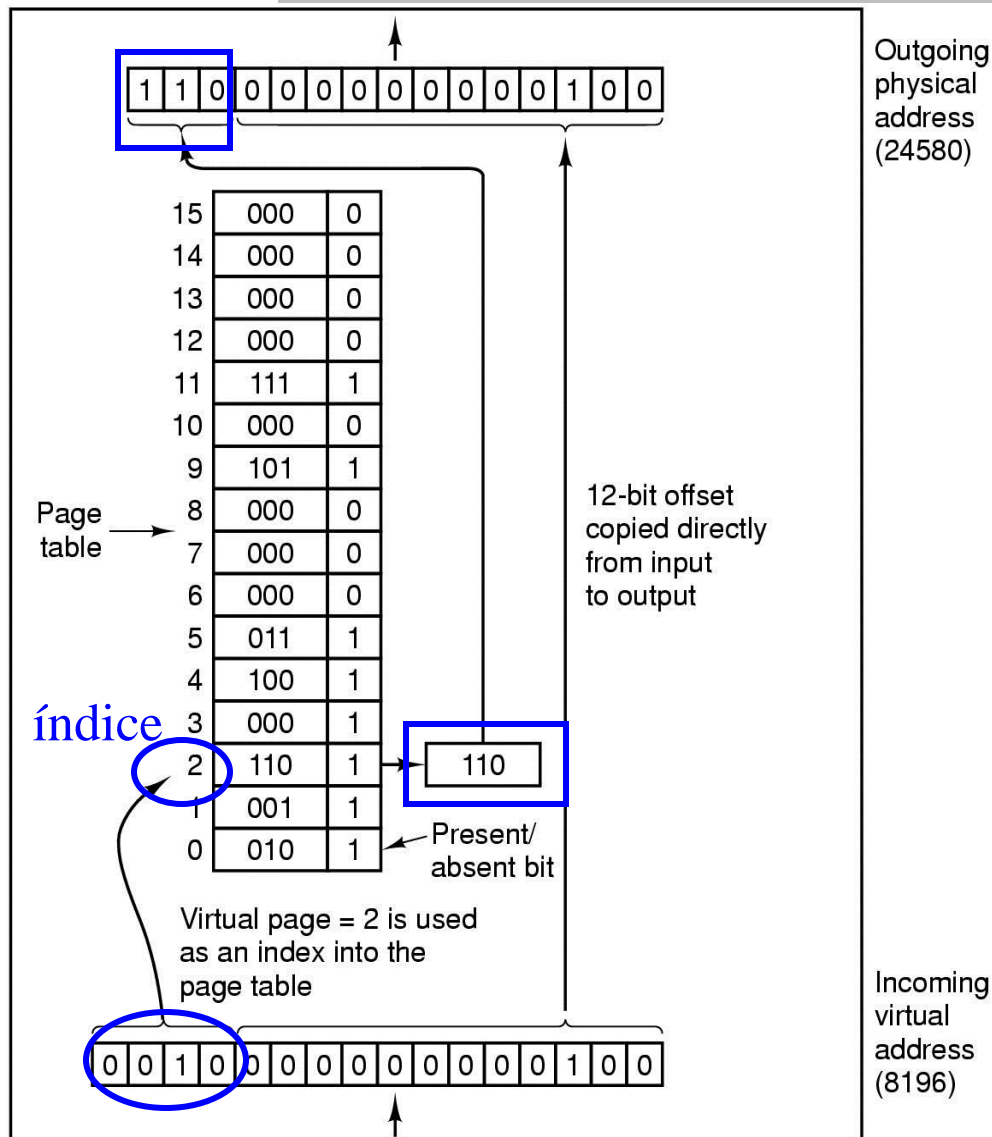
Mapeamento da MMU



- Operação interna de uma MMU com 16 páginas de 4Kb;
- Endereço virtual de **16 bits**: 4 bits para nº de páginas e 12 para deslocamento;
- Com 4 bits é possível ter 16 páginas virtuais (2^4);
- 12 bits para deslocamento é possível endereçar os 4096 bytes;

Gerenciamento de Memória

Mapeamento da MMU



- Número da página virtual é usado como índice;
- Se página está na memória RAM, então o nº da página real (110) é copiado para os três bits mais significativos do endereço de saída (real), juntamente com o deslocamento sem alteração;
- Endereço real com 15 bits é enviado à memória;

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



**Identifica a página real;
Campo mais importante;**

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



Bit de Residência:

Se valor igual 1, então entrada válida para uso;

Se valor igual 0, então entrada inválida, pois
página virtual correspondente não está na memória;

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



Bits de Proteção:

Indicam tipos de acessos permitidos:

1 bit → 0 – leitura/escrita

1 – leitura

3 bits → 0 – Leitura

1 – Escrita

2 - Execução

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



Bit de Modificação (Bit M):

Controla o uso da página;

Se valor igual a 1, página foi escrita;

página é copiada para o disco

Se valor igual a 0, página não foi modificada;

página não é copiada para o disco;

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



Bit de Referência (Bit R):

Controla o uso da página;

Auxilia o SO na escolha da página que deve deixar a MP (RAM);

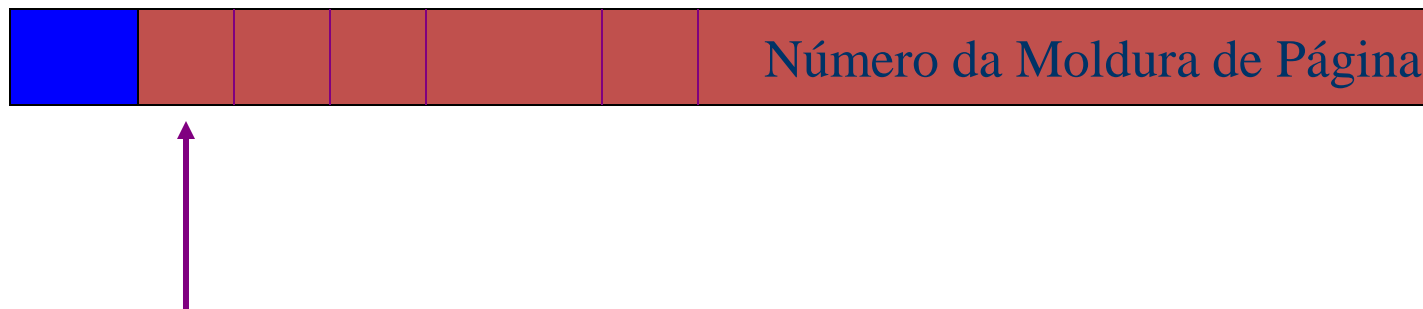
Se valor igual a 1, página foi referenciada (leitura/escrita);

