

Eletrônica Digital

Gerência de Memória

Prof. Rômulo Calado Pantaleão Camara

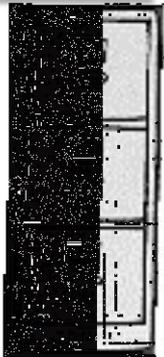
Carga Horária: 60h

Acessando documentos



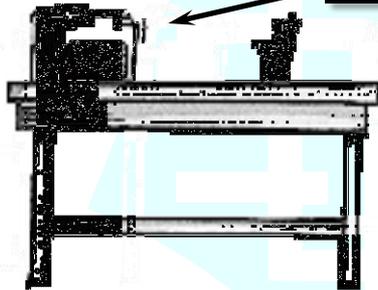
© Can Stock Photo - csp5800314

Fichário



Quadro

Pasta



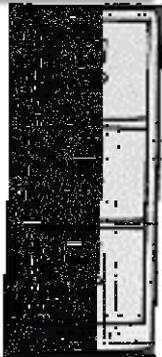
- ✓ Documento sendo processado é composto por várias pastas que estão no fichário.
- ✓ As pastas possuem uma certa quantidade de folhas numeradas sequencialmente.
- ✓ Cada folha é uma unidade de processamento.
- ✓ Endereço da folha que o funcionário procura:
 - Nr. pasta e nr. da folha

Acessando documentos



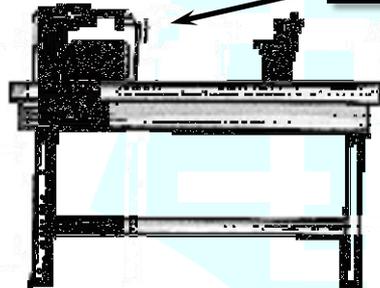
© Can Stock Photo - csp5800314

Fichário



Quadro

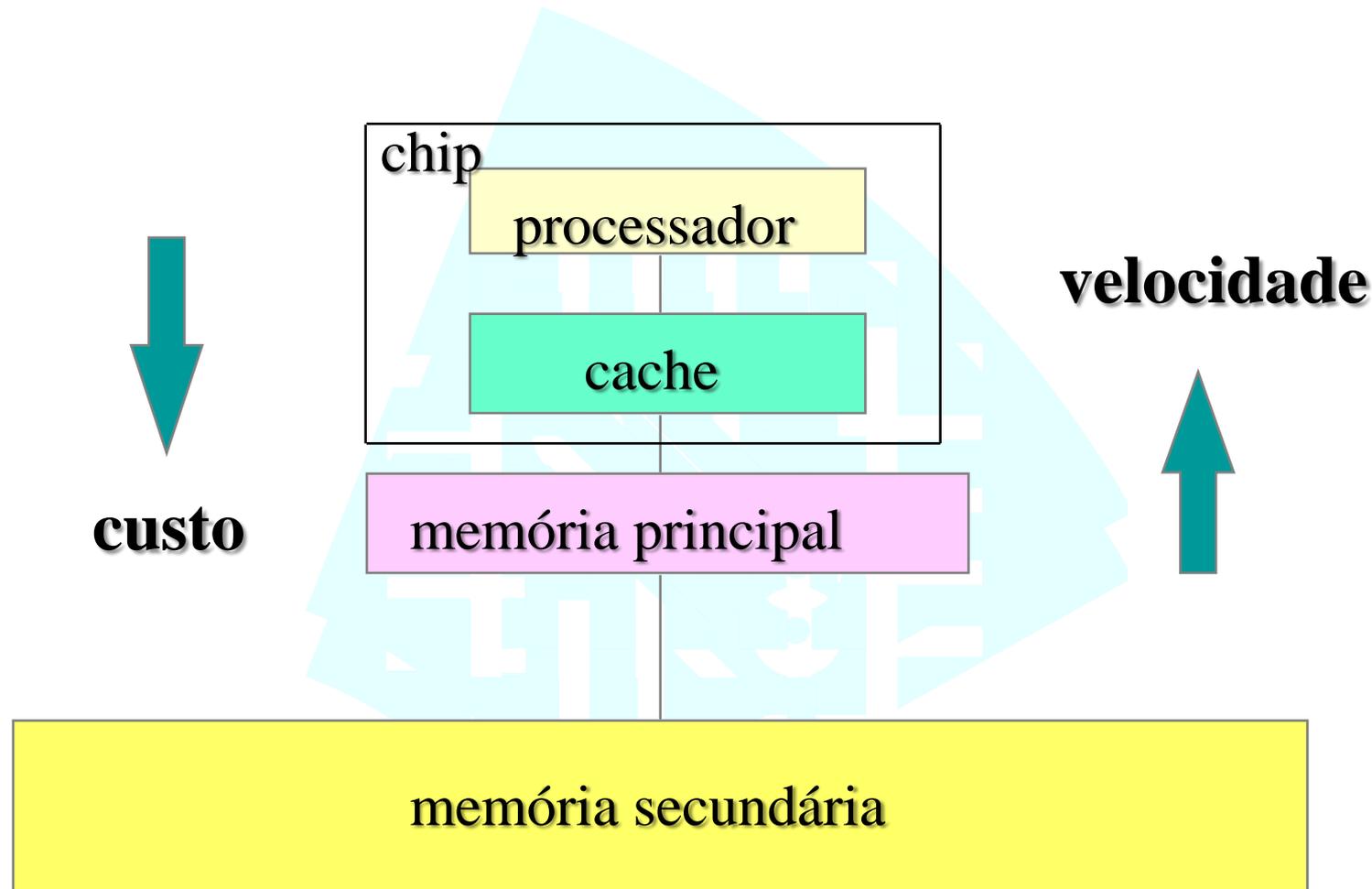
Pasta



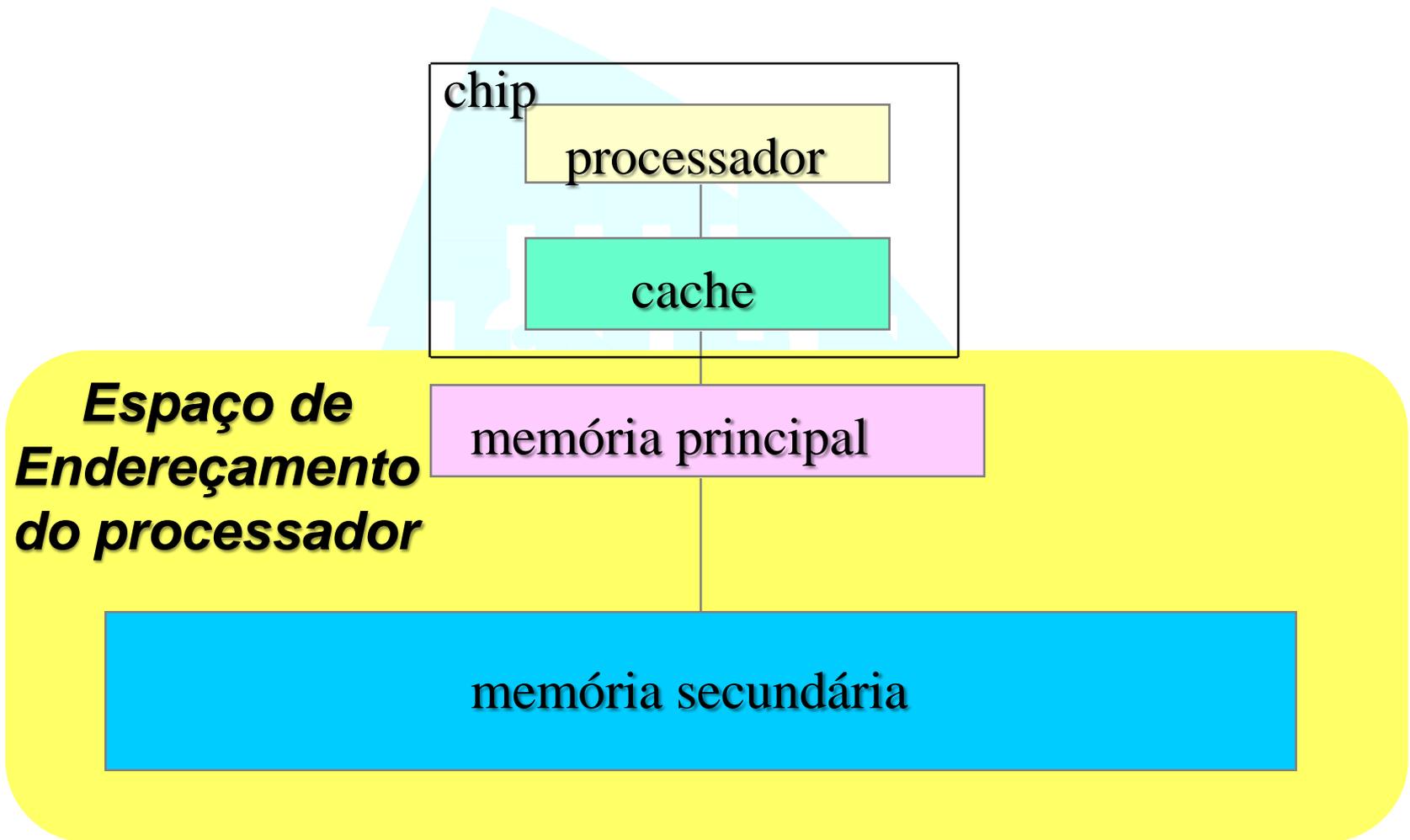
✓ Acessando a folha a ser processada:

1. Saber se a pasta está no fichário ou na mesa
2. Se estiver na mesa, descobrir se ela está no quadro
3. Se não estiver na mesa, deve ser trazida para a mesa.

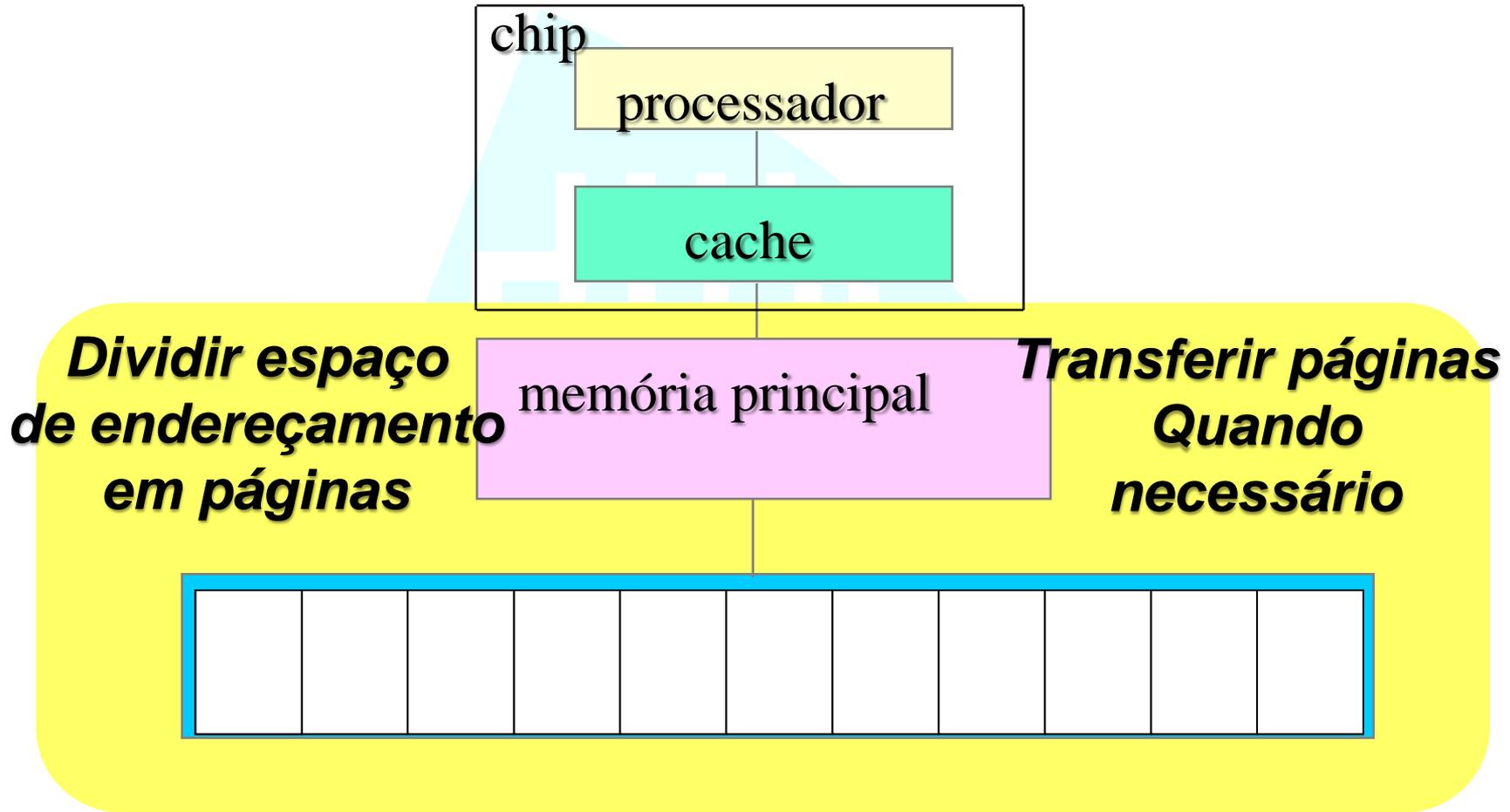
Hierarquia de memória



Memória Virtual



Como transferir a informação a ser processada para a memória?



