

Sistemas Operacionais

Gerenciamento de Memória Virtual Algoritmos de Paginação

Norton Trevisan Roman
Marcelo Morandini
Jó Ueyama

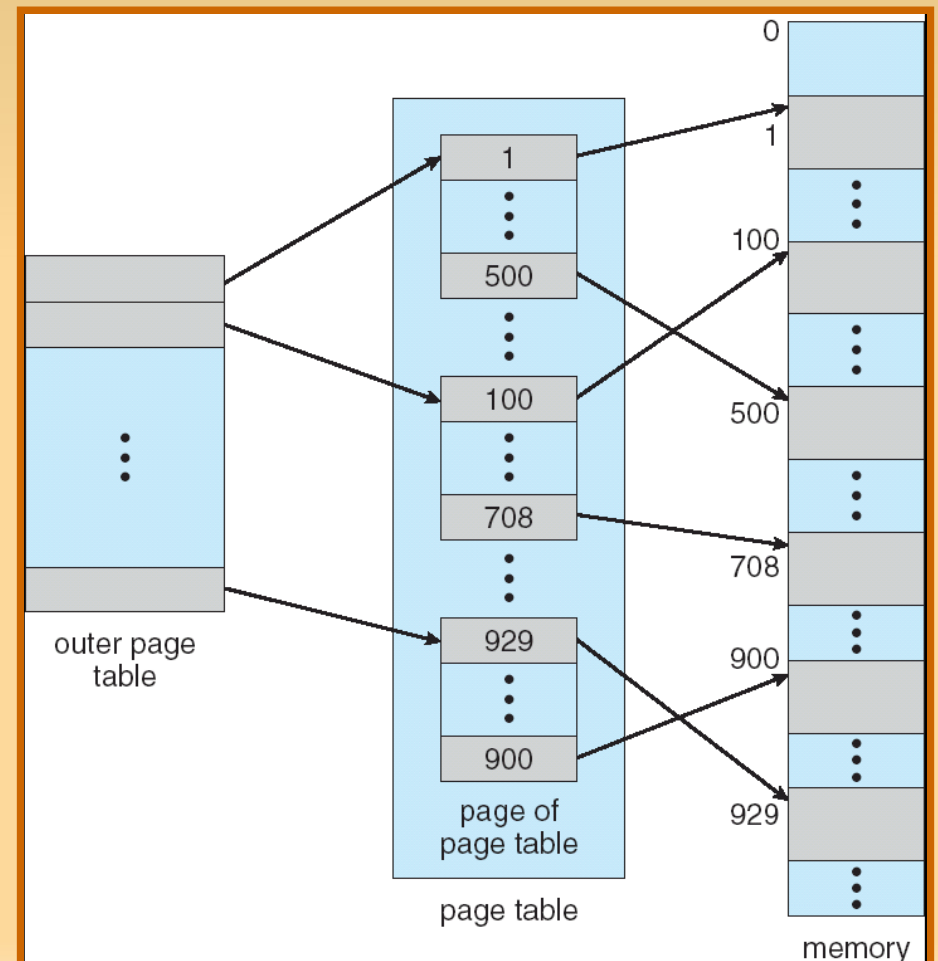
Apostila baseada nos trabalhos de Kalinka Castelo Branco, Antônio Carlos Sementille, Luciana A. F. Martimiano e nas transparências fornecidas no site de compra do livro "Sistemas Operacionais Modernos"

Tabelas de Páginas

- TLBs aceleram a tradução entre endereços virtuais e reais
- Mas esse é apenas um problema → como traduzir rapidamente
- Resta ainda como lidar com grandes espaços de endereços virtuais → Como organizar a tabela de páginas?
 - Paginação hierárquica (multi-nível)
 - Tabelas de página com hash
 - Tabelas de página invertidas

Paginação Multi-Nível

- Quebre o espaço de endereço lógico em múltiplas tabelas de página
 - Uma técnica simples é uma tabela de página em dois níveis



Paginação em Dois Níveis

- A idéia é evitar manter na memória todas as tabelas de página o tempo todo
 - Apenas as necessárias devem estar na memória
 - Ex.
 - Suponha que um programa esteja dividido em:
 - Uma tabela para o código (4MB da base da memória)
 - Uma para os dados (4MB seguintes)
 - Uma para a pilha (4MB do topo)
 - Há um buraco gigante vazio, na tabela (usou apenas 3 páginas)

