

Hierarquias de Memórias e Memória Virtual

José A. Cardoso e Cunha
DI-FCT/UNL

1.

As memórias: colecções de unidades dispersas no espaço e em função, com tempos de acesso, capacidade e custo variáveis.

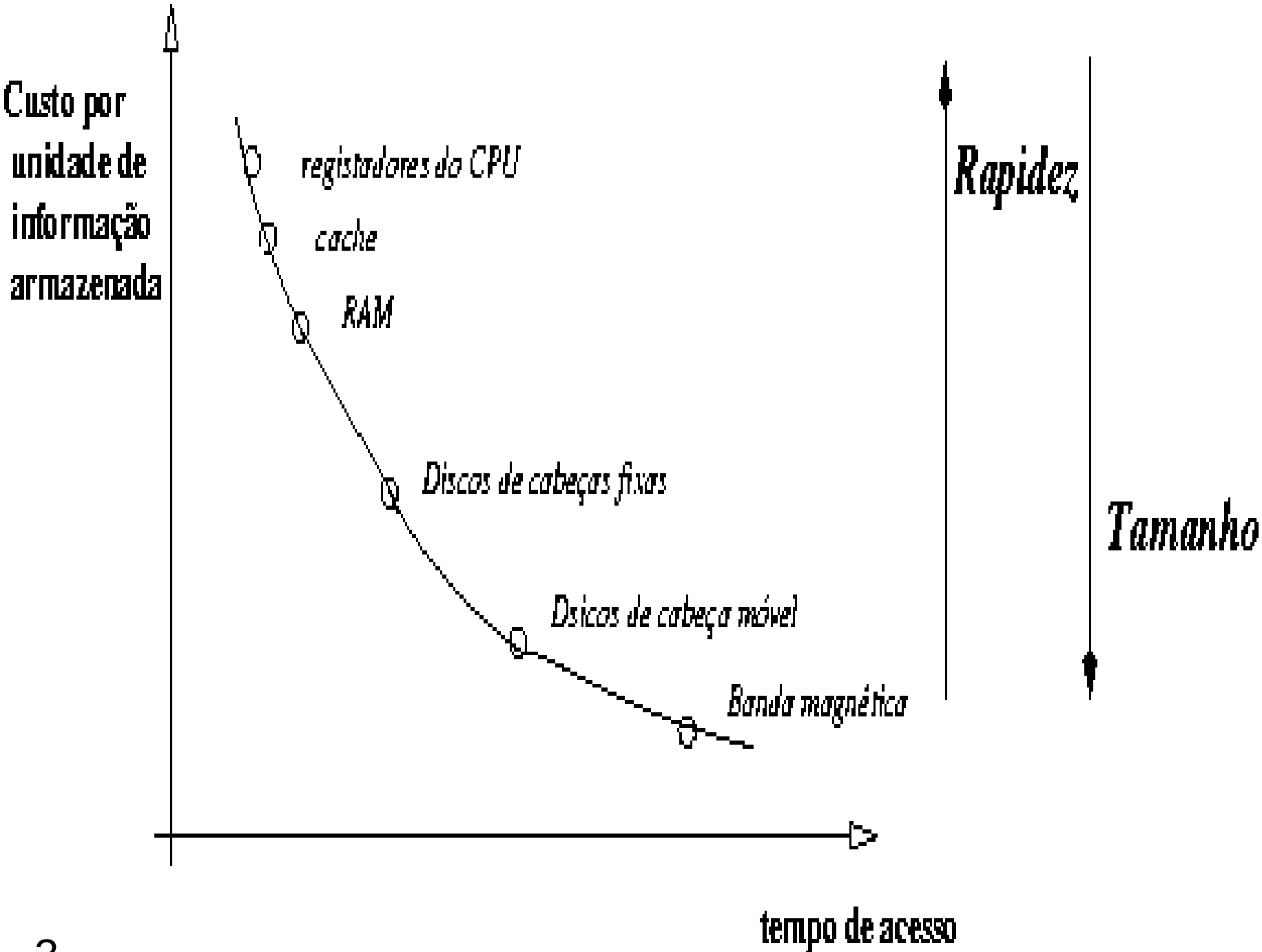
Memória: colecção finita de células, logicamente organizadas.

Cada célula:

- Endereço ou chave
- Conteúdo ou valor

Tipos de acesso:

- Por endereço
- Por chave / conteúdo (acesso associativo)



Memória central ou principal: acessível directamente pelo CPU, no passo FETCH e na execução de instruções de referência de memória.

Memória secundária ou auxiliar: acessível pelas instruções de entrada e saída; suportadas por dispositivos para armazenamento de ficheiros.

Unidades de acesso:

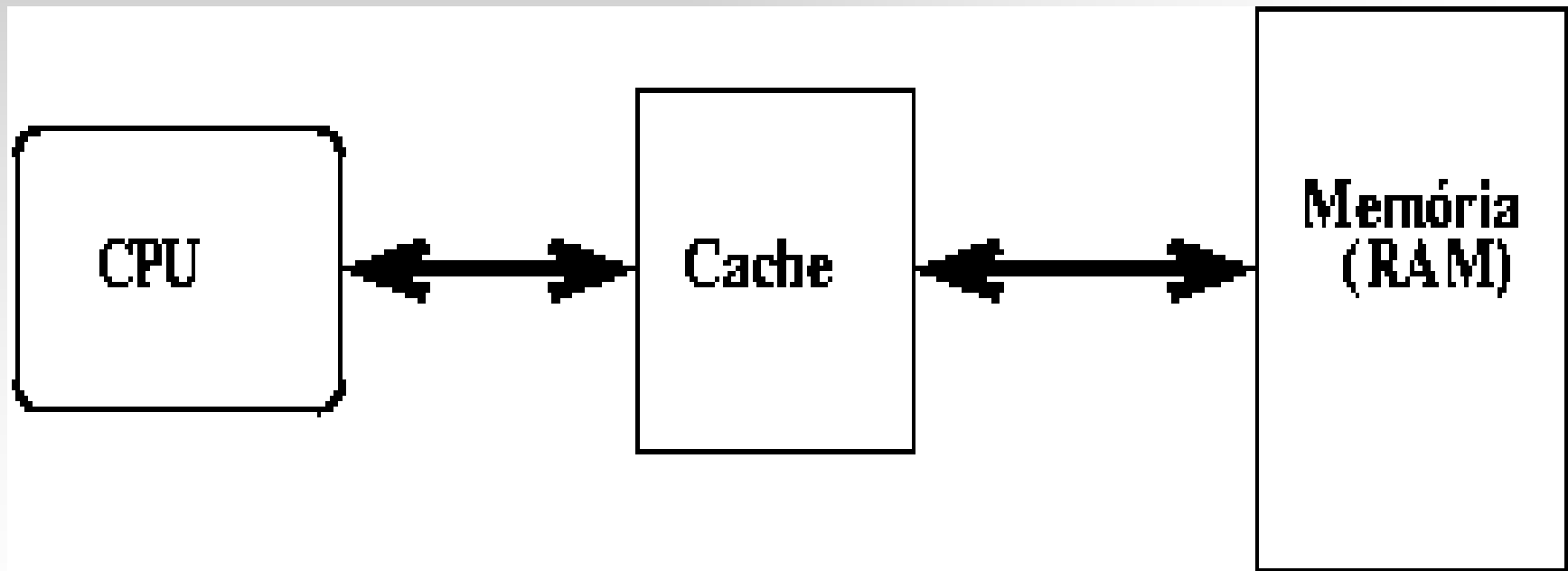
bytes, múltiplo de bytes

blocos de bytes (páginas, sectores ou blocos)

Memória central hierarquizada em 2 níveis:

- um tem acesso mais rápido e menor capacidade
- outro tem maior tempo de acesso e maior capacidade

→ O CPU deve “ver” uma memória de capacidade igual à RAM e com tempo de acesso igual à Cache.

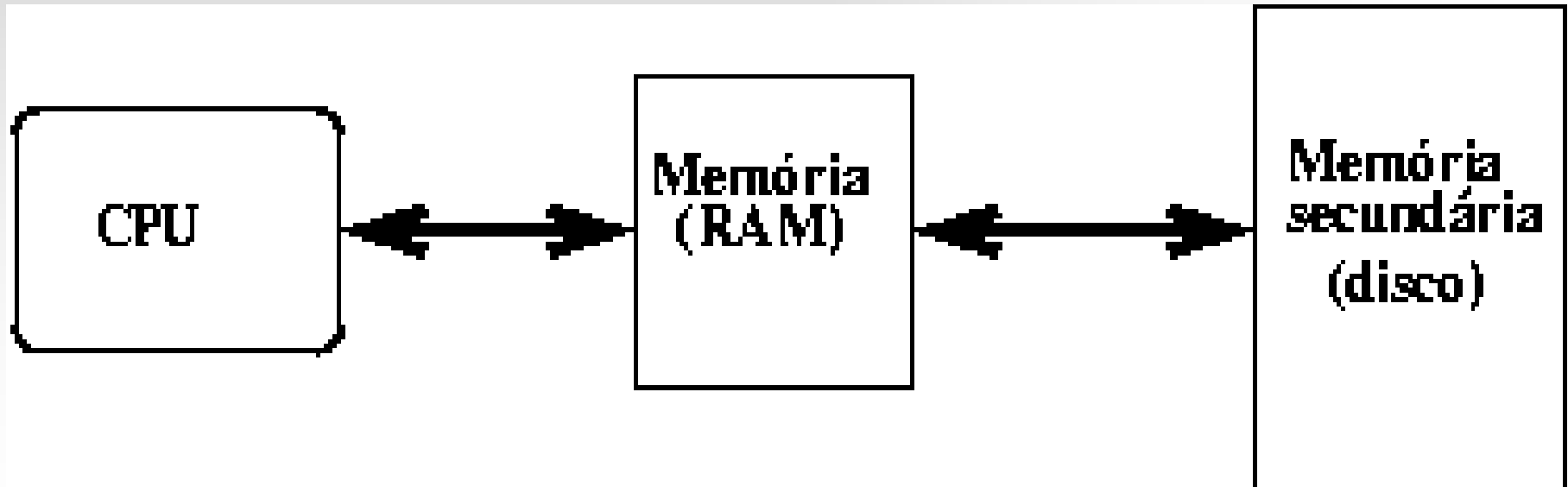


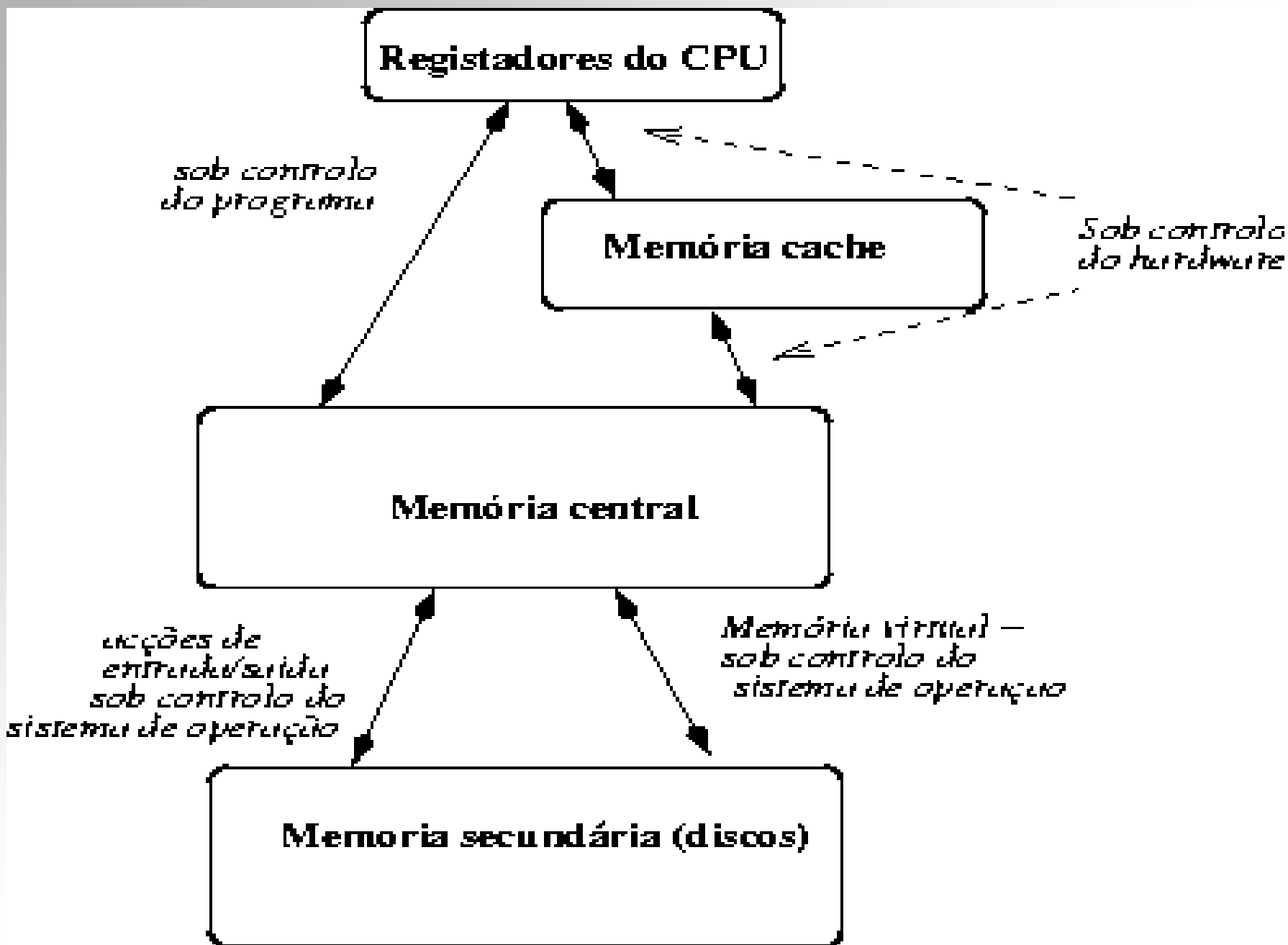
Memória central hierarquizada em mais 2 níveis

RAM-- Disco:

- um tem acesso mais rápido e menor capacidade
- outro tem maior tempo de acesso e maior capacidade

→ O CPU deve “ver” uma memória de capacidade igual ao DISCO e com tempo de acesso igual à RAM.