Sumário

- ✓ Introdução
- ✓ Máquina pura
- ✓ Monitor residente
- ✓ *Swapping*
- ✓ Partições múltiplas
- ✓ Paginação
- ✓ Segmentação
- ✓ Segmentação com paginação

Introdução

- ✓ Multiprogramação permite que a CPU possa ser compartilhada por um conjunto de processos;
 - Compartilhamento de memória entre vários processos;
 - Incremento na complexidade de comunicação entre os componentes computacionais;
- ✓ Para padronizar a multiprogramação foram desenvolvidos alguns mecanismos de Gerência de Memória;

Máquina Pura

- ✓ Sistema onde não existe gerência de memória;
- ✓ O usuário lida diretamente com o hardware e possui total controle sobre a memória;
 - Onde podemos ver exemplos?
- ✓ Vantagens:
 - Maior flexibilidade para o usuário;
 - Maior simplicidade;
 - Menor custo: não requer software nem hardware adicional;

Máquina Pura

✓ Desvantagens:

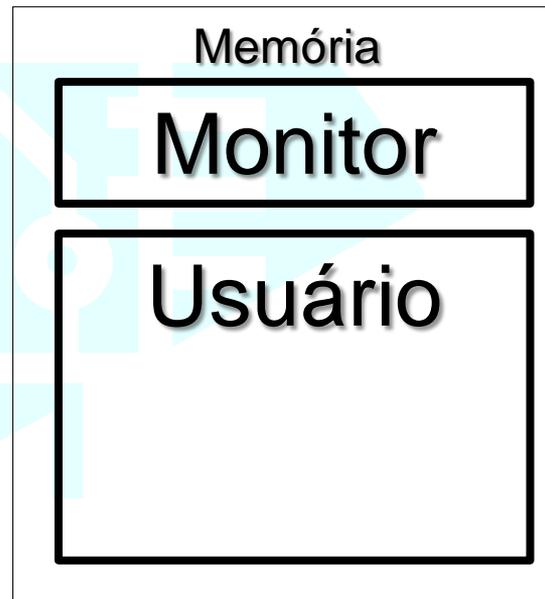
- Ausência de serviços: sistema operacional não controla interrupções;
- Não há monitor residente para processar chamadas do sistemas ou erros;

✓ Onde utilizar:

- Sistemas dedicados: eletrodomésticos, sistemas embarcados, circuitos integrados específicos;

Monitor Residente

- ✓ Sistema onde a memória é dividida em duas partes: área do usuário e área do sistema operacional;
- ✓ Na área do sistema operacional é guardada a tabela de interrupções;



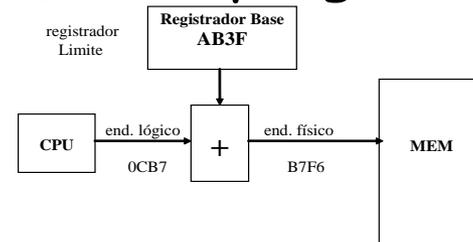
Monitor Residente

- ✓ Utilizado apenas para programação monousuário;
- ✓ Funcionamento:
 - Sistema operacional ocupa a parte inicial da memória;
 - Usuário trabalha com os endereços limitados pelo registrador de segurança;
 - Hardware de proteção para o SO - registrador de segurança indica o limite de endereço que o usuário pode utilizar;
 - O mecanismo de acesso a memória só aceita endereços maiores que o identificado no registrador limite;
 - Sempre será somado o valor do registrador limite no endereçamento lógico do programa;

Monitor Residente

✓ Relocação:

- **Estática:** endereços relativos são transformados em endereços absolutos no momento da carga do programa. Para isto, o código executável deverá conter os códigos que referenciam todos os endereços - **informação de relocação**.
- **Dinâmica:** os endereços permanecem lógicos o tempo inteiro. Dentro da CPU fica um registrador com o valor limite (**Registrador base** ou **Registrador de realocação**) e em tempo de execução do programa faz a soma com o endereço lógico;

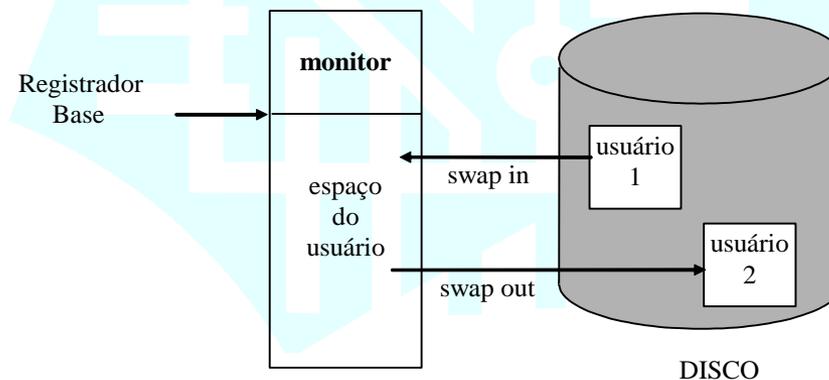


Monitor Residente

- ✓ **Relocação Dinâmica:**
 - Permite modificar o espaço de memória do SO a qualquer momento;
 - Movimentar um programa com facilidade;
 - Incluir controladores de dispositivos;
- ✓ **Com o uso de registrador base, passam a haver dois tipos de endereços:**
 - **endereços lógicos**, entre $0 \dots MAX$ (endereços vistos pelo programa);
 - **endereços físicos**, entre $RB+0 \dots RB+MAX$ (vistos pelo sistema operacional).

Swapping

- ✓ Utiliza o disco para transformar o Monitor Residente monousuário em multiusuário;
- ✓ O programa que perde a CPU é copiado para o disco, enquanto o programa que ganha é transferido do disco para a memória principal;
- ✓ Códigos executáveis ficam no disco e vão para a memória quando forem chamados;



Partições Múltiplas

- ✓ Vários programas compartilham a memória ao mesmo tempo;
- ✓ A memória é dividida pelo número de programas existentes;
- ✓ A memória pode ser dividida em:
 - Partições fixas: estáticas;
 - Partições variáveis: regiões dinâmicas;

Partições Múltiplas

✓ Partições fixas:

- A memória é dividida em partições que não são alteradas durante a execução do sistema. Por exemplo, uma memória de 256K palavras poderia ser dividida da seguinte forma:
 - Monitor residente: 64K palavras;
 - Espaço para processos pequenos: 16K palavras;
 - Espaço para processos médios: 48K palavras;
 - Espaço para processos grandes: 128K palavras.

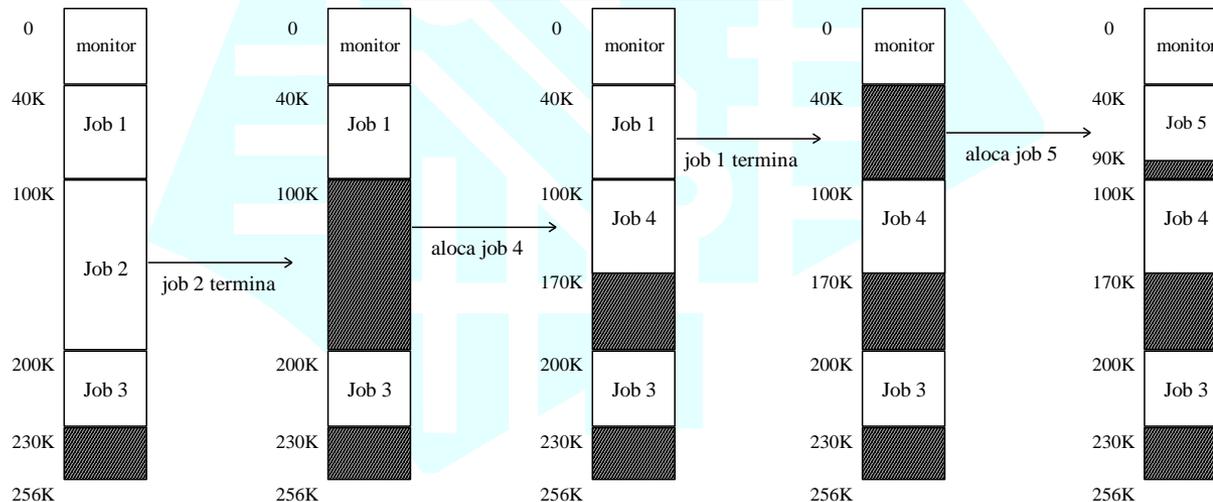
✓ Qual problema poderá ocorrer?

Subutilização de memória

Partições Múltiplas

- ✓ Partições Variáveis: o sistema operacional mantém informações sobre quais partes da memória estão disponíveis e quais estão ocupadas.

