

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

5

Tecn. em Análise e Desenvolvimento de Sistemas **Sistemas Operacionais** Gerência de Memória

Prof. Ramide Dantas



Objetivos

- Entender a importância da gerência de memória
- Compreender os principais mecanismos envolvidos



Gerência de Memória

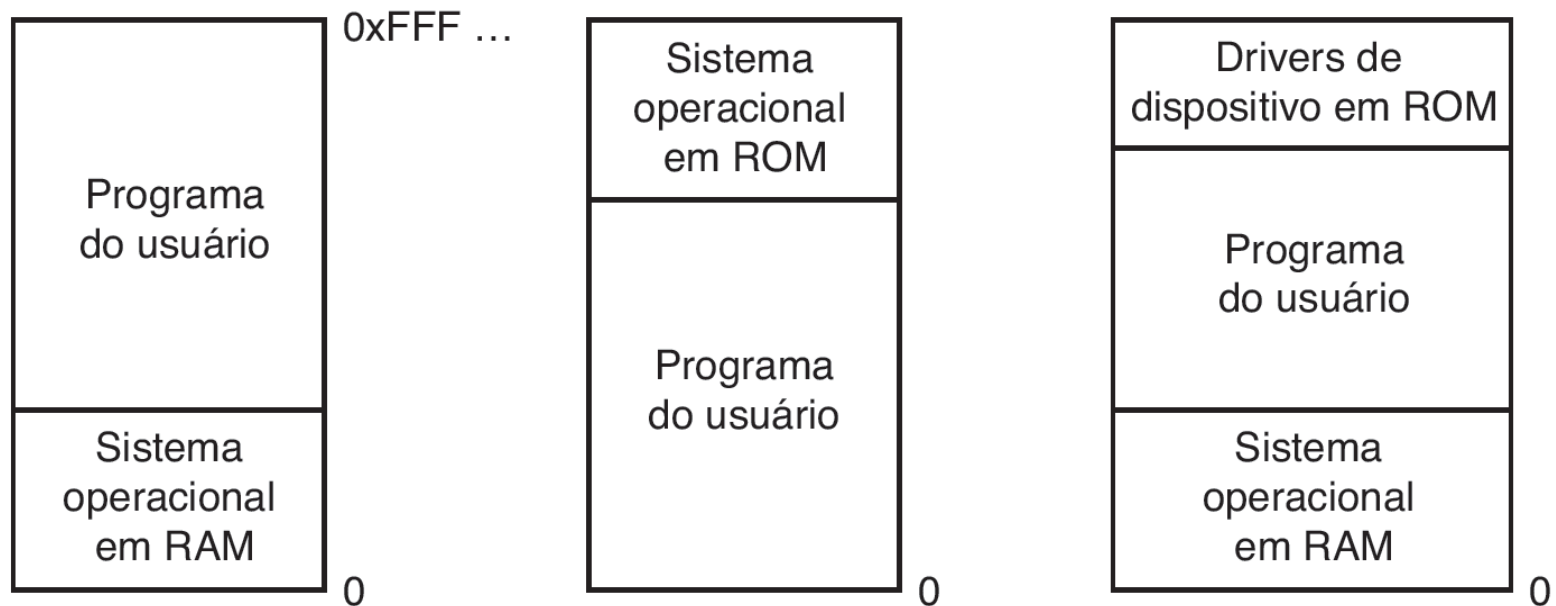
Por que gerenciar a memória?

- Recurso limitado e caro
- Essencial para o bom desempenho do sistema
- Questões de segurança e robustez do sistema



Gerência de Memória

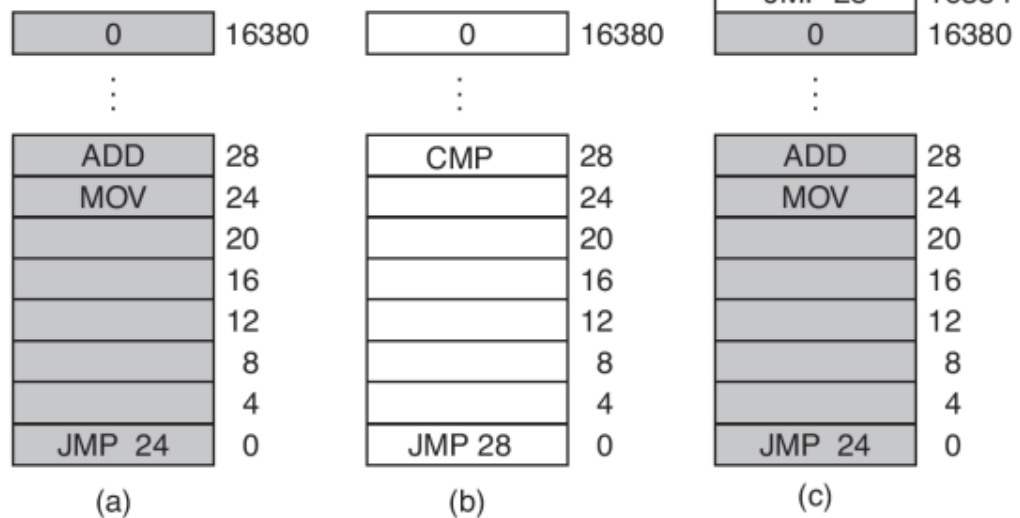
Sem abstração de memória



Gerência de Memória

Múltiplos programas sem abstração de memória

- Como ficam os endereços?
- Como proteger os programas uns dos outros?



Gerência de Memória

Relocação estática

Baseada em software (SO)

Programas são modificados em tempo de carga pelo SO

- Endereços de memória do código fonte são ajustados para a área reservada ao programa

Problemas

Não previne totalmente o acesso a endereços errados

Nem todos os acessos a memória são traduzidos pelo SO

- E.g.: ponteiros a relocação estática quebra os ponteiros deixando-os inúteis



Gerência de Memória

Relocação dinâmica

- Baseada em hardware e software

- Usa dos registradores: base e limite

- Cada acesso a memória é feito com relação à base

 - Endereço é somado à base; é verificado se o endereço resultante ultrapassa o limite

- Permite a carga dos programas em posições arbitrárias da memória e a proteção entre programas

Problema

- Custo computacional de somar e verificar a cada acesso
- a soma é uma especie de verificação sequencial, pois os valores das variáveis depende do valor das anteriores, que a torna lenta

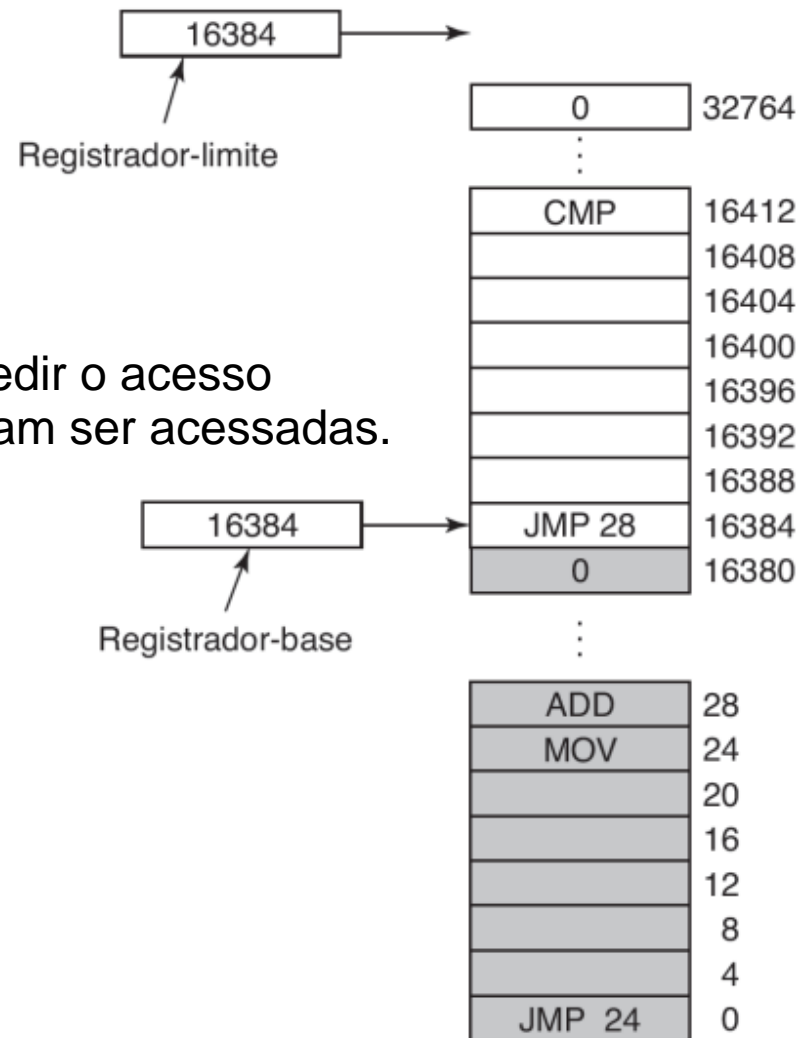


Gerência de Memória

Registradores base e limite

o SO seta a base e a cpu usa

o registrador de limite serve para impedir o acesso a áreas de memórias que não deveriam ser acessadas.



Gerência de Memória

Como carregar ter mais programas rodando do que a memória suporta?

Troca ou Permuta (swap)

Processos são colocados no disco (HD ou pendrive) **memória auxiliar** temporariamente

Processos são movidos por inteiro sempre

Problemas

Desempenho: carga/escrita de grandes processos

Fragmentação da memória

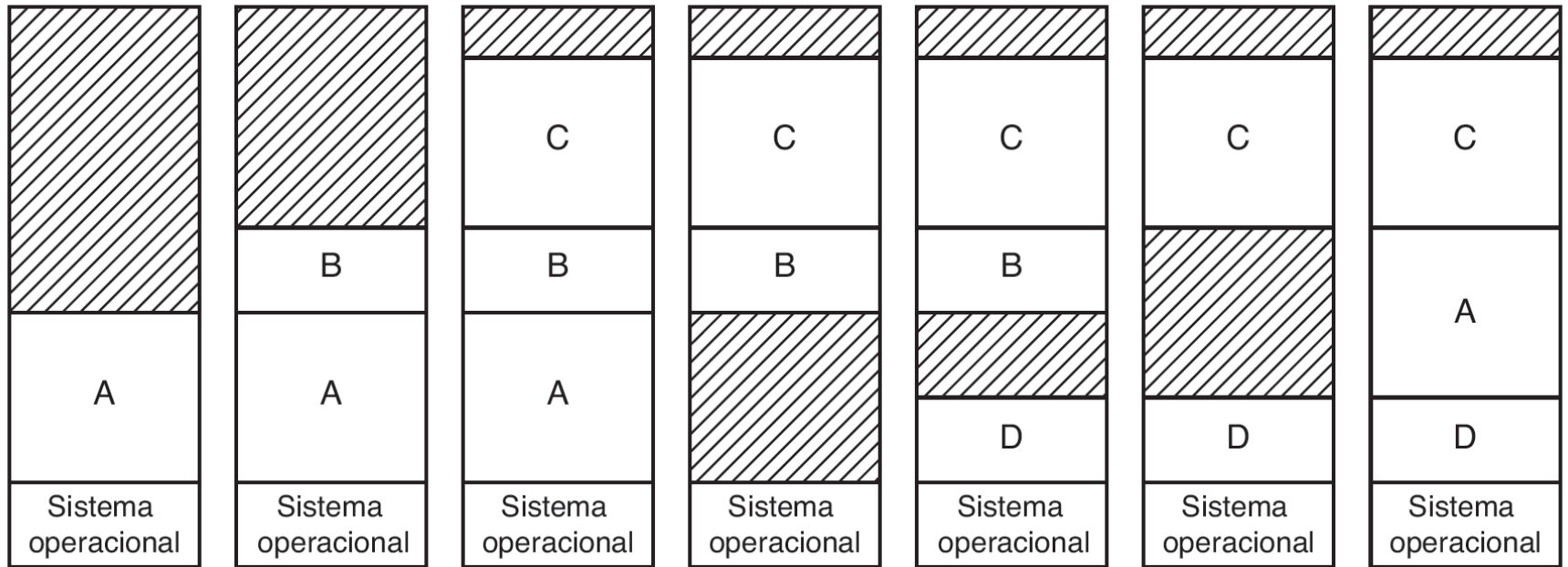


Gerência de Memória

Troca/Swap

o programa pode ir mudando a posição de memória ao decorrer de sua vida, troca-se programas da memória ram p/HD liberando memória para o outro programa rodar

Tempo →



a fragmentação aki causa problema de desperdicio pois as mprtes do programa estão muito espalhadas sem preencher as memórias em sua totalidade

Gerência de Memória

Processos podem solicitar mais memória após a carga

SO deve antecipar isso, deixando espaço livre junto a área do processo

Como os processos devem gerenciar a memória que é alocada para eles?