Se valor igual a 0, página não referenciada;

Gerenciamento de Memória

Memória Virtual - Paginação

- Tabela de Páginas: 32 bits (mais comum)



Bit de Cache:

Necessário quando os dispositivos de entrada/saída são mapeados na memória e não em um endereçamento específico de E/S;

Gerenciamento de Memória

Memória Virtual - Paginação

- A Tabela de páginas pode ser armazenada de três diferentes maneiras:
 - Registradores, se a memória for pequena;
 - Na própria memória RAM → MMU gerencia utilizando dois registradores:
 - Registrador Base da tabela de páginas (PTBR – *page table base register*): indica o endereço físico de memória onde a tabela está alocada;
 - Registrador Limite da tabela de páginas (PTLR – *page table limit register*): indica o número de entradas da tabela (número de páginas);
 - Dois acessos à memória;

Gerenciamento de Memória

Memória Virtual - Paginação

- Em uma memória *cache* na MMU chamada Memória Associativa;
 - Também conhecida como TLB (*Translation Lookaside Buffer* - *buffer* de tradução dinâmica);
 - Hardware especial para mapear endereços virtuais para endereços reais sem ter que passar pela tabela de páginas na memória principal;
 - Melhora o desempenho;

Gerenciamento de Memória

Memória Virtual - Paginação

Translation Look-Aside Buffer (TLB)

Um conjunto de registradores especiais:

- Super rápido (são registradores)
- Cada registrador tem duas partes:
 - chave e valor
- Dada a chave, retorna o valor correspondente
- A busca é feita em todos os registradores simultaneamente
- É uma espécie de cache da tabela de páginas

Porquê é mais barato que colocar a tabela de páginas em registradores?

- Porquê é um cache. Armazena parte da tabela!

Gerenciamento de Memória

Memória Associativa

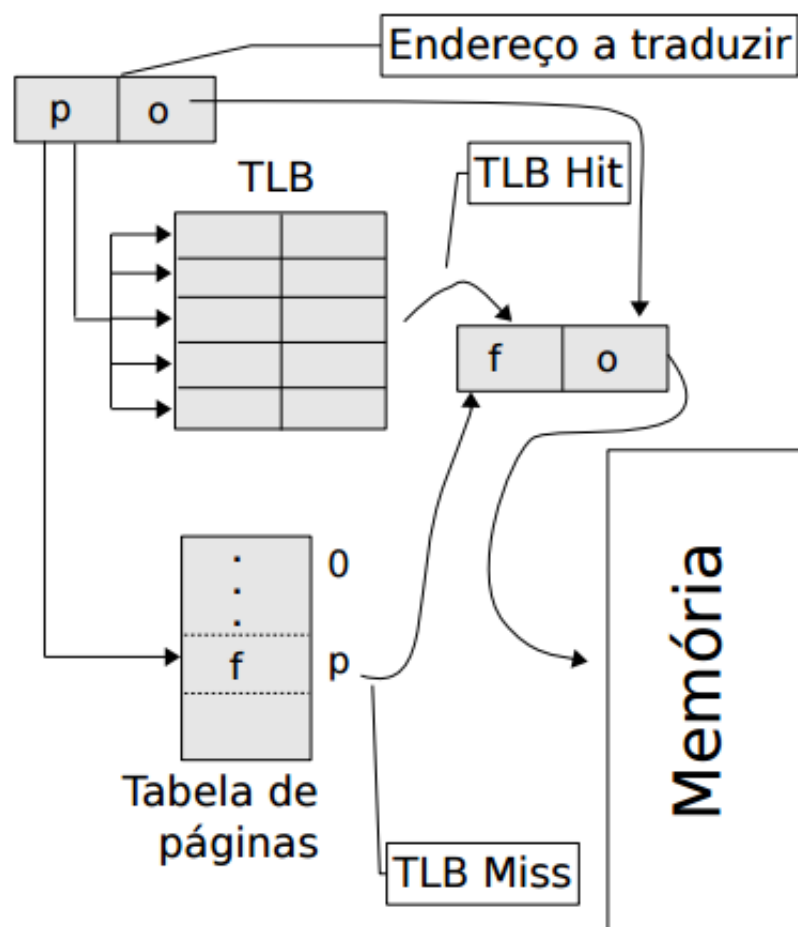
<i>Bit R</i>	Página Virtual	<i>Bit M</i>	<i>Bits de Proteção</i>	Página Física
1	140	1	RW	31
1	20	0	R X	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	R X	50
1	21	0	R X	45
1	860	1	RW	14
1	861	1	RW	75

Até 32/64 entradas

Gerenciamento de Memória

Memória Virtual - Paginação

TLB



O número de página é olhado no TBL;

Se achar, **TLB HIT** e retorna a página física

Se não achar, **TLB MISS**:

- Busca número da página na tabela de páginas (memória principal)
- Escolhe e limpa uma entrada no TLB
- Guarda o par (pag. Lógica e pag. física no TLB)
- Retorna a página física

Gerenciamento de Memória

Memória Virtual - Paginação

Tempo de Acesso:

Suponha que se gaste 20ns para olhar o TLB e 100ns para ler da memória:

TLB hit gasta: $20 + 100 = 120\text{ns}$

TLB miss gasta: $20 + 100 + 100 = 220\text{ns}$

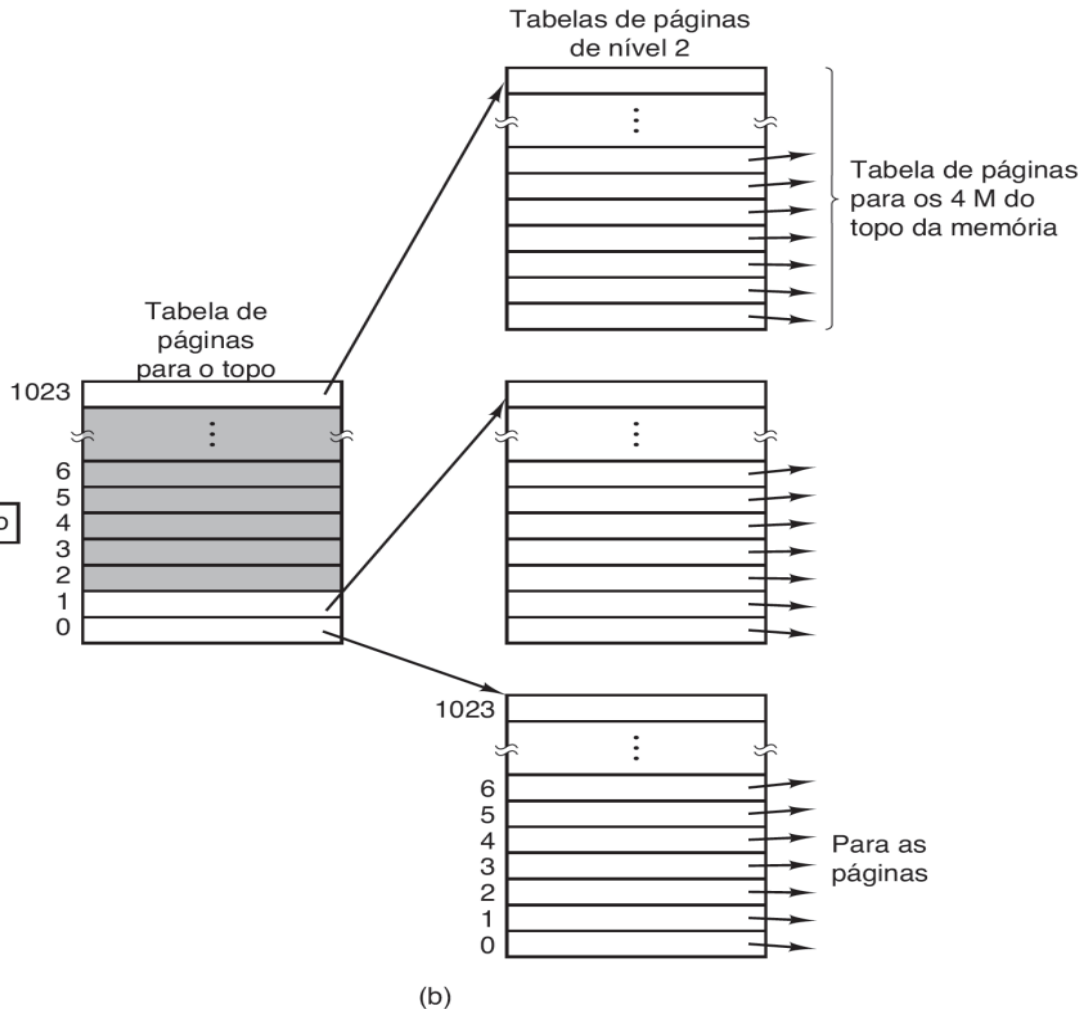
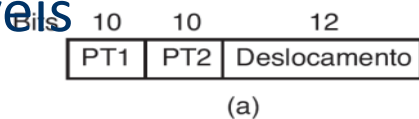
O hit rate diz a frequência com que achamos a página desejada no TLB. Com um hit rate de 80%, o tempo de acesso será:

$$0.8 * 120 + 0.2 * 220 = 140\text{ns}$$

Gerenciamento de Memória

Paginação Multinível

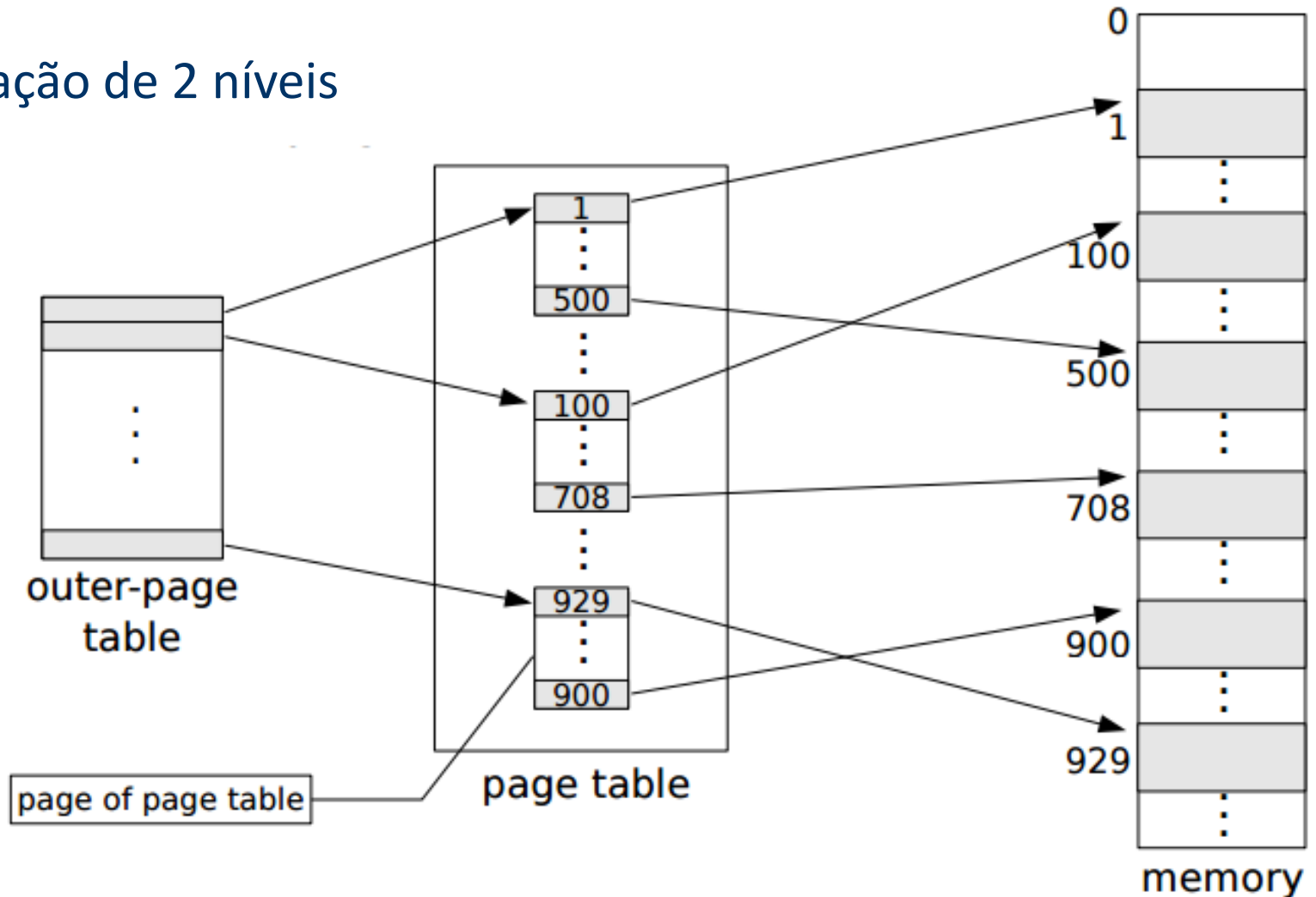
- Endereço de 32 bits com 2 campos para endereçamento de tabelas
- Tabelas de páginas com 2 níveis