JOB	Memória	Tempo
1	60K	10
2	100K	5
3	30K	20
4	70K	8
5	50K	15



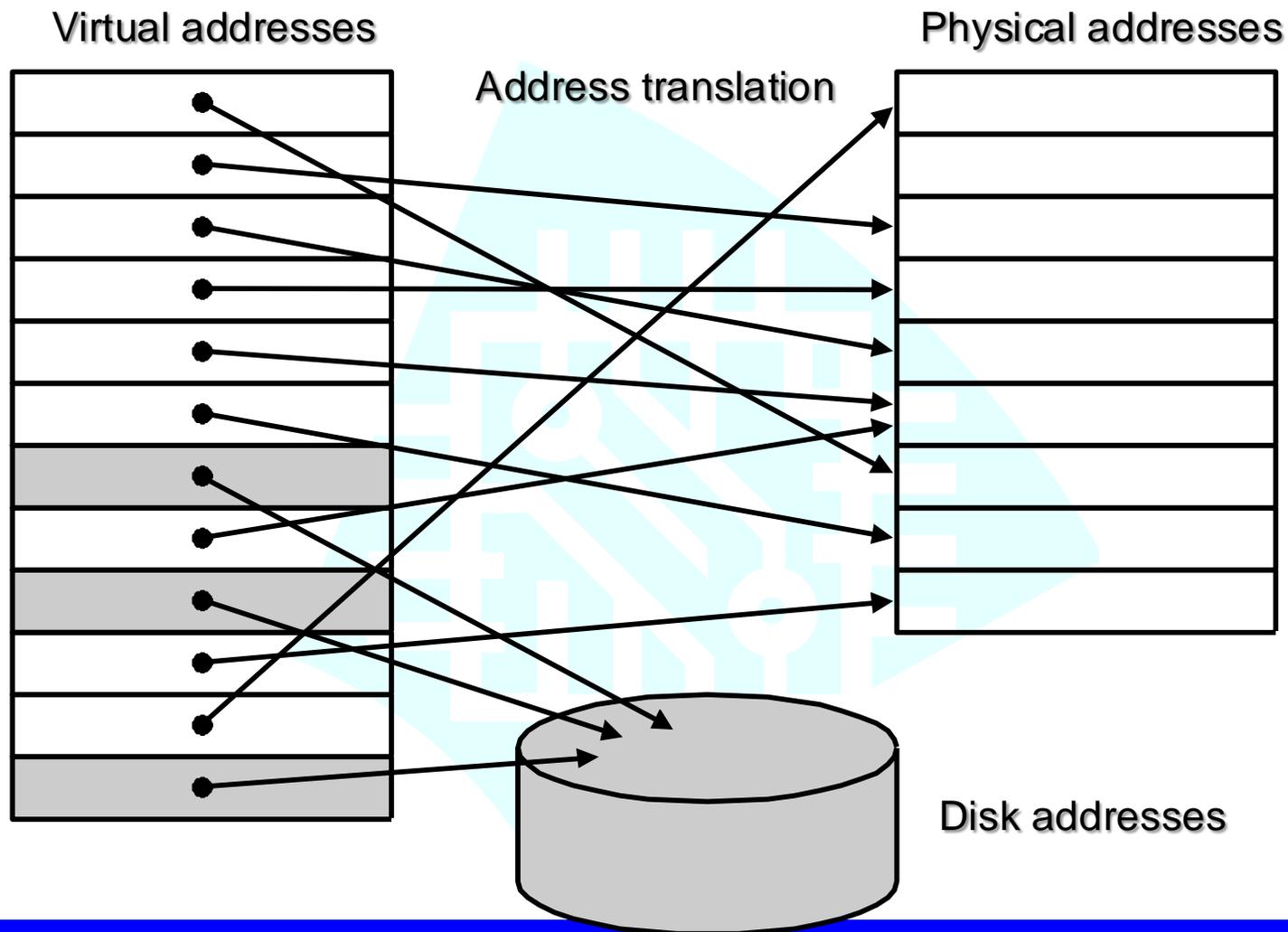
Partições Múltiplas: variáveis

- ✓ Características das partições variáveis:
 - Sempre existe um conjunto de áreas livres na memória;
 - O conjunto é pesquisado e parte da memória é entregue ao programa;
 - Se a área for maior o restante ficará livre;
 - Quando o processo termina a área da memória é liberada;
 - Unidade de alocação diferencia do byte: 2kb, 4kb,...

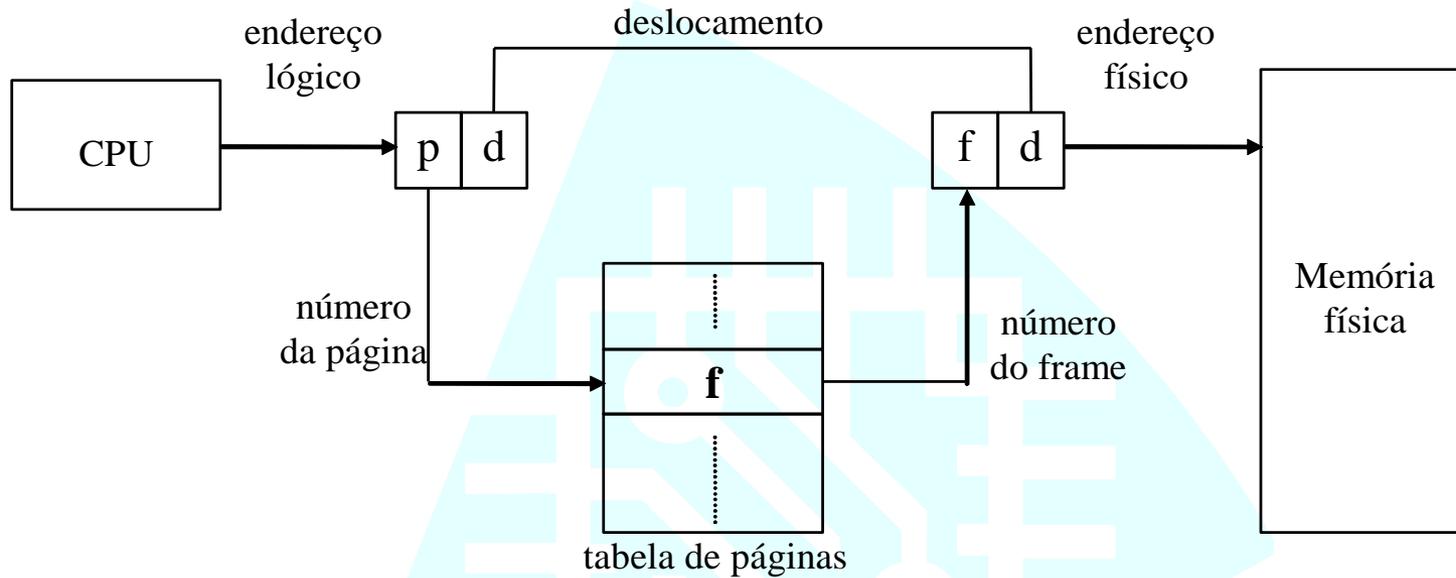
Paginação

- ✓ Partições variáveis apresenta o problema da fragmentação externa;
- ✓ Cada endereço gerado pela CPU é quebrado em duas partes: um número de página p e um deslocamento d ;
- ✓ A memória lógica de um processo está dividida em blocos denominados **páginas lógicas**;
- ✓ a memória física do computador está dividida em blocos denominados **páginas físicas** ou *frames* ;
- ✓ Páginas físicas e lógicas possuem o mesmo tamanho, definido a nível de hardware;

Paginação



Paginação



Paginação

✓ Exemplo:

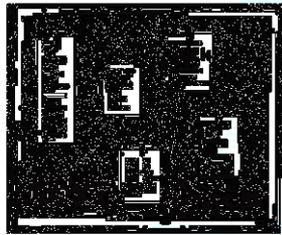
- Vamos supor que o computador utilize endereços de 16 bits e que a memória seja organizada em palavras;
 - Logo temos 2^{16} (64k) palavras para processos;
- Podemos então ter 5 bits para **p** e 11 bits para **d**



- Essa divisão é decidida em tempo de projeto do computador.

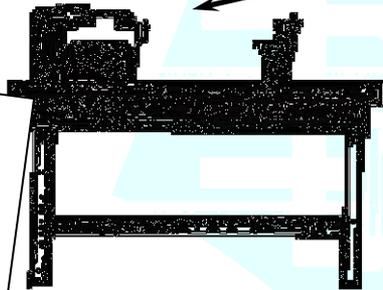
Localizando a informação

Fichário



Quadro

Pasta



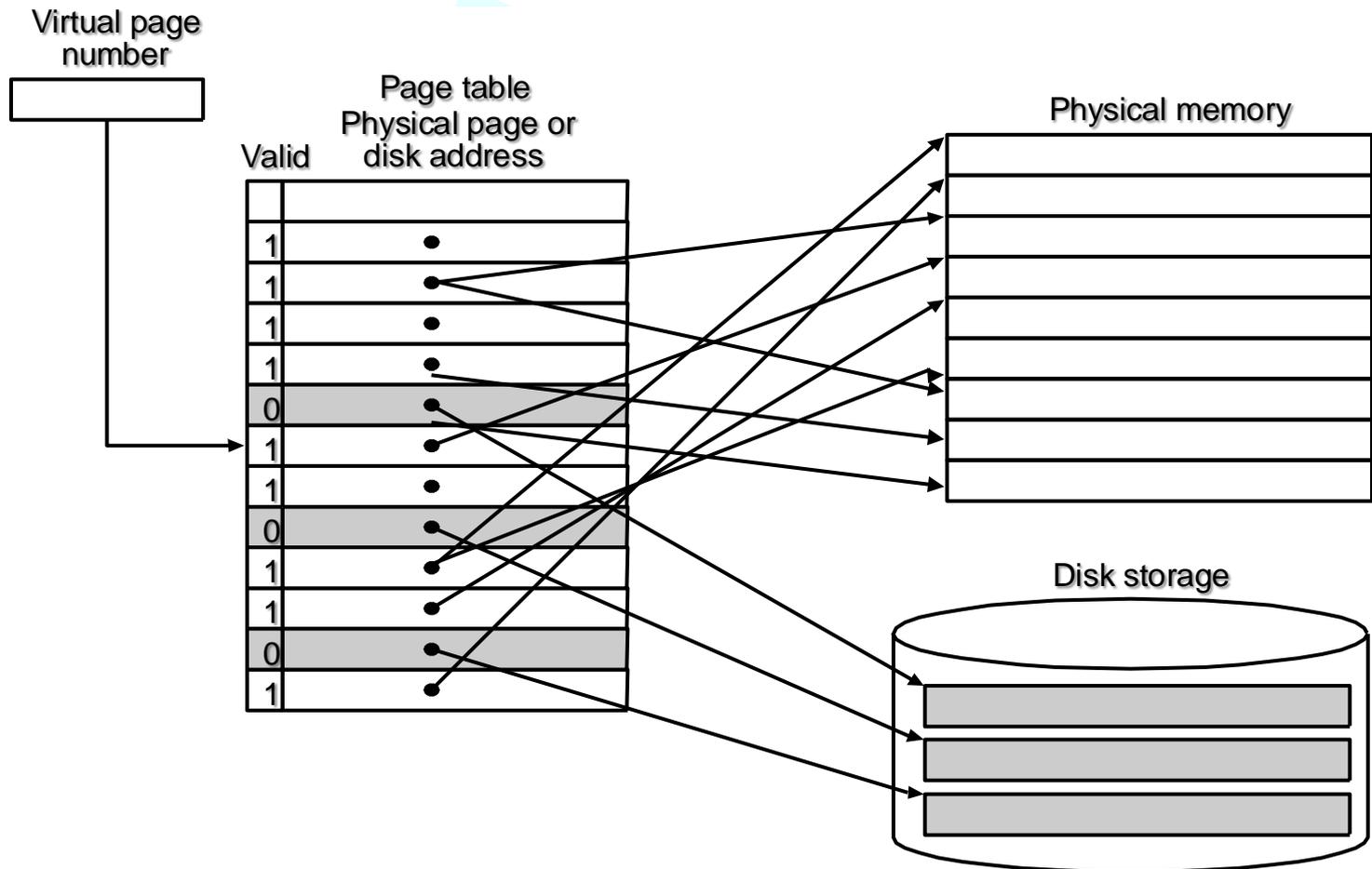
Onde encontraremos essa
informação?

✓ Tabela de localização:

- Informa se a pasta está na mesa
- E em qual posição do suporte de mesa
- Endereço na mesa= nr. do suporte e nr. da página