Necessidades de memória

suporte para a execução de programas e acesso aos dados pelo CPU, durante a execução dos programas (curtos períodos de tempo)

suporte para arquivo de ficheiros (longos períodos de tempo, memória não volátil)

Memória Virtual

esconder as limitações do espaço de endereços reais

simular uma memória dedicada a cada processo, com dimensão adequada e acesso rápido

Factores condicionantes

- esquemas de endereçamento hardware

Factores condicionantes

- esquemas de endereçamento hardware
- limitações do espaço (real e físico) de memória

Factores condicionantes

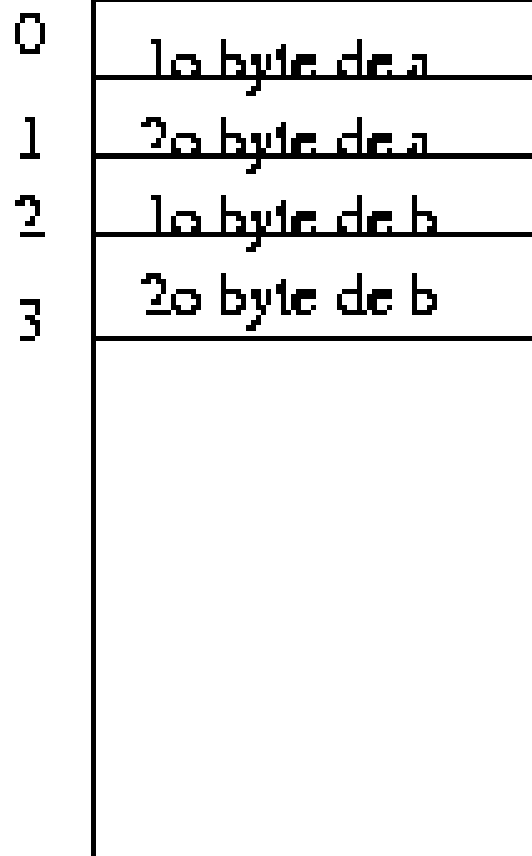
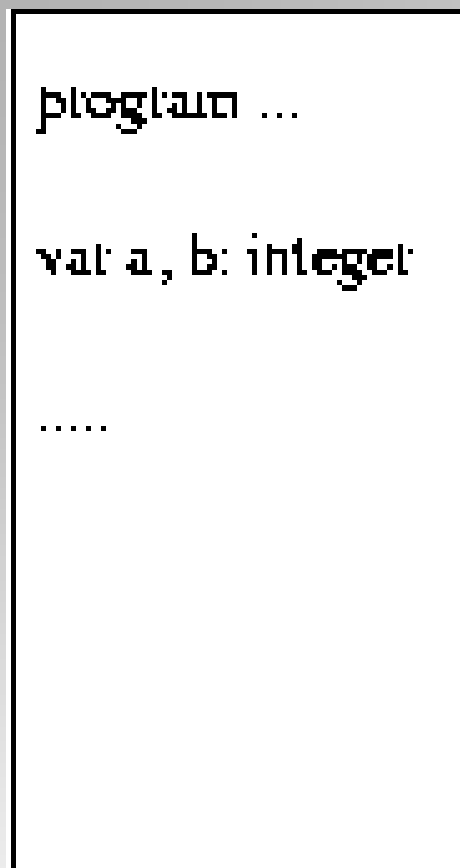
- esquemas de endereçamento hardware
- limitações do espaço (real e físico) de memória
- necessidade de partilhar a memória central por múltiplos programas utilizadores (devido a multiprogramação)

Funções do SO

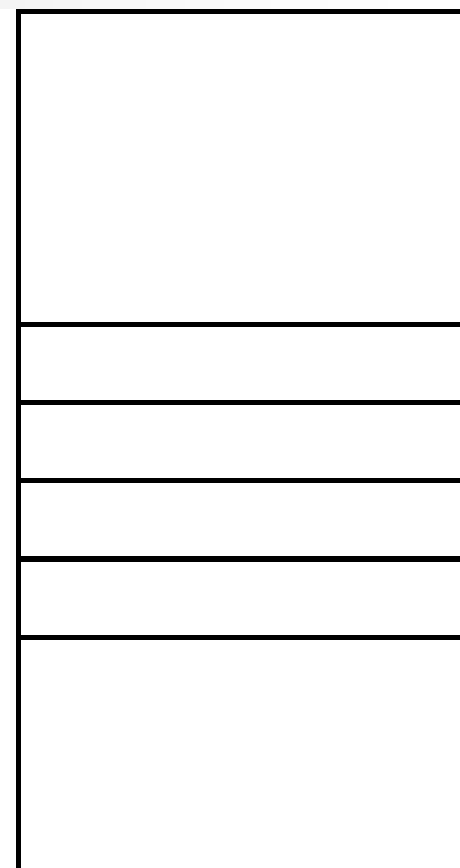
manter estruturas de dados sobre o estado da memória

estratégias de atribuição de memória aos processos (carregar em M/ remover para disco)

protecção entre os mapas de memória de processos diferentes (com suporte do hardware)



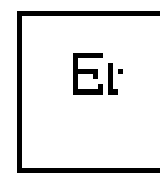
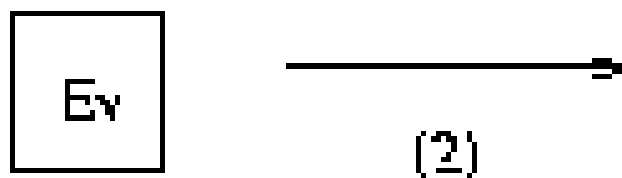
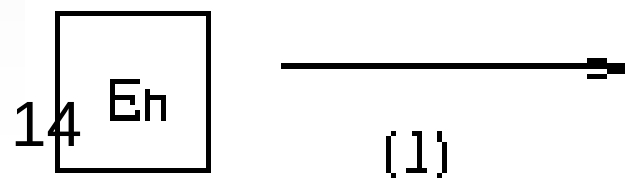
1000
1001
1002
1003



Espaco de nomes
do programa
Pascal (En)

Espaco de
enderecos
virtuais (Ev)

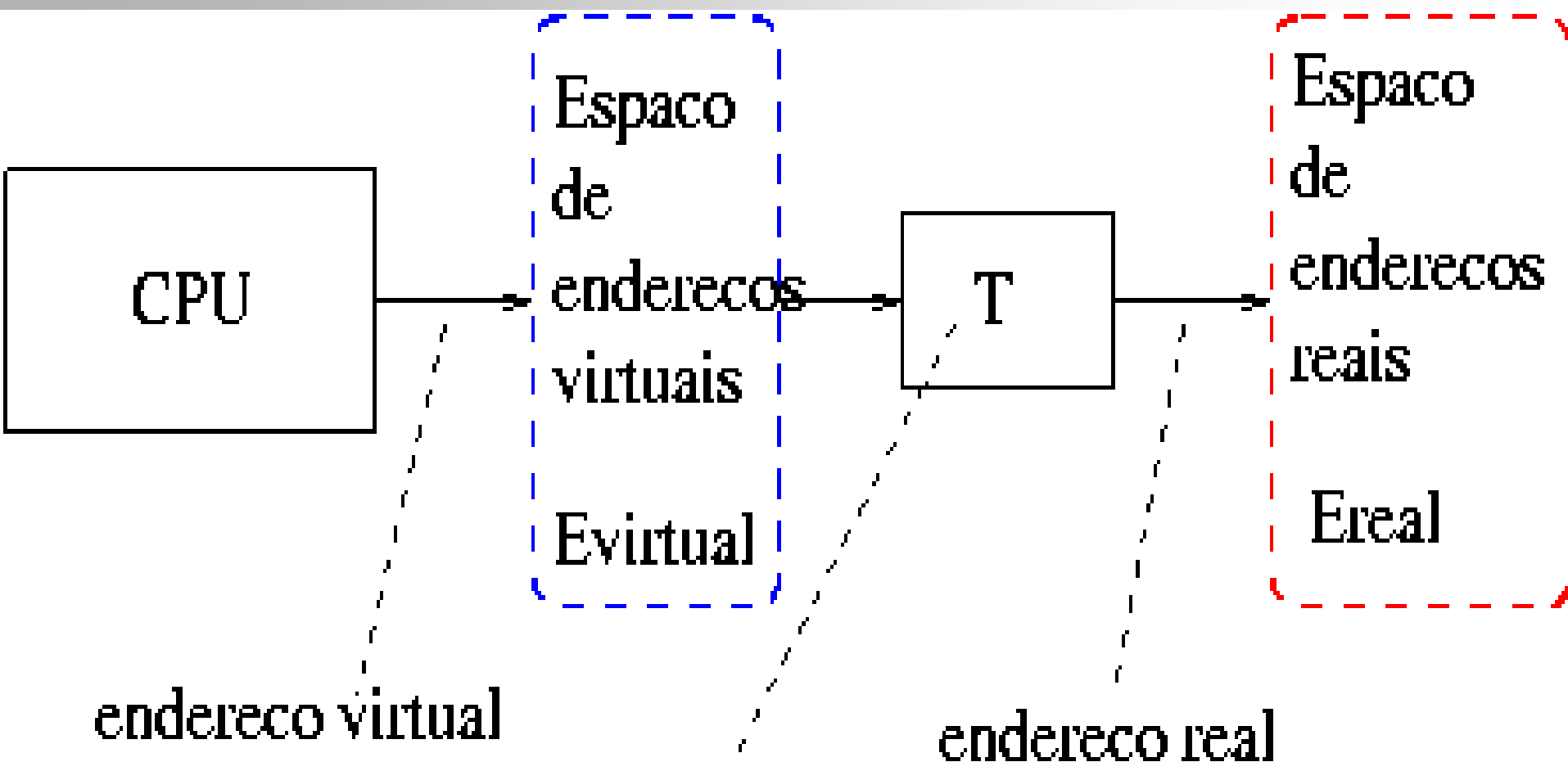
Espaco de
enderecos
reais (Er)



Transformação de endereços

Diversas abordagens:

- (1) os endereços de memória são já os endereços absolutos reais, quando se escreve o programa;
- (2) os endereços de memória são relocáveis e só receberão valores absolutos reais por acção de um programa Carregador, antes de se iniciar a execução, passando a ser válidos durante toda a execução do programa (Recolocação Estática);
- (3) os endereços de memória são relocáveis e só serão calculados os valores reais absolutos, a cada referência de memória, pelo CPU, durante a execução (Recolocação Dinâmica).



Unidade de Transformacao
de Enderecos
(MMU—Memory Management Unit)

Endereços virtuais: definidos pelo programa a nível das instruções de referência de memória e dependendo dos modos de endereçamento.

A dimensão máxima e a organização do Espaço de Endereços Virtuais (EV) é determinada pelo endereço efectivo gerado pelas instruções máquina.

Endereços reais: definidos pelas linhas de endereço do Bus que dão acesso às células físicas de Memória Central.

A dimensão máxima do Espaço de Endereços Reais (ER) é definida pelo número de linhas de endereço do Bus.

Espaço de **endereços físicos (EF):** definidos pela capacidade de memória central instalada em cada computador.

Em geral $EV \geq ER \geq EF$

Separação de Espaços EV e ER

torna independentes as organizações dos dois espaços;

permite que o Espaço Virtual dum processo seja independente do Espaço Real, face a:

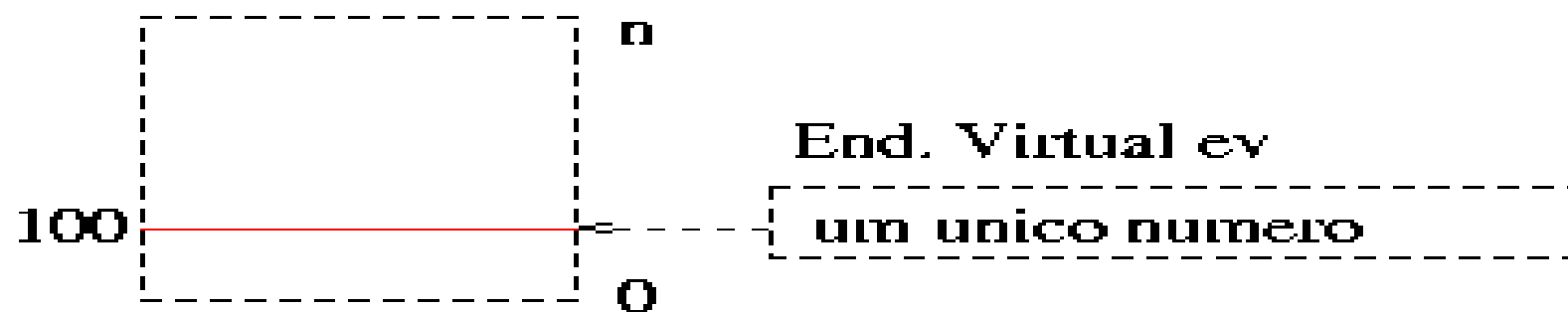
dimensão máxima → Memória Virtual

localização de endereços reais de memória que é atribuída a cada processo

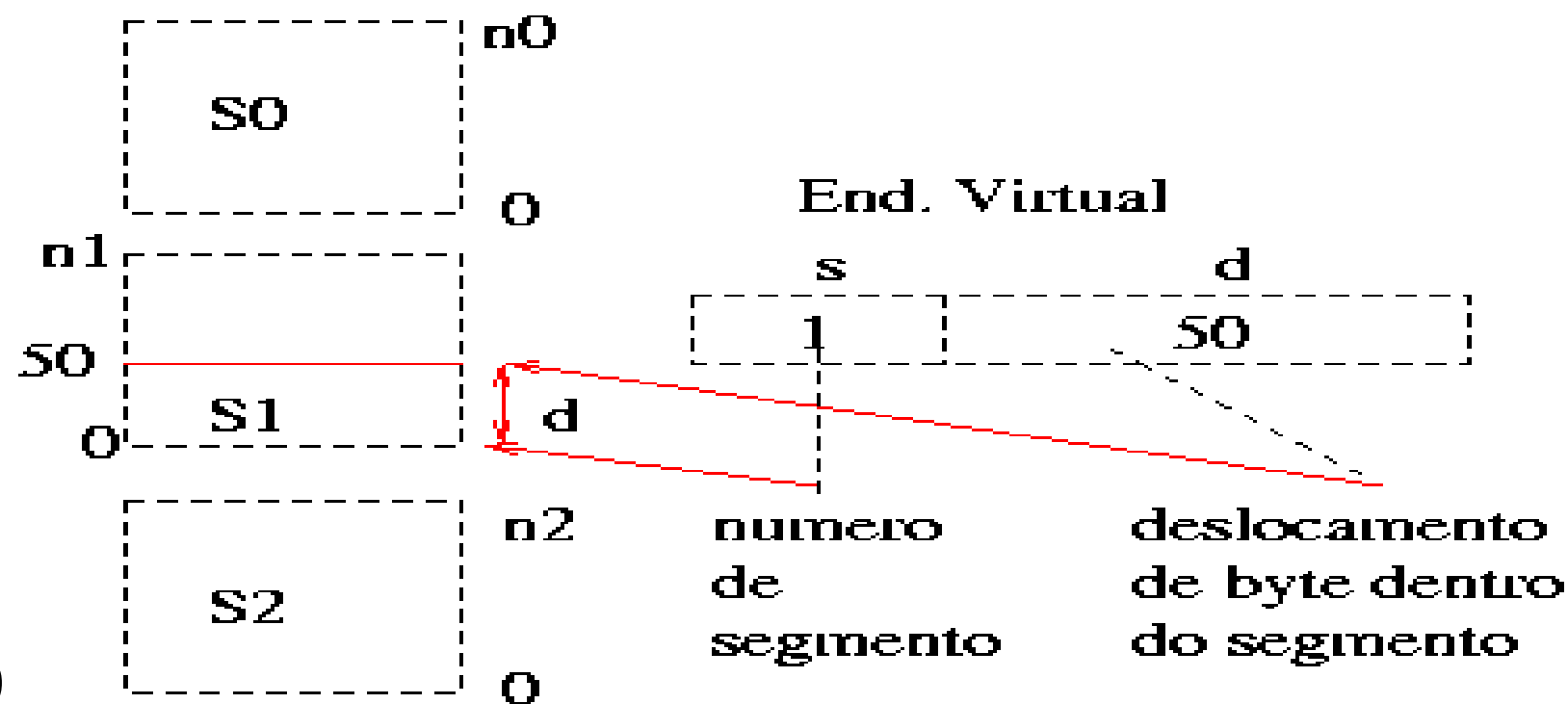
→ Recolocação dinâmica

Organizacao do Espaco de Enderecos Virtuais

a) Espaço Virtual Linear: um unico segmento logico
posicoes logicamente contiguas