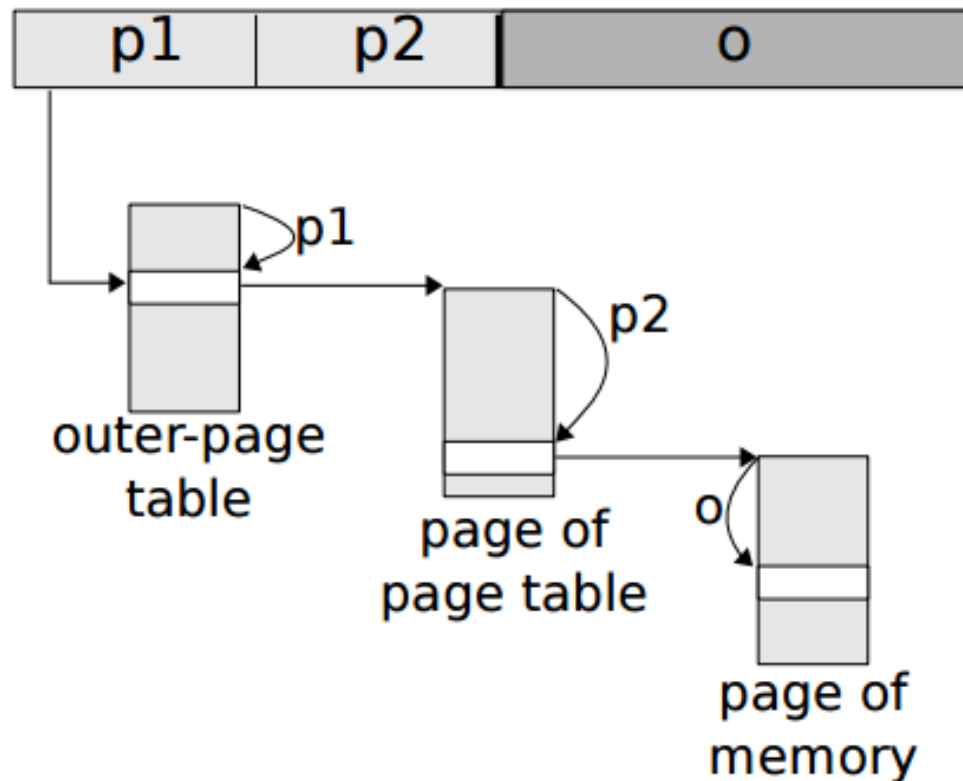
Gerenciamento de Memória

Paginação Multinível

- Paginação de 2 níveis



- Paginação de 2 níveis



Tempo de acesso

Suponha que:

- TLB hit gasta $20 + 100 = 120\text{ns}$
- Hit rate de 98%

Com um nível de paginação:

- TLB miss gasta $20 + 100 + 100 = 220\text{ns}$
- Tempo de acesso: $.98 * 120 + .02 * 220 = 122\text{ns}$

Com dois níveis de paginação:

- Tempo de acesso: $.98 * 120 + .02 * 320 = 124\text{ns}$

Com três níveis de paginação:

- Tempo de acesso: $.98 * 120 + .02 * 420 = 126\text{ns}$

Gerenciamento de Memória

Alocação de Páginas

- Quantas páginas reais serão alocadas a um processo;
- Duas estratégias:
 - Alocação fixa ou estática: cada processo tem um número máximo de páginas reais, definido quando o processo é criado;
 - O limite pode ser igual para todos os processos;
 - Vantagem: simplicidade;
 - Desvantagens:
 - (i) número muito pequeno de páginas reais pode causar muita paginação;
 - (ii) número muito grande de páginas reais causa desperdício de memória principal;

Gerenciamento de Memória

Alocação de Páginas

- Alocação variável ou dinâmica: número máximo de páginas reais alocadas ao processo varia durante sua execução;
 - Vantagem:
 - (i) processos com elevada taxa de paginação podem ter seu limite de páginas reais ampliado;
 - (ii) processos com baixa taxa de paginação podem ter seu limite de páginas reais reduzido;
 - Desvantagem: monitoramento constante;

Gerenciamento de Memória

Busca de Página

- **Paginação simples:**
 - Todas as páginas virtuais do processo são carregadas para a memória principal;
 - Assim, sempre todas as páginas são válidas;
- **Paginação por demanda (*Demand Paging*):**
 - Apenas as páginas efetivamente acessadas pelo processo são carregadas na memória principal;
 - Quais páginas virtuais foram carregadas → Bit de controle (bit de residência);
 - Página inválida;

Gerenciamento de Memória

Busca de Página

- Página inválida: MMU gera uma interrupção de proteção e aciona o sistema operacional;
 - Se a página está fora do espaço de endereçamento do processo, o processo é abortado;
 - Se a página ainda não foi carregada na memória principal, ocorre uma **falta de página** (*page fault*);

Gerenciamento de Memória

Busca de Página

- Falta de Página:
 - Processo é suspenso e seu descritor é inserido em uma **fila especial** – fila dos processos esperando uma página virtual;
 - Uma página real livre deve ser alocada;
 - A página virtual acessada deve ser localizada no disco;
 - Operação de leitura de disco, indicando o endereço da página virtual no disco e o endereço da página real alocada;

Gerenciamento de Memória

Busca de Página

- Após a leitura do disco:
 - Tabela de páginas do processo é corrigida para indicar que a página virtual agora está válida e está na página real alocada;
 - *Pager*: carrega páginas específicas de um processo do disco para a memória principal;
 - O descritor do processo é retirado da **fila especial** e colocado na fila do processador;

Gerenciamento de Memória

Substituição de Páginas

- Política de Substituição Local: páginas dos próprios processos são utilizadas na troca;
 - Dificuldade: definir quantas páginas cada processo pode utilizar;
- Política de Substituição Global: páginas de todos os processos são utilizadas na troca;
 - Problema: processos com menor prioridade podem ter um número muito reduzido de páginas, e com isso, acontecem muitas faltas de páginas;

Gerenciamento de Memória

Substituição de Páginas

- Algoritmos de substituição local alocam uma fração fixa de memória para cada processo; enquanto que algoritmos de substituição global alocam molduras de páginas entre os processos em execução, variando o número de páginas no tempo;

Gerenciamento de Memória

Substituição de Páginas

Algoritmos de substituição de páginas

- A falta de página força uma escolha
 - Qual página deve ser removida?
 - Alocação de espaço para a página a ser trazida para a memória
- A página modificada deve primeiro ser salva
 - Se não tiver sido modificada é apenas sobreposta
- Melhor não escolher uma página que está sendo muito usada
 - Provavelmente precisará ser trazida de volta logo.

