



Gerenciamento de Memória

Walter Fetter Lages

w.fetter@ieee.org

Universidade Federal do Rio Grande do Sul

Escola de Engenharia

Departamento de Engenharia Elétrica

ENG04008 Sistemas de Tempo Real

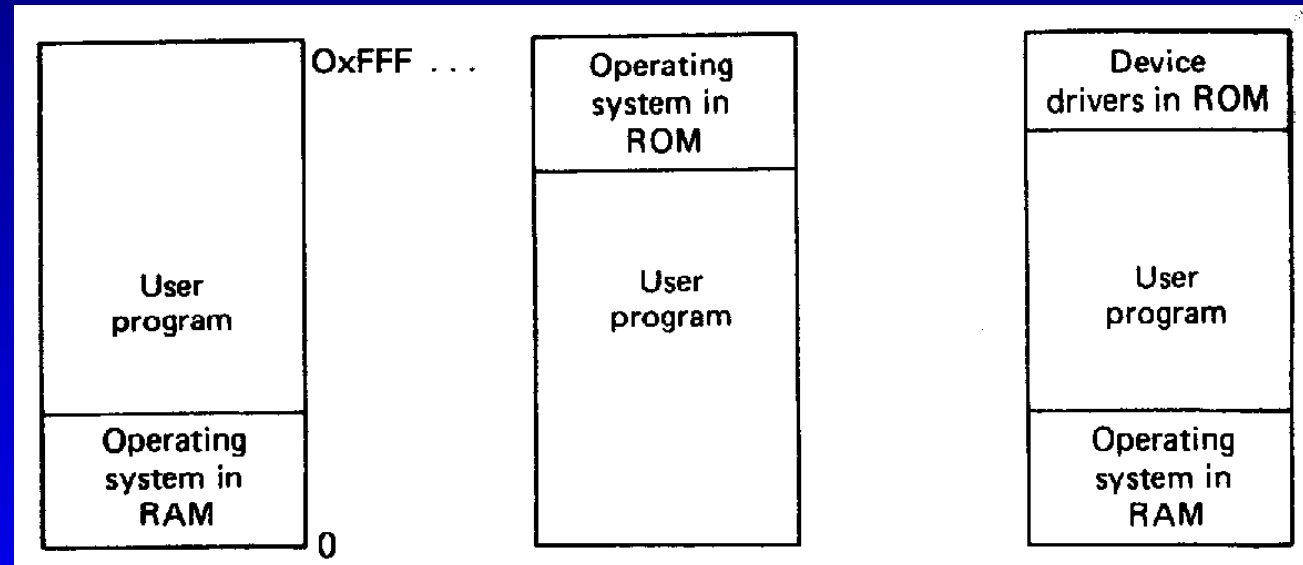


Introdução

- Alocação e desalocação de memória
 - Estática
 - Dinâmica
- Memória virtual
 - Overlay
 - Paginação
- Segmentação de memória

Sistemas Monotarefa

- Um único programa na memória
 - O programa utiliza toda a memória
 - O programa deve ter todos os drivers de I/O
- Divisão da memória entre o programa e o sistema operacional