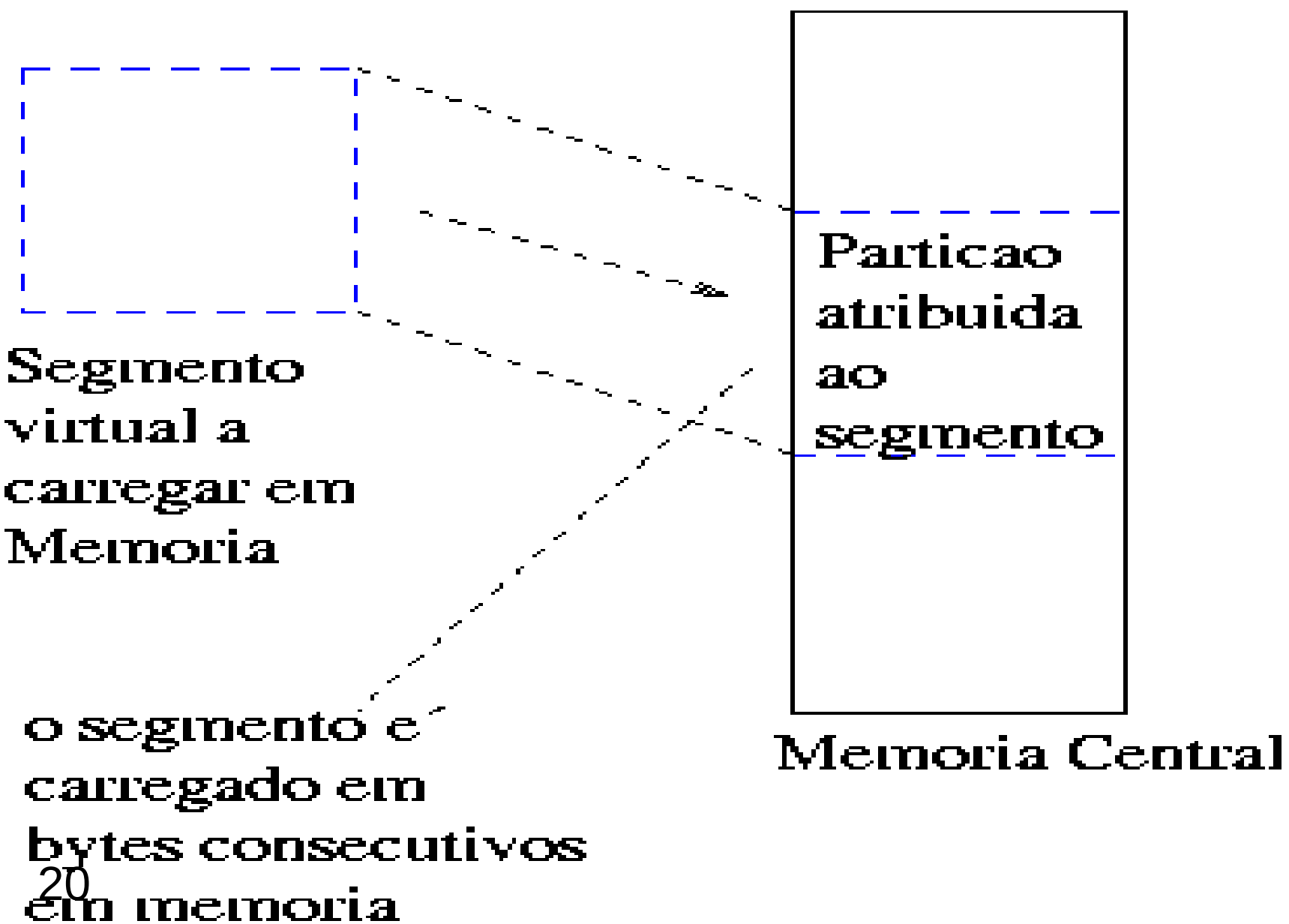


b) Espaço Virtual Segmentado



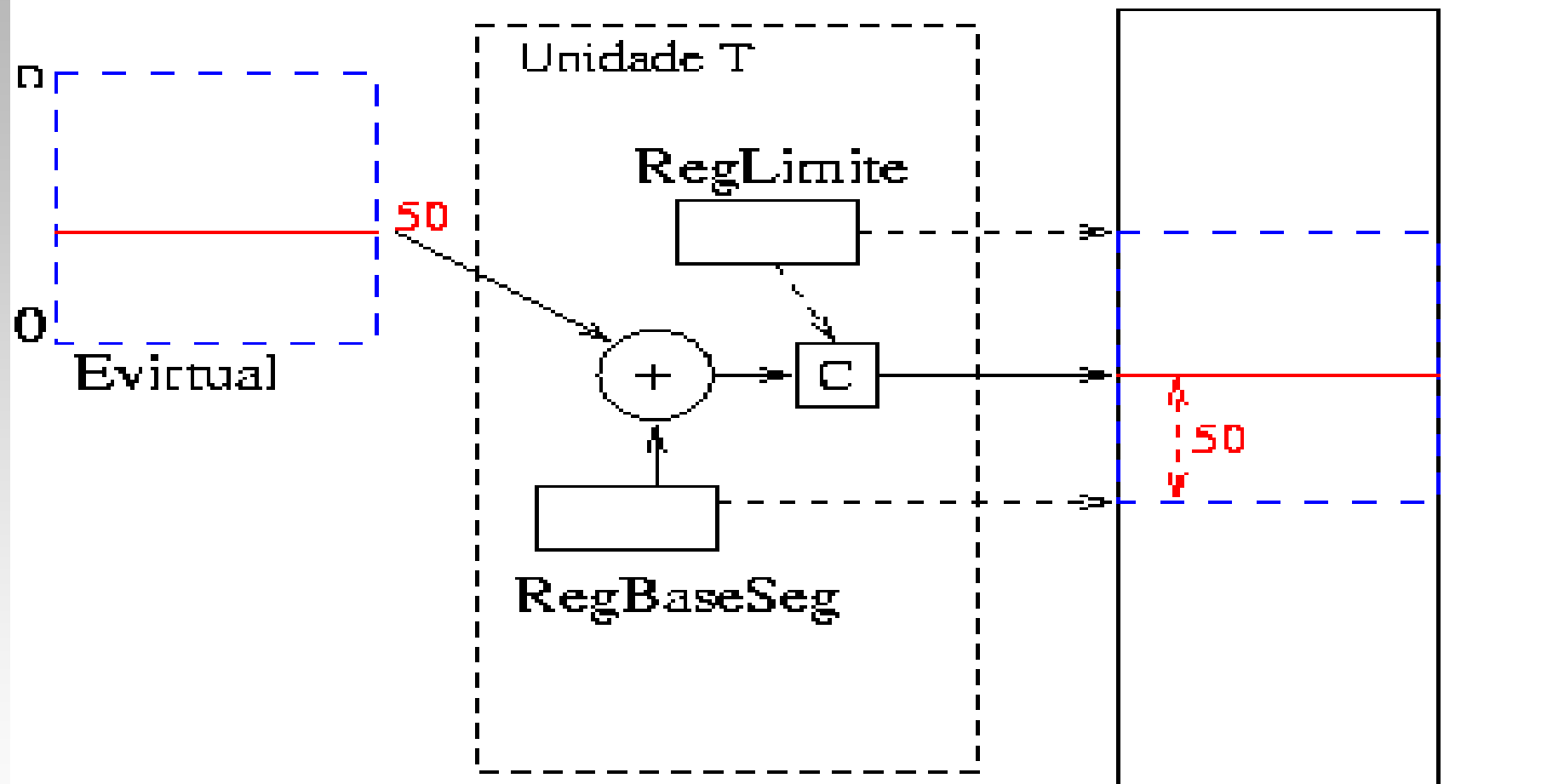
Organizacao do Espaco de Enderecos Reais

a) Particoes de memoria de tamanho variavel



Organizacao do Espaco de Enderecos Reais

a) Particoes de memoria de tamanho variavel



Transformacao de enderecos
feita pela unidade T, a cada
referencia de memoria gerada
pelo CPU

Memoria Central

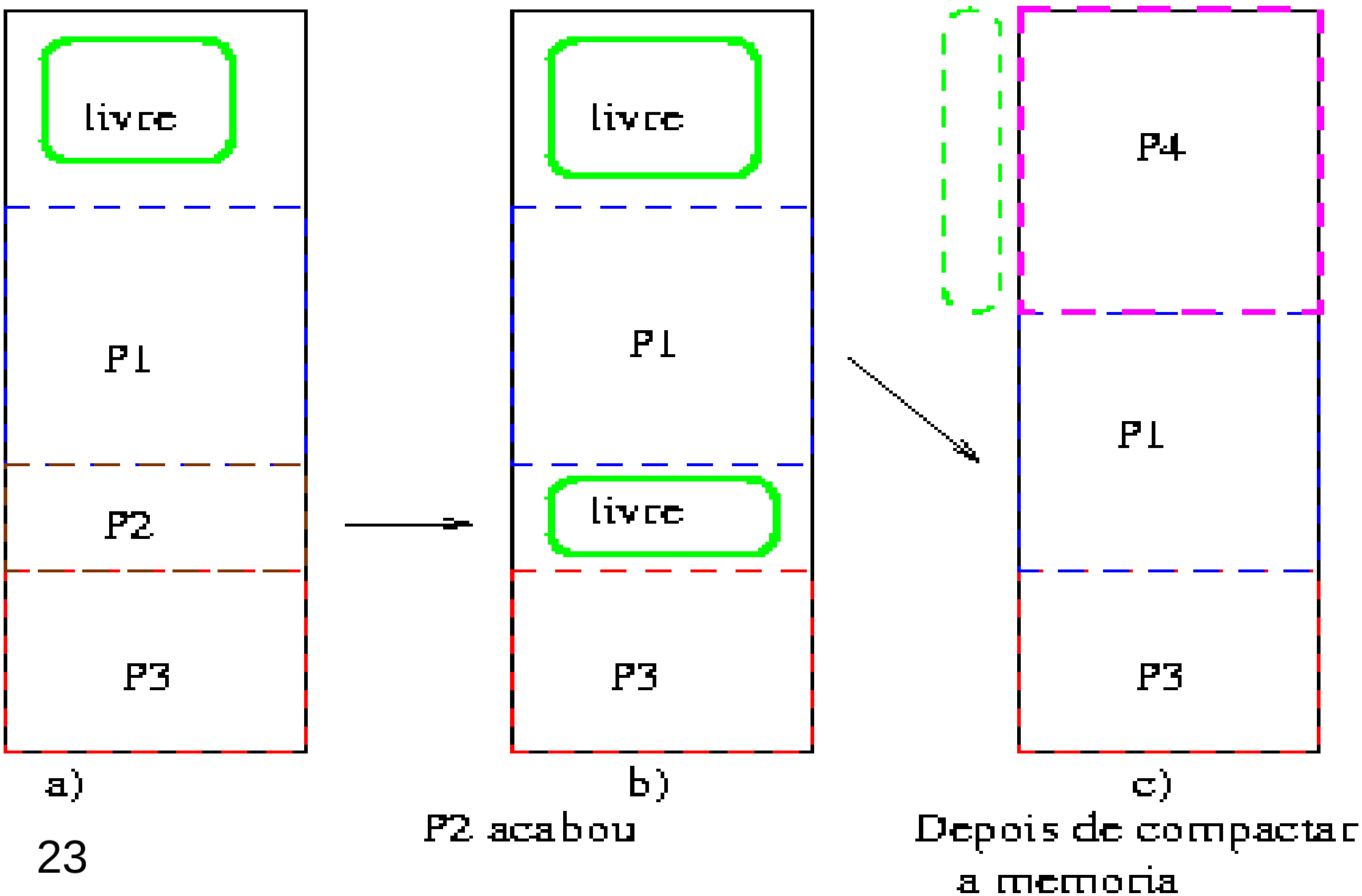
Transformação de endereços T1

EV linear → ER com partições de tamanho variável

- garante fácil recolocação
- garante independência de EV em relação à localização física em memória
- gestão elaborada do espaço de memória central
- limita a dimensão máxima de EV a ser menor ou igual à máxima memória central disponível → exige o carregamento em memória de todo o módulo executável.

Organizacao do Espaco de Enderecos Reais

a) Particoes de memoria de tamanho variavel



Transformação de endereços T2

EV linear → ER com páginas de tamanho fixo por hardware

Paginação: técnica de gestão de memória que subdivide o Espaço de endereços reais ER em zonas iguais:

→ blocos ou **páginas reais** de memória

para facilitar a gestão de memória.

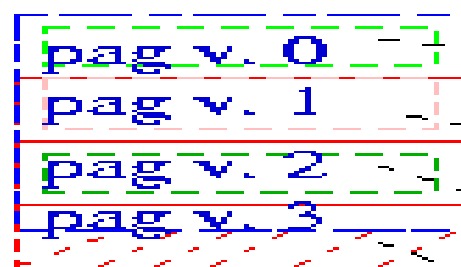
O espaço EV, para efeitos do carregamento em Memória, é considerado subdividido, em **páginas virtuais** (ou lógicas) de dimensão igual à das páginas reais de memória.

As páginas virtuais ficam contíguas no espaço virtual EV.

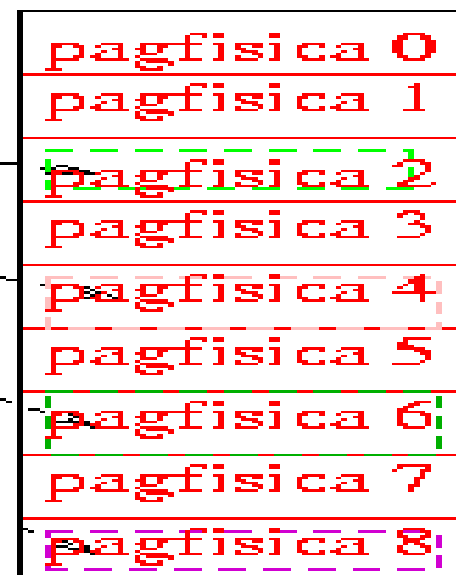
As páginas reais podem ficar espalhadas pela Memória, consoante as zonas de memória livres: não necessariamente contíguas.

Organizacao do Espaco de Enderecos Reais

b) Paginas de memoria com tamanho fixo



Segmento
virtual a
carregar em
Memoria



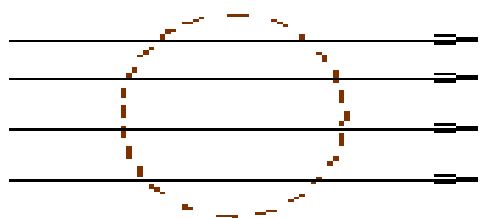
pagfisica 0: enderecos
desde 0
ate' 1023

pagfisica 1: enderecos
desde 1024
ate' 2047

Memoria Central

(paginas fisicas de
1 KiloByte

pag v. 0
pag v. 1
pag v. 2
pag v. 3



pagfisica 2
pagfisica 4
pagfisica 6
pagfisica 8

Evirtual

T

Ereal

Interpretação do endereço virtual

O endereço virtual ev é gerado pelo CPU.

Interpretação do endereço virtual

O endereço virtual ev é gerado pelo CPU.

Antes de ser enviado para a Memória, o endereço ev é interpretado pela unidade T de transformação de endereços.

Interpretação do endereço virtual

O endereço virtual ev é gerado pelo CPU.

Antes de ser enviado para a Memória, o endereço ev é interpretado pela unidade T de transformação de endereços.

A unidade T calcula, com base no endereço ev , qual é o número de página virtual correspondente e qual é o deslocamento do byte dentro dessa página.