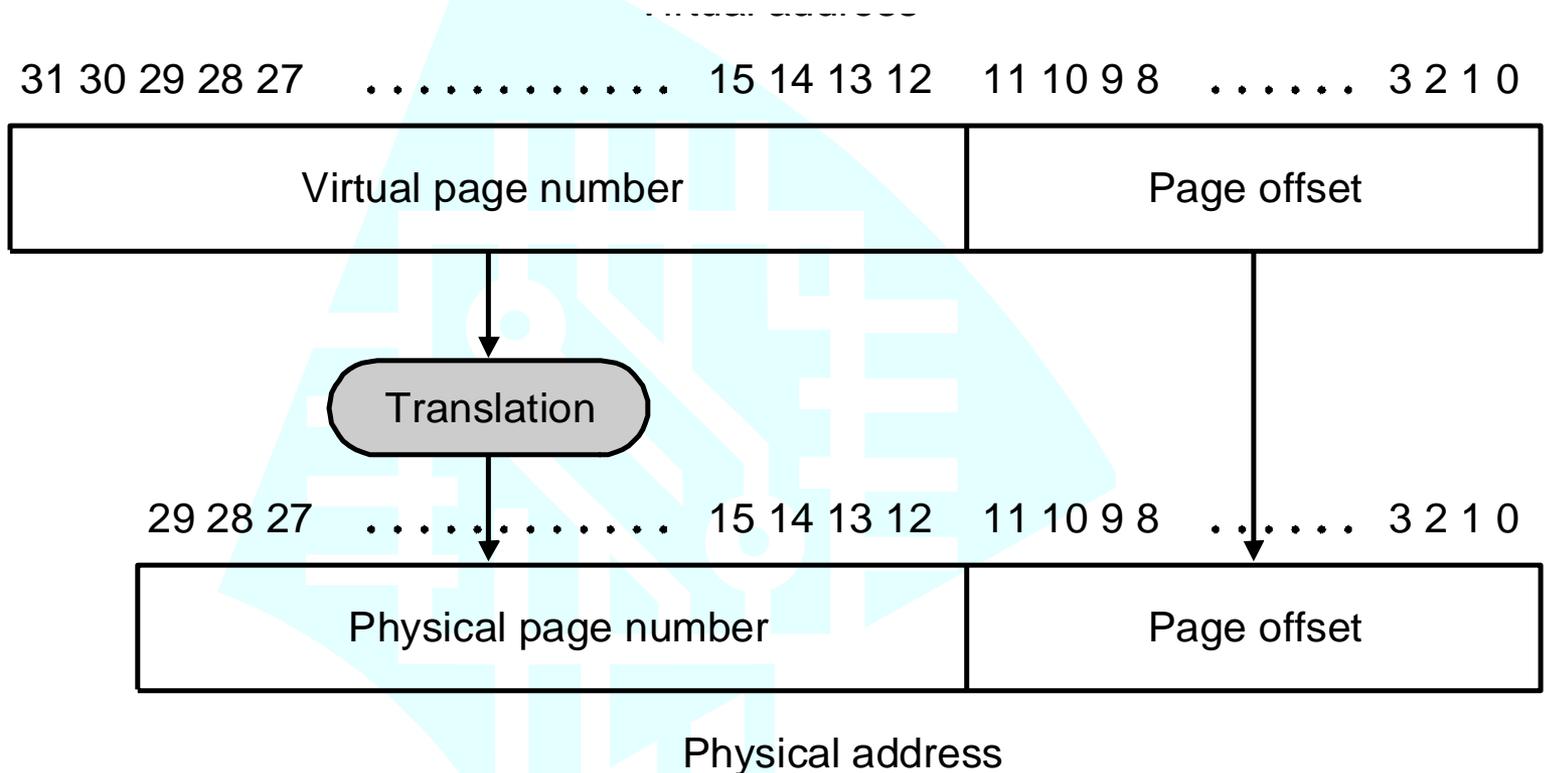
Paginação

✓ Localizando a página

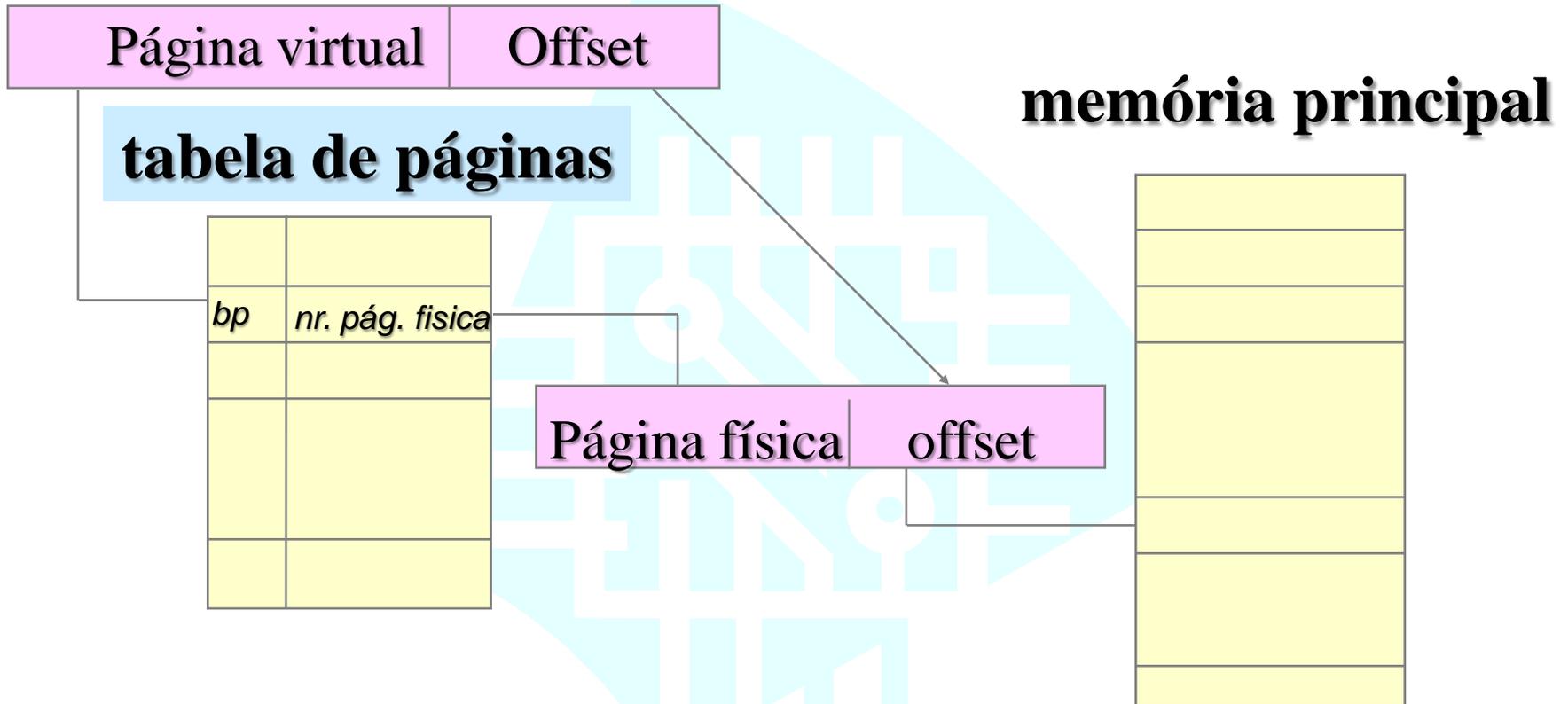


Paginação

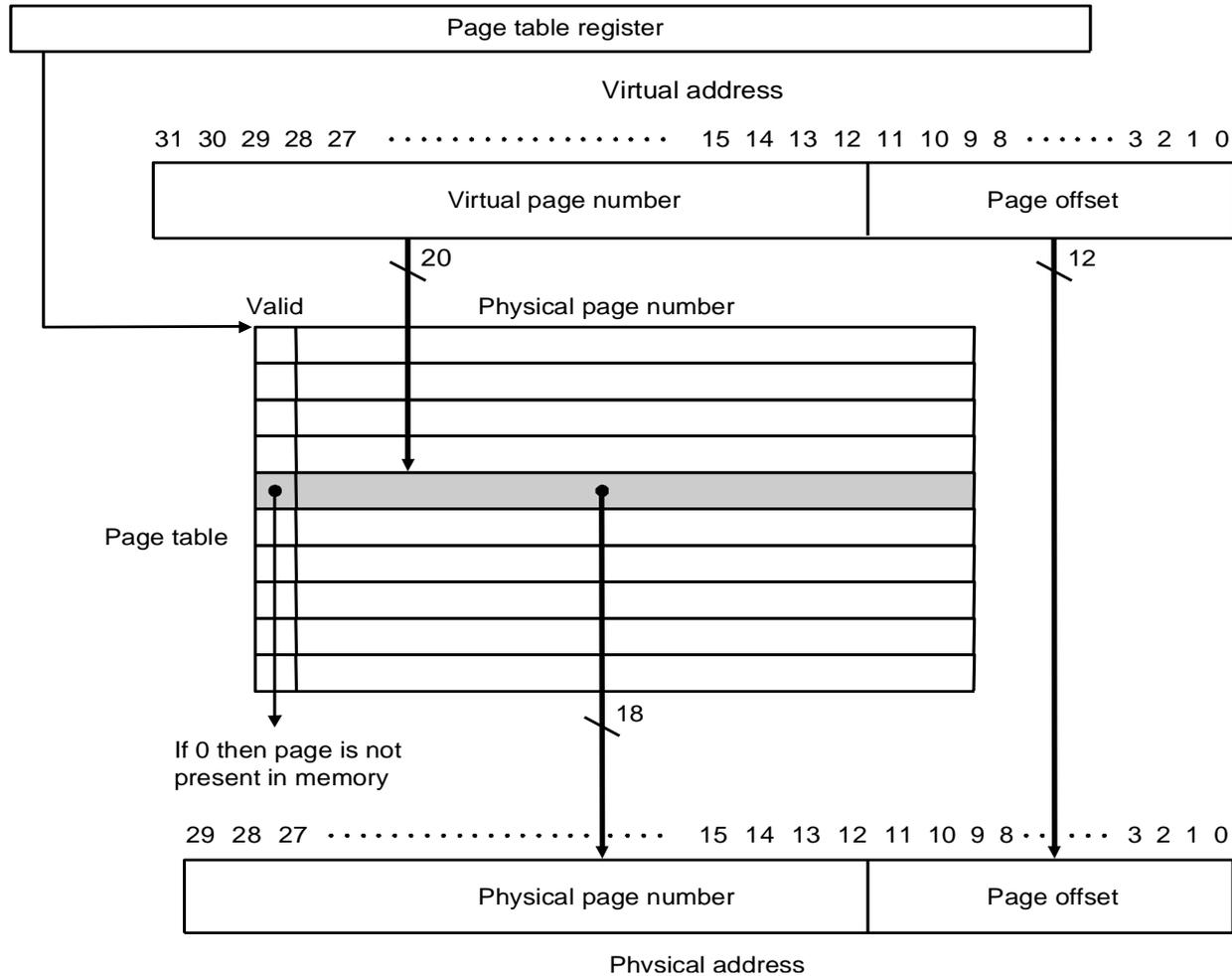
✓ Localizando a página na memória



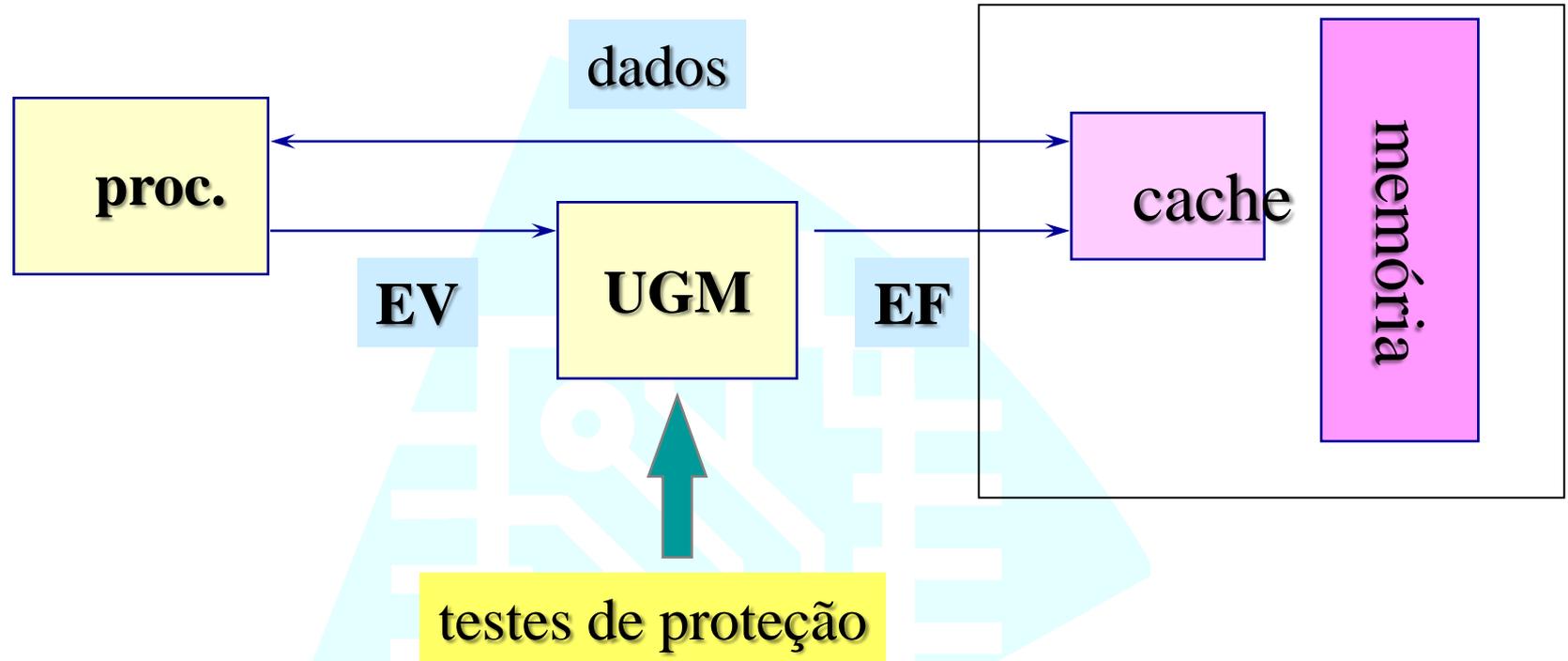
Paginação



Paginação



Unidade de Gerenciamento de Memória



✓ Converte endereço lógico em endereço físico.