Três áreas: código (texto), pilha (stack) e dados (heap)

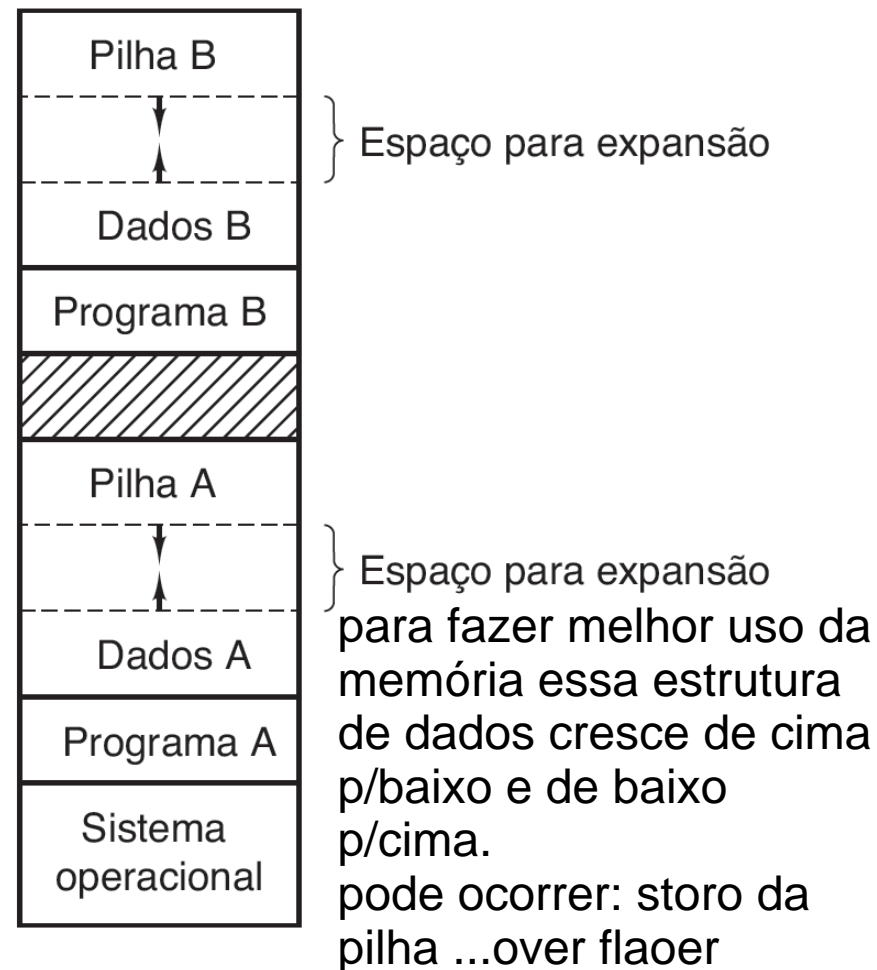
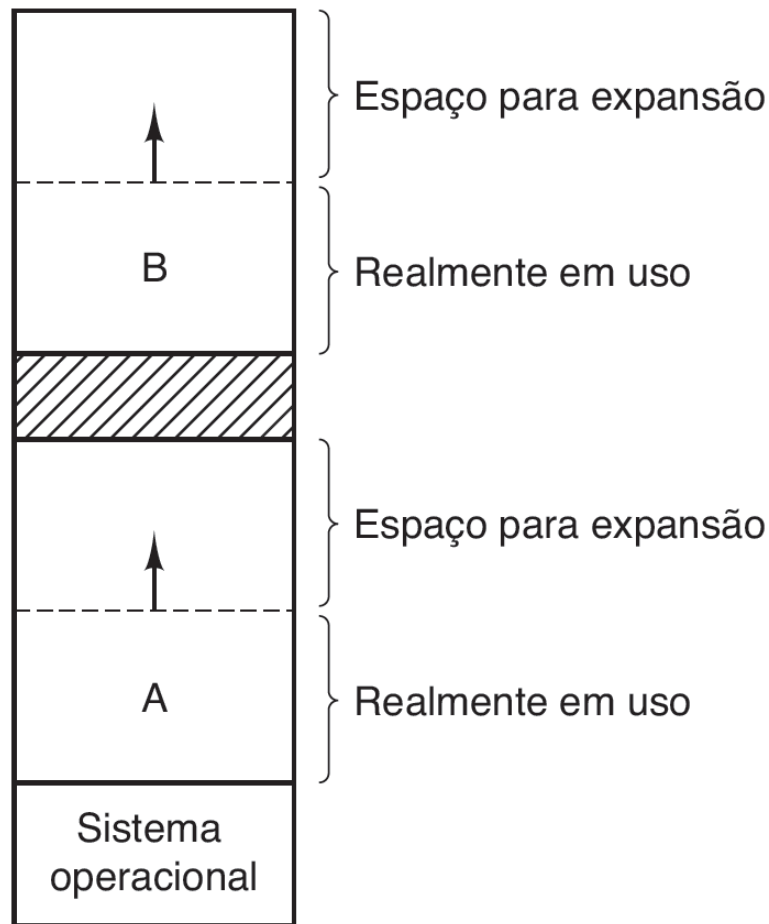
Código tem tamanho fixo; pilha e heap tem tamanho variável

Pilha e heap estão em ponto opostos da memória do processo, crescendo uma em direção a outra

dados são mais difíceis de lidar



Gerência de Memória



Gerência de Memória

Gerenciamento da memória livre

Mapa de bits:

- Um bit é para cada região de memória (0=livre, 1=usada)

- Permite descobrir facilmente se uma região está livre

- Dificulta a busca por blocos livres de tamanho arbitrário

Lista encadeada de blocos livres:

- Regiões livres contíguas são encadeados numa lista

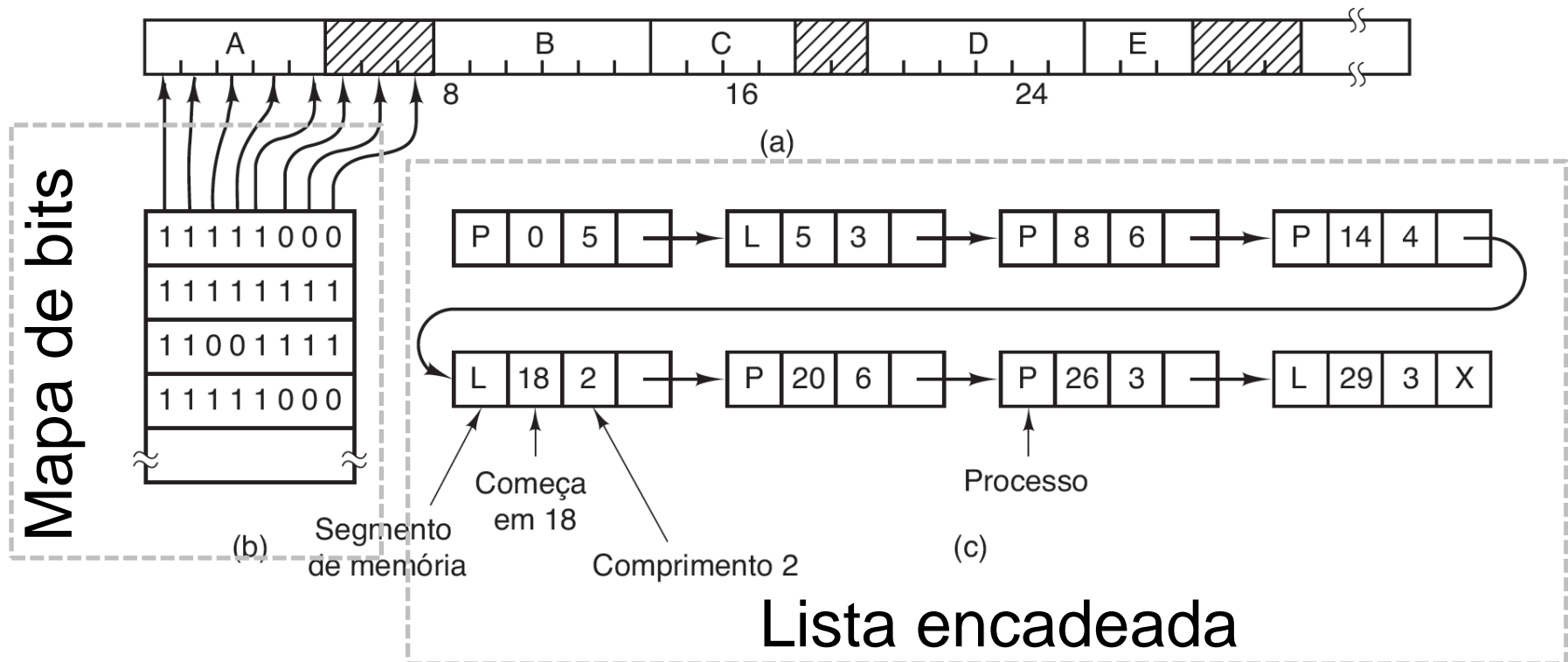
- Melhor para a alocação de blocos



Gerência de Memória

Gerenciamento de memória livre

Memória



Gerência de Memória

Gerenciamento de memória com listas encadeadas

consolidação de memória - relocação das memórias para deixar memórias livres
proximas diminuindo a fragmentação

Antes de X terminar



torna-se



torna-se

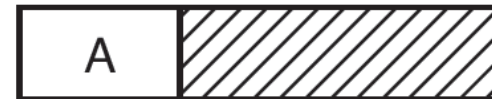


torna-se



torna-se

Após X terminar



Gerência de Memória

Dado a requisição por um bloco de K Mbytes, que espaço alocar?

Algoritmos de alocação de memória (alguns):

- First fit: 1º bloco livre na lista $\geq K$ pega o 1º que couber
- Next fit: próximo livre que caiba K
pega de onde parou da última vez melhor que o primeiro pois não precisa ver novamente os blocos já vistos
- Best fit: bloco com tamanho mais próximo $\geq K$
- Worst fit: maior bloco livre
- ...