Paginação em Dois Níveis

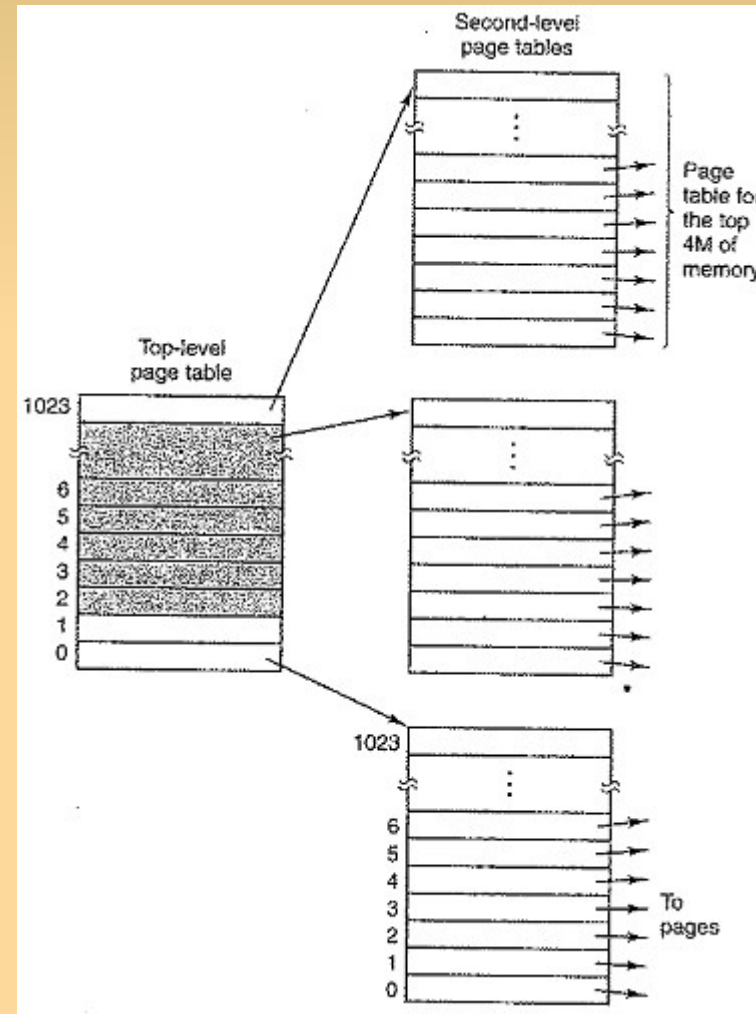
- A idéia é evitar manter na memória todas as tabelas de página o tempo todo
 - Ex.
 - Podemos dividir o programa em:
 - Uma tabela para o código (4MB da base da memória)
 - Uma para os dados (4MB seguintes)
 - Uma para a pilha (4MB do topo)
 - Há um buraco gigante vazio, na tabela (usou apenas 3 páginas)
 - Mesmo que a tabela inteira seja enorme, para esse programa precisaríamos apenas manter 4 (a principal, e essas outras 3)
 - As demais (possivelmente não usadas), não estariam na memória.

Paginação em Dois Níveis

Tabela Única: 20 bits
→ 1M entradas

X
X
X
X
7
X
5
X
X
X
3
4
0
6
1
2

Duas Tabelas:
Principal: 10 bits → 1K entradas
Secundárias: Ad hoc

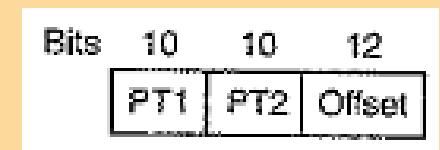


Em nosso exemplo, teríamos mais 3 páginas de 10 bits → 3 x 1K.

O tamanho total ocupado seria 4K

Paginação em Dois Níveis

- Endereçamento:
 - Um endereço lógico (em máquinas de 32 bits) é dividido em:
 - um número de página contendo 20 bits
 - um deslocamento de página contendo 12 bits
 - Como a tabela de página é paginada, o número de página é dividido ainda em:
 - um número de página PT1, de 10 bits
 - um número de página PT2, de 10 bits
- PT1 é um índice para a tabela de página mais externa, e PT2 é o deslocamento da página dentro da tabela de página mais externa

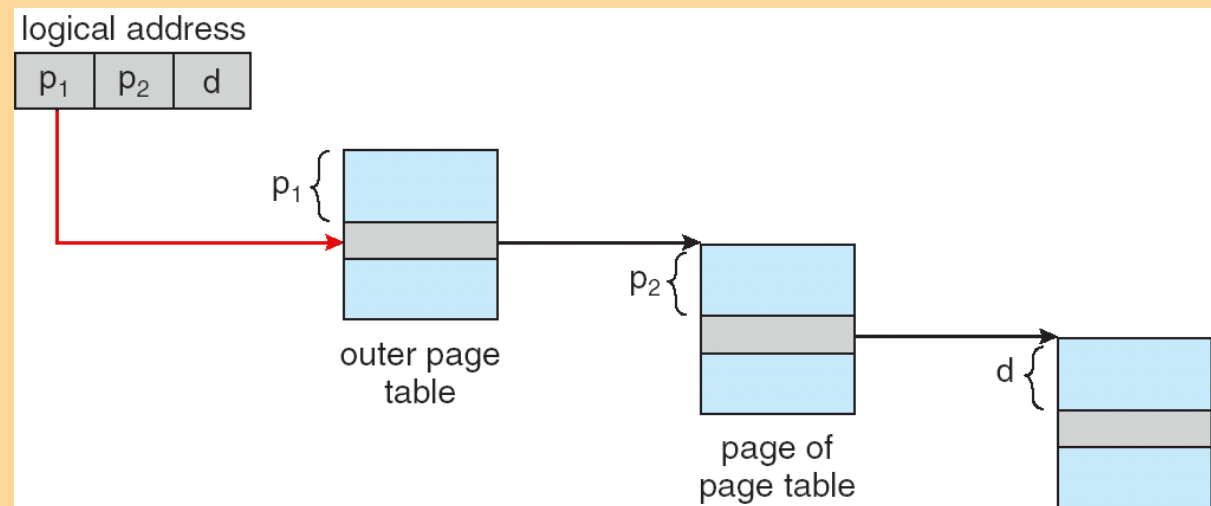


Paginação em Dois Níveis

- Uma vez que o offset é 12 bits
 - As páginas têm 4KB (o máximo em 12 bits)
- Como temos $32 - 12 = 20$ bits para endereçar páginas:
 - Temos $2^{20} = 1\text{M}$ possíveis páginas

- Tradução de endereço:

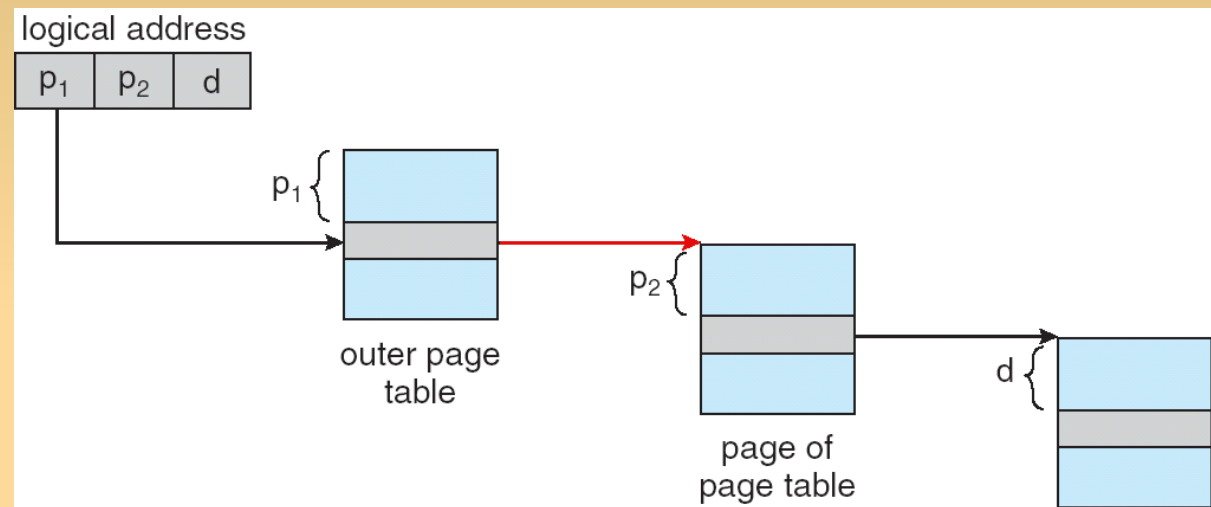
- A MMU extrai o campo PT1, usando seu valor como índice na página mais alta



Paginação em Dois Níveis

- Tradução de endereço:

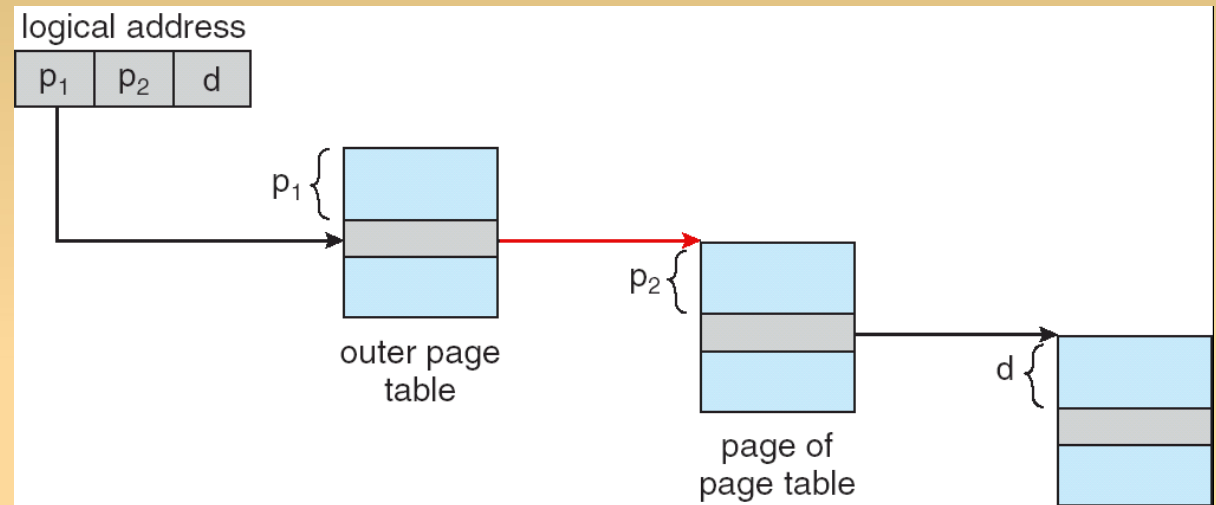
- A entrada nessa posição dá o endereço (ou número da moldura) da tabela de página no segundo nível