Paginação

- ✓ O tamanho da memória física é determinado pelo número de bits utilizados para representar os endereços físicos;
 - UGM utiliza 20 bits para representar endereços físicos. Qual o tamanho da memória física? 2^{20}
 - Qual o espaço de endereçamento? $(2^{20} - 1)$ até 0
 - Se esse sistema utiliza páginas de tamanho 1 Kwords, qual o tamanho da memória física? 1024 páginas
- ✓ Os valores p e d correspondentes a um endereço lógico U são obtidos da seguinte forma:
 - $p = U \text{ div } P$ onde P é o tamanho da página;
 - $d = U \text{ mod } P$

Paginação

✓ Preço da penalidade

Tecnologia	Tempo de acesso	Custo por Mbyte (97)
SRAM	5-25 ns	\$100 – 250
DRAM	60-120 ns	\$5 - \$10
Disco	10-20 million ns	\$0.10 - \$0.20

Paginação

- ✓ Localização da página
 - Em qualquer posição
- ✓ Política de escrita
 - Write-back
- ✓ Falta de páginas
 - tratamento por software (Sistema Operacional)

Implementação da Tabela de Páginas

- ✓ Há 3 formas de implementação da Tabela de Páginas:
 - Conjunto de Registradores Dedicados;
 - Memória principal;
 - Memória Associativa;

Implementação da Tabela de Páginas

- ✓ Conjunto de Registradores Dedicados;
 - Na mudança do processo em execução, estes registradores são carregados com os valores correspondentes ao processo que recebe a UCP;
 - O acesso aos registradores da tabela é feito através de instruções privilegiadas;
 - Como estes registradores são caros, sua utilização não é viável para tabelas grandes.

Implementação da Tabela de Páginas

- ✓ Memória principal:
 - O registro descritor de cada processo contém o endereço da sua respectiva tabela de páginas;
 - O hardware inclui um registrador que aponta para a tabela de páginas em uso (*PTBR - Page Table Base Register*);
 - Este registrador é acessado através de instruções privilegiadas;
- ✓ Desvantagem:
 - Tempo de acesso: é necessário realizar dois acessos à memória, um para fazer o mapeamento (acesso à tabela) e outro para acessar os dados.

Implementação da Tabela de Páginas

✓ Memória Associativa:

- A memória associativa é uma memória de alta velocidade onde cada posição contém 2 campos: uma **chave** e um **valor**.
- Quando a memória associativa é consultada, ela recebe uma chave, a qual é comparada com o campo chave de todas as posições, simultaneamente.
- Se alguma chave igual é encontrada, o campo valor correspondente é devolvido.
- Se aquela entrada na tabela não está na memória associativa, então um acesso à memória principal deve ser feito

Paginação

- ✓ Localidade da execução:
 - A probabilidade de uma pesquisa à memória associativa resultar em sucesso é grande graças a propriedade conhecida como **localidade de execução**:
 - Todo programa avança acessando o seu espaço lógico (buscando de lá instruções e dados).
 - Durante esse avanço, a cada intervalo de tempo, utiliza apenas pequenas regiões do seu espaço de endereçamento (muitas regiões sequer são acessadas durante toda a execução).
 - É essa propriedade que mantém pequenas, regiões do espaço de endereçamento denominada localidade de execução.

Paginação

✓ Tempo de acesso efetivo:

- O número de vezes em que o número da página é encontrado na memória associativa depende do tamanho desta.
- Memória associativa de 8 ou 16 bits tem taxa de acerto de 80% ou 90% (*hit ratio*).

✓ Páginas compartilhadas:

- É possível compartilhar código entre os processos;
- O sistema operacional deve garantir (via hardware) que nenhum processo altere (escreva) estas páginas.
- As páginas com código devem ter os mesmos números para todos os processos.

Paginação

✓ Proteção:

- Cada entrada na tabela de páginas possui alguns bits adicionais para implementar proteção.
- Como todos os acessos passam pela tabela, o hardware pode verificar a legitimidade do acesso apenas verificando estes bits.
 - um bit para indicar que a página é de apenas leitura;
 - um bit para indicar página válida/inválida.

✓ Fragmentação:

- Fragmentação interna na última página do programa.
- Para um dado tamanho de memória lógica, quanto menor o tamanho da página, maior será o tamanho da tabela de páginas.

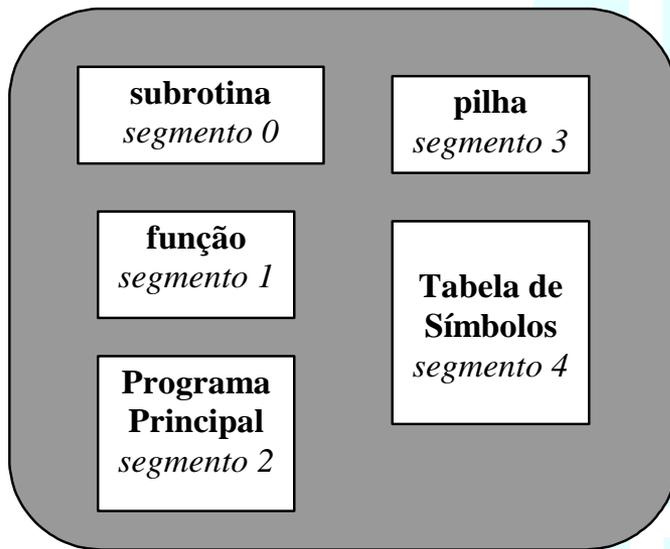
Segmentação

- ✓ É uma forma de gerência de memória que suporta esta visão do usuário (segmentos de tamanhos variados).
- ✓ O espaço de endereçamento lógico é uma coleção de segmentos.
- ✓ Cada segmento tem um número(nome) e um tamanho.
- ✓ Cada posição da memória lógica é especificada por um segmento e um deslocamento dentro deste segmento;
- ✓ Os compiladores e montadores automaticamente criam segmentos para refletir a organização do programa.

Segmentação: Hardware

- ✓ É necessário mapear os endereços lógicos, do tipo **segmento/deslocamento**, para endereços físicos, que continuam sendo (e sempre serão) índices de um *array* unidimensional.
- ✓ Isto é feito através de uma tabela de segmentos.

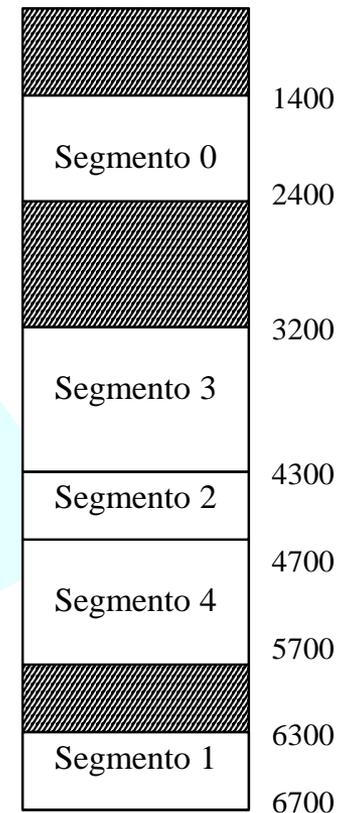
Segmentação



Espaço de Endereçamento Lógico

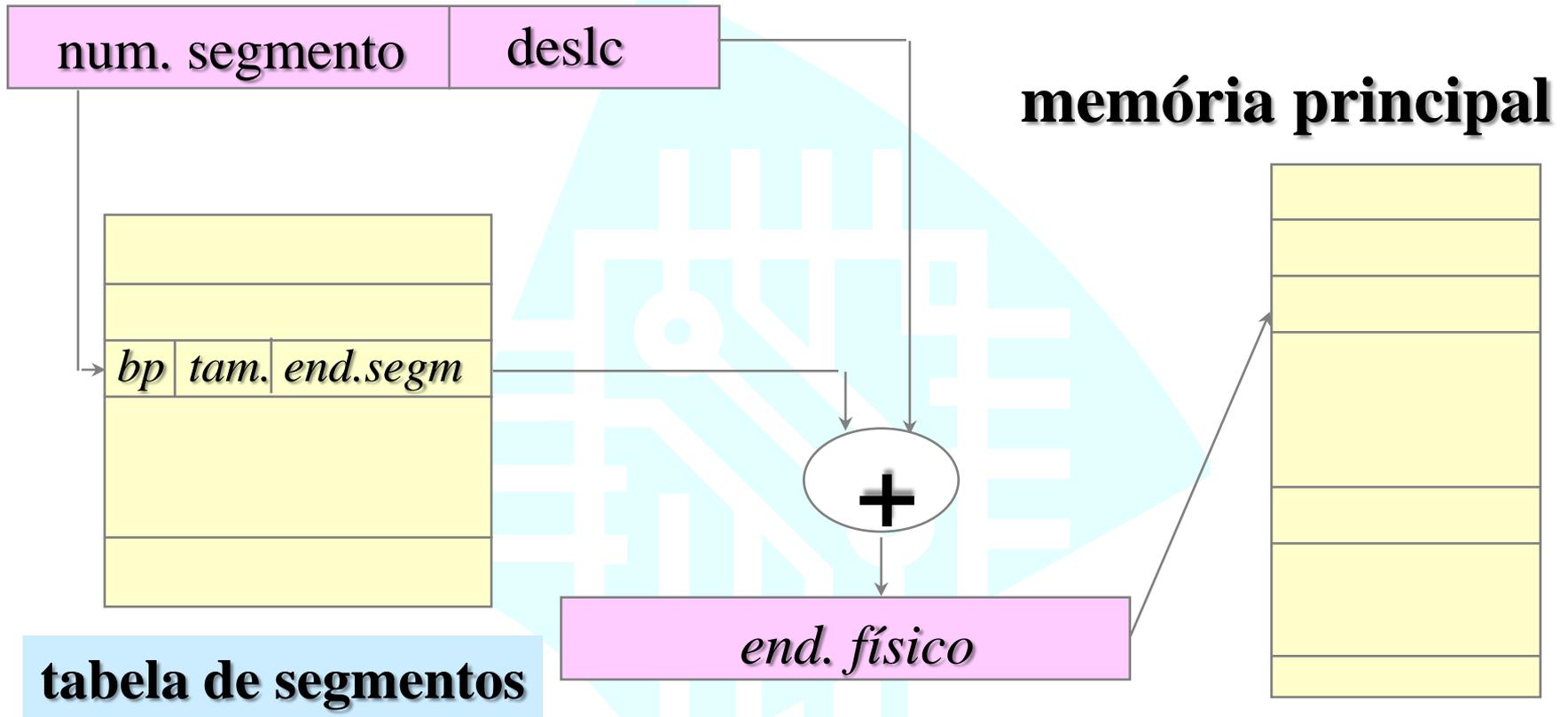
	base	limite
0	1400	1000
1	6300	400
2	4300	400
3	3200	1100
4	4700	1000

Tabela de segmentos



Memória Física

Segmentação



Segmentação

✓ Vantagens:

- Melhor uso da memória
- Permite associar proteção a informação

✓ Desvantagens:

- Todo o segmento é carregado na memória
- Fragmentação externa

Segmentação

- ✓ Implementação da tabela de segmentos:
 - Semelhante a tabela de páginas;
- ✓ Proteção e compartilhamento:
 - Segmentos podem ser compartilhados entre usuários, desde que a restrição dos segmentos de código ocupem as mesmas posições relativas dentro dos espaços lógicos dos processos compartilhantes.
- ✓ Fragmentação:
 - **fragmentação externa:** a sucessiva alocação e liberação de segmentos com diferentes tamanhos faz surgir pequenos espaços de memória livre.

Segmentação X Paginação

✓ Fragmentação:

- Segmentação:

- Fragmentação Externa;

- Paginação:

- Fragmentação interna (em média, perda de $\frac{1}{2}$ página por processo); Melhor opção se as páginas forem pequenas.

✓ Administração de Memória:

- Paginação mais simples já que qualquer página física pode ser usada para carregar qualquer página lógica.