Gerência de Memória

Memória virtual

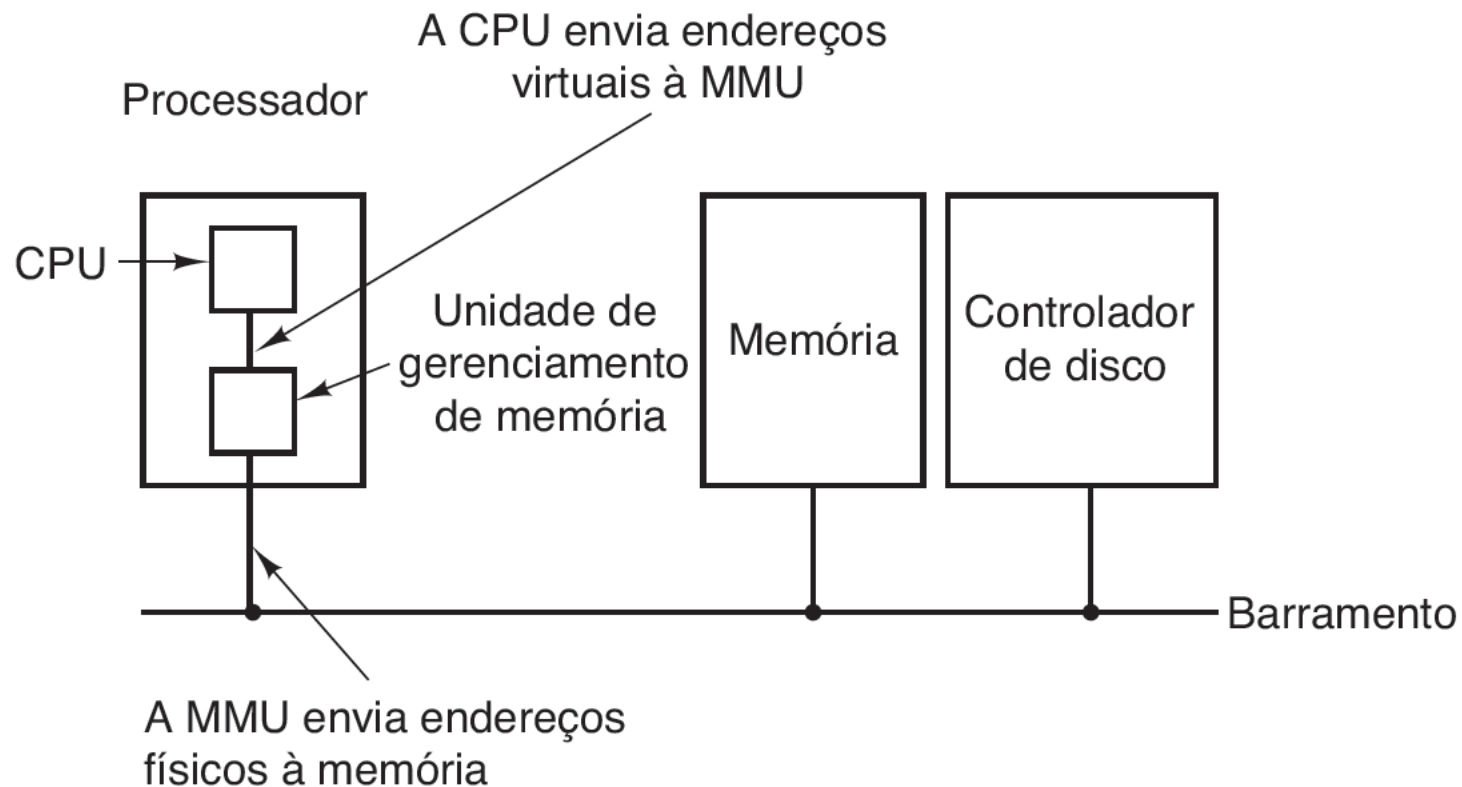
Os processos executam sobre uma abstração da memória real:

- Memória começa do 0 e tem tamanho máximo definido pelo endereçamento
 - E.g., máquinas de 32bits → 4GB
- Memória pertence totalmente ao processo
 - Não há outros processos competindo por essa memória



Gerência de Memória

Memória virtual via Paginação



Gerência de Memória

é como se o software fosse enganado ele acha que tudo está na memória, mas parte está na memória e parte está no disco.

Memória virtual via Paginação

Hardware e SO cooperam para proporcionar a ilusão de “memória infinita” aos processos

- MMU (Hardware): realiza o mapeamento entre páginas virtuais em páginas físicas

Tradução é realizada a cada acesso à memória

- SO (Software): configura o mapeamento e resolve eventuais “problemas”

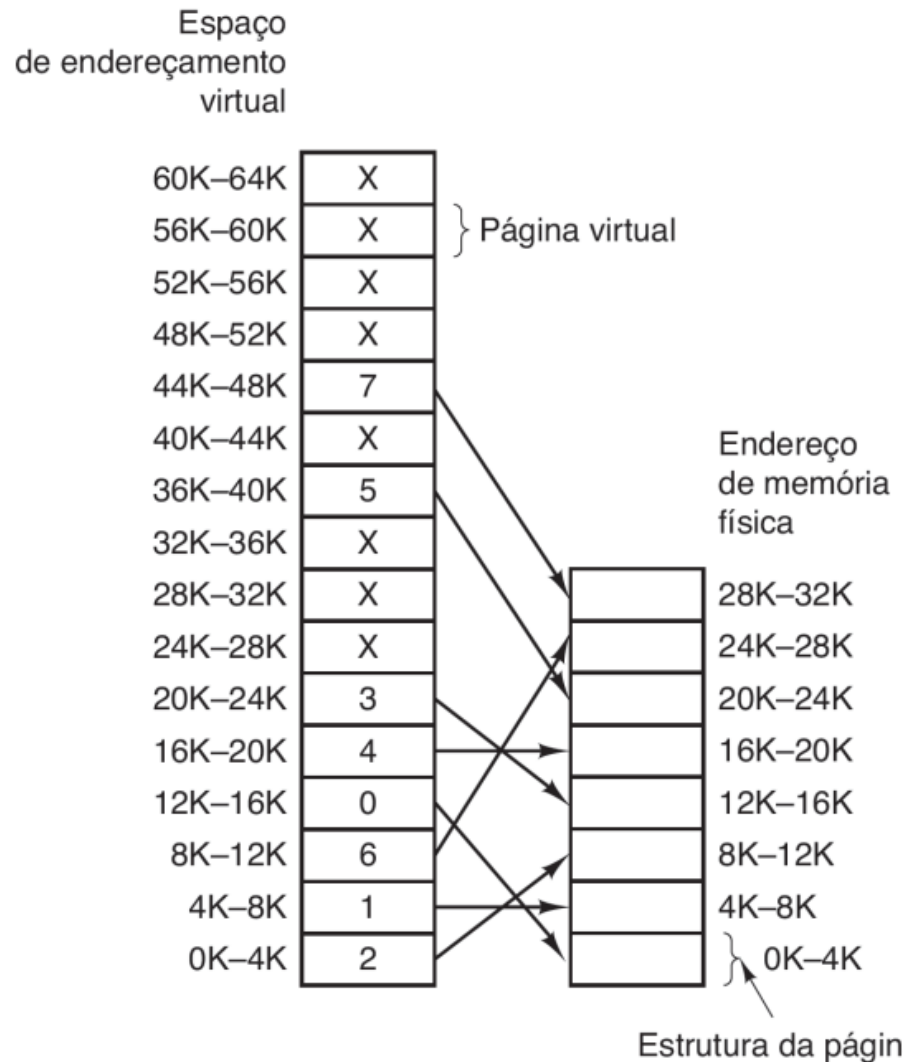
Manutenção do mapeamento é feita sempre que for preciso acomodar mais processos, por exemplo.



Gerência de Memória

Exemplo de mapeamento de páginas virtuais em físicas (quadros ou molduras)

Veja que a memória real é muito menor (menos páginas)

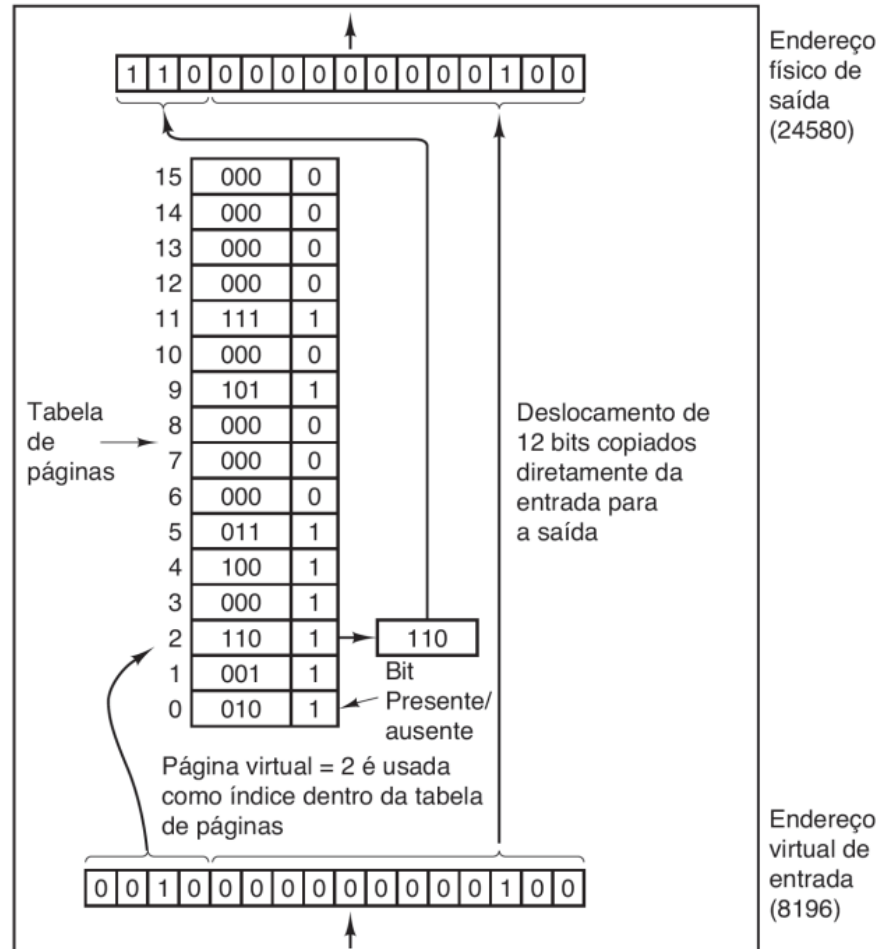


Gerência de Memória

Operação interna da MMU com 16 páginas de 4KB

memória virtual é sempre maior que a real, permitindo separar um bit na virtual para informar se a pág virtual está alocada na memória ou não