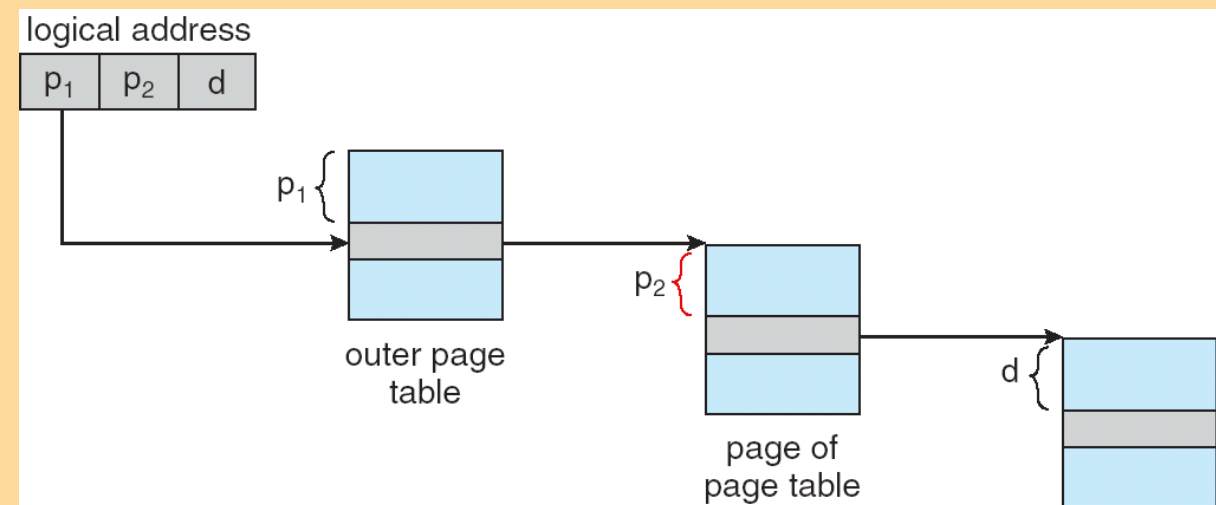
Paginação em Dois Níveis

- Tradução de endereço:

- A entrada nessa posição dá o endereço (ou número da moldura) da tabela de página no segundo nível



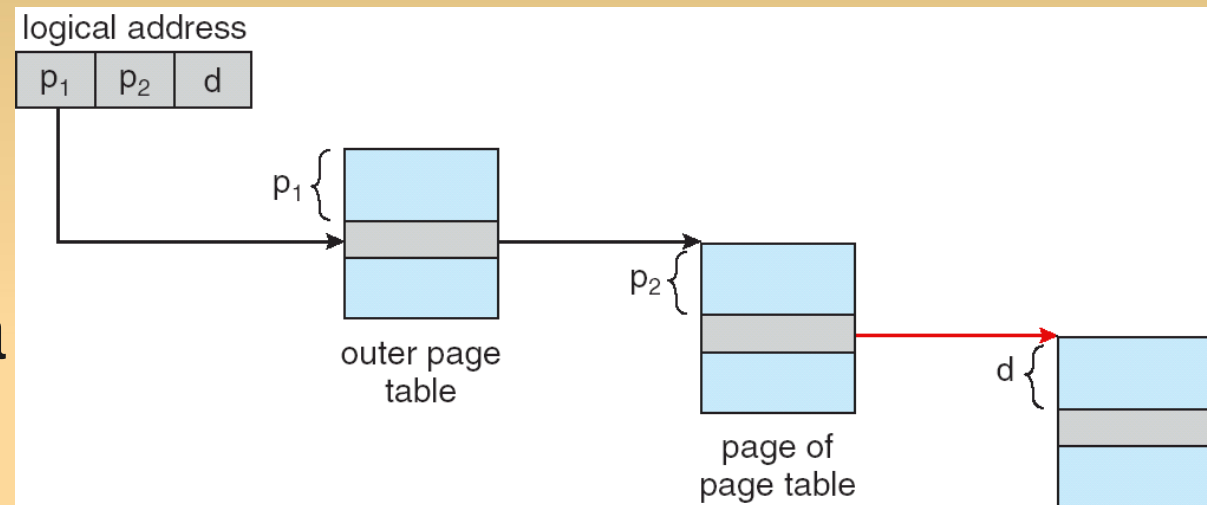
- O campo PT2 é então usado como um índice na página de segundo nível



Paginação em Dois Níveis

- Tradução de endereço:

- O número da moldura física é obtido dessa entrada na tabela
 - Checa o bit de residência