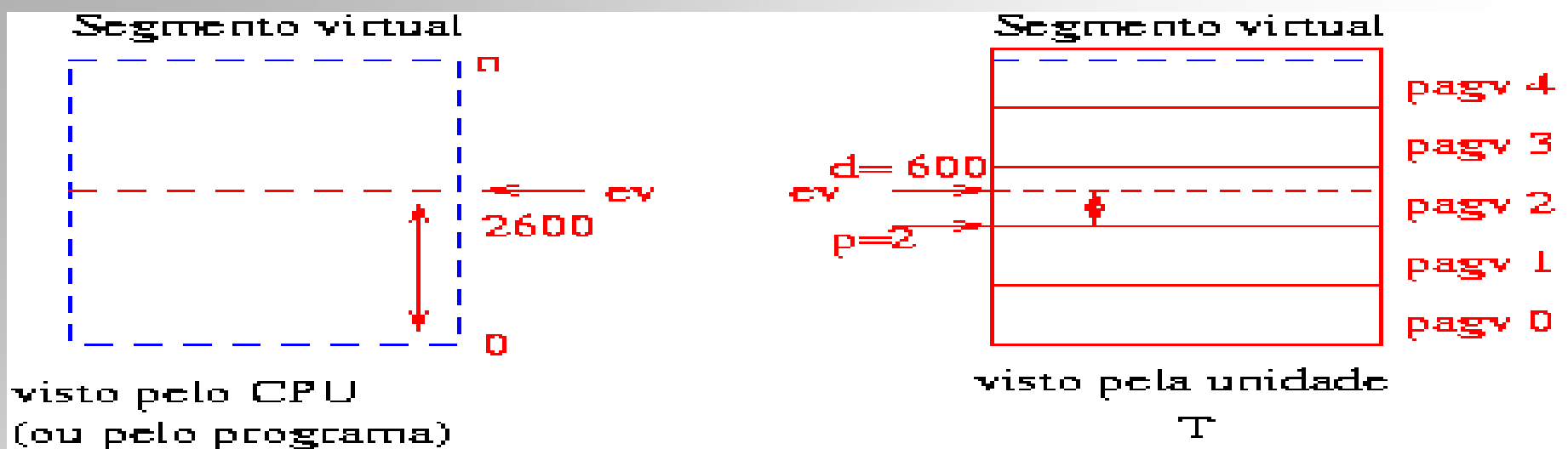
Interpretação do endereço virtual

O endereço virtual ev é gerado pelo CPU.

Antes de ser enviado para a Memória, o endereço ev é interpretado pela unidade T de transformação de endereços.

A unidade T calcula, com base no endereço ev , qual é o número de página virtual correspondente e qual é o deslocamento do byte dentro dessa página.

Esta interpretação é completamente invisível ao Programa.



Tamanho da pagina = DIM = 1000 (base 10)

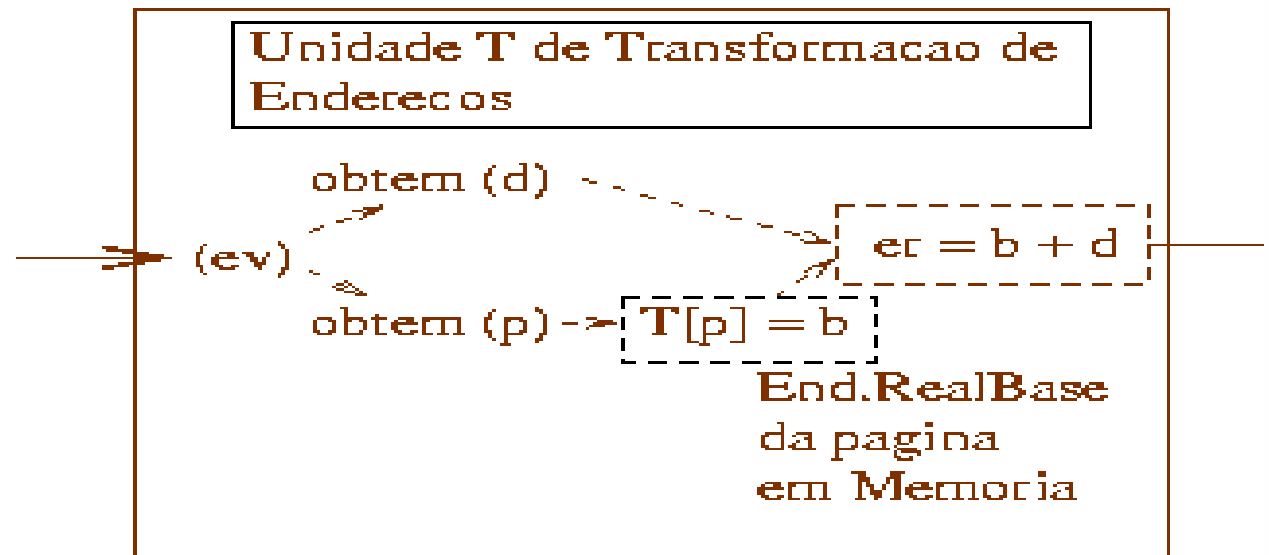
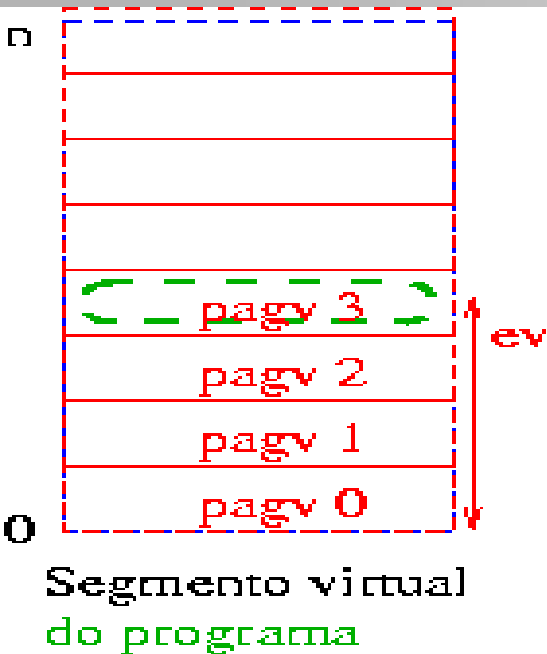
ev = 2600

interpretado como

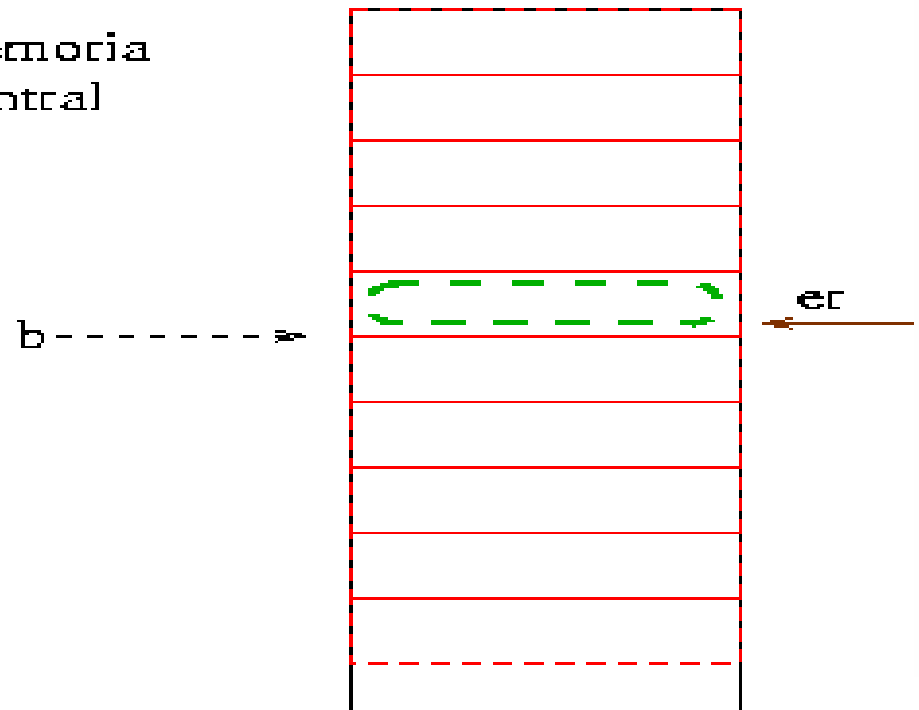
$p = ev \text{ div } 1000 = 2 \longrightarrow \text{pagina virtual 2}$

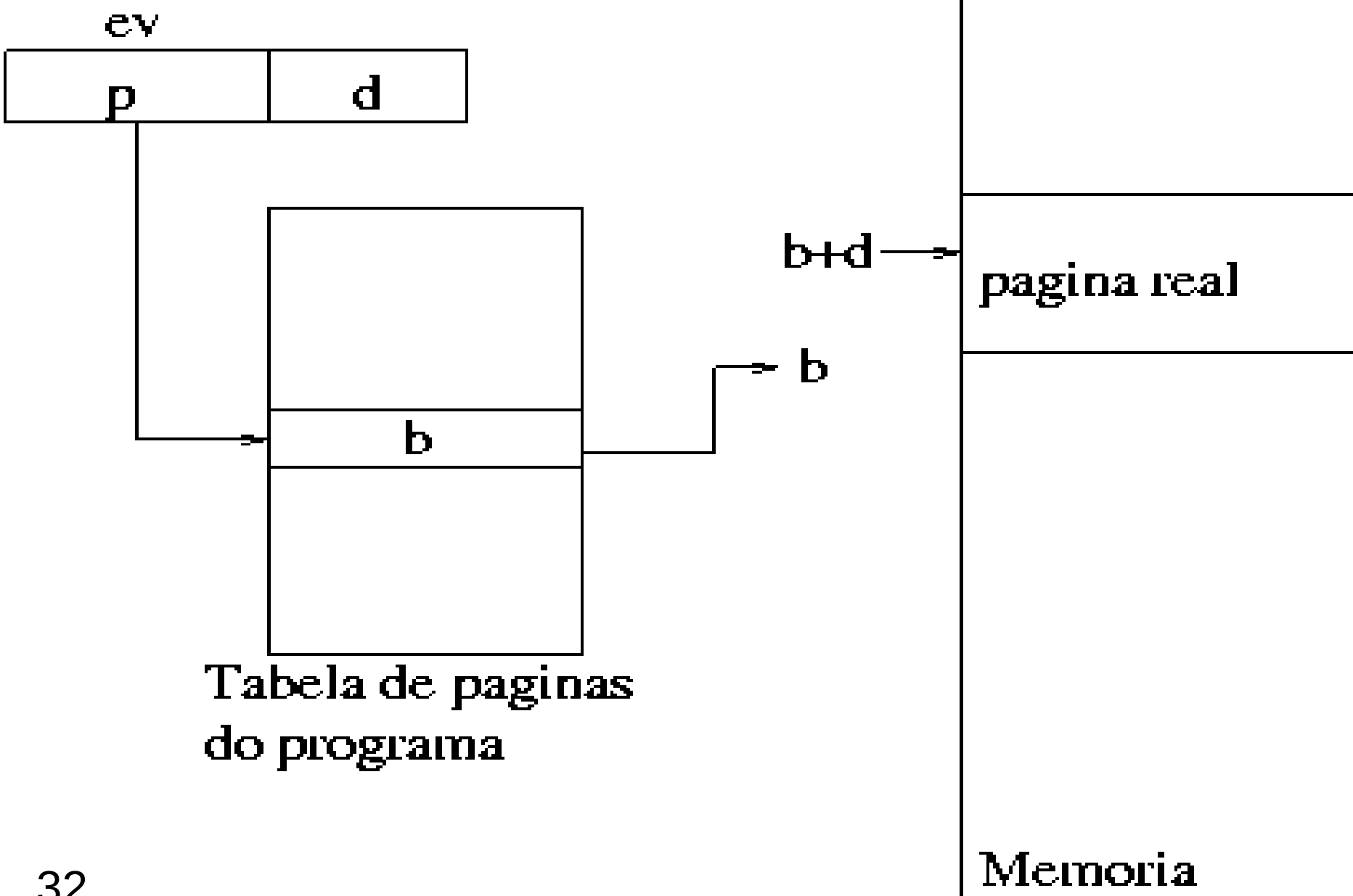
$d = ev - p * 1000 = 600 \longrightarrow \text{deslocamento}$
de byte dentro da pagina

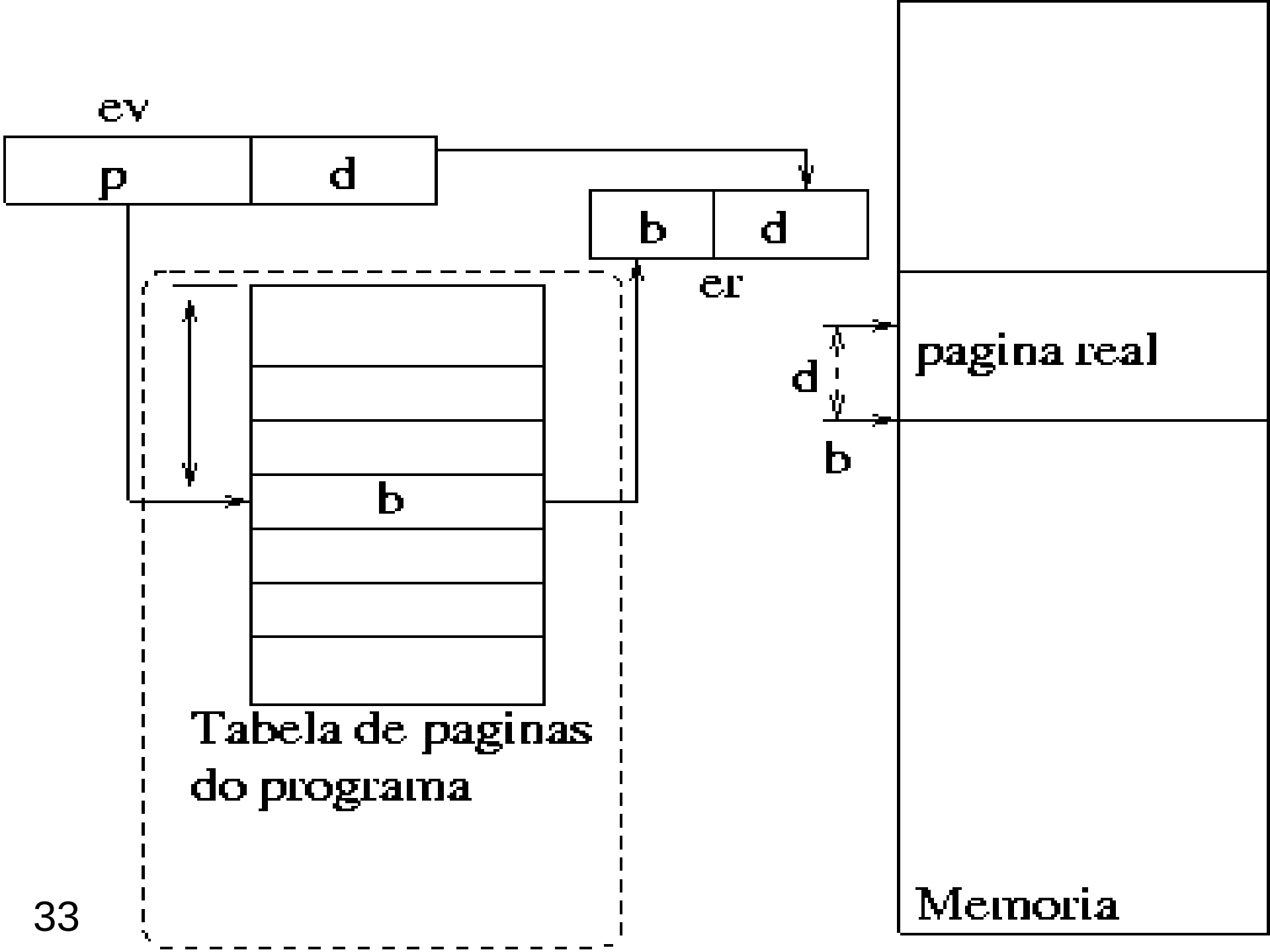
A unidade T usa 2 como indice na Tabela de Paginas do Programa, onde foi colocado o Endereco Real de Base da Pagina Fisica em Memoria onde o SO carregou a Pagina Virtual 2.