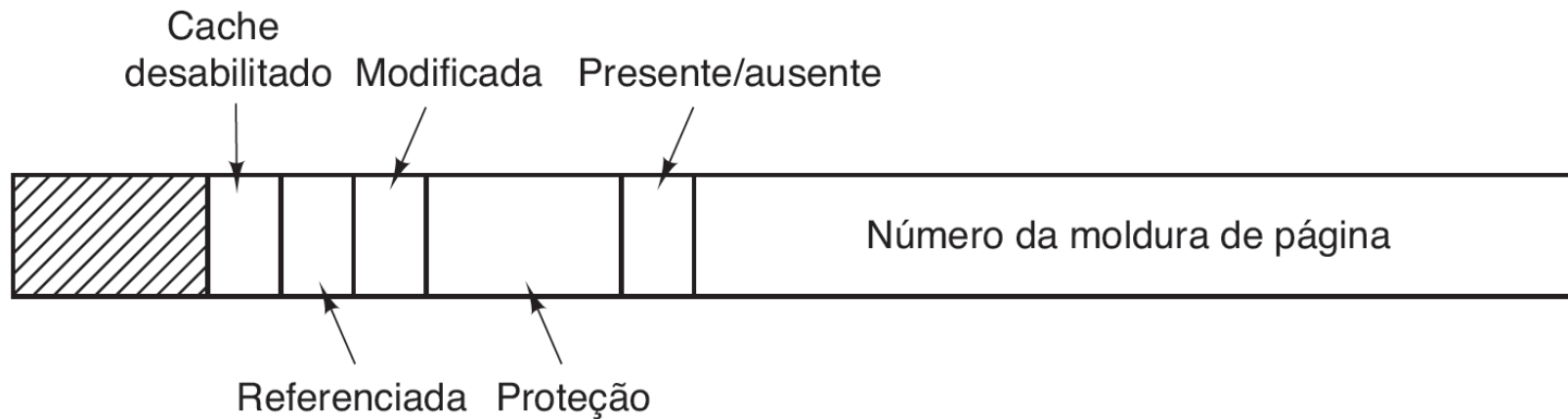
Saída: endereço físico



Entrada: endereço virtual

Gerência de Memória

Estrutura de uma entrada de uma tabela de páginas



Gerência de Memória

Acelerando a paginação

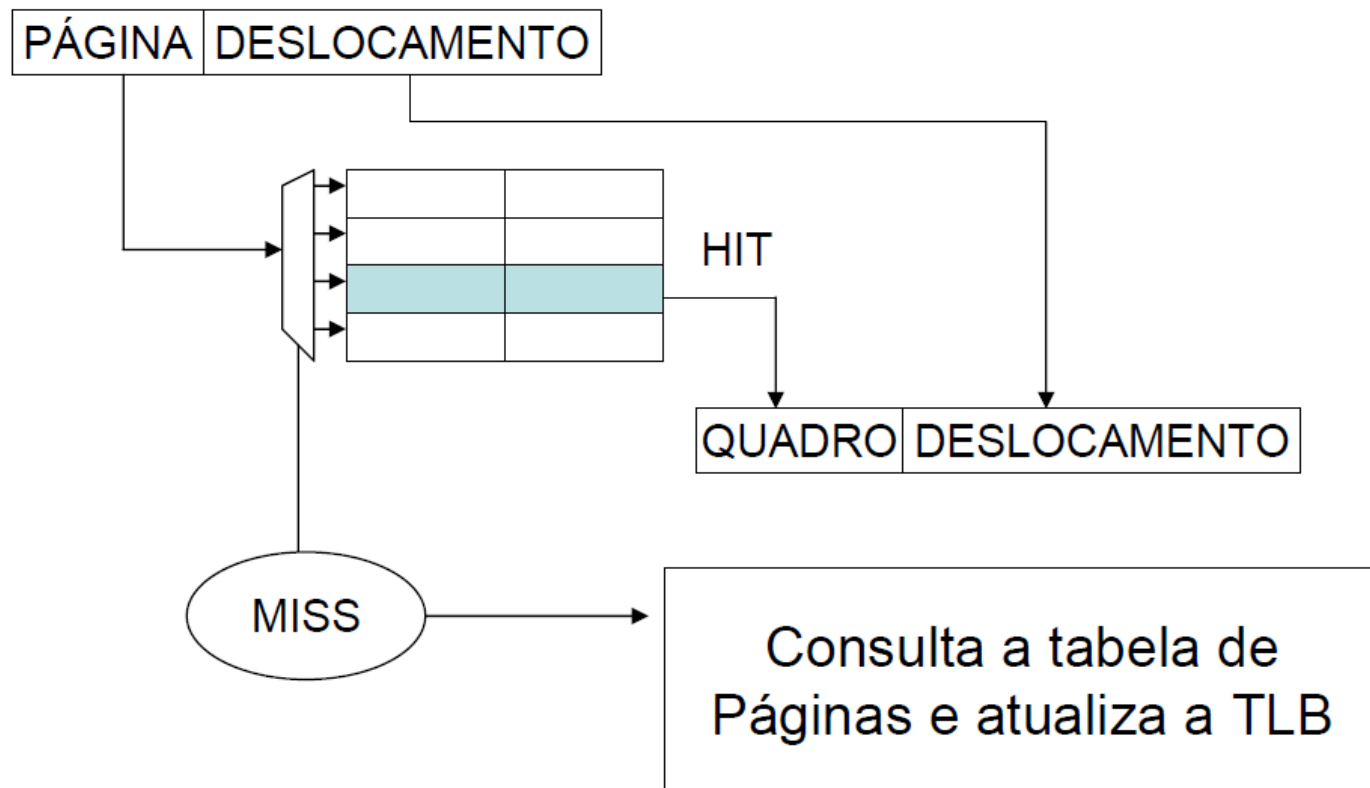
Problemas na implementação da paginação:

- O mapeamento do endereço virtual para o endereço físico deve ser rápido.
- Se o espaço virtual for grande, a tabela de páginas será grande.



Gerência de Memória

TLB – Translation Lookaside Buffer (Buffers para tradução de endereços)



Gerência de Memória

Exemplo de TLB

Válida	Página virtual	Modificada	Proteção	Moldura da página
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



Gerência de Memória

Tabelas de páginas multinível

Usa dois ou mais níveis de indireção para encontrar a página física

Tabelas de níveis intermediários não precisam existir em memória

Útil quando o espaço de endereçamento é grande

Evita tabela de páginas grande na memória

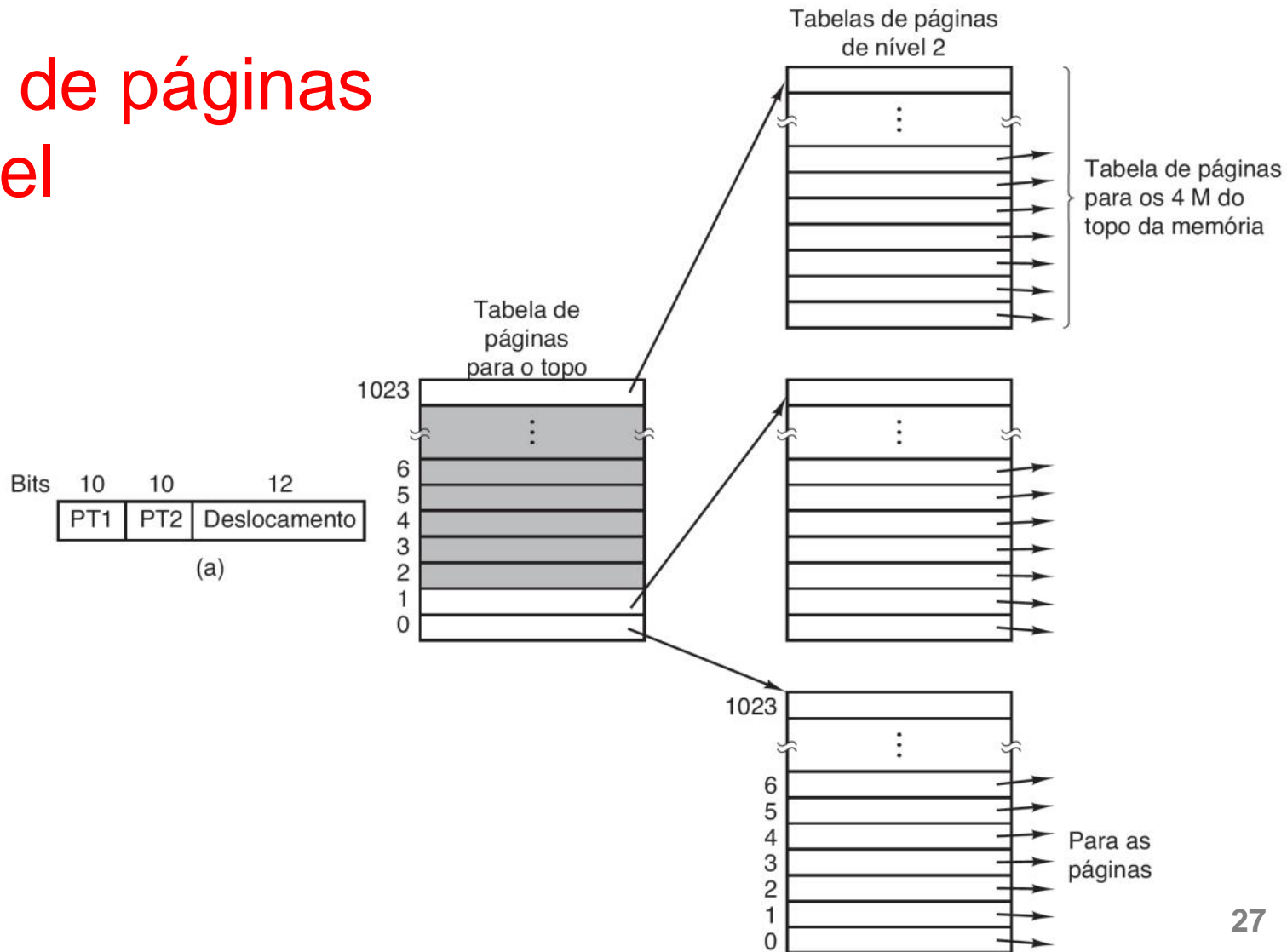
Para uma memória virtual de 1GB com páginas de 4KB seriam ~250K entradas na tabela de páginas