Segmentação X Paginação

✓ Proteção e compartilhamento

- Segmentação bem melhor. Segmentos constituem as unidades lógicas do programa (isto é, cada segmento tem um determinado significado ou função);
- Cada segmento poderá ser compartilhado e ter associada a si uma determinada proteção.
- Paginação: partes de um programa estão aglutinadas, formando um bloco único (em uma mesma página podem estar funções, subrotinas e dados, por exemplo).

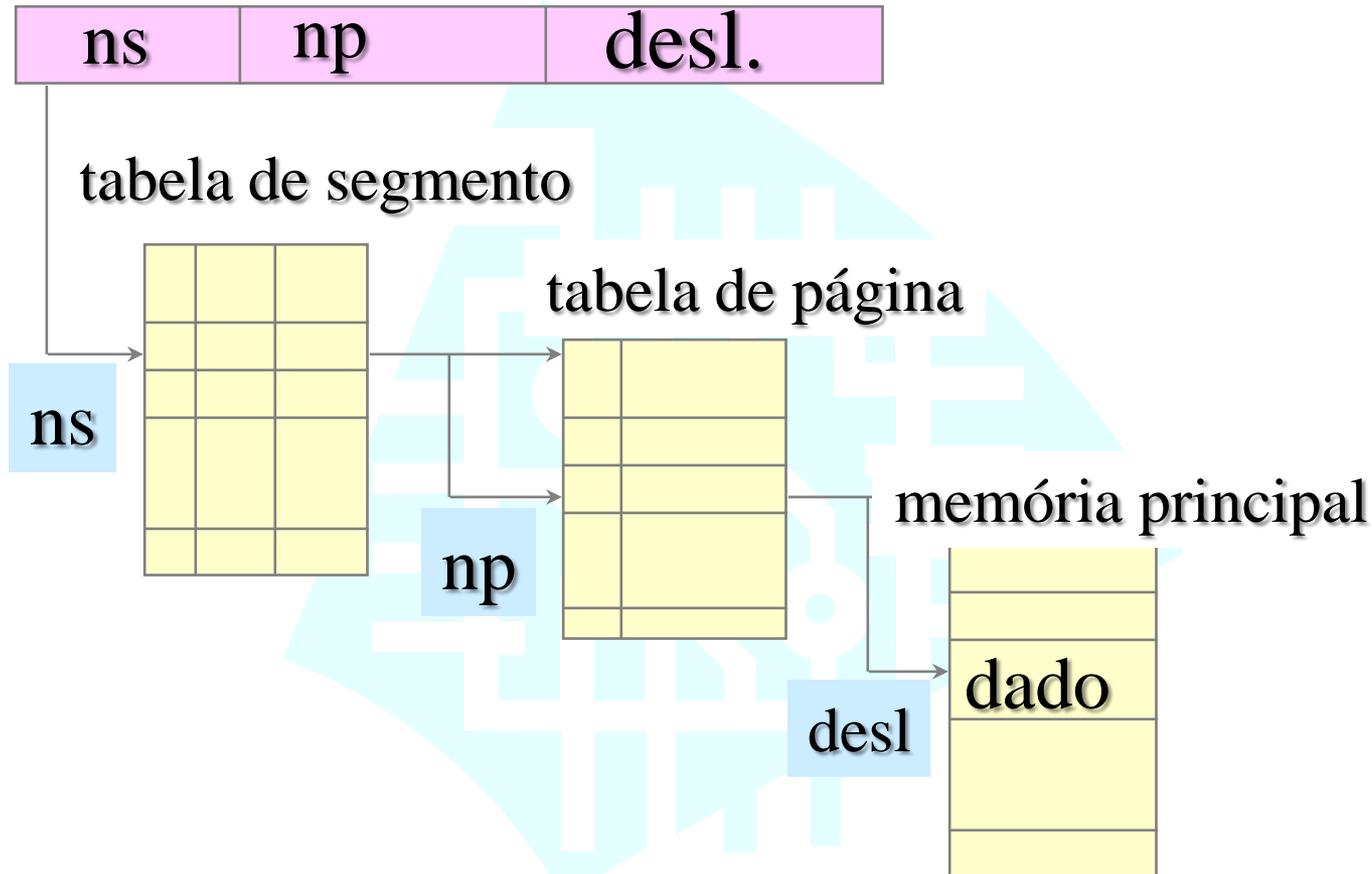
Segmentação Paginada

- ✓ Segmentação tem problema de fragmentação externa;
- ✓ Como resolver? Dividir os segmentos em páginas
- ✓ Com isso o problema da fragmentação externa é completamente eliminado;
- ✓ Porém existirá fragmentação interna!
- ✓ O que fazer? Nada, pois a fragmentação é pequena, meia página.

Segmentação Paginada

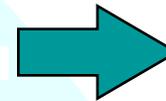
- ✓ Se as tabelas de segmentos e de páginas estão na memória principal, então para cada acesso à memória serão necessários dois acessos adicionais.
- ✓ Para acessar o endereço (s,p,d) tem-se que acessar inicialmente a entrada s da tabela de segmentos para lá pegar o endereço da tabela de páginas correspondente, após isso, acessar a entrada p da tabela de páginas para pegar o endereço da página física f e, finalmente, acessar o que interessa, que está na posição $f+d$.

Segmentação Paginada



Segmentação paginada

- ✓ Falta do Segmento
- ✓ Falta de Página
- ✓ Falta de Proteção
- ✓ Falta de Limite



chamada do
sistema
operacional

chaveamento
de contexto

Qual a solução para o duplo acesso a memória?

Memória Acossiativa com tabela (s,p) e como
valor o número da página física **f**
correspondente a esse par;

Paginação por demanda

- ✓ Programa Completo X parte do programa
- ✓ Muitas partes dos programas não são necessárias durante a execução;
 - Exemplo: Editor de texto;
- ✓ Portanto, não é necessário carregar todo o programa na memória para poder utilizá-lo;
- ✓ Na paginação por demanda, apenas as páginas efetivamente acessadas pelo processo são carregadas para a memória física; (bit Valid)

Memória Virtual

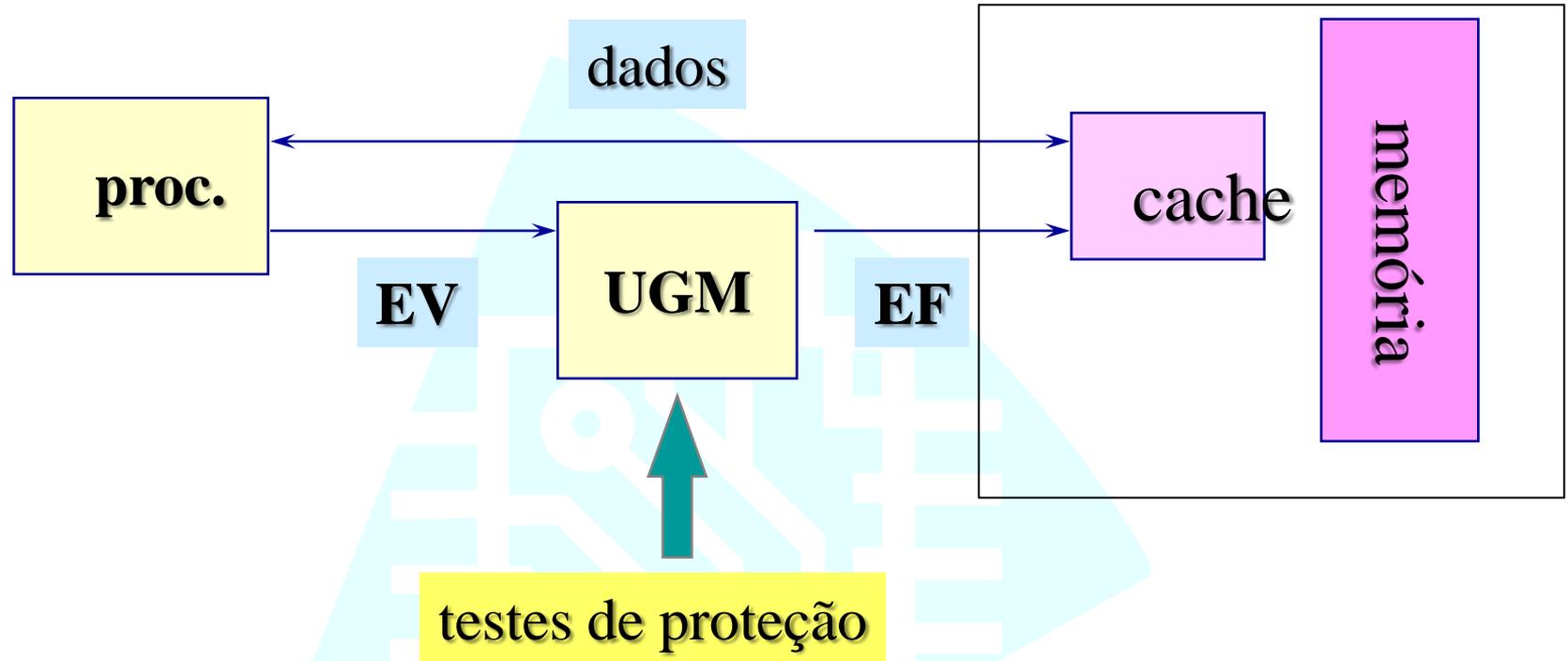
Grandes Memórias Virtuais



Grandes Tabelas de Páginas
Grandes Tabelas de Segmentos

- ✓ Como considerar tabela de páginas tão grandes?

Unidade de Gerenciamento de Memória

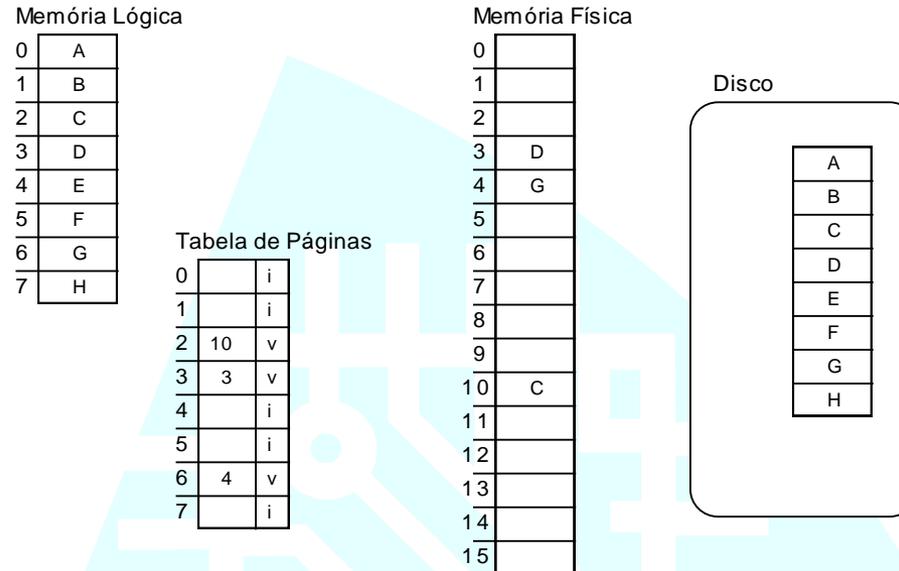


✓ Converte endereço lógico em endereço físico.

Unidade de Gerenciamento de Memória

- ✓ UGM Gera uma interrupção de proteção e aciona o sistema operacional.
- ✓ O SO consulta o descritor do processo em questão.
- ✓ Caso a página acessada esteja fora do espaço de endereçamento do processo, o processo é abortado.
- ✓ Caso contrário tem-se que a página referenciada ainda não foi carregada para a memória, e é dito que ocorreu uma interrupção por **falta de página** (*page fault*).