Memoria Central







Exemplo: Espaço Virtual EV de 4 GigaBytes

Endereço virtual de 32 bits

31 30 2912 11 100

Exemplo: Espaço Virtual EV de 4 GigaBytes

Endereço virtual de 32 bits

31 30 2912 11 100

Páginas de 4 KiloBytes: (definidas pelo *Hardware*)

O deslocamento do byte é indicado por um nº de 12 bits

Exemplo: Espaço Virtual EV de 4 GigaBytes

Endereço virtual de 32 bits

31 30 2912 11 100

Páginas de 4 KiloBytes:

O deslocamento do byte é indicado por um nº de 12 bits

O máximo EV contém ??? páginas = ??? páginas:

Exemplo: Espaço Virtual EV de 4 GigaBytes

Endereço virtual de 32 bits

31 30 2912 11 100

Páginas de 4 KiloBytes:

O deslocamento do byte é indicado por um nº de 12 bits

O máximo EV contém $2^{32} / 2^{12}$ páginas = 2^{20} páginas:

Exemplo: Espaço Virtual EV de 4 GigaBytes

Endereço virtual de 32 bits

31 30 2912 11 100

Páginas de 4 KiloBytes:

O deslocamento do byte é indicado por um nº de 12 bits

O máximo EV contém $2^{32} / 2^{12}$ páginas = 2^{20} páginas:

O nº de página tem 20 bits

31 30 2912 11 100



Número da Página

Deslocamento Byte

Vantagens da Paginação

Facilita a gestão eficiente de memória: quaisquer páginas reais de memória livres servem para carregar páginas virtuais.

Vantagens da Paginação

Facilita a gestão eficiente de memória: quaisquer páginas reais de memória livres servem para carregar páginas virtuais.

Não se exige atribuição em zonas contíguas de memória: não há fragmentação da memória.

Vantagens da Paginação

Facilita a gestão eficiente de memória: quaisquer páginas reais de memória livres servem para carregar páginas virtuais.

Não se exige atribuição em zonas contíguas de memória: não há fragmentação da memória.

Para o carregamento de programas em memória, permite 2 casos

a) Pré-carregamento de todas as páginas virtuais de um programa em Memória, antes de iniciar a execução.

Vantagens da Paginação

Facilita a gestão eficiente de memória: quaisquer páginas reais de memória livres servem para carregar páginas virtuais.

Não se exige atribuição em zonas contíguas de memória: não há fragmentação da memória.

Para o carregamento de programas em memória, permite 2 casos

a) Pré-carregamento de todas as páginas virtuais de um programa em Memória, antes de iniciar a execução.

b) Carregamento a pedido (*On-demand paging*): esquema dinâmico de carregamento de páginas:

Vantagens da Paginação

Facilita a gestão eficiente de memória: quaisquer páginas reais de memória livres servem para carregar páginas virtuais.

Não se exige atribuição em zonas contíguas de memória: não há fragmentação da memória.

Para o carregamento de programas em memória, permite 2 casos

- a) Pré-carregamento de todas as páginas virtuais de um programa em Memória, antes de iniciar a execução.
- b) Carregamento a pedido (*On-demand paging*): esquema dinâmico de carregamento de páginas:

só carrega uma dada página virtual, no momento da 1ª referência, durante a execução: quando o endereço é gerado pelo CPU, a unidade T de transformação de endereços detecta se a correspondente página virtual já está carregada em memória ou não:

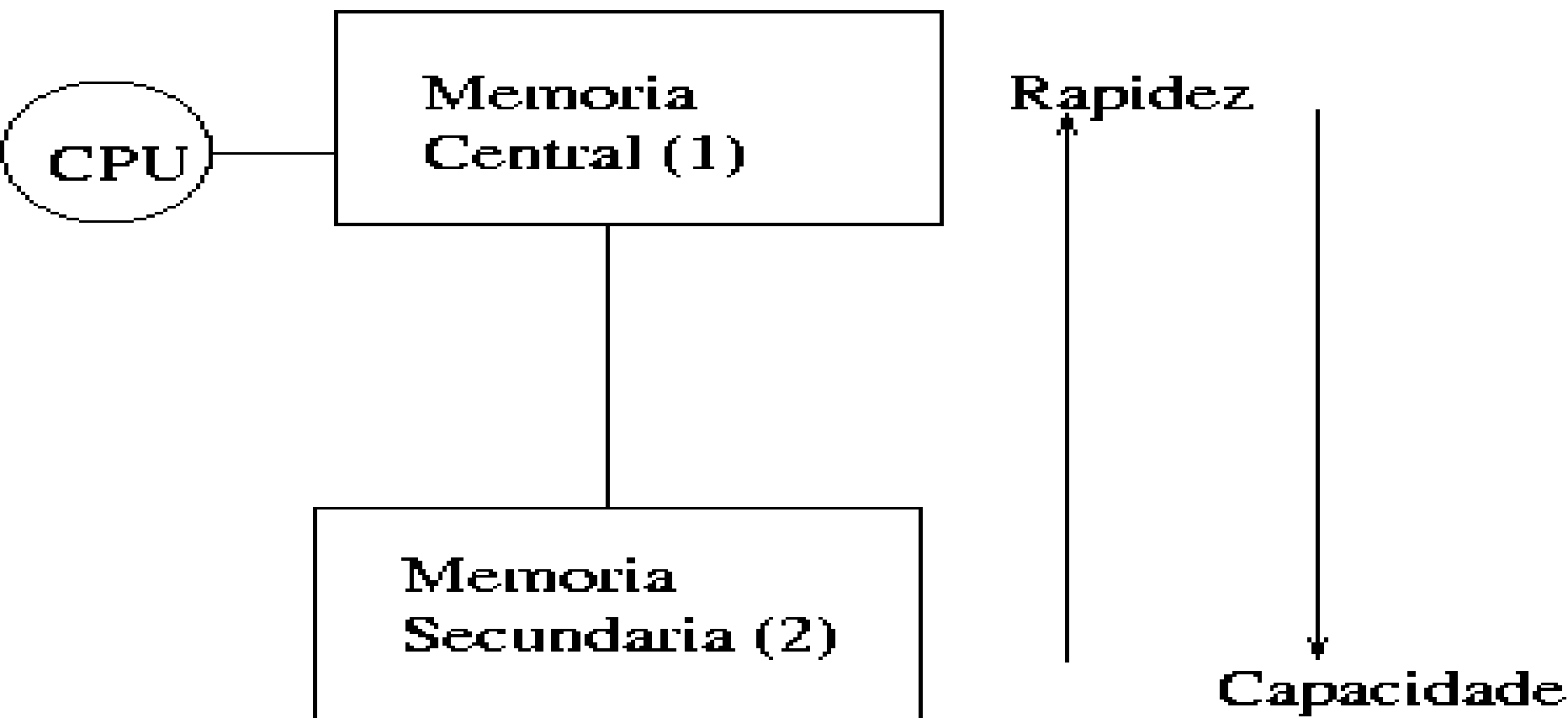
- Se a página virtual referida não estiver ainda em memória, a unidade T gera um pedido de interrupção ao CPU (*Page-fault interrupt*):

- a rotina de serviço de interrupções deste tipo:

- Desencadeia o carregamento da página virtual pedida, de disco para memória central

- Consegue-se, assim, simular uma memória virtual cuja capacidade é definida pelo tamanho dos endereços virtuais (por exemplo 4 GigaBytes), que pode ser superior à capacidade do Espaço de endereços reais e à capacidade da Memória central instalada.

Dois níveis na hierarquia de memória



Um esquema dinâmico:

- detecção de referências a páginas ausentes de memória
- gestão de transferências entre a Memória e o Disco

Paginação dinâmica, por pedido:

Hardware adicional:

na Tabela de Páginas: um bit indicador de Presença / Ausência de página em Memória

(inicializado pelo SO, por omissão, todos os bits a 0 no início...)

Paginação dinâmica, por pedido:

Hardware adicional:

na Tabela de Páginas: um bit indicador de Presença / Ausência de página em Memória

(inicializado pelo SO, por omissão, todos os bits a 0 no início)

suporte para tipo de interrupções por Falta de Página:

- a unidade T deve poder determinar qual o endereço virtual ev que gerou a situação de 'página ausente';

Paginação dinâmica, por pedido:

Hardware adicional:

na Tabela de Páginas: um bit indicador de Presença / Ausência de página em Memória

(inicializado pelo SO, por omissão, todos os bits a 0 no início)

suporte para tipo de interrupções por Falta de Página:

- a unidade T deve poder determinar qual o endereço virtual ev que gerou a situação de 'página ausente';

- a instrução que gerou essa referência deve poder ser repetida, desde o seu início.

Paginação dinâmica, por pedido:

Hardware adicional:

na Tabela de Páginas: um bit indicador de Presença / Ausência de página em Memória

(inicializado pelo SO, por omissão, no início da execução de um processo)

suporte para tipo de interrupções por Falta de Página:

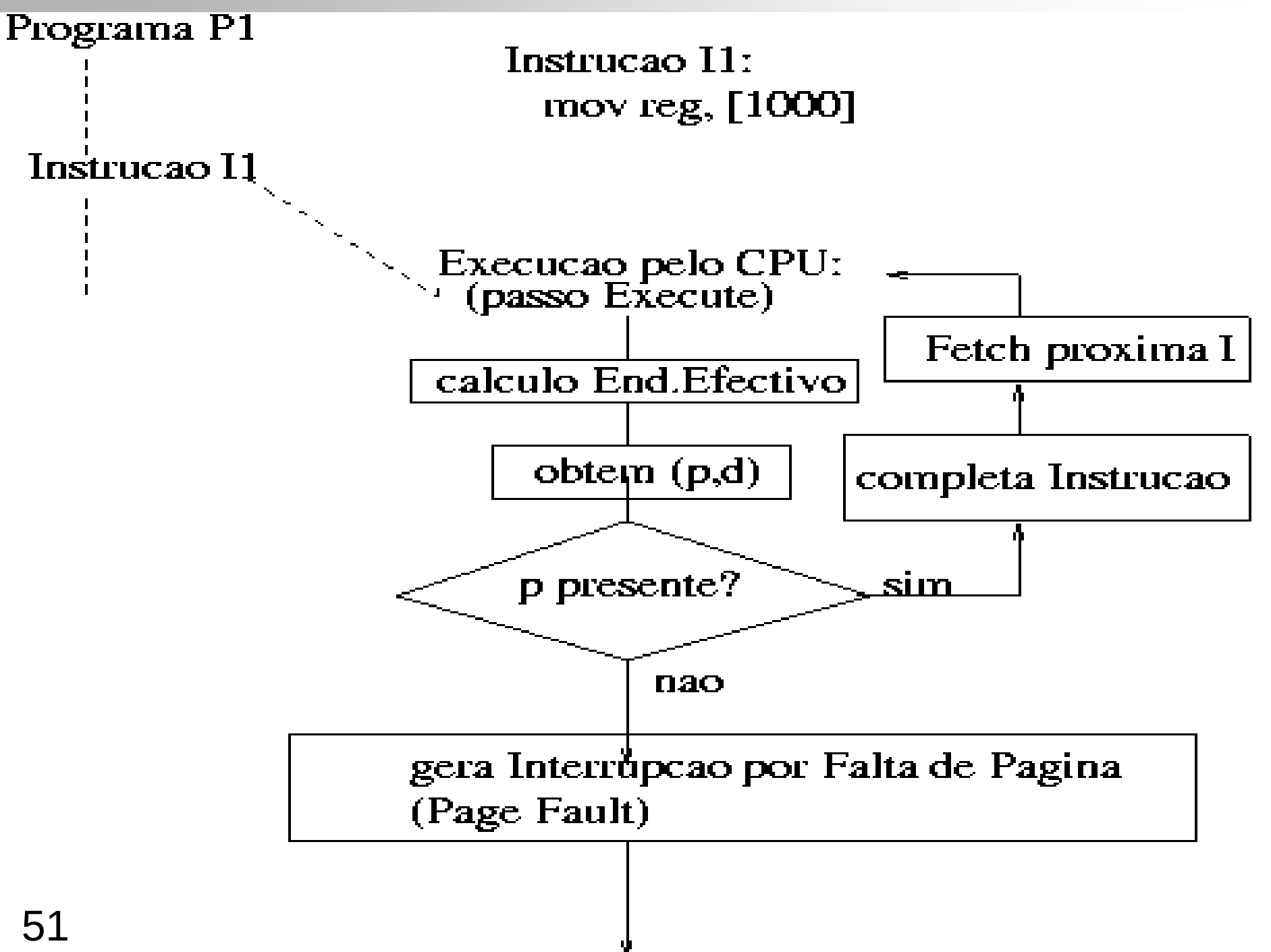
- a unidade T deve poder determinar qual o endereço virtual ev que gerou a situação de 'página ausente';
- a instrução que gerou essa referência deve poder ser repetida, desde o seu início.

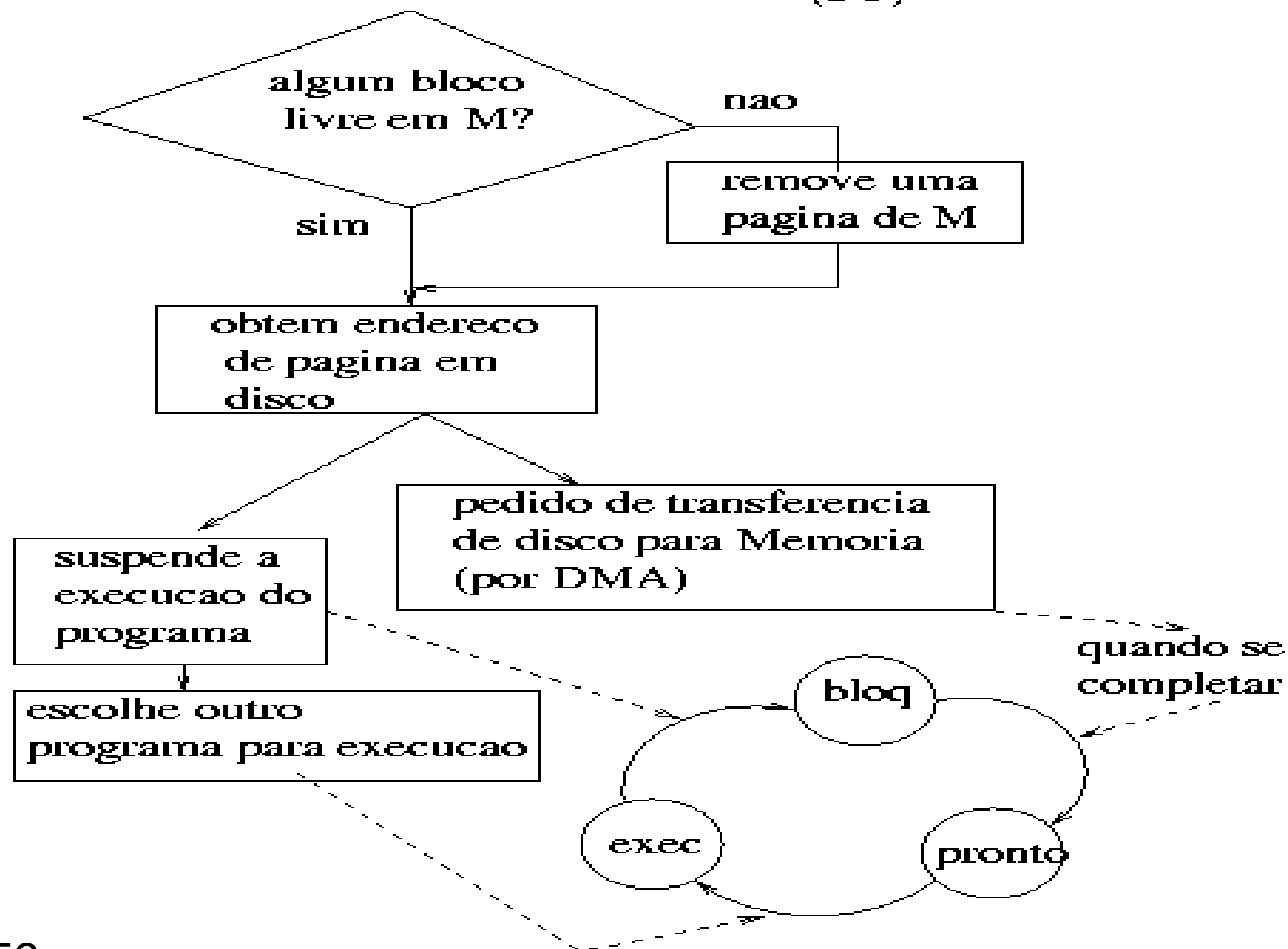
Software adicional (do SO):

rotina de serviço de interrupções por falta de página

rotinas de controlo da transferência de páginas disco-Mem

rotinas de gestão do espaço de memória, em termos de páginas





Características da Paginação:

- Elimina fragmentação externa de memória
-

Características da Paginação:

- Elimina fragmentação externa de memória
- Apresenta fragmentação interna do módulo do programa, se este tiver um tamanho que não é múltiplo do tamanho da página

Características da Paginação:

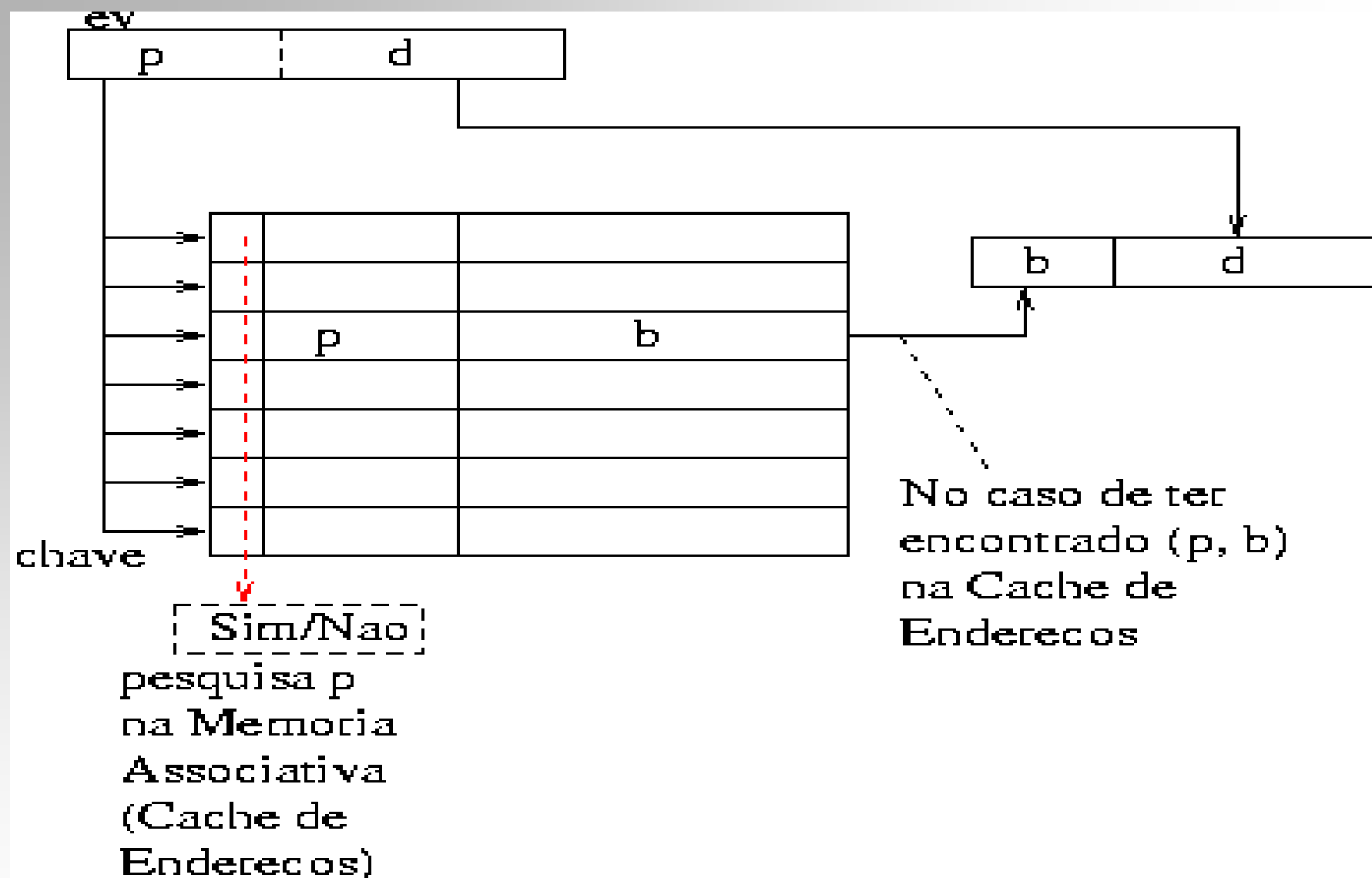
- Elimina fragmentação externa de memória
- Apresenta fragmentação interna do módulo do programa, se este tiver um tamanho que não é múltiplo do tamanho da página
- Com paginação por pedido, suporta Memória Virtual, tal que, a nível do programa, o espaço de endereçamento "visto" é uma memória linear, de dimensão superior à da memória real

Características da Paginação:

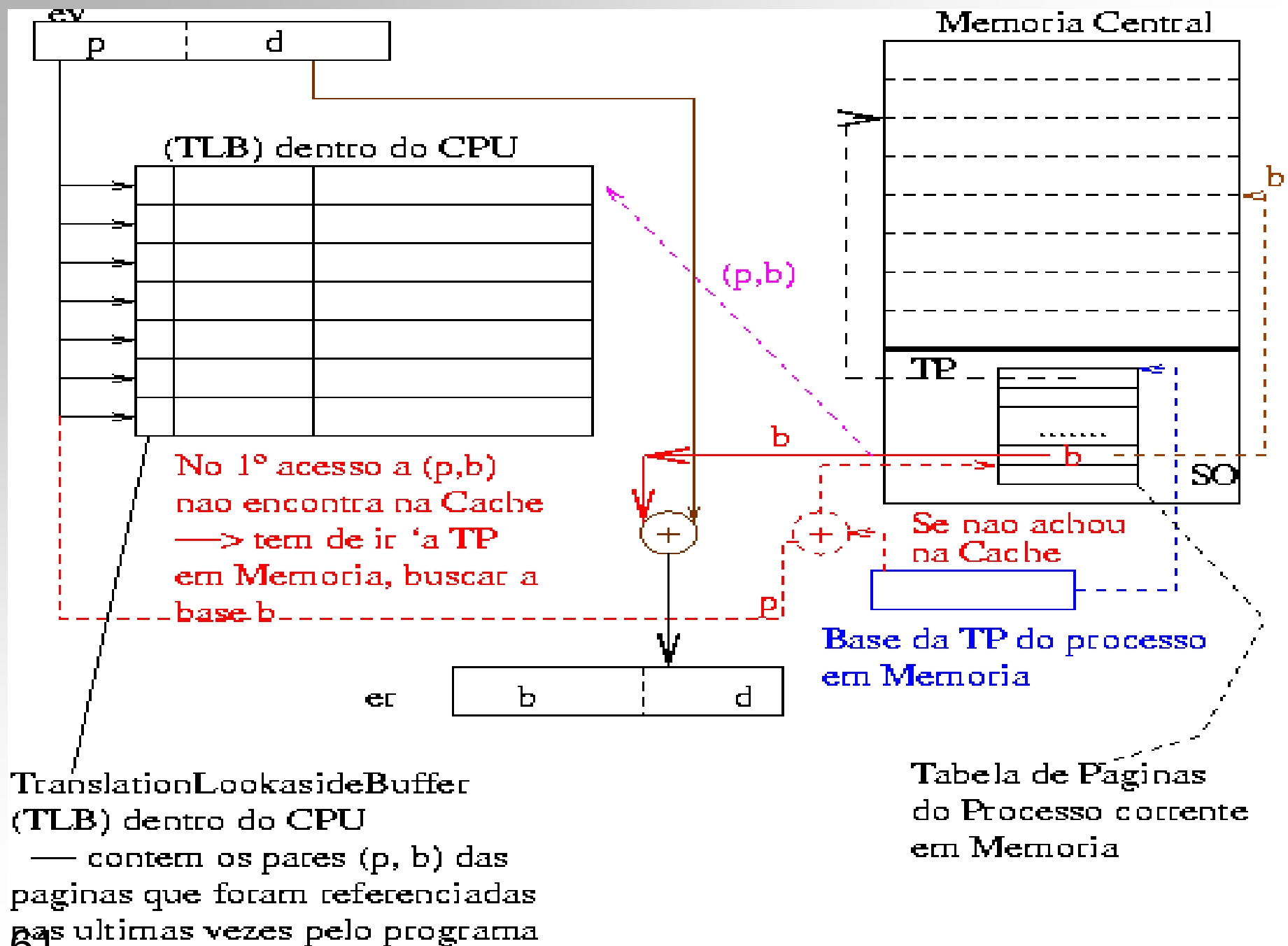
- Elimina fragmentação externa de memória
- Apresenta fragmentação interna do módulo do programa, se este tiver um tamanho que não é múltiplo do tamanho da página
- Com paginação por pedido, suporta Memória Virtual, tal que, a nível do programa, o espaço de endereçamento “visto” é uma memória linear, de dimensão superior à da memória real
- Exige suporte hardware adicional e eficiência nas transferências entre memória central e secundária

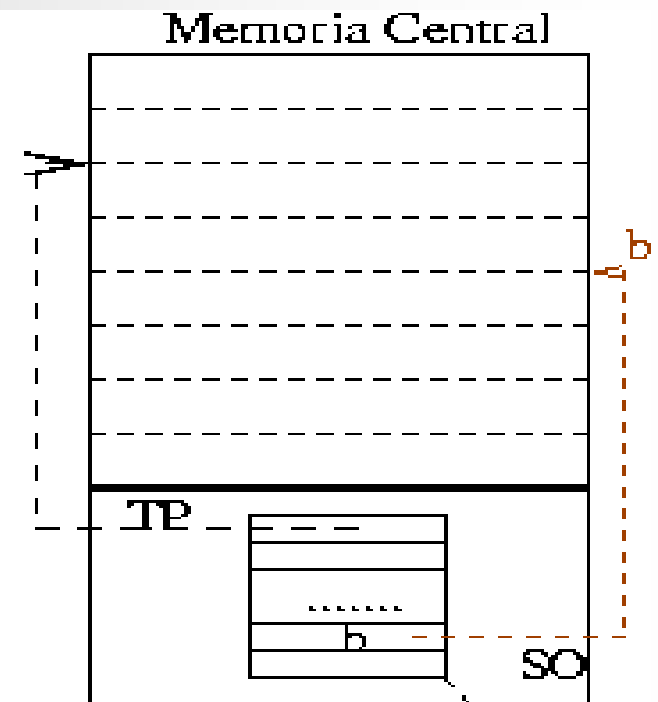
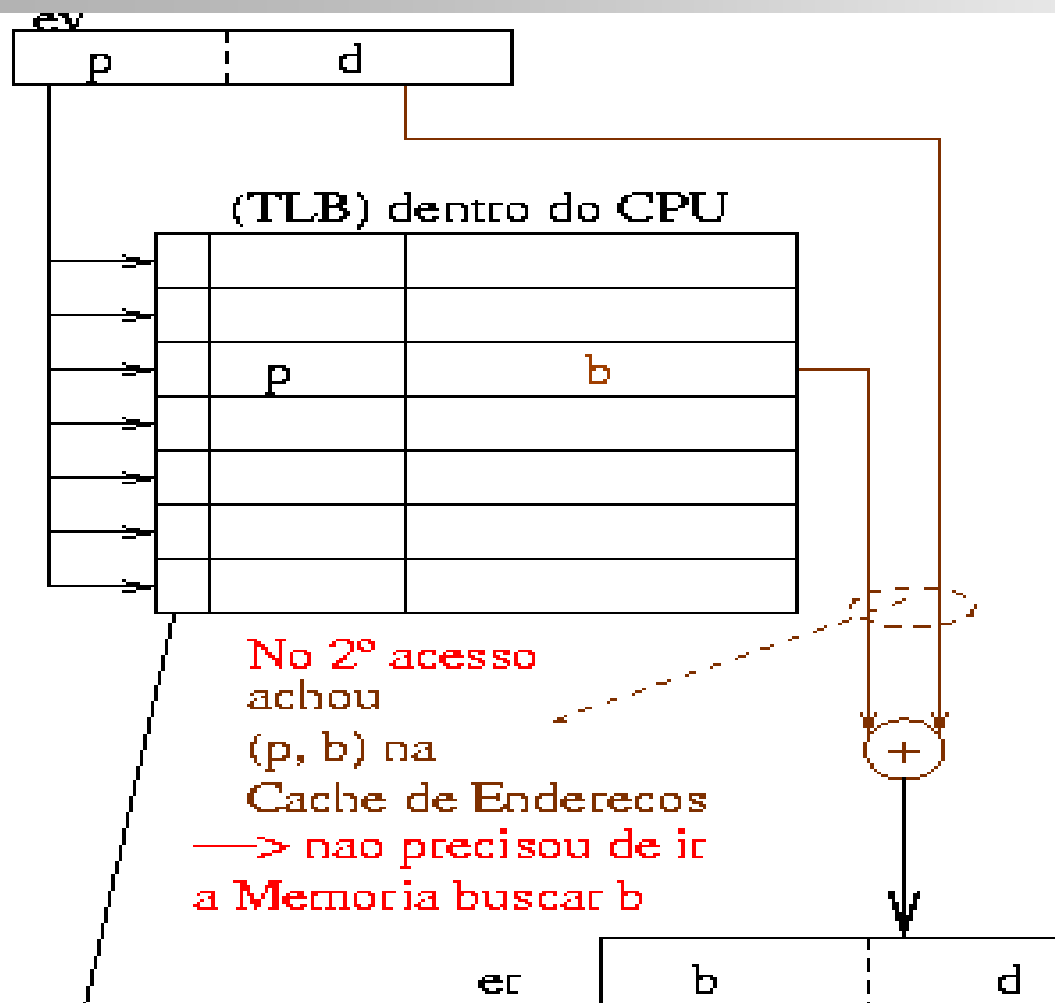
Características da Paginação:

- Elimina fragmentação externa de memória
- Apresenta fragmentação interna do módulo do programa, se este tiver um tamanho que não é múltiplo do tamanho da página
- Com paginação por pedido, suporta Memória Virtual, tal que, a nível do programa, o espaço de endereçamento ``visto`` é uma memória linear, de dimensão superior à da memória real
- Exige suporte hardware adicional e eficiência nas transferências entre memória central e secundária
- Exige suporte software, a nível do SO, com algoritmos que tentem otimizar a utilização do espaço de memória e escolher o melhor conjunto de páginas a manter carregadas em memória, a cada momento.