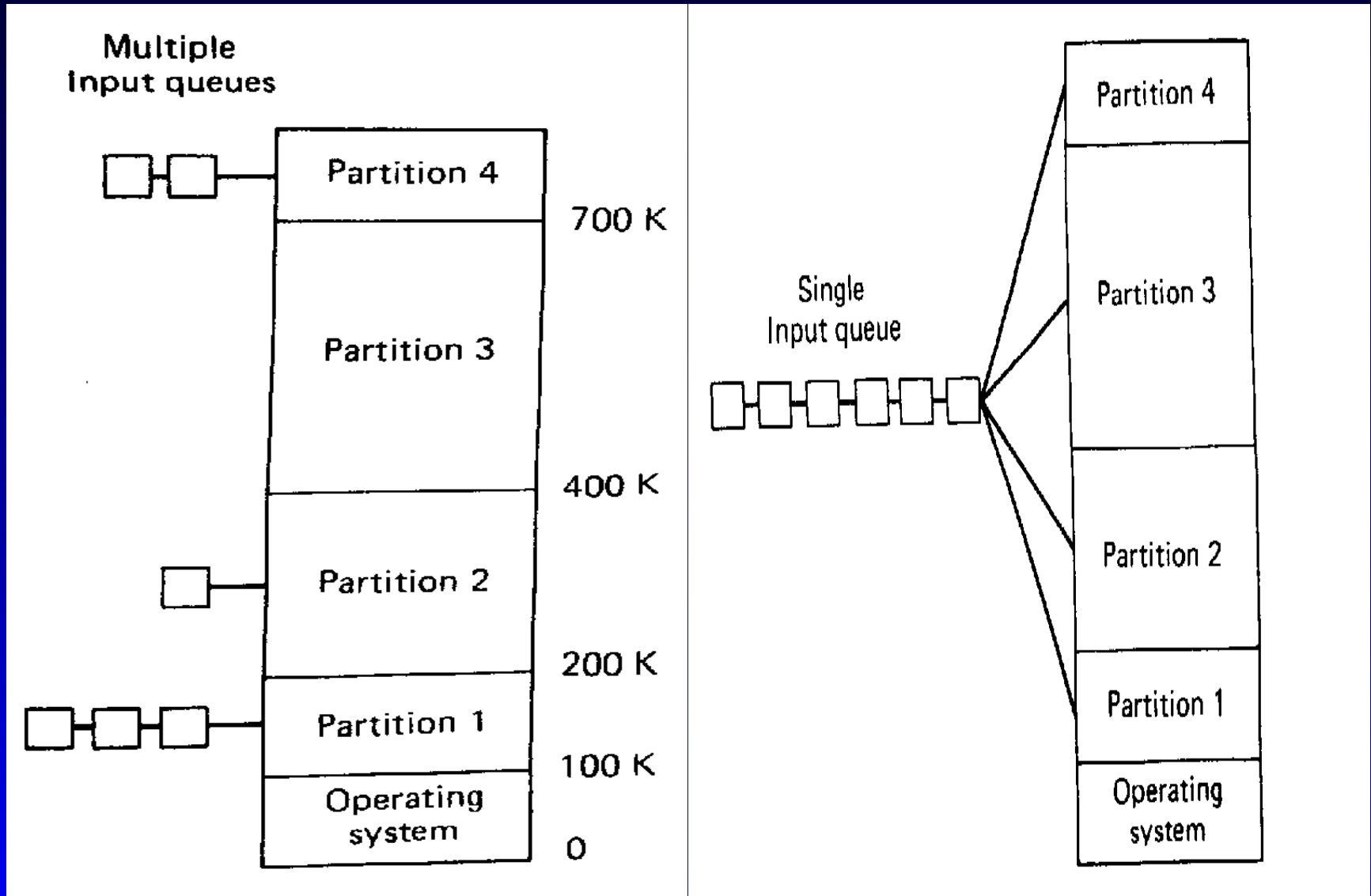




Sistemas Multitarefa

- Partições fixas
 - Partições definidas manualmente pelo operador
 - Uma partição de tamanho suficiente é alocada para cada tarefa
 - Em geral, o número de tarefas $>$ número de partições \Rightarrow diversas políticas para utilização das partições
 - Uma fila de tarefas para cada partição
 - Uma fila única
 - Fila única com *best-fit*
 - Fila única com *best-fit* e máximo de *skip-over*