Com 16B por entrada, seriam 4MB só para a tabela



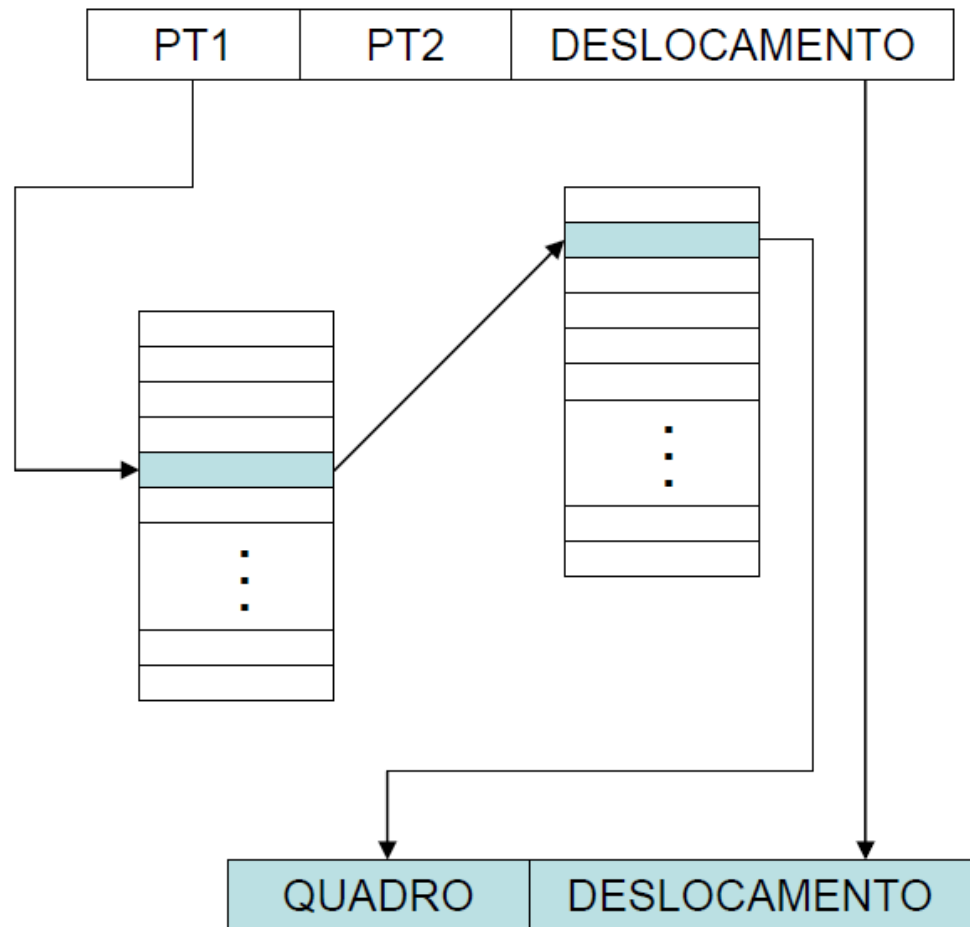
Gerência de Memória

Tabelas de páginas multinível



Gerência de Memória

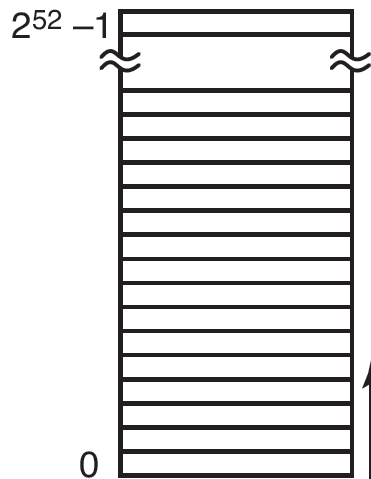
Tabelas de páginas multinível



Gerência de Memória

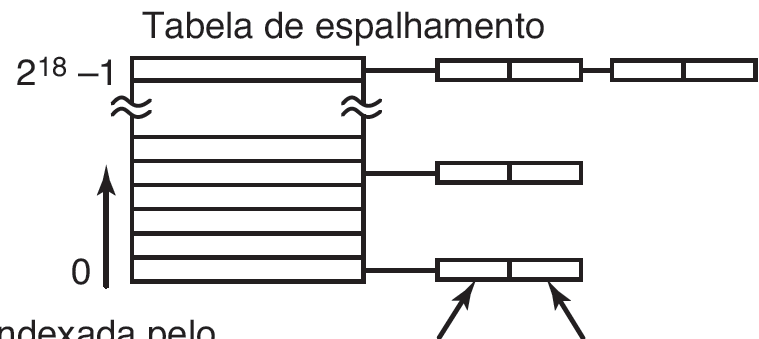
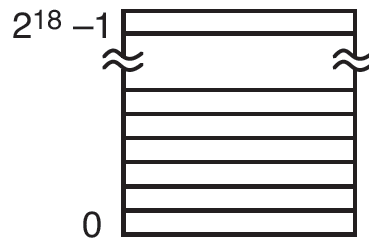
Tabelas de páginas invertidas

Tabela de páginas tradicional com uma entrada para cada uma das 2^{52} páginas



Indexada pela página virtual

Memória física de 1 GB
tem 2^{18} molduras de
página de 4 KB



Indexada pelo
espalhamento na
página virtual

Página virtual
Moldura da página

Gerência de Memória

um processo com muitos pages fault significa que o mesmo não está perdendo rendimento.

Falha (ou falta) de página (page fault)

Ocorre quando a página física não se encontra carregada em memória

O SO deve buscar a página da memória auxiliar (disco), carregar em memória física e “reiniciar” a instrução que gerou a falha

Normalmente uma página que estava carregada terá de ser movida para o disco para liberar memória (substituição de página) **problema é escolher que programa de ser movido.**

O SO deve escolher a melhor página para substituir

Existem vários algoritmos para essa escolha



Gerência de Memória

Algoritmo de substituição de páginas

- Algoritmo ótimo de substituição de página.
- Algoritmo de substituição de página não usado recentemente.
- Algoritmo de substituição de página primeiro a entrar, primeiro a sair.
- *Algoritmo de substituição de página segunda chance.*
- *Algoritmo de substituição de página de relógio.*
- *Algoritmo de substituição de página usado menos recentemente.*
- Algoritmo de substituição de página de conjunto de trabalho.
- *Algoritmo de substituição de página WSClock.*



Gerência de Memória

- Algoritmo ótimo de substituição de página
Escolhe página que vai demorar mais (em número de instruções) para ser acessada
Inviável na prática
- Algoritmo de substituição de página não usada recentemente (NRU)
Escolher página baseado nos bits R e M da tabela de página
Dá preferência a páginas não referenciadas e não modificadas
Bit R é zerado periodicamente pelo SO
a modificada é mais importante que a lida



Gerência de Memória

- Algoritmo de substituição de página primeira a entrar, primeira a sair (FIFO)

Escolhe as páginas apenas pela “idade” delas

Ruim pois não considera utilização

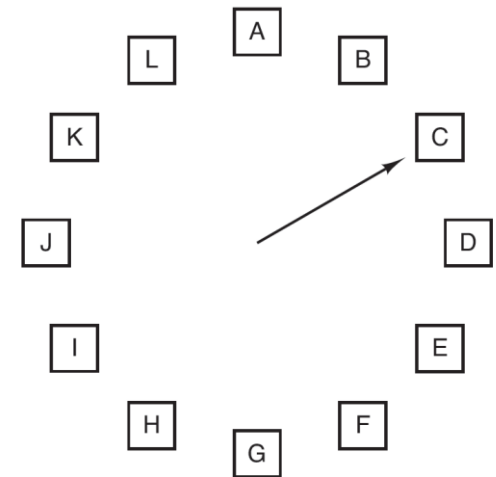
- Algoritmo segunda chance

Combina FIFO com NRU

- Algoritmo de substituição de página de relógio

Segunda chance implementado usando

uma lista circular **se tiver sido referenciado será substituído**



Gerência de Memória

- Algoritmo de substituição de página usada menos recentemente (LRU) *list riste and use - substitui a pag que está parada a mais tempo*
Boa aproximação mas implementação pode ser complexa
Manter a lista de página ordenada por acesso é custoso computacionalmente
Outra solução: cada página mantém um “hora de acesso” (contador da CPU)
Nem todo hardware implementa essas soluções, então aproximações em software são empregadas



Gerência de Memória

uma melhoria do NFU o envelhecimen uma forma de envelhecimento é dá um shift a direitoto, isso faz com que as pag q a muito não são usada envelhecam e sejam fechadas

- Algoritmo de substituição de página usada menos frequentemente (NFU) no

É mantido um contador de uso das páginas

Páginas pouco usadas não são substituídas

Problema: páginas muito usadas no passado não são desalocadas

Solução: Envelhecimento

Contador é decrementado com o passar do tempo
contador alto pois foi muito usada



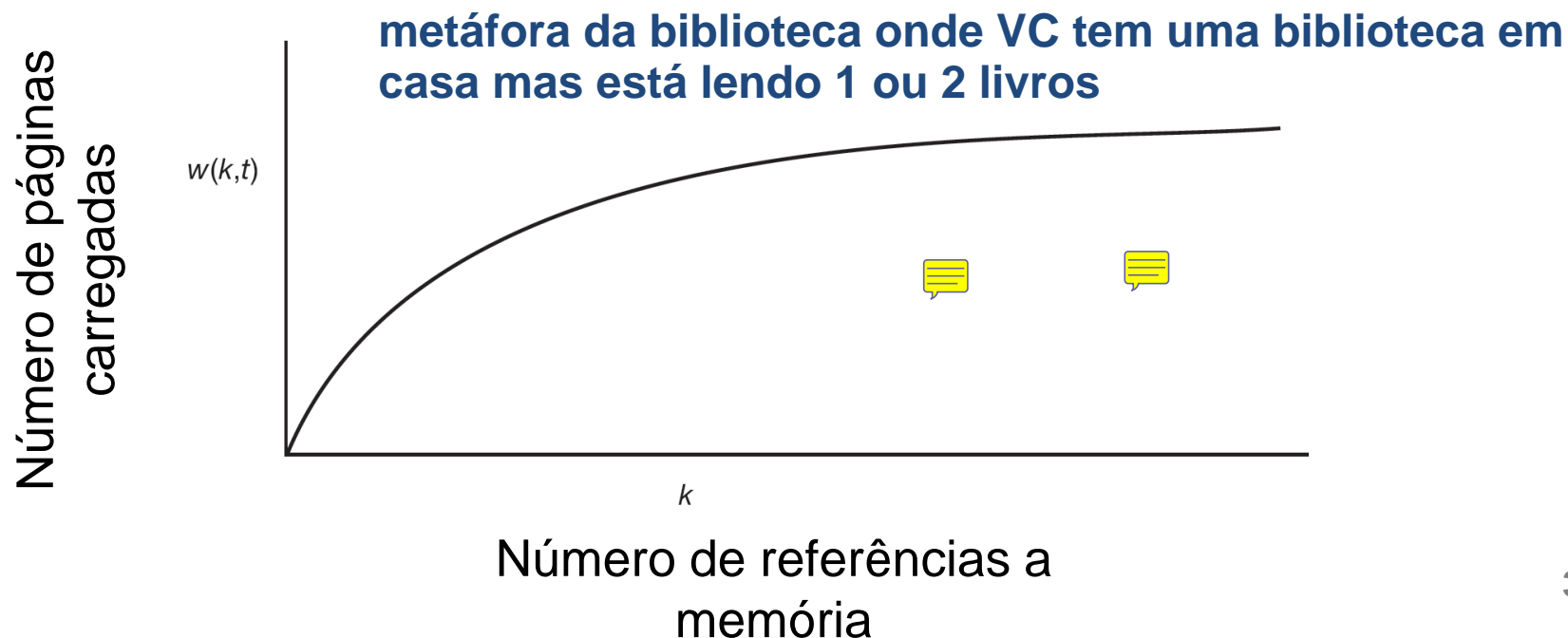
Gerência de Memória

obs a memória virtual é usada o tempo todo, mas a memória de página

- Algoritmo de substituição de página de conjunto de trabalho (WS) **80% do tempo esta rodando 20% da memória, se soubermos q pedaço é esse que está rodando, deixa-se apenas ele**

Mantém na memória o conjunto de páginas de trabalho de um processo

Usa informação histórica do processo para determinar o WS



Gerência de Memória

questão de prova: memorizar as siglas:

Resumo dos algoritmos de substituição de página

Algoritmo	Comentário
Ótimo	Não implementável, mas útil como um padrão de desempenho
NRU (não usada recentemente)	Aproximação muito rudimentar do LRU
FIFO (primeiro a entrar, primeiro a sair)	Pode descartar páginas importantes
Segunda chance	Algoritmo FIFO bastante melhorado
Relógio	Realista
LRU (usada menos recentemente)	Excelente algoritmo, porém difícil de ser implementado de maneira exata
NFU (não frequentemente usada)	Aproximação bastante rudimentar do LRU
Envelhecimento (<i>aging</i>)	Algoritmo eficiente que aproxima bem o LRU
Conjunto de trabalho	Implementação um tanto cara
WSClock	Algoritmo bom e eficiente



Gerência de Memória

Falta de página (Page Fault)

- O hardware cria uma interrupção, salvando o contador do programa na pilha.
- Uma rotina do kernel é iniciada para salvar o conteúdo dos registradores de uso geral e outras informações voláteis.
- O sistema operacional tenta descobrir qual página virtual é necessária.
- Uma vez conhecido o endereço virtual que causou a falta da página, o sistema verifica se esse endereço é válido e se a proteção é consistente com o acesso.



Gerência de Memória

- Uma página física é selecionada para ser retirada da memória para dar espaço a página em disco.
- Se a página selecionada estiver “suja”, a página é escalonada para ser transferida para o disco e será realizado um chaveamento de contexto.
- Se estiver limpa, não é preciso fazer mais nada (i.e., a cópia em disco ainda é válida)
- Após retirar a página, o sistema operacional buscará o endereço em disco onde está a página em falta solicitada e escalonará uma operação para trazê-la.