Chave: nº da pagina virtual
 Conteudo: base da pagina em Memoria





TranslationLookasideBuffer
(TLB) dentro do CPU
— contem os pares (p, b) das
paginas que foram referenciadas
nas ultimas vezes pelo programa

Tabela de Paginas
do Processo corrente
em Memoria

Pesquisa p na Memória
Associativa (TLB)

Achou p?

Sim

Nao

Acede a TP em Mem

p
presente ?

Sim

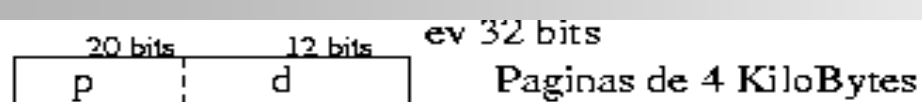
Actualiza
(p,b) em TLB

Obtem
 $er = (b, d)$

Nao

**GERA UMA
INTERRUPCAO
POR FALTA DE
PAGINA**

er
para aceder
ao byte d da
pagina p



(TLB) dentro do CPU

→		
→	0	32 * K
→		
→	1	4 * K
→		
→	3	28 * K
→		

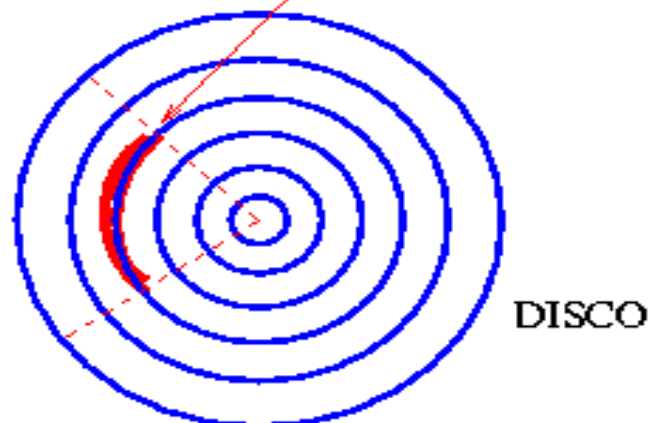
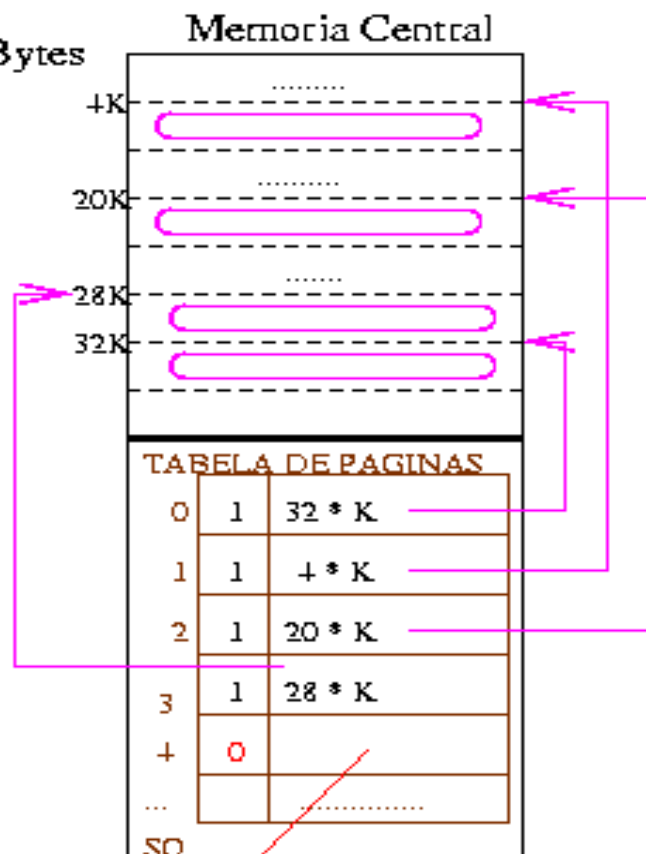
Sequencia de referencias:

(p = 0, d = 1)

(p = 1, d = 2)

(p = 3, d = 1)

(p = 4, d = 1)



20 bits 12 bits ev 32 bits
 0....100 0.....01
 Paginas de 4 KiloBytes

(TLB) dentro do CPU

	0	32 * K
	1	4 * K
	3	28 * K

Sequencia de referencias:

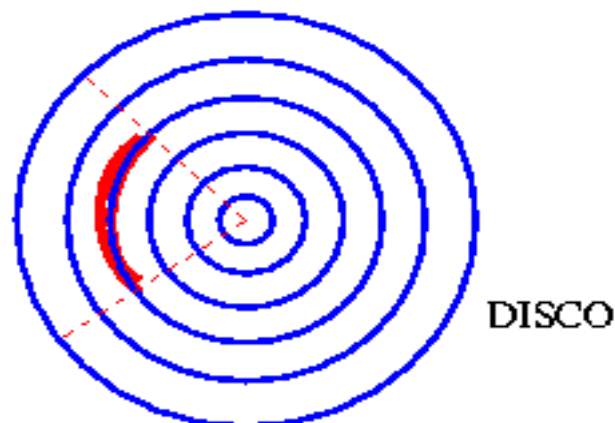
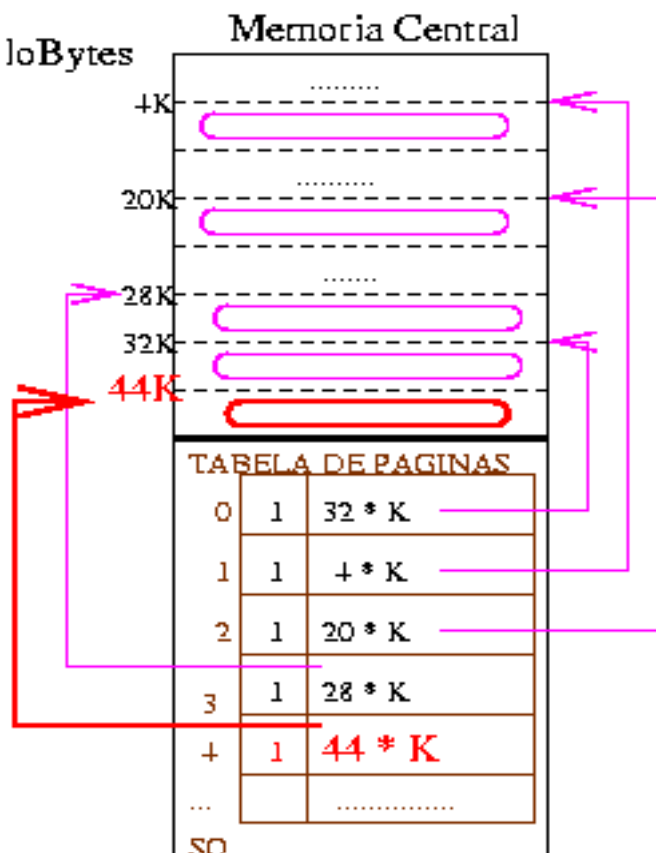
(p = 0, d= 1)

(p = 1, d=2)

(p = 3, d=1)

(p, = 4, d=1)

depois de completada
 a transferencia da pagina 4
 de disco para memoria



20 bits 12 bits ev 32 bits
 0....100 00.....01
 Paginas de 4 KiloBytes

(TLB) dentro do CPU

	0	32 * K
	1	4 * K
	4	44 * K
	3	28 * K

Sequencia de referencias:

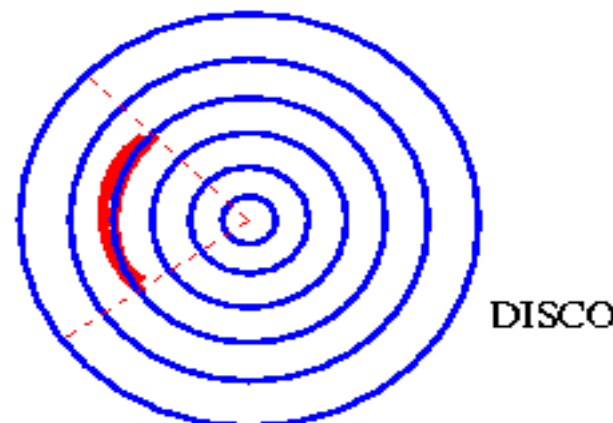
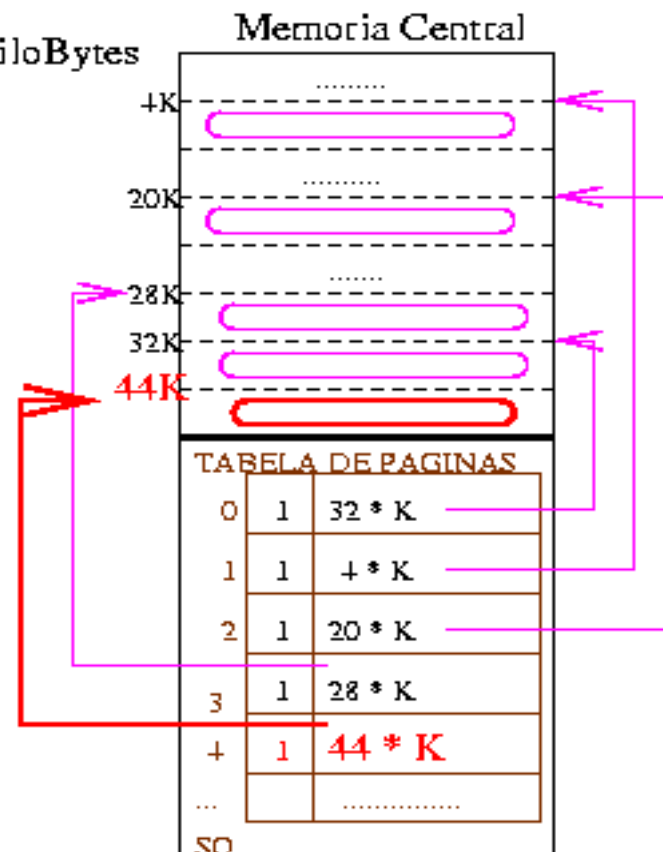
(p = 0, d= 1)

(p = 1, d=2)

(p = 3, d=1)

(p, = 4, d=1)

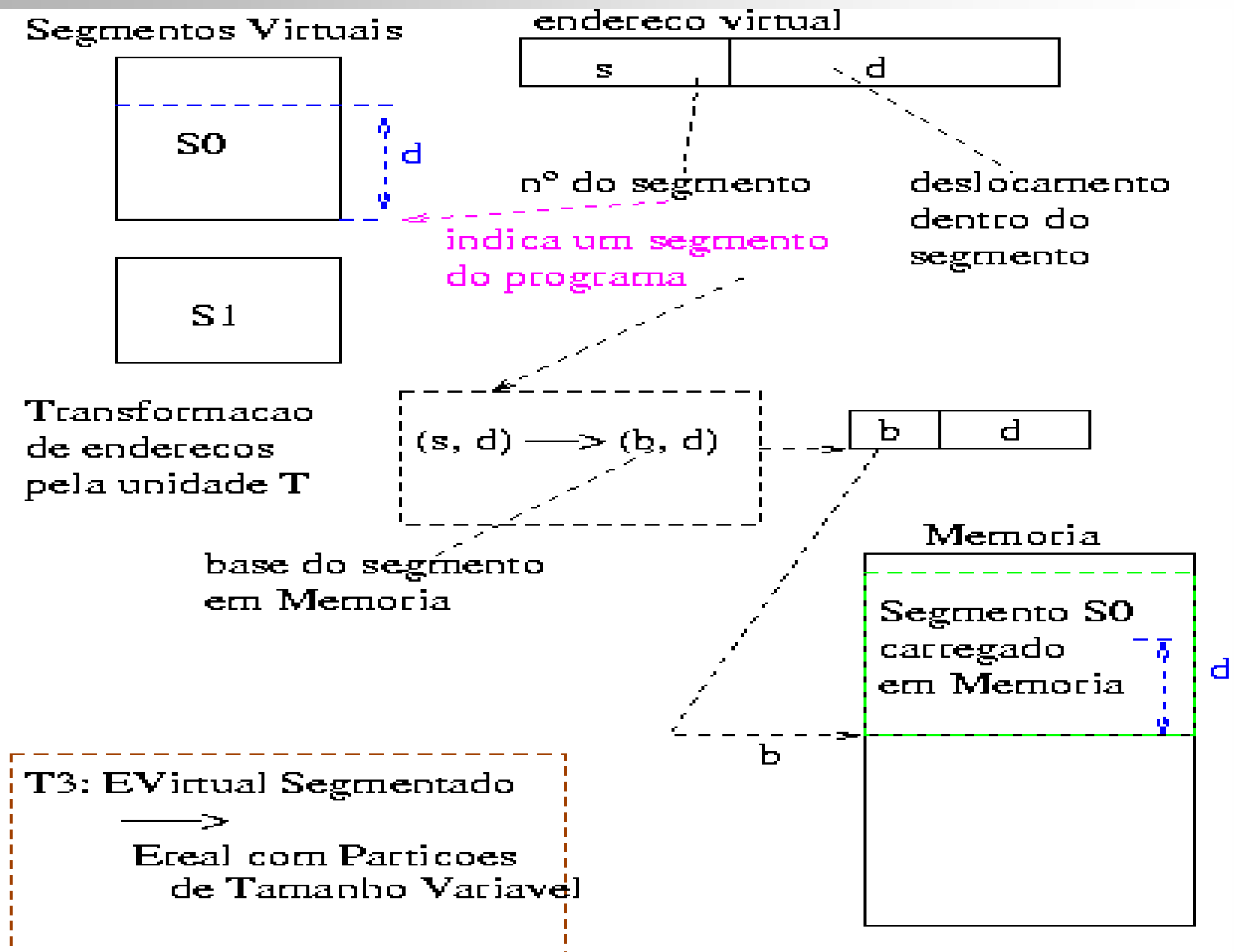
depois de repetida a instrucao
 que gerou a referencia 'a
 pagina 4