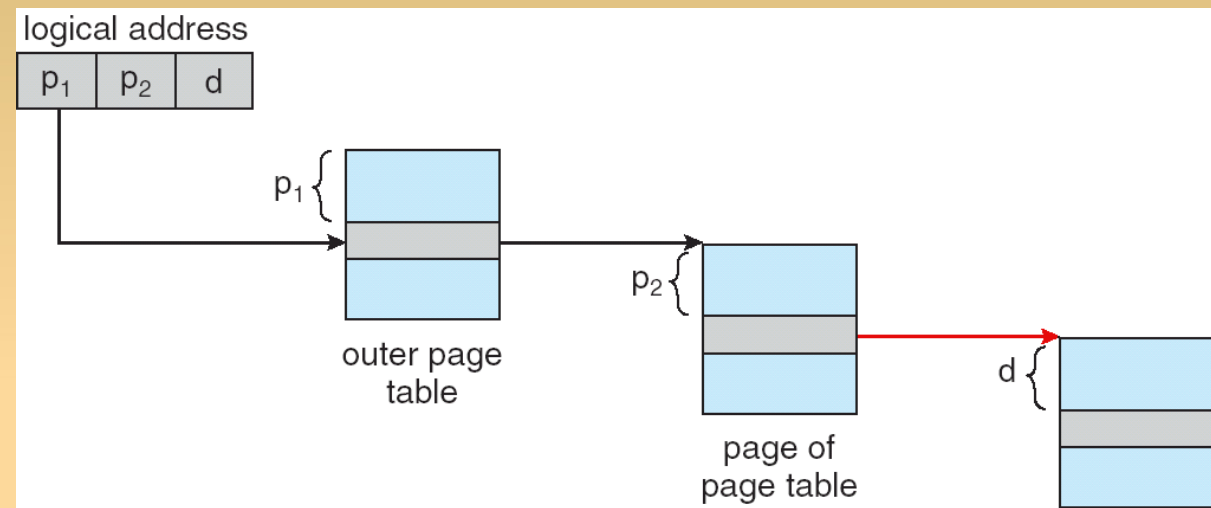


Paginação em Dois Níveis

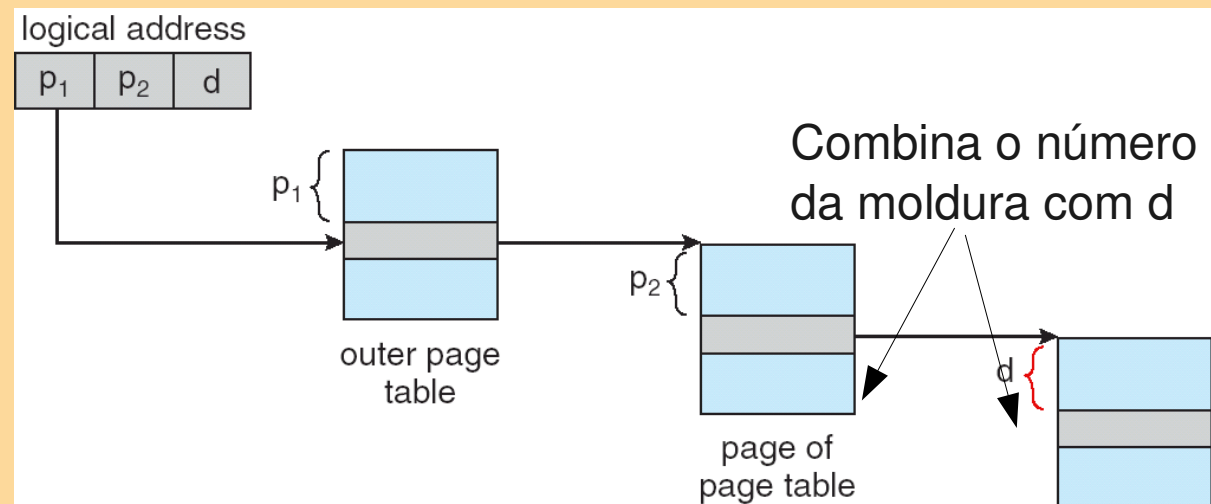
- Tradução de endereço:

- O número da moldura física é obtido dessa entrada na tabela

- Checa o bit de residência



- Finalmente o deslocamento é usado para compor o endereço físico



Paginação em Três Níveis

- Mais níveis podem existir

- Dois:

outer page	inner page	offset
p_1	p_2	d
42	10	12

- Três:

2nd outer page	outer page	inner page	offset
p_1	p_2	p_3	d
32	10	10	12

- Na prática, não se usa mais que 3

Tabelas de Página em Hash

- Comuns em espaços de endereço > 32 bits
- O número de página virtual é usado para cálculo do valor de hash
- Cada entrada na tabela contém uma lista ligada de elementos, consistindo de:
 - O número da página virtual
 - O valor da moldura
 - Um ponteiro para o próximo elemento da lista

Tabelas de Página em Hash

- Funcionamento:
 - O endereço de página virtual é usado para cálculo do hash
 - Esse endereço é comparado a cada elemento
 - Se for encontrado, o campo da moldura é usado no endereço físico