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmos:
 - Ótimo;
 - NRU (Não usada recentemente);
 - FIFO;
 - Segunda Chance;
 - Relógio;
 - LRU;
 - *Working set*;
 - *WSClock*;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmo Ótimo:
 - Retira da memória a página que tem menos chance de ser referenciada;
 - Praticamente impossível de se saber;
 - Impraticável;
 - Usado em simulações para comparação com outros algoritmos;

Gerenciamento de Memória

Troca de Páginas - Paginação

- **Algoritmo *Not Recently Used Page Replacement* (NRU)** → troca as páginas não utilizadas recentemente:
 - 02 bits associados a cada página → R e M
 - Classe 0 → não referenciada, não modificada;
 - Classe 1 → não referenciada, modificada;
 - Classe 2 → referenciada, não modificada;
 - Classe 3 → referenciada, modificada;
 - R e M são atualizados a cada referência à memória;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **NRU:**

- Periodicamente, o bit R é limpo para diferenciar as páginas que não foram referenciadas recentemente;
 - A cada *tick* do relógio ou interrupção de relógio;
 - Classe 3 → Classe 1;
- Vantagens: fácil de entender, eficiente para implementar e fornece bom desempenho;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Algoritmo *First-in First-out Page Replacement* (FIFO)**
 - SO mantém uma lista das páginas correntes na memória;
 - A página no início da lista é a mais antiga e a página no final da lista é a mais nova;
 - Simples, mas pode ser ineficiente, pois uma página que está em uso constante pode ser retirada;
 - Pouco utilizado;

Gerenciamento de Memória

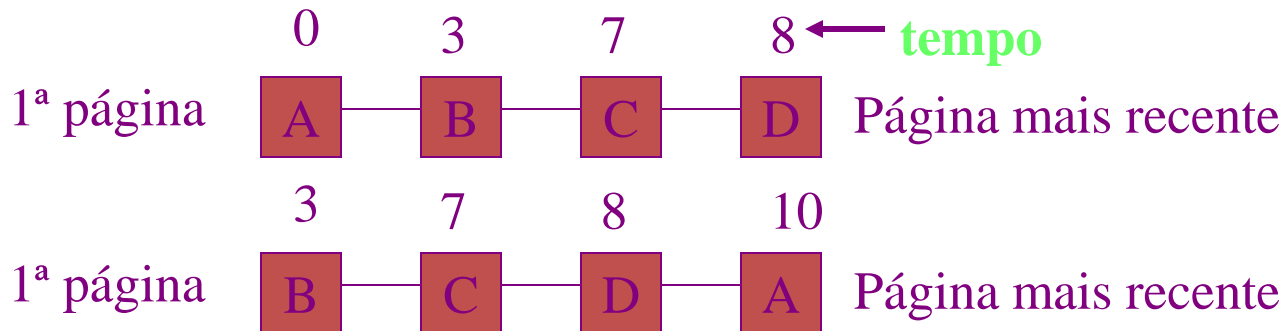
Substituição de Páginas - Paginação

• Algoritmo da Segunda Chance

- FIFO + *bit* R;
- Página mais velha é candidata em potencial;

Se o bit $R=0$, então página é retirada da memória, senão, $R=0$

e se dá uma nova chance à página colocando-a no final da lista;



Se página A com $R=1$; e falta de página em tempo 10; Então $R=0$ e página A vai para final da lista;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Algoritmo do Relógio**
 - Lista circular com ponteiro apontando para a página mais antiga
 - Algoritmo se repete até encontrar $R=0$;

Se $R=0$

- troca de página
- desloca o ponteiro

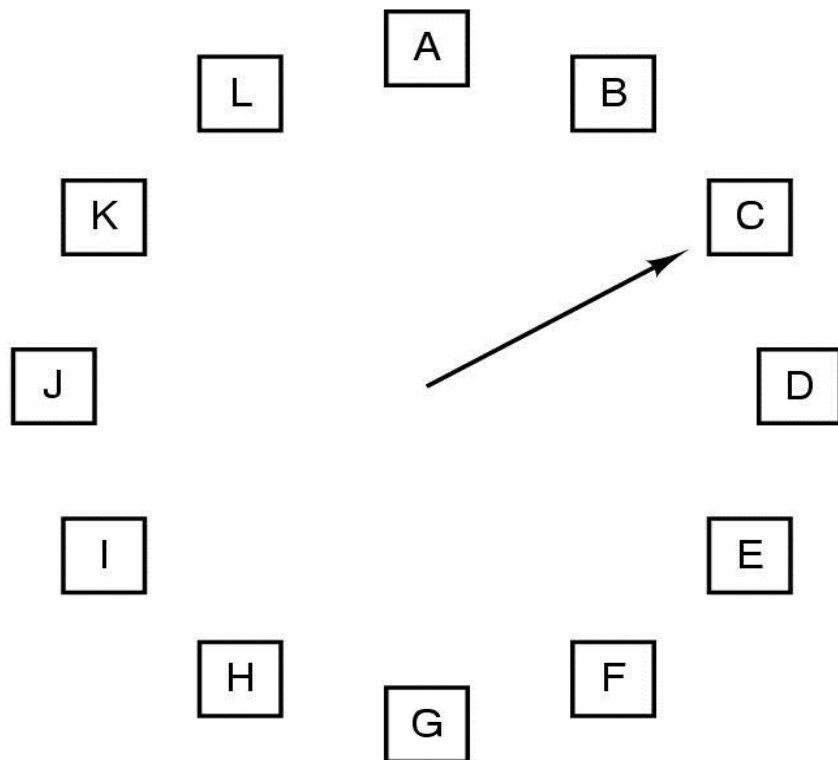
Se $R=1$

- $R = 0$
- desloca o ponteiro
- continua busca

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmo do Relógio



When a page fault occurs, the page the hand is pointing to is inspected. The action taken depends on the R bit:

R = 0: Evict the page

R = 1: Clear R and advance hand

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmo *Least Recently Used Page Replacement* (LRU)
 - Troca a página menos referenciada/modificada recentemente;
 - Alto custo
 - Lista encadeada com as páginas que estão na memória, com as mais recentemente utilizadas no início e as menos utilizadas no final;
 - A lista deve ser atualizada a cada referência da memória;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmo *Least Recently Used Page Replacement* (LRU)
 - Pode ser implementado tanto por hardware quanto por software:
 - Hardware: MMU deve suportar a implementação LRU;
 - Contador em hardware (64 *bits*);
 - Tabela de páginas armazena o valor desse contador para saber quantas vezes a página foi usada;
 - Software: duas maneiras
 - NFU (*Not frequently used*);
 - *Aging* (Envelhecimento);

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Software: NFU**

- Para cada página existe um contador → iniciado com zero e incrementado a cada referência à pagina;
- Página com menor valor do contador é candidata a troca;
- Como esse algoritmo não se esquece de nada
 - Problema: pode retirar páginas que estão sendo referenciadas com frequência;
 - » Compilador com vários passos: passo 1 tem mais tempo de execução que os outros passos → páginas do passo 1 terão mais referências armazenadas;

Gerenciamento de Memória

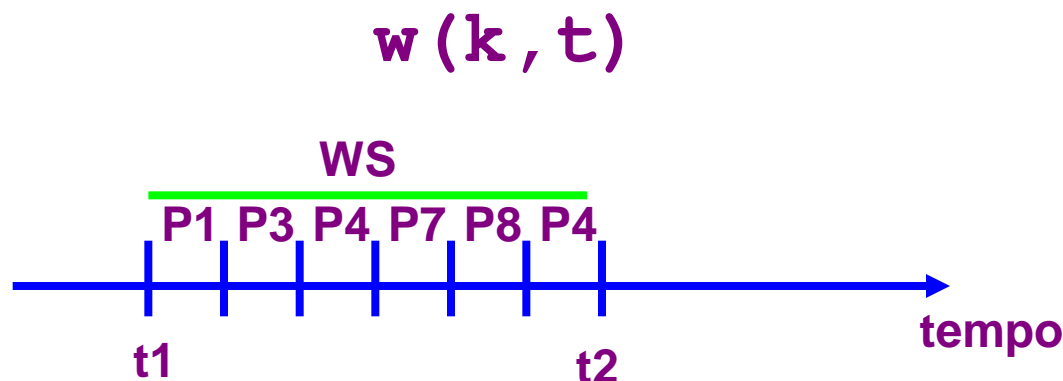
Substituição de Páginas - Paginação

- **Software: Algoritmo *aging***
 - Modificação do NFU, resolvendo o problema descrito anteriormente;
 - Além de saber quantas vezes a página foi referenciada, também controla quando ela foi referenciada;
 - Geralmente, 8 bits são suficientes para o controle se as interrupções de relógio (*clock ticks*) ocorrem a cada 20ms (10^{-3});

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Algoritmo *Working Set* (WS):**
 - Paginação por demanda → páginas são carregadas na memória somente quando são necessárias;
 - Pré-paginação → *Working set*
 - Conjunto de páginas que um processo está efetivamente utilizando (referenciando) em um determinado tempo t ;



Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Algoritmo *Working Set* (WS):**
 - Objetivo principal: reduzir a falta de páginas
 - Um processo só é executado quando todas as páginas necessárias no tempo t estão carregadas na memória;
 - SO gerencia quais páginas estão no *Working Set*;
 - Para simplificar \rightarrow o *working set* pode ser visto como o conjunto de páginas que o processo referenciou durante os últimos t segundos de tempo;
 - Utiliza *bit* R e o tempo de relógio (tempo virtual) da última vez que a página foi referenciada;

Gerenciamento de Memória

Substituição de Páginas - Paginação

■ Algoritmo *Working Set*:

Tempo virtual atual (CVT): 2204
 $age = CVT - TLU$
 (Ex.: $2204 - 2084 = 120$)
 $\tau = \text{múltiplos } clock\ ticks$

* Se todas as páginas estiverem com $R=1$, uma página é escolhida
 Randomicamente;
 ** Se todas as páginas estiverem no WS, a página mais velha com $R=0$ é escolhida;

		Bit R
Tempo do último		
Uso (TLU)	2084	1
	2003	1
	1980	1
	1213	0
	2014	1
	2020	1
	2032	1
	1620	0

Tabela de Páginas

Percorrer as páginas examinando bit R;
 Se ($R==1$)*
 página foi referenciada;
 faz TLU da página igual ao CVT;
 Se ($R==0$ e $age > \tau$)
 página não está no *working set*;
 remove a página;
 Se ($R==0$ e $age \leq \tau$) **
 página está no *working set*;
 guarda página com maior *age*;