Mecanismo de substituição de páginas



Fonte: AMBIENTES OPERACIONAIS – Prof. Simão Toscani

Mecanismo de substituição de páginas

Memória Lógica do Processo 1

0	A
1	B
2	C
3	D

Tabela de Páginas do Processo 1

0	1	v
1	5	v
2		i
3	0	v

Memória Lógica do Processo 2

0	E
1	F
2	G
3	H

Tabela de Páginas do Processo 2

0	3	v
1	2	v
2	4	v
3		i

Memória Lógica do Processo 1

0	A
1	B
2	C
3	D

Tabela de Páginas do Processo 1

0	1	v
1	5	v
2		i
3	0	v

Memória Lógica do Processo 2

0	E
1	F
2	G
3	H

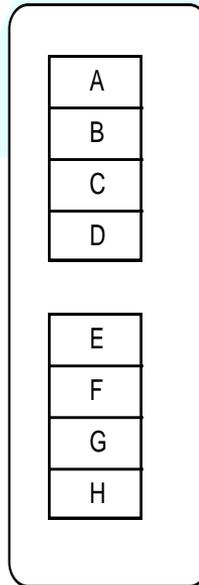
Tabela de Páginas do Processo 2

0	3	v
1	2	v
2		i
3	4	v

Memória Física

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

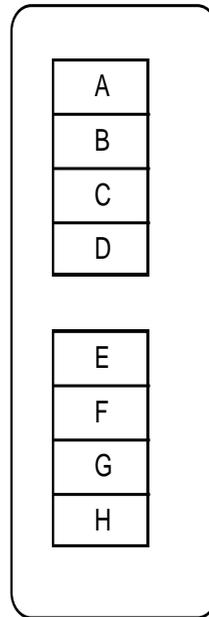
Disco



Memória Física

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

Disco



Fonte: AMBIENTES OPERACIONAIS – Prof. Simão Toscani

Algoritmos de substituição de página

- ✓ FCFS (*First Come First Served*)
 - A página escolhida como vítima é sempre aquela que está há mais tempo na memória.
- ✓ ÓTIMO
 - Página lógica que sai e a que ocorrerá o próximo acesso mais remotamente no futuro;
 - Esse algoritmo não existe! 😊
- ✓ LRU (*Least Recently Used*)
 - Página acessada mais remotamente no passado
- ✓ Existem outros: segunda chance, FIFO com segunda chance...

Interação entre *cache* e *memória virtual*

- ✓ Cache pode operar com endereços físicos ou virtuais;
- ✓ Quando utilizando endereços físicos, o acesso ao *cache* é realizado apenas ao final da tradução do endereço (*virtual para físico*) pela UGM;
- ✓ Com endereços virtuais pode-se permitir concorrência, porém tem que usar o mecanismo de *flushing*;

Tabela de páginas sob demanda

- ✓ Alocar área pré-definida na memória e permitir crescimento em uma direção
 - como gerenciar área de programas/dados + pilha?
- ✓ Alocar duas áreas pré-definidas permitindo crescimento da tabela em duas direções
 - MIPS

Tabela de páginas sob demanda

- ✓ Pagar tabela de páginas
 - área do sistema operacional
 - algumas páginas são carregadas inicialmente
 - Falta de página ou de Tabela de página (Page fault)
 - Chamada do sistema operacional

Tabela hierarquizada

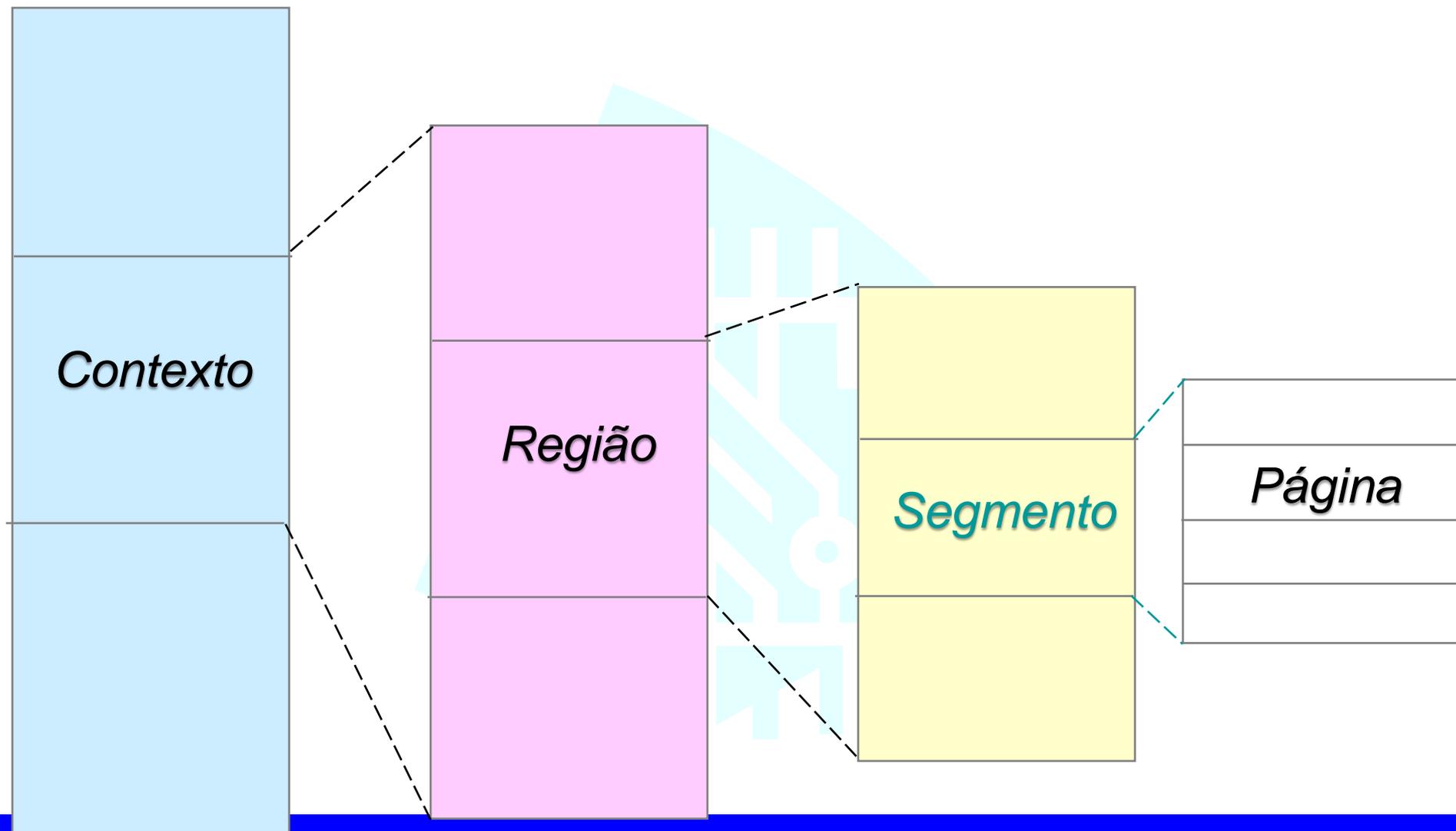
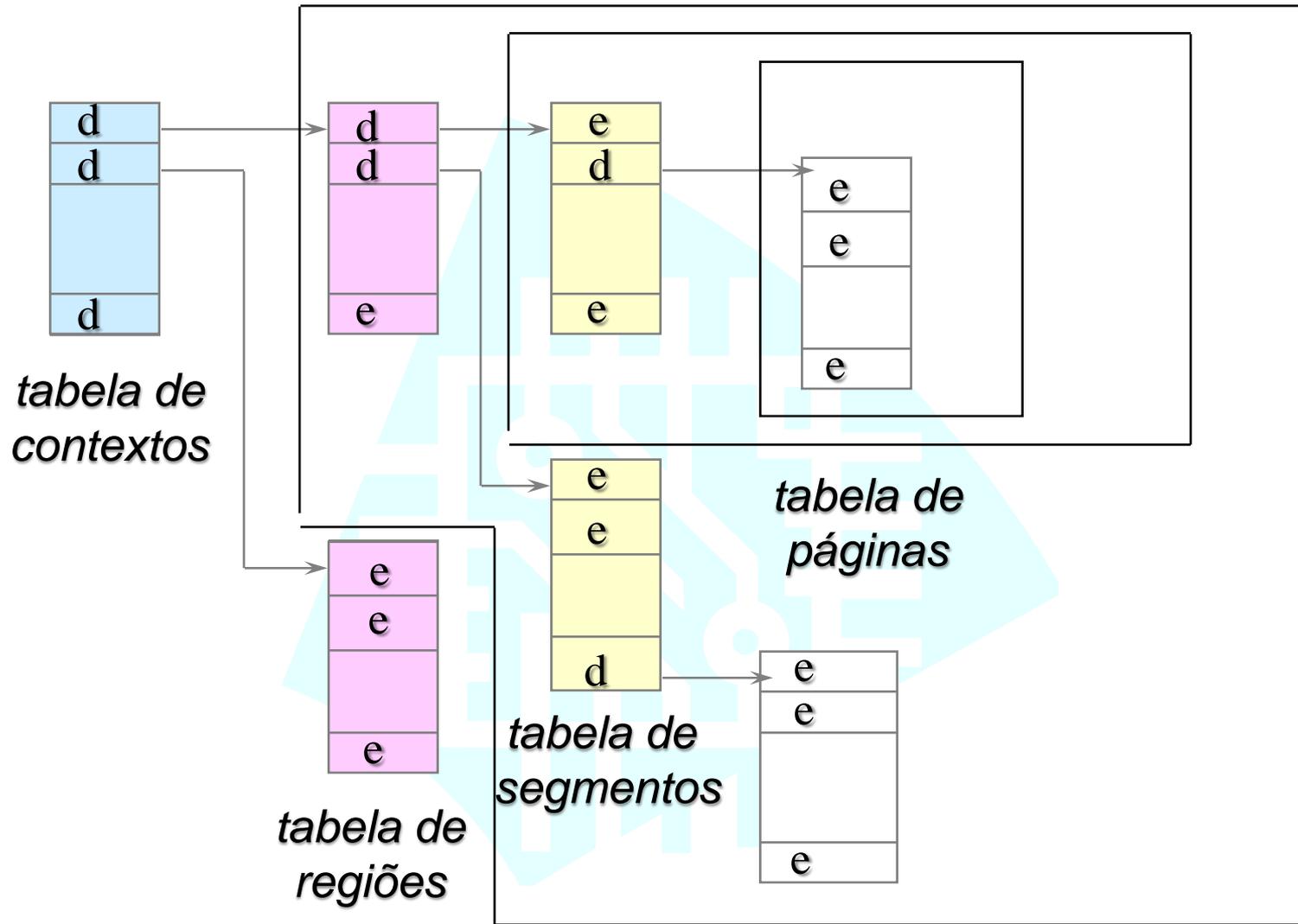


Tabela de Página Hierarquizada



Endereço Físico e Virtual

endereço físico

num. pag. física	desloc.
------------------	---------

endereço virtual

i1	i2	i3	desloc
----	----	----	--------

Memória Virtual em Máquinas de 32 e 64 bits

Grandes Memórias Virtuais



Grandes Tabelas de Páginas
Grandes Tabelas de Segmentos

Tabela de Páginas **Invertidas** (Power)
Tabelas **Hierarquizadas** (Sparc, Alpha, Motorola)