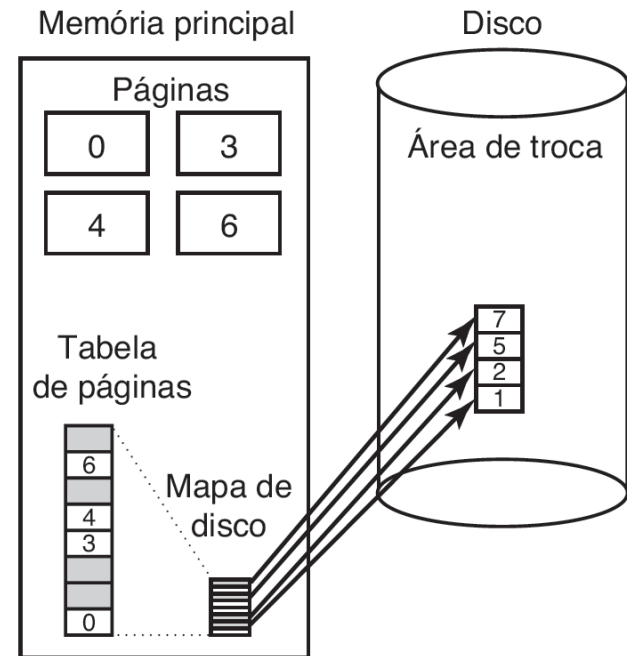
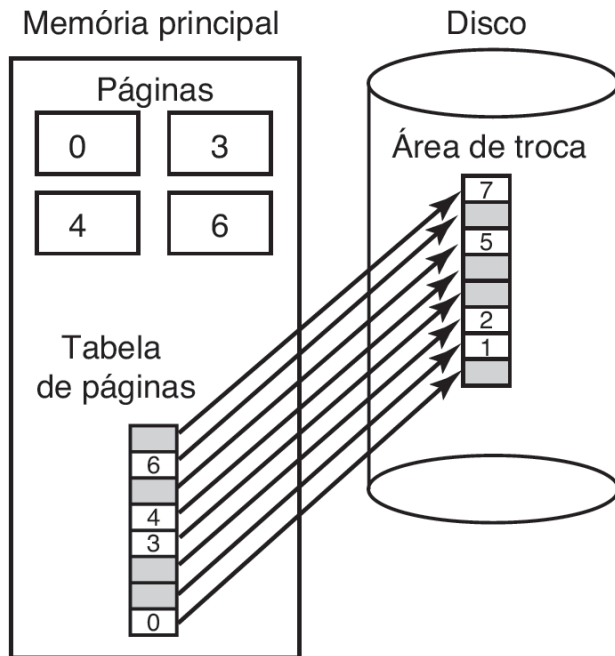
Gerência de Memória

- Quando a interrupção de disco indicar que a página chegou na memória, as tabelas de páginas serão atualizadas para refletir sua posição, e será indicado que a moldura de página está normal.
- A instrução que gerou o “page fault” é recuperada para o estado em que se encontrava quando começou
 - Contador de programa é reiniciado a fim de apontar para aquela instrução.
- Quando o processo é escalonado, o SO recarrega os registradores e outras informações de estado e retorna ao espaço de usuário para continuar a execução como se nada tivesse ocorrido.



Gerência de Memória

Memória secundária



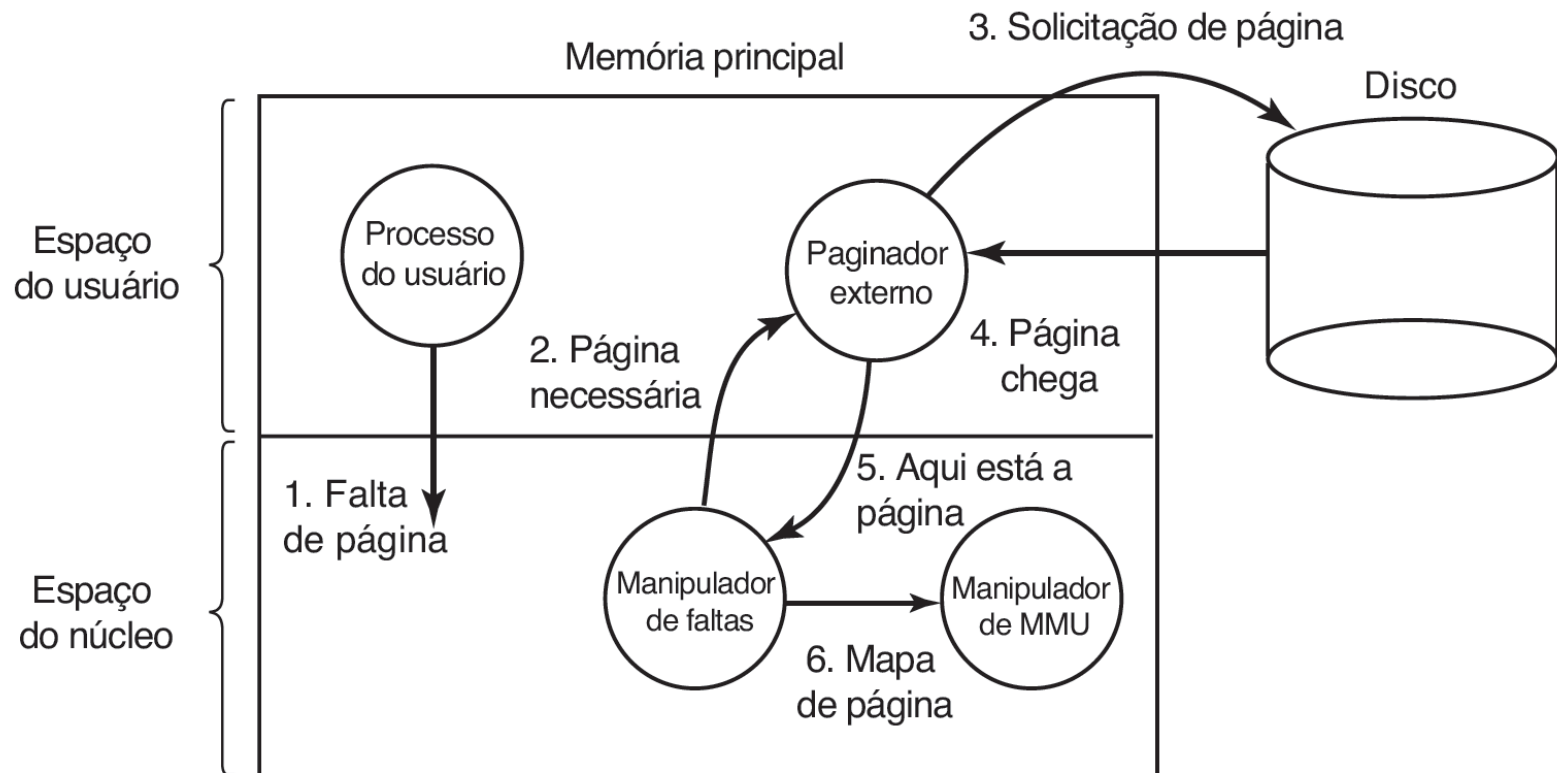
Gerência de Memória

Sistema de gerenciamento de memória é dividido em três partes:

- Um manipulador de MMU de baixo nível.
- Um manipulador de falta de página que faz parte do núcleo.
- Um paginador externo executado no espaço do usuário.



Gerência de Memória



Gerência de Memória

Espaços separados de instrução de dados

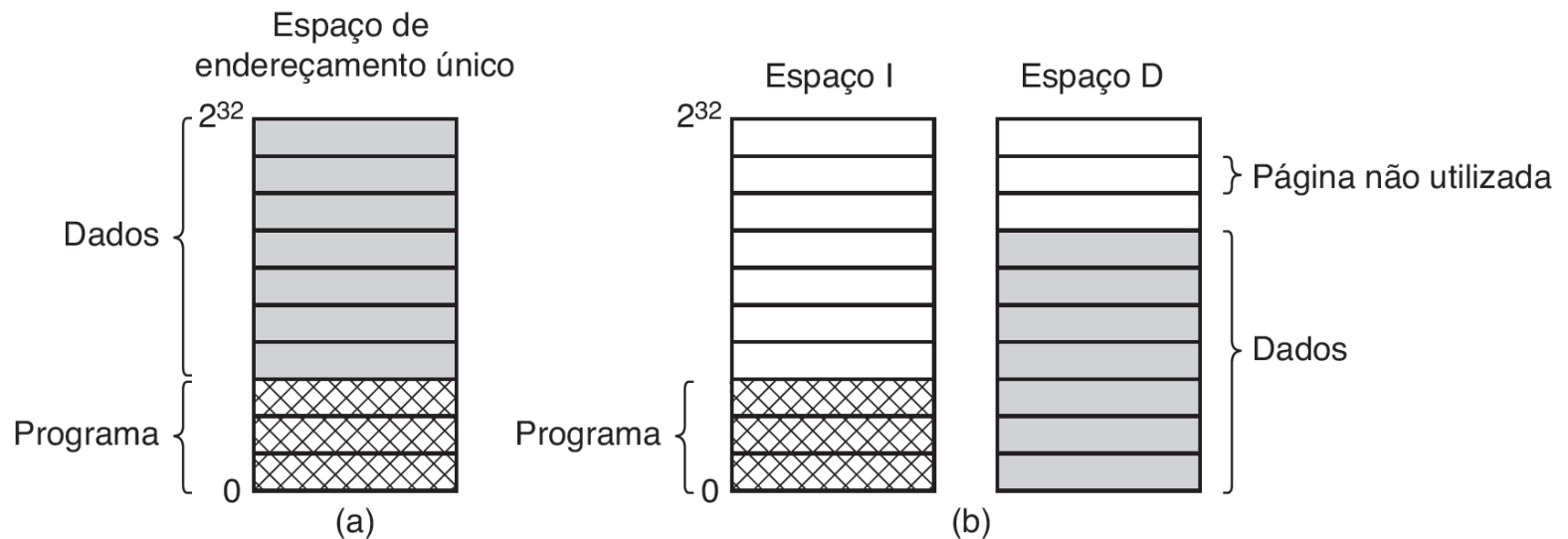


Figura 3.23 (a) Um espaço de endereçamento. (b) Espaços I e D independentes.