Gerenciamento de Memória

Substituição de Páginas - Paginação

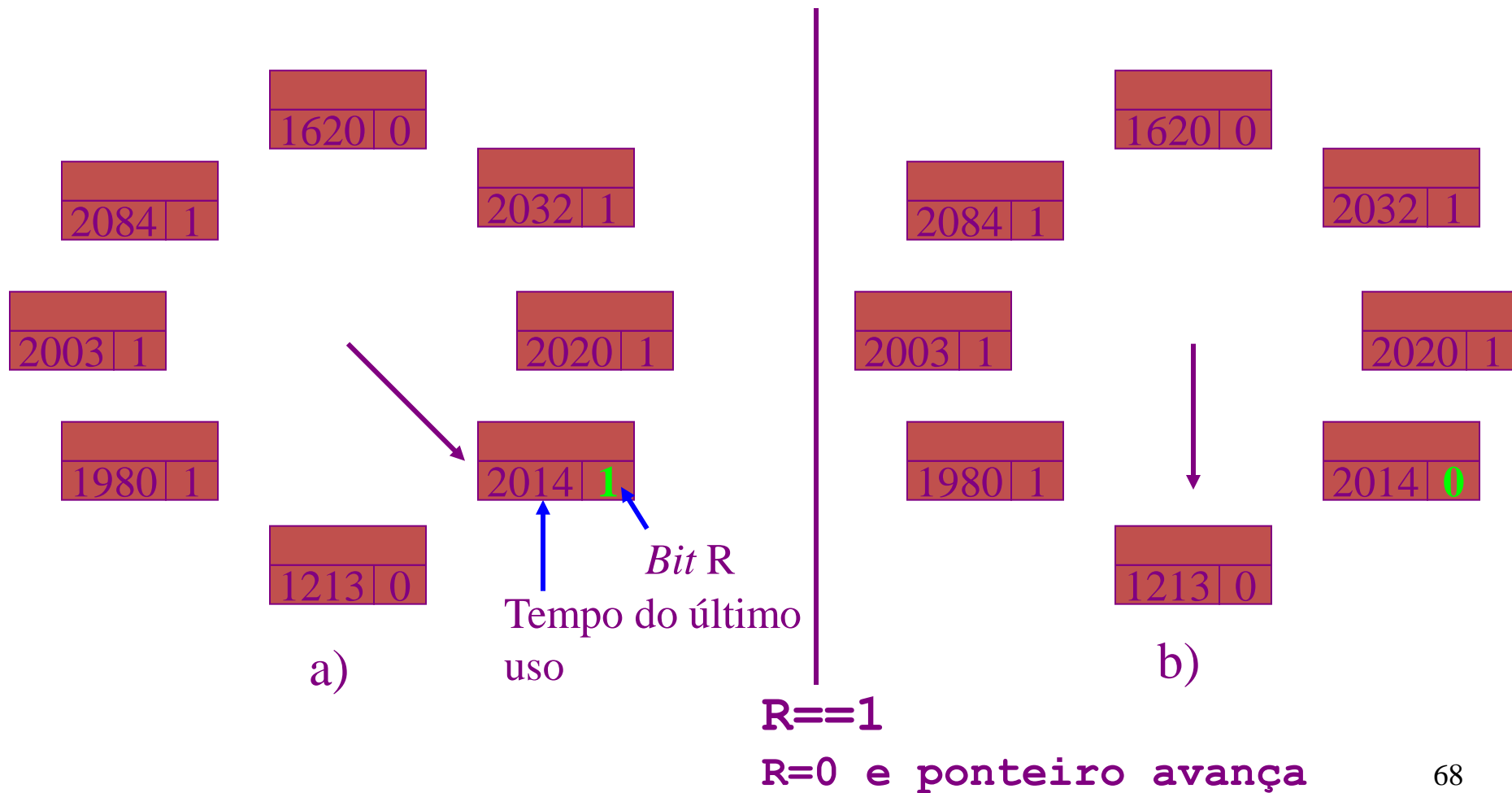
- Algoritmo *WSClock*:
 - *Clock + Working Set*;
 - Lista circular de páginas formando um anel a cada página carregada na memória;
 - Utiliza *bit R* e o tempo da última vez que a página foi referenciada;
 - *Bit M* utilizado para agendar escrita em disco;

Gerenciamento de Memória

Substituição de Páginas - Paginação

Algoritmo *WSClock*:

Tempo virtual atual: 2204

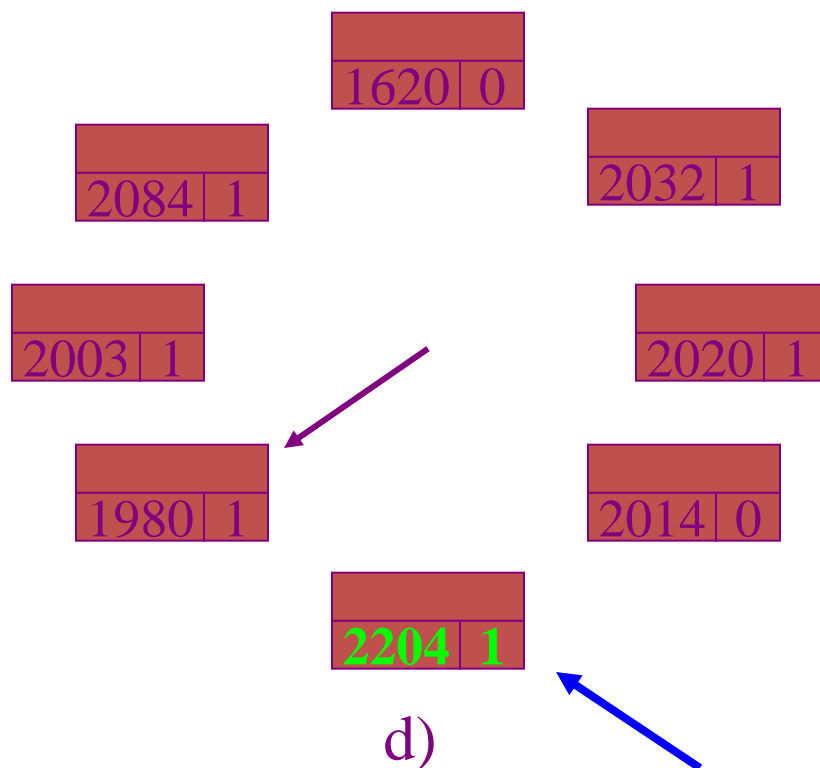
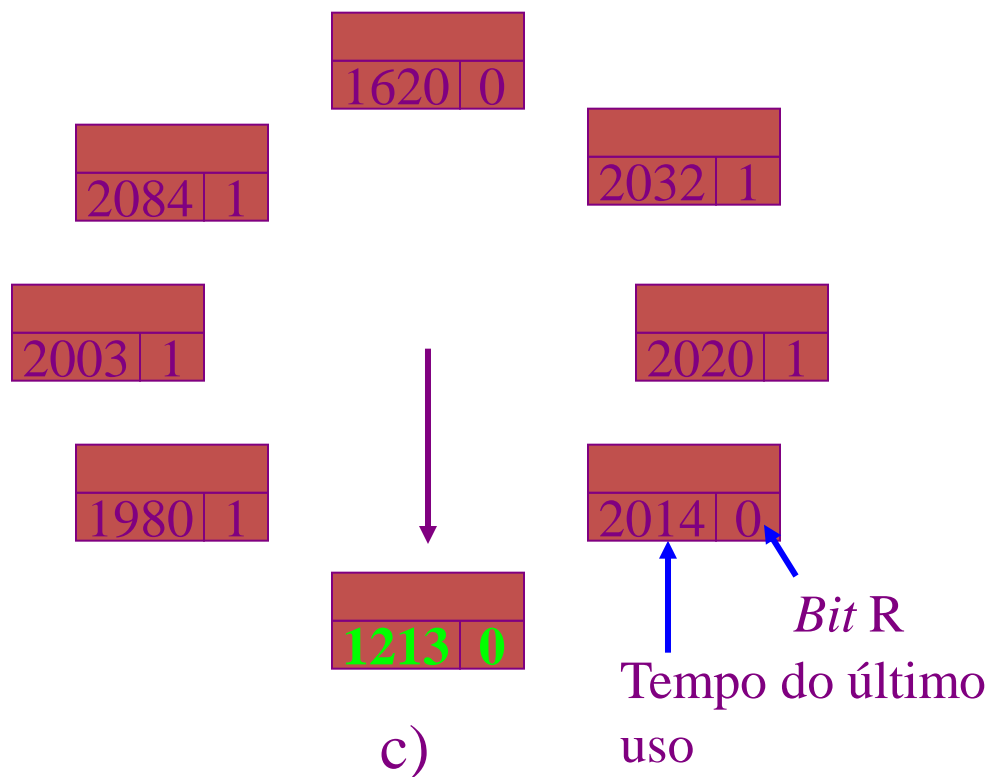