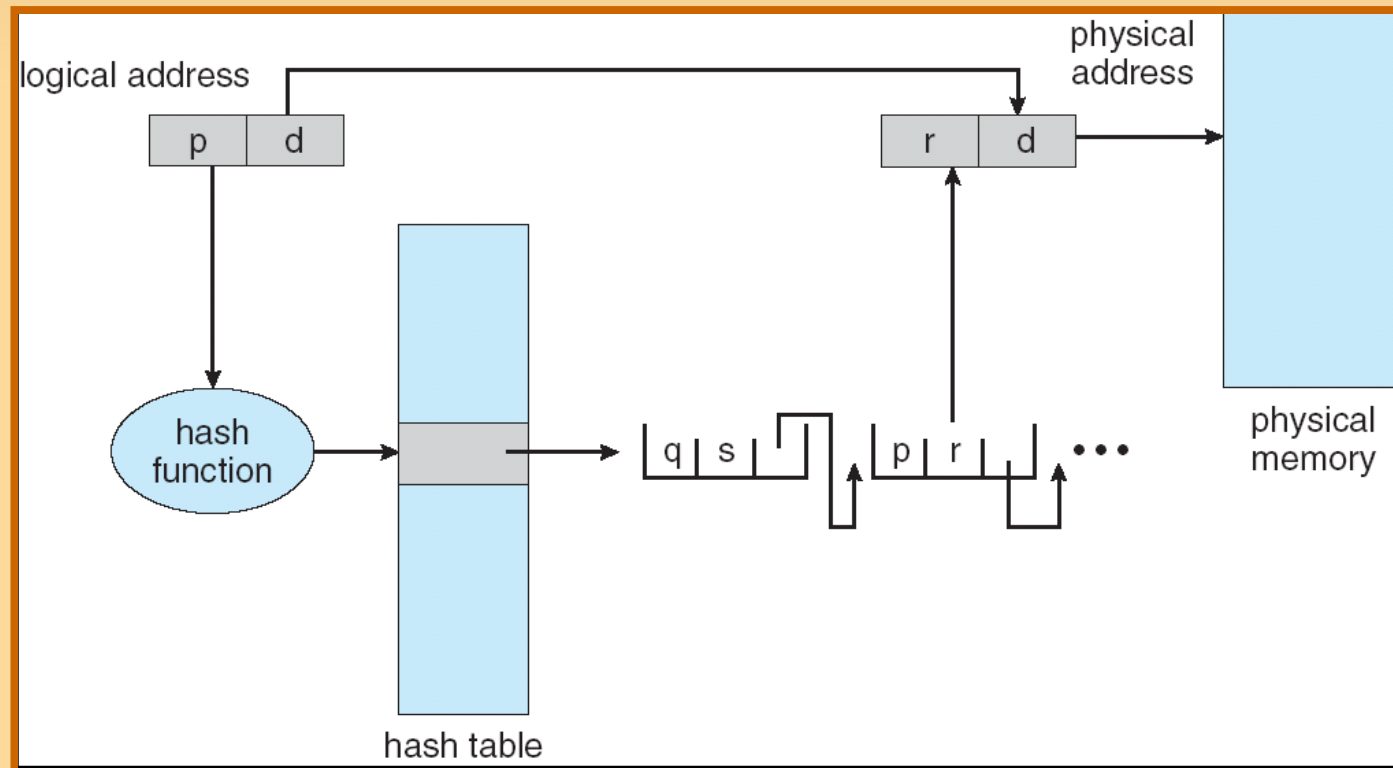


Tabela de Página Invertida

- Com 32 bits, multi-nível funciona bem
- E com 64?
 - Se a página tiver 4K (2^{12}), teremos uma tabela com $2^{64} / 2^{12} = 2^{64-12} = 2^{52}$ entradas
 - Se cada entrada tiver 8 bytes, a tabela terá mais de 30.000.000GB
 - 30PB (peta bytes)
 - Provavelmente, não é uma boa idéia colocá-la na RAM
- Solução:
 - Tabela de página invertida

Tabela de Página Invertida

- Possui uma entrada por moldura na memória física
 - Em vez de uma entrada por página no espaço virtual
 - A entrada registra que página virtual esta localizada na moldura
 - Com informações sobre o processo que possui essa página
- Poupam muito espaço quando o espaço virtual é muito maior que o físico

Tabela de Página Invertida

- Modelo de única página (ou multi-nível)
 - Cada processo tem uma tabela de páginas associada a ele → classificação feita pelo endereço virtual;
- Modelo de página invertida:
 - SO mantém uma única tabela para as molduras de páginas da memória;
 - Cada entrada consiste no endereço virtual da página armazenada naquela página real, com informações sobre o processo dono da página virtual;

Tabela de Página Invertida

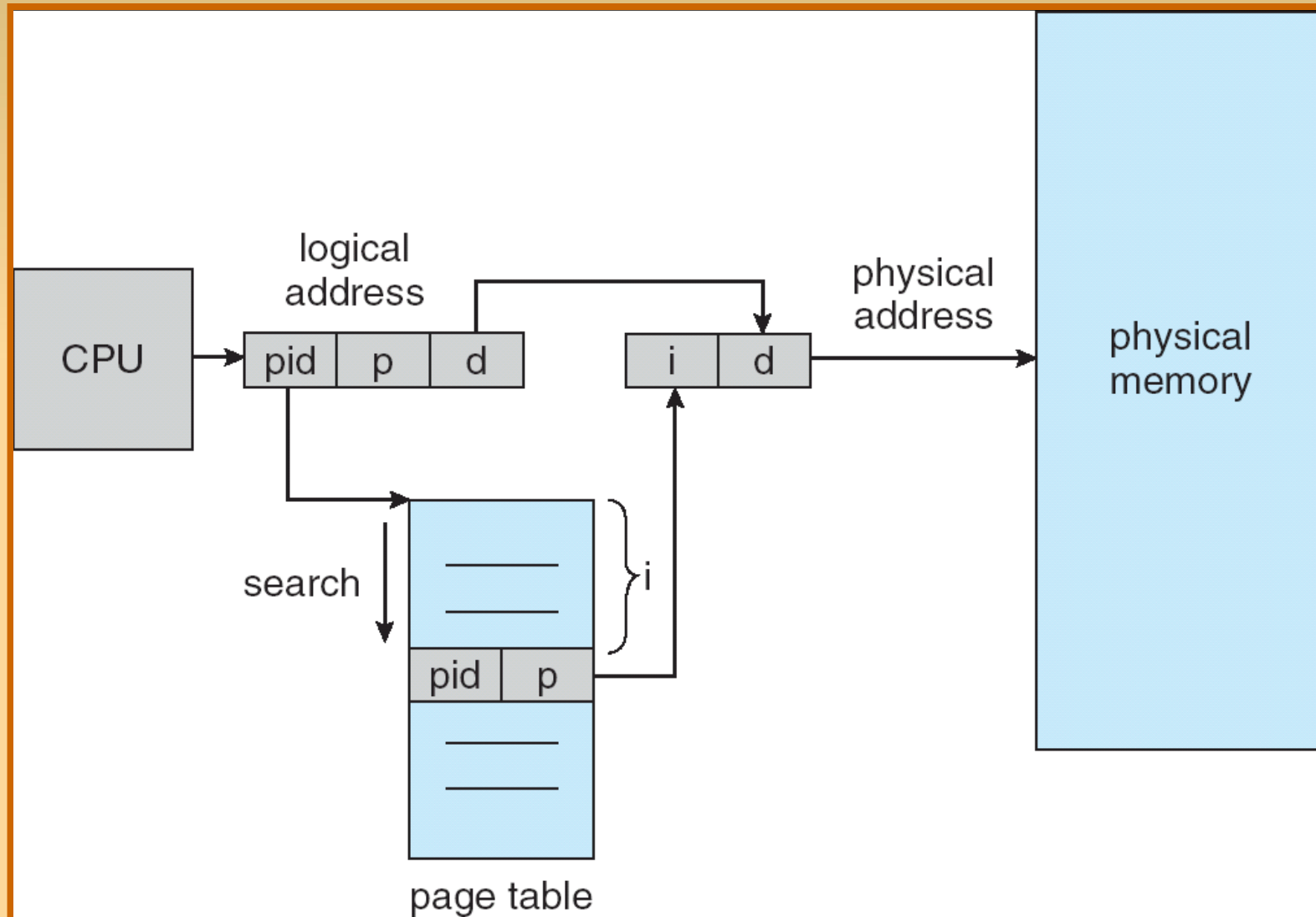
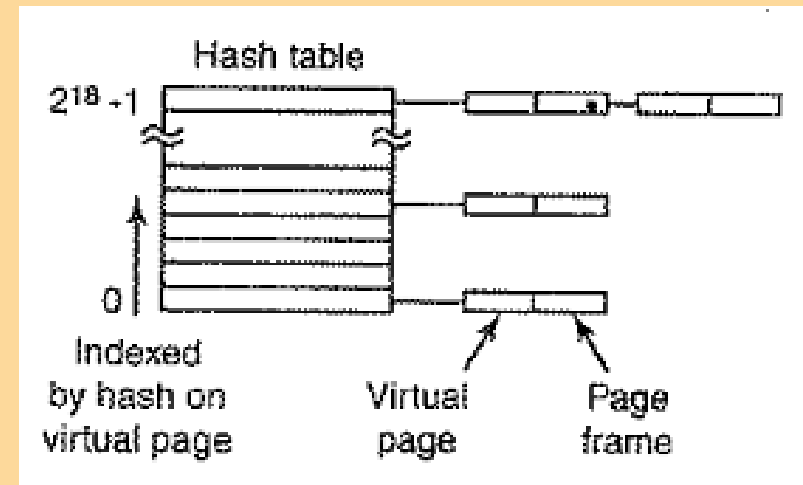