Gerência de Memória

Páginas compartilhadas

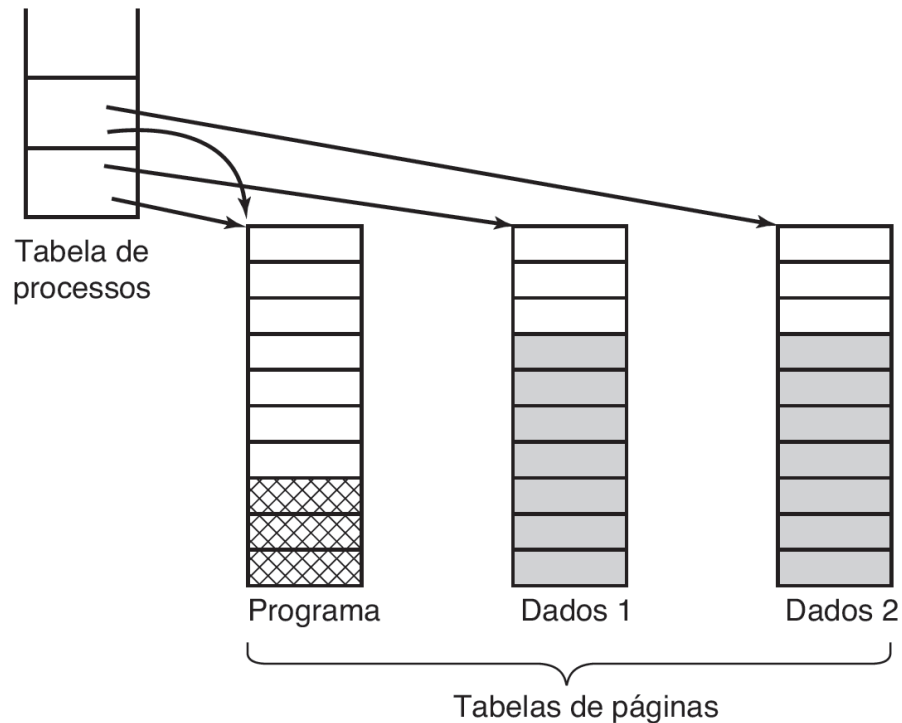
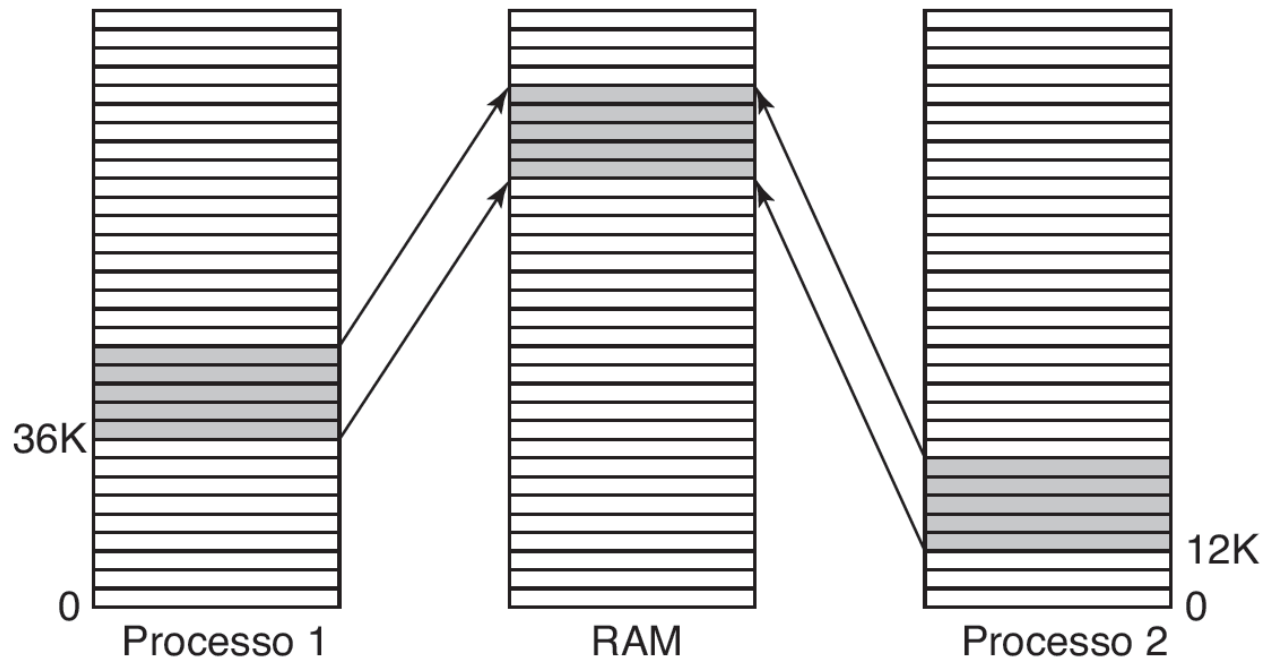


Figura 3.24 Dois processos que compartilham o mesmo programa compartilhando sua tabela de páginas.

Gerência de Memória

Bibliotecas compartilhadas



Gerência de Memória

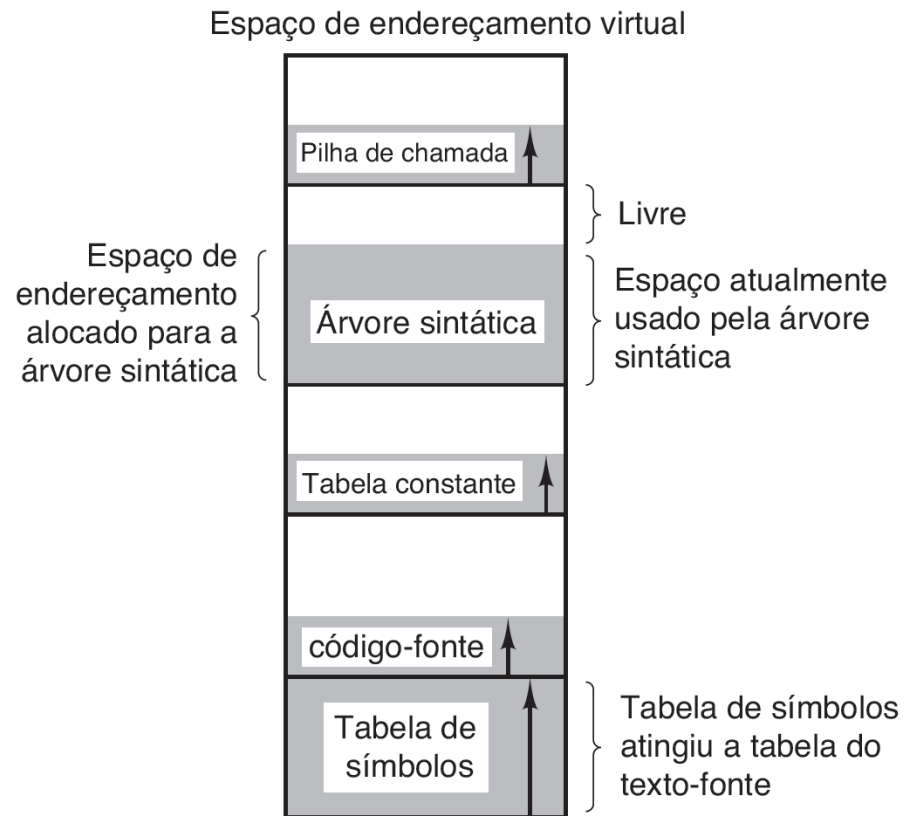
Segmentação

Um compilador tem muitas tabelas que são construídas conforme a compilação ocorre, possivelmente incluindo:

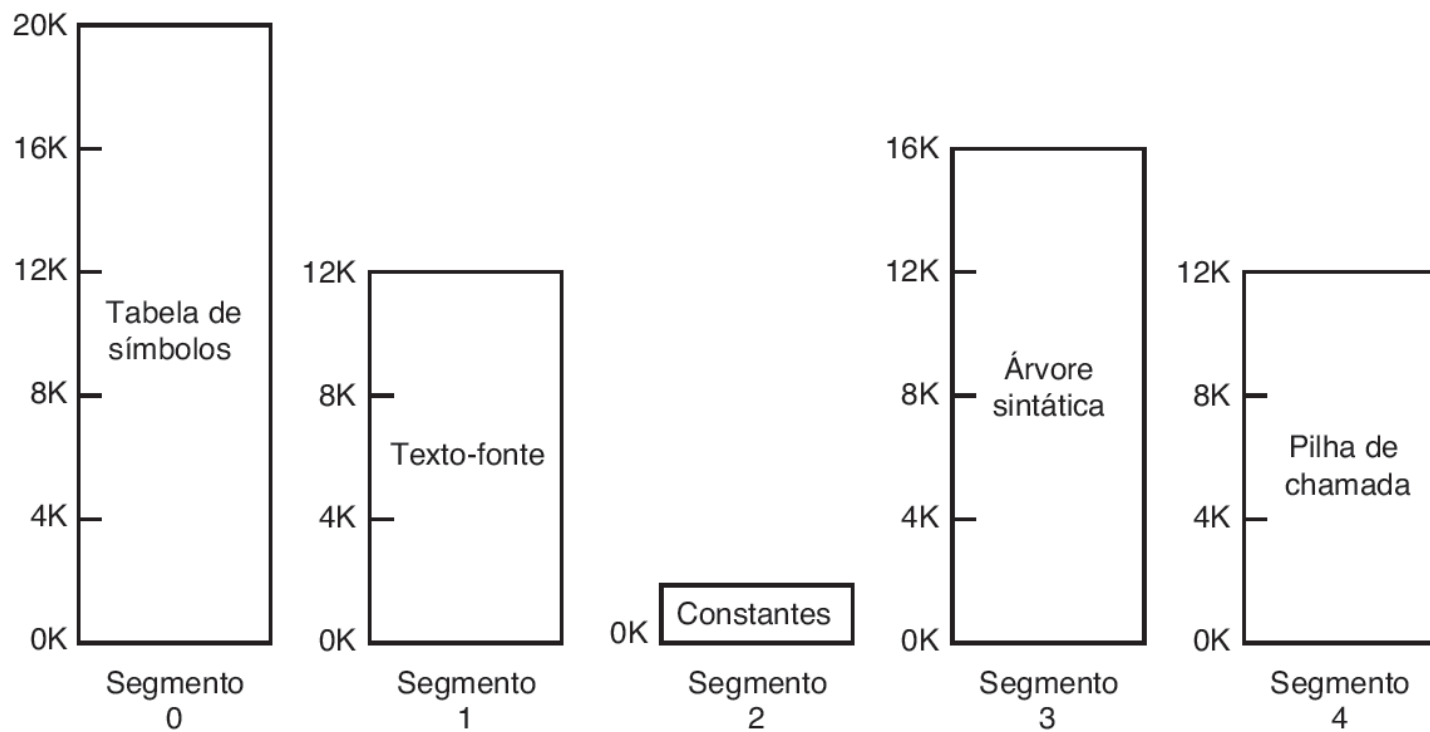
- O código-fonte sendo salvo para impressão
- A tabela de símbolos – os nomes e atributos das variáveis.
- A tabela com todas as constantes usadas, inteiras e em ponto flutuante.
- A árvore sintática, a análise sintática do programa.
- A pilha usada pelas chamadas de rotina dentro do compilador.



Gerência de Memória



Gerência de Memória



Gerência de Memória

Implementação da segmentação pura

Consideração	Paginação	Segmentação
O programador precisa saber que essa técnica está sendo usada?	Não	Sim
Há quantos espaços de endereçamento linear?	1	Muitos
O espaço de endereçamento total pode superar o tamanho da memória física?	Sim	Sim
Rotinas e dados podem ser distinguidos e protegidos separadamente?	Não	Sim
As tabelas cujo tamanho flutua podem ser facilmente acomodadas?	Não	Sim
O compartilhamento de rotinas entre os usuários é facilitado?	Não	Sim
Por que essa técnica foi inventada?	Para obter um grande espaço de endereçamento linear sem a necessidade de comprar mais memória física	Para permitir que programas e dados sejam divididos em espaços de endereçamento logicamente independentes e para auxiliar o compartilhamento e a proteção