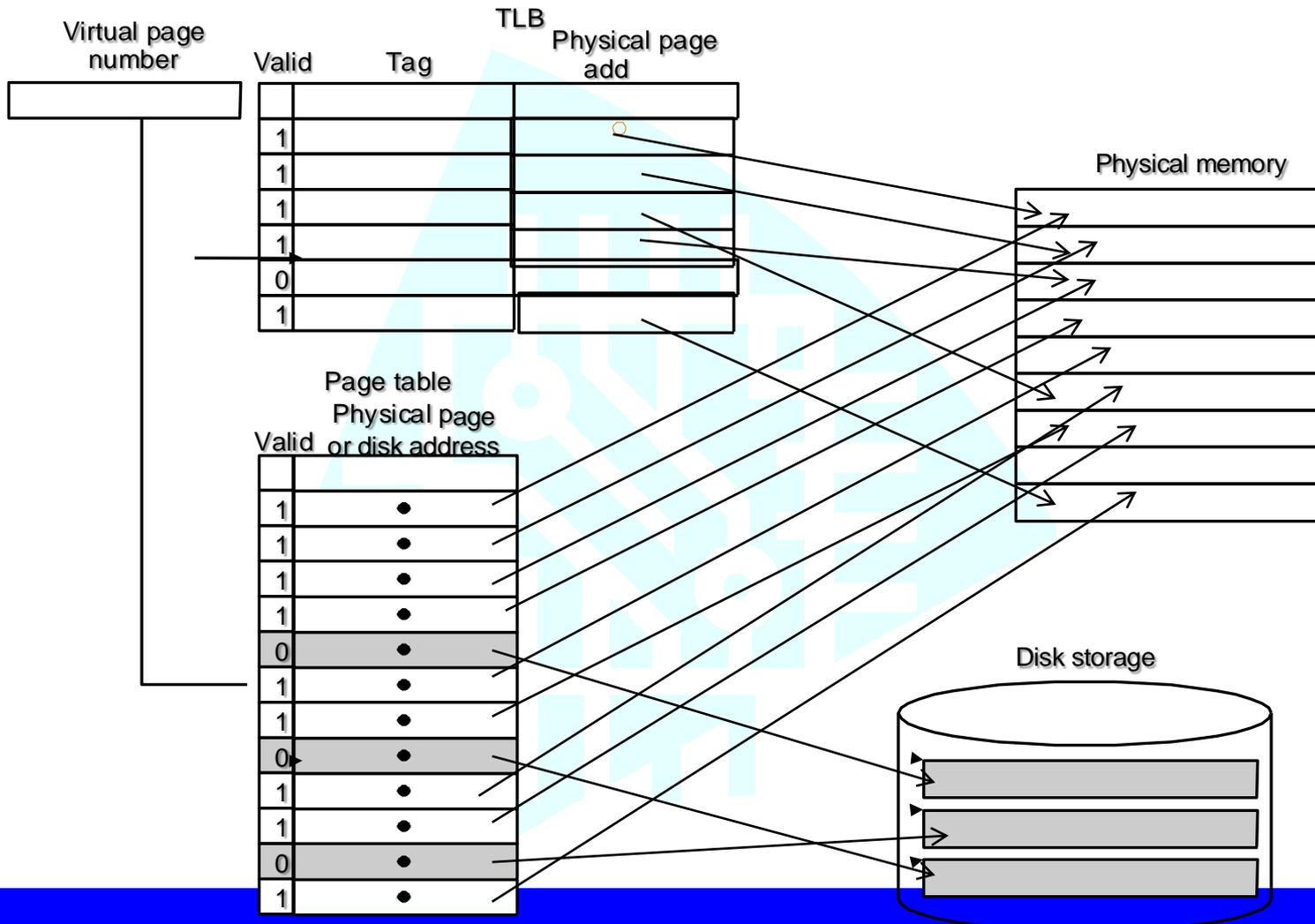
Acelerando a tradução...

- ✓ técnica tradicional:
 - mapeamento direto

nr. blocos da mem. virtual = nr. de entradas na tabela

- ✓ Onde guardar a tabela de tradução?
 - memória principal
 - TLB

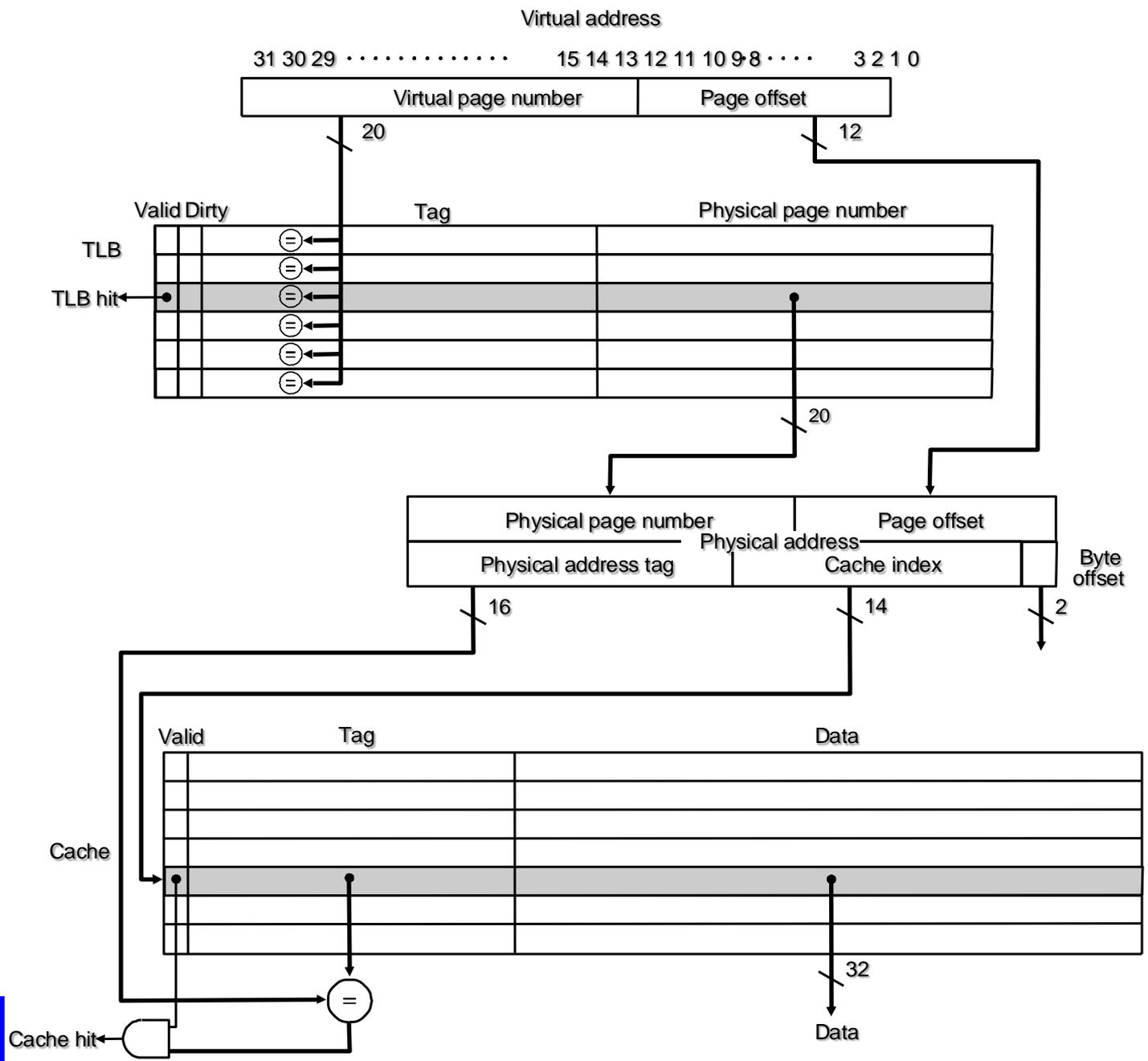
TLB (Translation lookaside buffers)

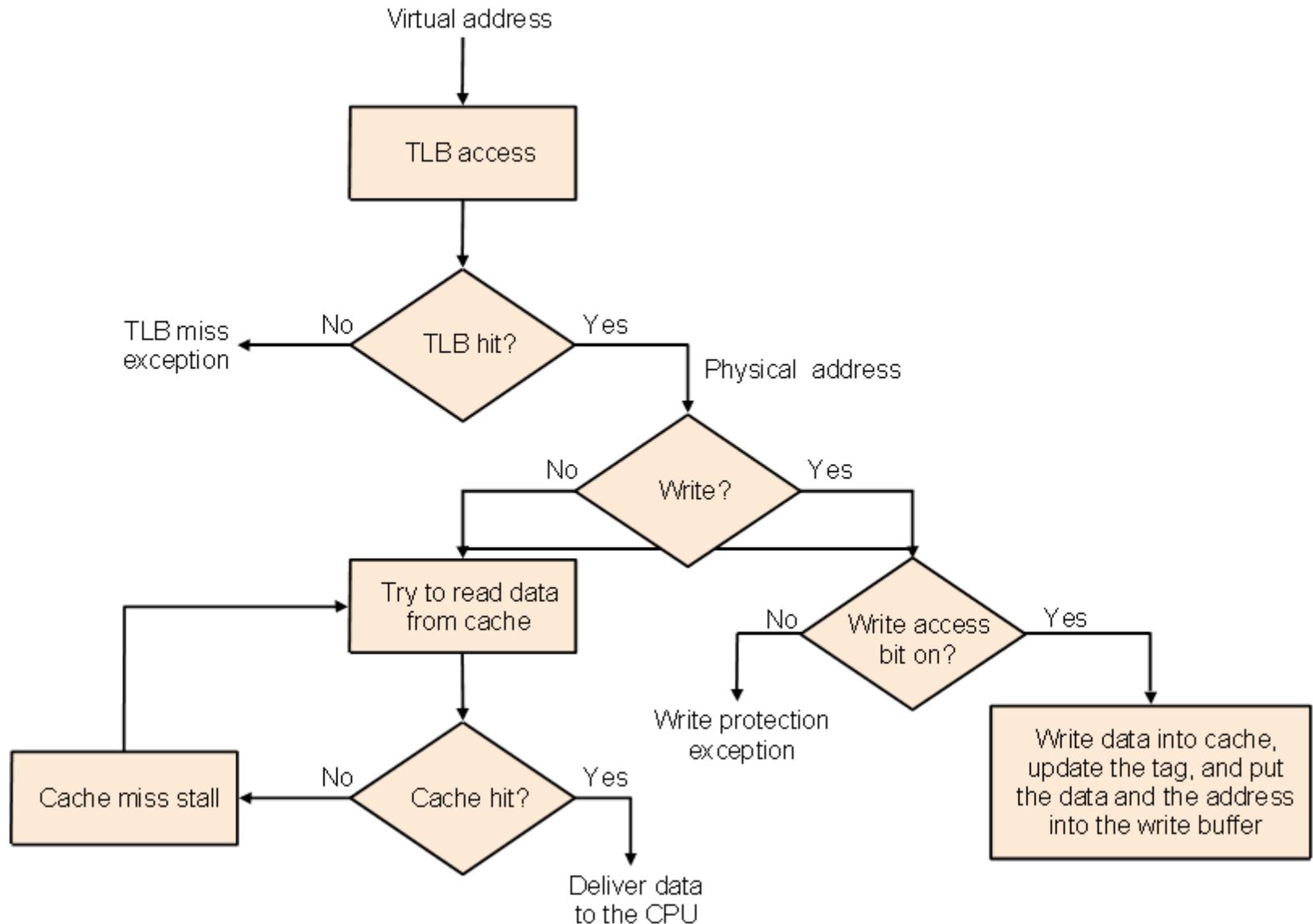


TLB

- ✓ Tamanho
 - 32 a 4.096 entradas
- ✓ Tamanho entrada:
 - 1-2 entrada da tabela (4-8 bytes)
- ✓ Tempo de acesso:
 - 0.5 a 1 ciclo
- ✓ Taxa de faltas:
 - 0.01 - 1%

MIPS





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

Resumindo

- ✓ Onde um bloco/pag. pode ser colocado?
 - Cache:
 - mapeamento direto
 - associativa por conjunto
 - associativa
 - Memória virtual:
 - associativa

Resumindo

- ✓ Como um bloco/pag. pode ser localizado?
 - Cache:
 - mapeamento direto: índice
 - associativa por conjunto: índice + comparação
 - associativa: comparação de toda a cache
 - Memória virtual:
 - tabela de páginas/segmentos

Resumindo

- ✓ Qual o bloco deve ser substituído numa falta?
 - Cache:
 - LRU (grau de associatividade 2-4)
 - randomica
 - Memória virtual:
 - LRU (a penalidade é muito grande)

Resumindo

- ✓ Qual a estratégia de escrita?
 - Cache:
 - Write-through
 - Write-back
 - Memória virtual:
 - Write-back

Exercício

Considere um sistema de memória virtual com as seguintes características:

- Endereço virtual de 32 bits
- Páginas de 8 Kbytes
- 64 M Bytes de memória principal

(a)- Qual o layout do endereço virtual?

(b)- Qual o layout e o tamanho da tabela de páginas em bytes?

Assuma que cada página virtual possui um bit de validade, 2 bits de proteção e um "dirty bit".

Considere ainda que os endereços na memória secundária não estão armazenados nesta tabela.

Bibliografia

- ✓ *Sistemas Operacionais*, Oliveira R., Carissimi A. e Toscani S., Ed. Sagra-Luzzatto, 2004
- ✓ *Organização e Arquitetura de computadores;* Henessy, J.L.; Patterson, D. A.;
- ✓ *Organização de Computadores*, Ricarte, I. L. M;
- ✓ *Ambientes Operacionais;* Toscani, S.;
- ✓ *Sistemas Operacionais*, Oliveira R., Carissimi A. e Toscani S., Ed. Sagra-Luzzatto, 2004
- ✓ *Arquitetura não convencional de computadores;* Slides; Barros, E. N. S;