Gerenciamento de Memória

Substituição de Páginas - Paginação

Algoritmo WSClock:

Tempo virtual atual: 2204



$R == 0$ e $age > t$

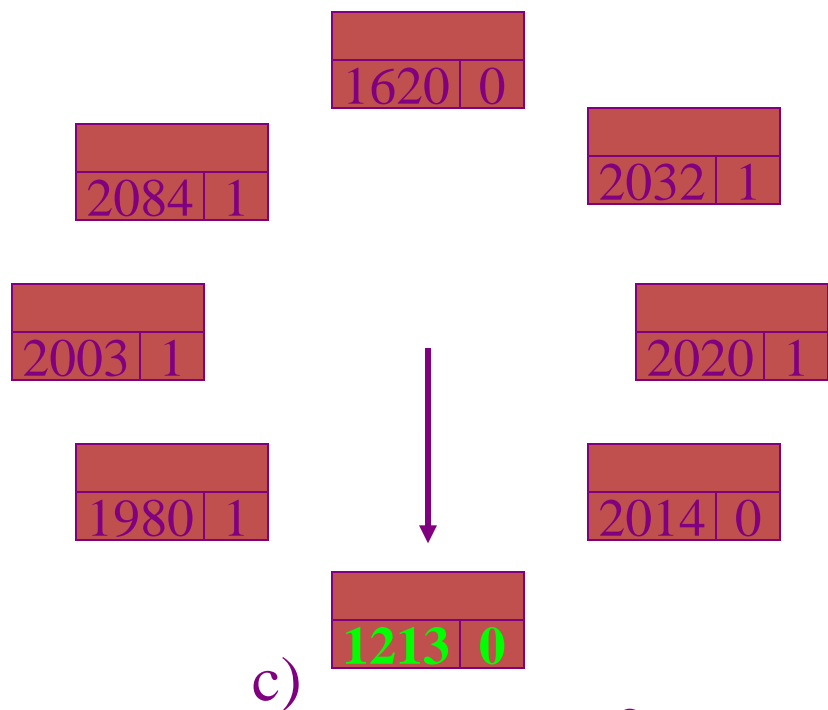
$M == 0$ (não agenda escrita) \rightarrow troca

Nova página

Gerenciamento de Memória

Substituição de Páginas - Paginação

 Algoritmo *WSClock*:

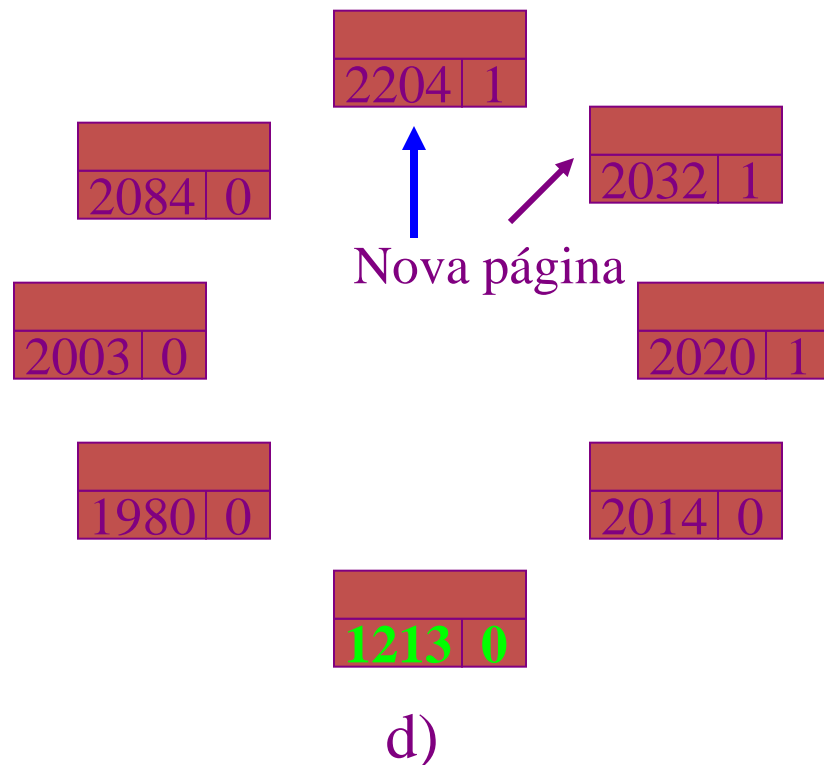


c)

$R == 0$ e $age > t$

$M == 1$ (agenda escrita e continua procura)

Tempo virtual atual: 2204



d)

Gerenciamento de Memória

Substituição de Páginas - Paginação

- **Algoritmo *WSClock*:**
 - Se todas estiverem com $M=1$; então escreve página atual no disco, e troca a página;
 - Melhor desempenho → menos acessos ao disco;

Gerenciamento de Memória

Substituição de Páginas - Paginação

- Algoritmos de substituição local:
 - *Working Set*;
 - *WSClock*;
- Algoritmos de substituição local/global:
 - Ótimo;
 - NRU;
 - FIFO;
 - Segunda Chance;
 - LRU;
 - Relógio;

- *Capítulo 3 - Tanenbaum*