Gerência de Memória

Pentium: segmentação com paginação

GDT: Global Descriptor Table

Segmentos do sistema

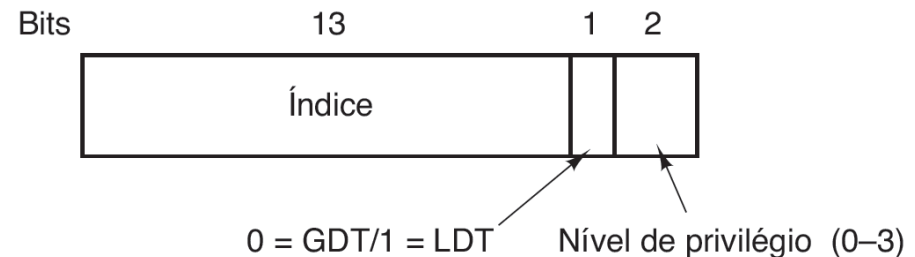
Única para todo o sistema

LDT: Local Descriptor Table

Segmentos dos processos

Uma por processo

Seletor de Segmento



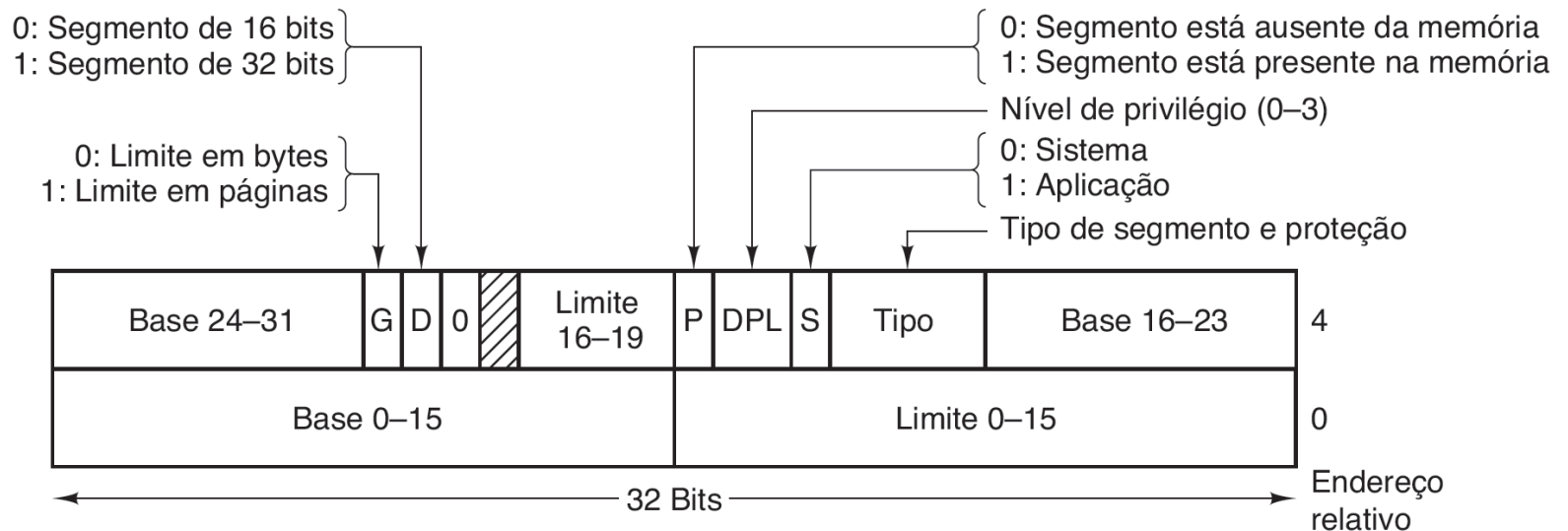
Funcionamento

Seletor de segmento é carregado pelo programa em registrador especial, usado nas referências à memória

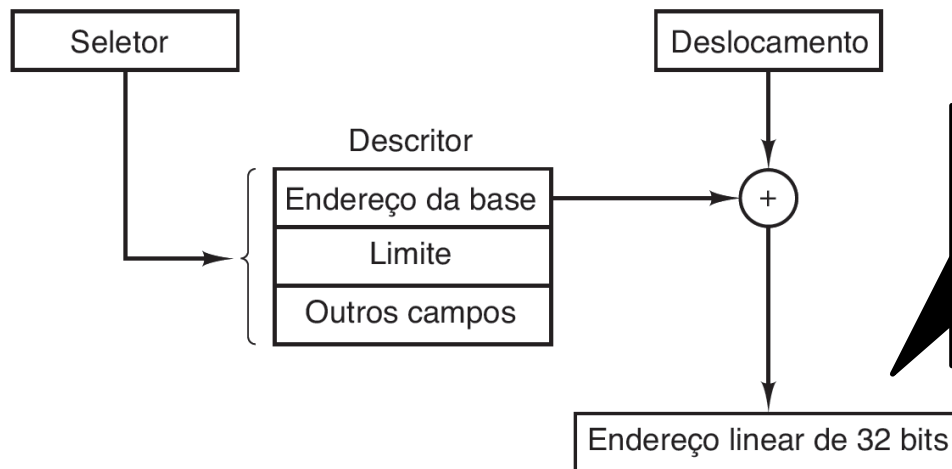
CS: Segmento de Código; DS: Segmento de Dados; existem outros

Gerência de Memória

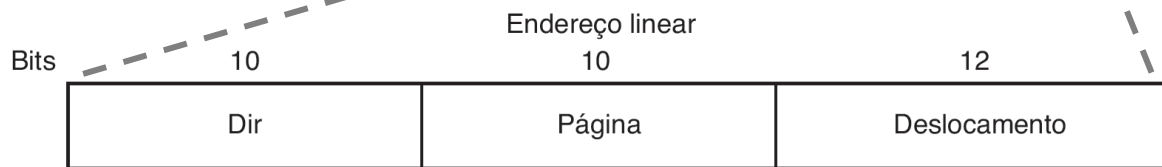
Descritor de segmento na GDT ou LDT



Resolvendo o segmento

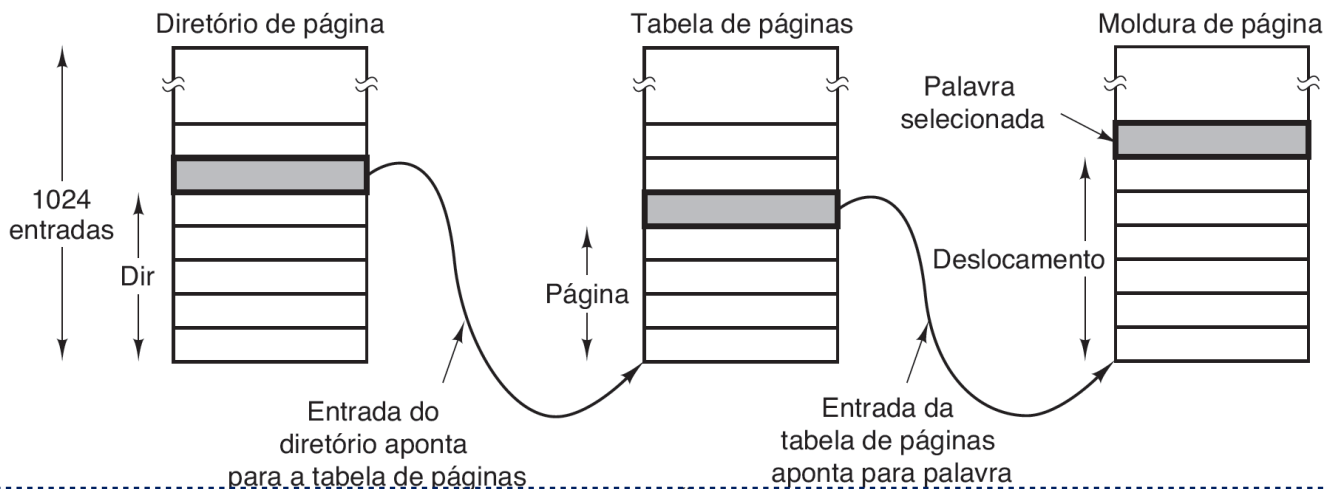


Sem paginação, este já é o endereço real



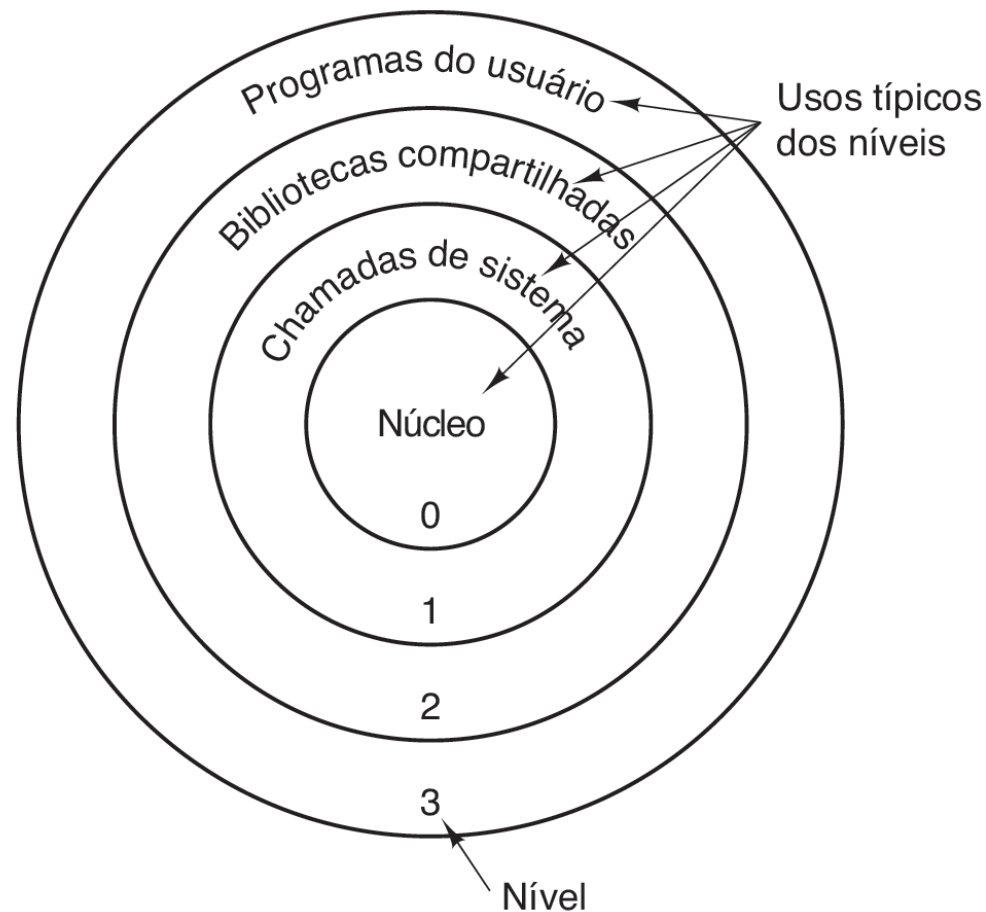
(a)

Resolvendo a página



Gerência de Memória

Níveis de proteção do Pentium



Gerência de Memória

Windows

32 bits → 4 GB

2GB para o processo, 4GB para o sistema

64 bits → 8 TB

Processos podem optar por trabalhar com 2GB apenas

Working Set + Aging

Processos podem saber o tamanho e modificar o conjunto de páginas de trabalho

Páginas são “envelhecidas” (aging) de acordo com uso e removidas do working set



Gerência de Memória

Linux

- Não usa segmentação (segmentos fixos)

- Copy-on-write

 - Páginas são compartilhadas entre processos o máximo possível

 - São copiadas sob demanda apenas quando escritas

- Usa aproximação de LRU

 - Listas de páginas ativas e inativas

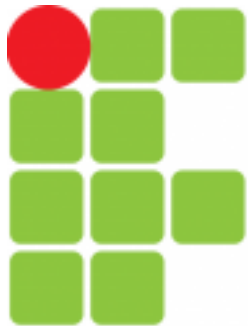
 - Página na lista de inativas são retiradas



Revisão

- Qual o problema de não trabalhar com abstração de memória?
- O que é memória virtual? Como a paginação implementa MV?
- O que é page fault? Como é realizado? Qual critério usar para substituir páginas?
- O que é segmentação? Qual a diferença de paginação? Podem ser usados em conjunto?





INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
PERNAMBUCO

5

Tecn. em Análise e Desenvolvimento de Sistemas **Sistemas Operacionais** Gerência de Memória

Prof. Ramide Dantas