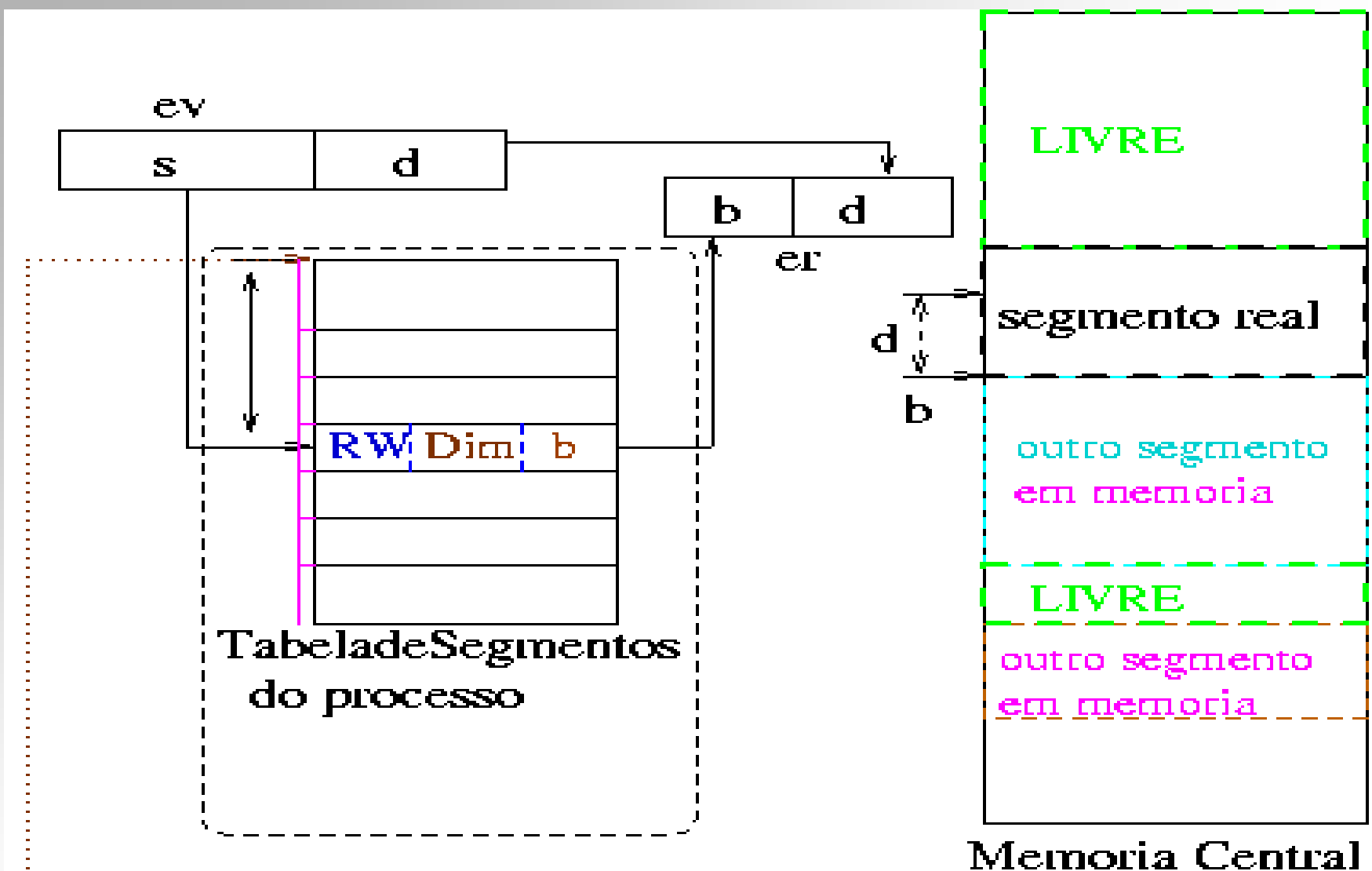


Transformação T3: EV Segmentado → ER: Partições Variáveis

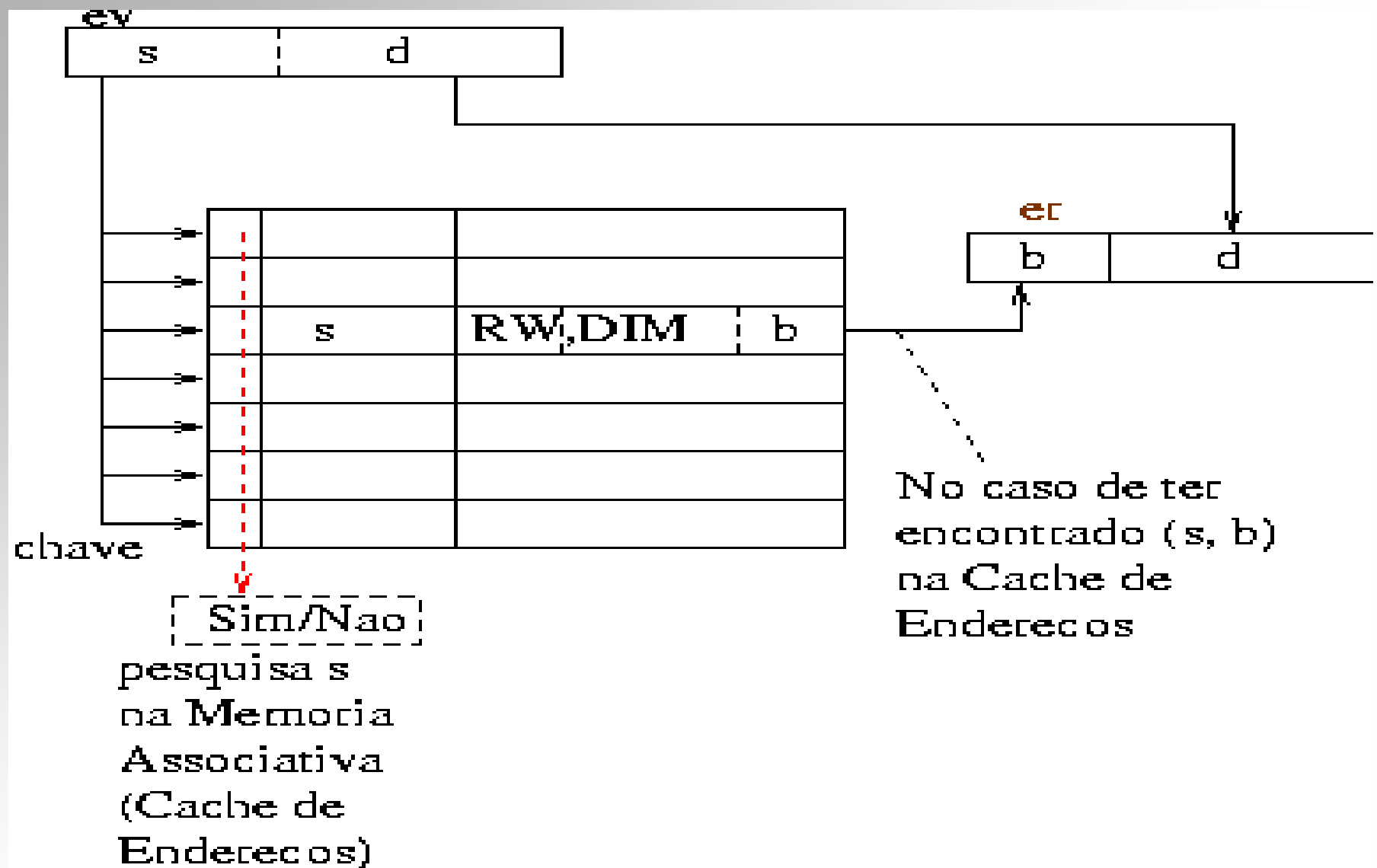
Permite:

- estruturação lógica do Espaço de Endereços Virtuais
- segmentos de dimensão variável
- possível protecção por hardware, a nível do segmento
- segmentos possivelmente partilhados
- pré-carregamento dos segmentos em memória
ou segmentação dinâmica, por pedido
- gestão de memória dificultada, devido a fragmentação externa (partições de tamanho variável)





RegBase da Tabela de Segmentos
do Processo em Memoria



Chave: nº do segmento virtual
 Conteudo: base do segmento em Memoria

Pesquisa s na Memória Associativa (TLB)

Achou s?

Sim

Nao

Acede a TS em Mem

s
presente?

Sim

Actualiza
(s,b) em TLB

Nao

GERA UMA
INTERRUPCAO
POR FALTA DE
SEGMENTO

$d \leq \text{DIM}$

Nao

GERA UMA
INTERRUPCAO
POR LIMITE DE
SEGMENTO

Sim

acesso
RW OK?

Nao

GERA UMA
INTERRUPCAO
POR PROTECCAO
DO SEGMENTO

Sim

Obtem
 $er = (b, d)$ para aceder
ao byte d do segmento s

Transformação T4: EV Segmentado → ER: Paginação

Reune as vantagens de:

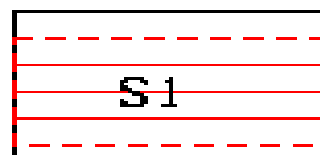
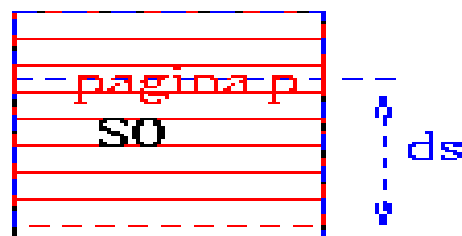
- segmentação: do ponto de vista da estrutura do Espaço de Endereços Virtuais.
- paginação: do ponto de vista da gestão de memória e do suporte de Memória Virtual.

O programa é estruturado em Segmentos.

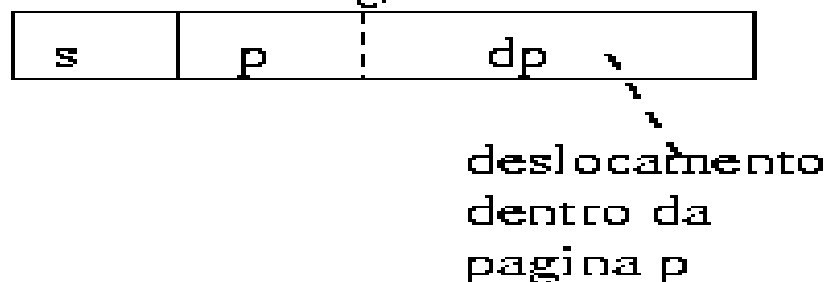
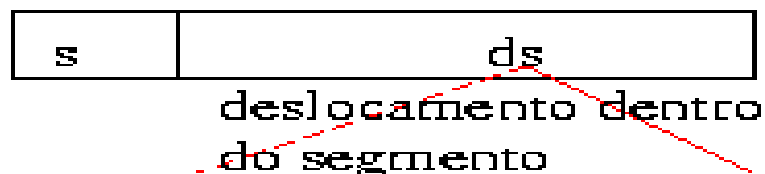
Os Segmentos são considerados subdivididos em Páginas pela unidade T:

- as páginas de cada segmento não precisam de ser carregadas em posições contíguas de memória central
- não exige que todo um segmento esteja carregado em memória central: só precisam de se manter em memória, as páginas correntemente em uso.

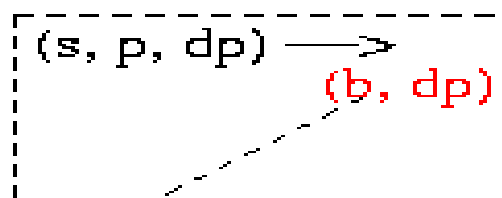
Segmentos Virtuais



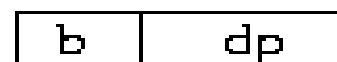
endereço virtual



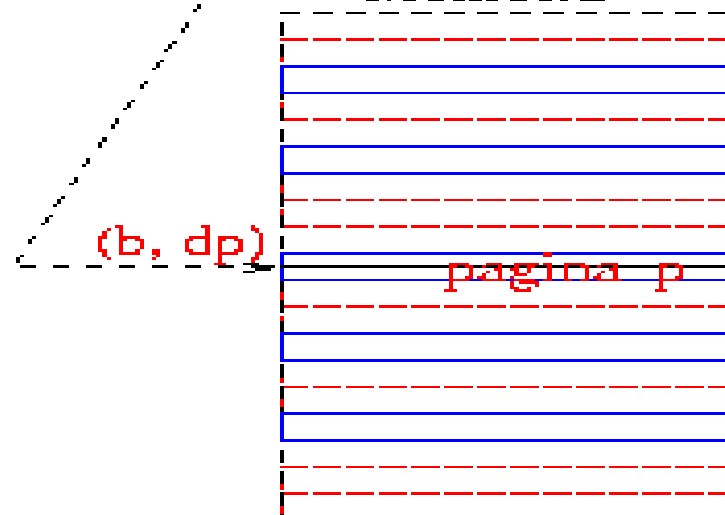
Transformação
de endereços
pela unidade T



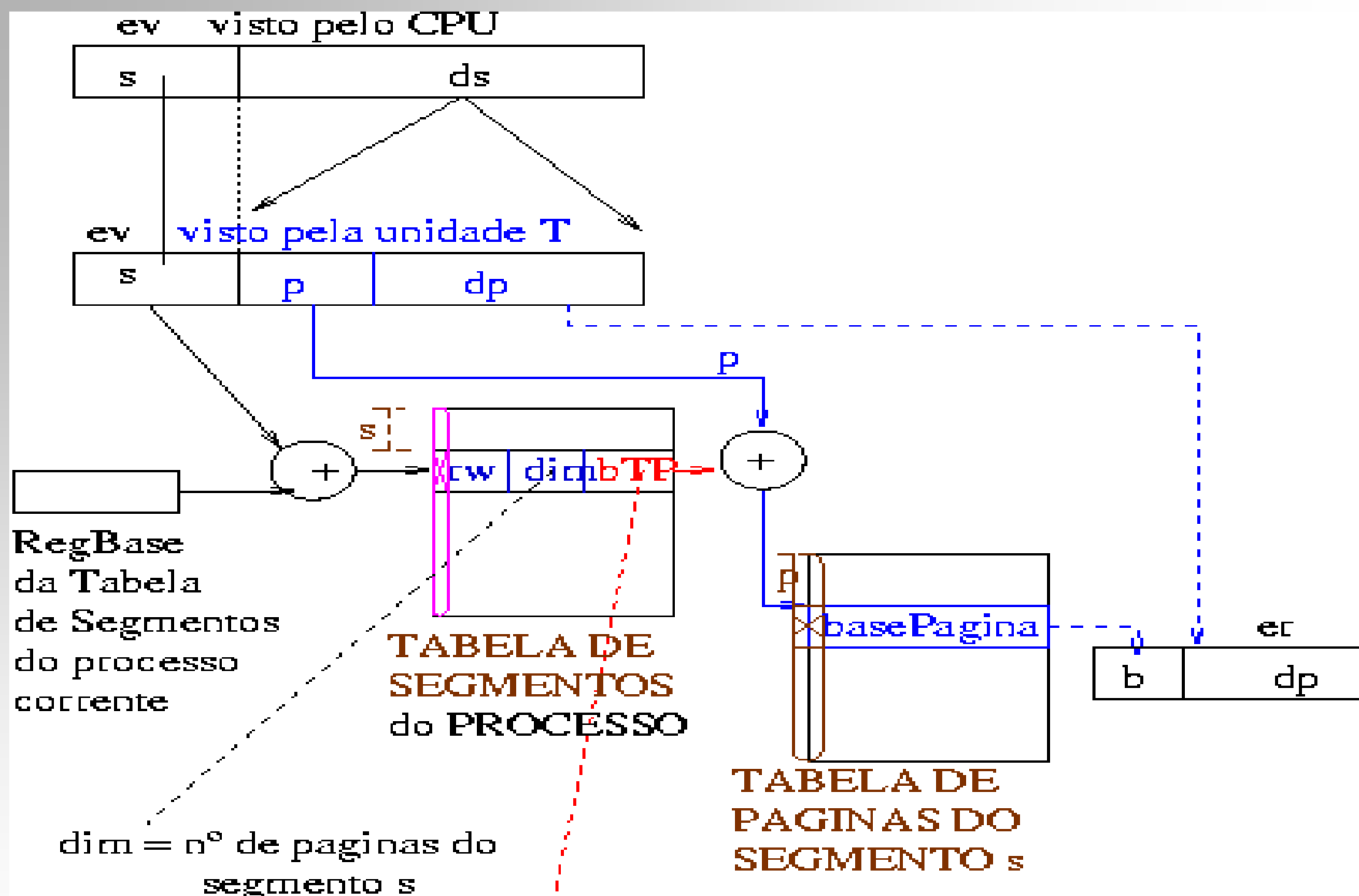
base da página p
em Memória



Memória



T4: EVirtual Segmentado
→
Ereal com Páginas



bTP = base da Tabela de Paginas do segmento **s** em Memoria

No acesso a segmento s

$(s, ds) \longrightarrow (s, p, dp)$

Accede a TS em Mem

s
presente ?

Nao

**GERA UMA
INTERRUPCAO
POR FALTA DE
SEGMENTO**

Sim

Accede a TP em Mem

p
presente ?

Sim

Obtem
(s,p,dp)

cria uma nova
Tabela de Paginas
com paginas
ausentes

Nao

**GERA UMA
INTERRUPCAO
POR FALTA DE
PAGINA**

Carrega Pagina
em Memoria
e actualiza Tabela
de Paginas