Tabela de Página Invertida

- Problema:
 - Traduções virtual → física é muito mais difícil
 - Quando o processo n referencia a página virtual p , o hardware não consegue mais encontrar a página física usando p como índice
 - Deve buscar em toda a tabela de páginas por uma entrada (n,p)
 - Se encontrar, gera o endereço físico
 - Faz isso em cada referência à memória → aumenta o tempo necessário para pesquisar a tabela
 - O que fazer?
 - Usar a TLB para guardar as mais acessadas
 - Caso a página buscada não esteja na TLB, devemos procurar em toda a tabela invertida

Tabela de Página Invertida

- Busca na tabela invertida:
 - Feita através de uma tabela de hash, com o valor de hash calculado a partir do endereço virtual
 - Todas as páginas virtuais atualmente na memória com o mesmo valor de hash são agrupadas
 - Uma vez encontrado o número de página buscado, o par (virtual, físico) é armazenado na TLB
- Páginas invertidas são bastante comuns em sistemas de 64 bits



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;

Alocação de Páginas

- Quantas páginas reais serão alocadas a um processo?
- Duas estratégias:
 - 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;

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):
 - Processos começam com nenhuma página na memória
 - Assim que a CPU tenta executar a primeira instrução, gera um page fault
 - O S.O. Traz a página que falta à memória

Busca de Página

- Paginação por demanda (Demand Paging):
 - Apenas as páginas efetivamente acessadas pelo processo são carregadas na memória principal;
 - Quais páginas virtuais foram carregadas?
 - Indicado pelo bit de residência
 - 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);

Busca de Página

- Paginação por demanda – 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;

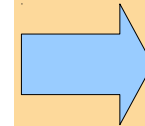
Busca de Página

- Paginação por demanda – Falta 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;
 - Processo executado pelo 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;

Troca de Páginas

- Política de Substituição Local:
 - Páginas dos próprios processos são utilizadas na troca;
 - Ex: page fault no processo A

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

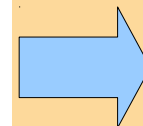


???

Troca de Páginas

- Política de Substituição Local:
 - Páginas dos próprios processos são utilizadas na troca;
 - Ex: page fault no processo A
 - Escolha a mais antiga dentre as páginas pertencentes a A
 - Aloca a cada processo fração fixa da memória

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6



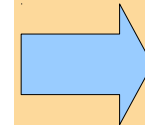
A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Troca de Páginas

- Política de Substituição Local:

- Dificuldade: definir quantas páginas cada processo pode utilizar
 - Se o processo usar menos que as previstas → desperdício
 - Se usar mais que o previsto (ou galgar mais memória enquanto executa) → muitas trocas, mesmo havendo páginas disponíveis

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

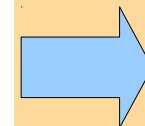


A0
A1
A2
A3
A4
A6
B0
B1
B2
B3
B4
B5
B6
C1
C2
C3

Troca de Páginas

- Política de Substituição Global:
 - Páginas de todos os processos são utilizadas na troca;
 - Ex: page fault no processo A

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6



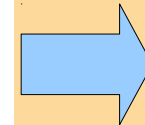
???

Troca de Páginas

- Política de Substituição Global:

- Páginas de todos os processos são utilizadas na troca;
 - Ex: page fault no processo A
 - Escolha a mais antiga dentre as páginas disponíveis
- Processos sem fração fixa da memória

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6

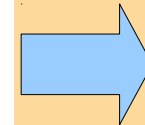


A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

Troca de Páginas

- Política de Substituição Global:
 - Problema: processos com menor prioridade podem ter um número muito reduzido de páginas, e com isso, acontecem muitas faltas de páginas;

	Age
A0	10
A1	7
A2	5
A3	4
A4	6
A5	3
B0	9
B1	4
B2	6
B3	2
B4	5
B5	6
B6	12
C1	3
C2	5
C3	6



A0
A1
A2
A3
A4
A5
B0
B1
B2
A6
B4
B5
B6
C1
C2
C3

Troca de Páginas

- Quando ocorre um page fault
 - Se todas as molduras estiverem ocupadas, o S.O. Escolhe uma página para eliminar – página vítima
 - A escolha pode ser aleatória
 - Será melhor escolher uma página que não seja muito usada
 - Se essa página foi modificada enquanto estava na memória, deve ser reescrita no disco
 - Se não tiver sido modificada, pode apenas ser sobrescrita

Troca de Páginas

- Exemplo:
 - Dois processos P1 e P2, cada um com 4 páginas virtuais;
 - Memória principal com 6 páginas;

Memória Virtual P1

0	A
1	B
2	C
3	D

Memória Virtual P2

0	E
1	F
2	G
3	H

Memória Principal

0	D
1	A
2	F
3	E
4	G
5	B

3 páginas de
cada processo

Troca de Páginas

Memória Virtual P1 Tabela de Páginas P1

0	A
1	B
2	C
3	D

Simplificada

0	1	v
1	5	v
2		i
3	0	v

Memória Principal

0	D
1	A
2	F
3	E
4	G
5	B

3 páginas de
cada processo

Memória Virtual P2 Tabela de Páginas P2

0	E
1	F
2	G
3	H

Simplificada

0	3	v
1	2	v
2	4	v
3		i

→ P2 tenta acessar página 3! Falta de Página!

Troca de Páginas

Memória Virtual P1

0	A
1	B
2	C
3	D

Tabela de Páginas P1

Simplificada

0	1	v
1	5	v
2		i
3	0	v

Memória Principal

0	D
1	A
2	F
3	E
4	H
5	B

3 páginas de cada processo

Memória Virtual P2

0	E
1	F
2	G
3	H

Tabela de Páginas P2

Simplificada

0	3	v
1	2	v
2		i
3	4	v

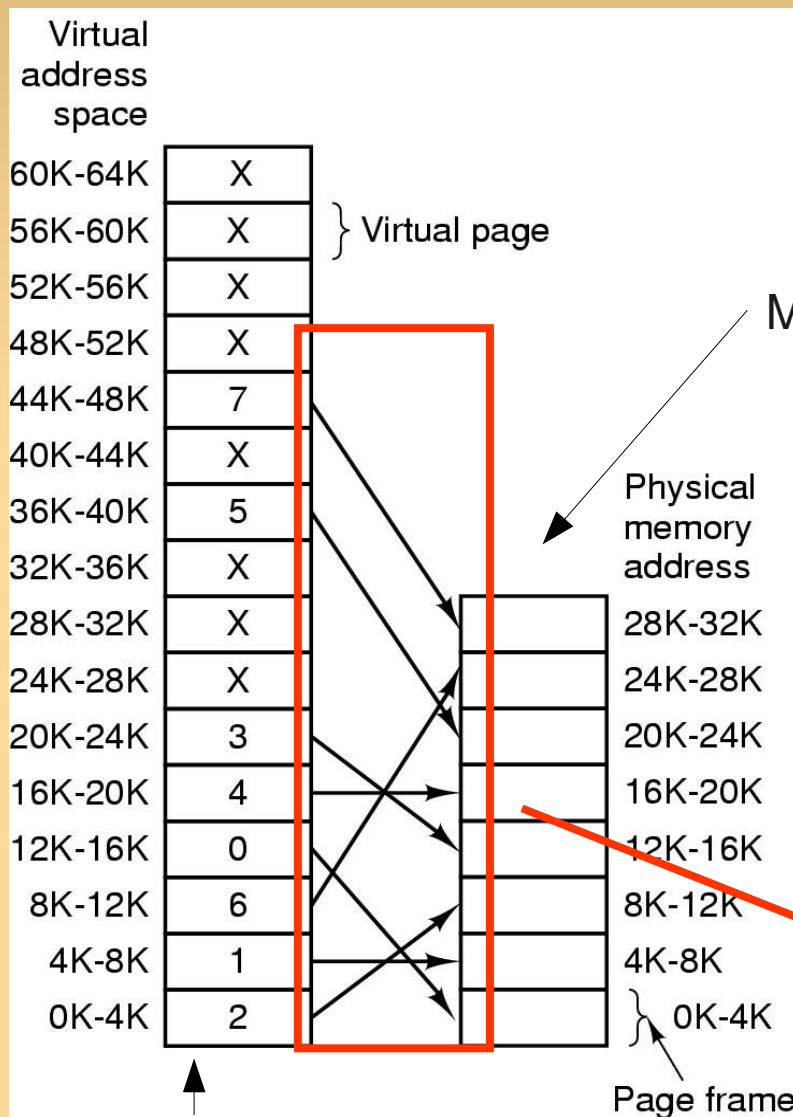
→ Página 2 (virtual) é escolhida como vítima!

Algoritmos de Troca de Página

- Ótimo;
- NRU;
- FIFO;
- Segunda Chance;
- Relógio;
- LRU;
- Working set;
- WSClock;
- Veremos cada um em detalhes

Memória Virtual

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



▪ Ex:

- Computador que gera 64K de endereços virtuais
 - Tem apenas 32K de endereços físicos
 - Programas de 64K podem ser escritos, mas não carregados inteiramente na memória

▪ MMU realiza o mapeamento

Endereços virtuais (gerados pelo computador)