Partições Fixas

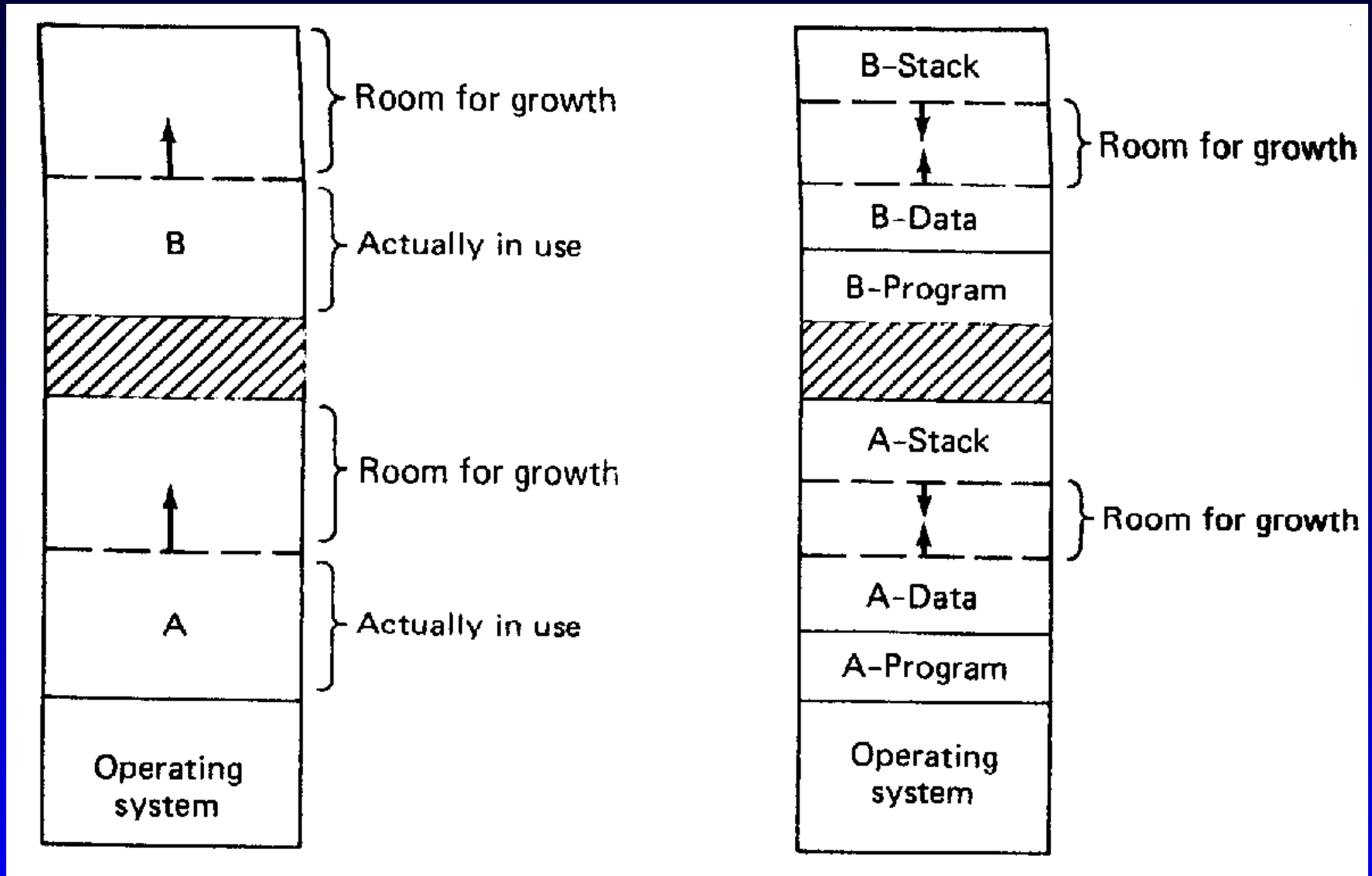




Partições Variáveis

- O tamanho das partições varia conforme a criação/destruição das tarefas
- Fragmentação de memória
- Compactação de memória
 - Consome muito tempo -> normalmente feito durante a alocação/desalocação de memória em sistemas não tempo real
- Espaço para a memória das tarefas crescer
 - Alocação dinâmica de memória para dados

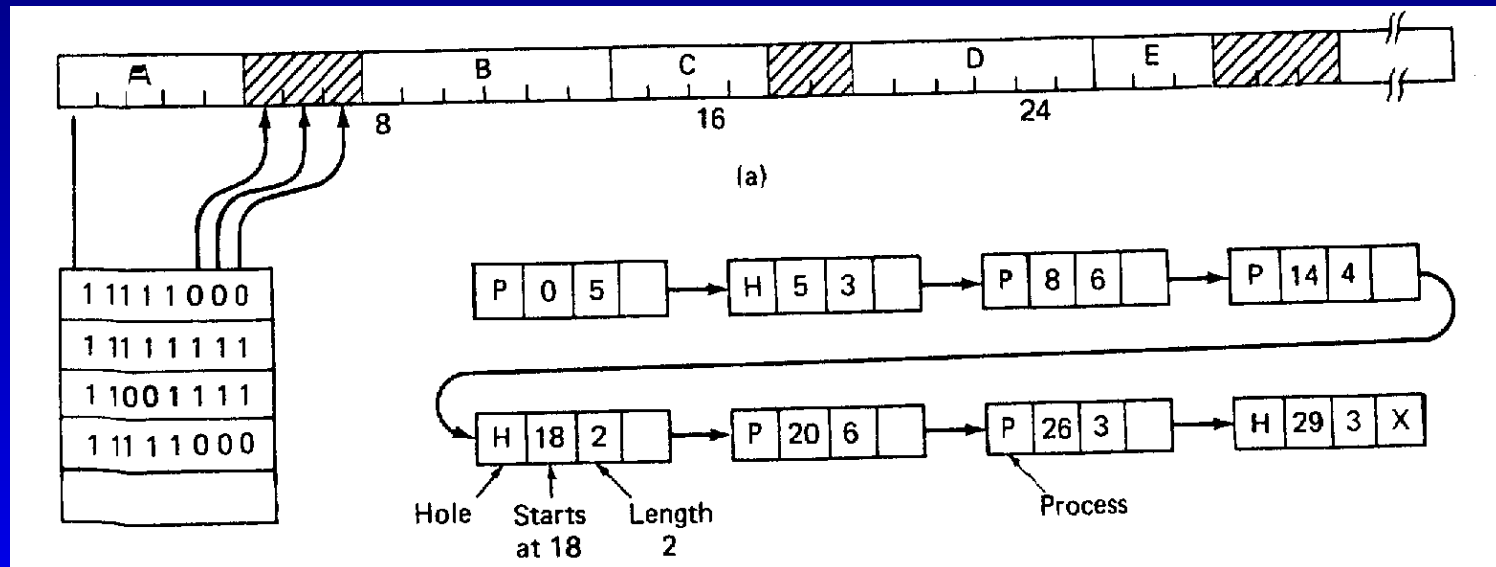
Lay-out de Memória das Tarefas



Métodos de Gerenciamento



- Bit-map
 - Memória dividida em unidades de alocação
 - Existe um bit-map para sinalizar se cada unidade de alocação está alocada ou livre
 - Procura por espaços livres não é eficiente





Métodos de Gerenciamento

- Listas ligadas
 - Existe uma lista ligada que relaciona os blocos de memória alocados e livres
 - Normalmente a lista é ordenada por endereço
- Algoritmos de Alocação de memória
 - First-fit
 - Next-fit
 - Best-fit
 - Worst-fit
 - Quick-fit

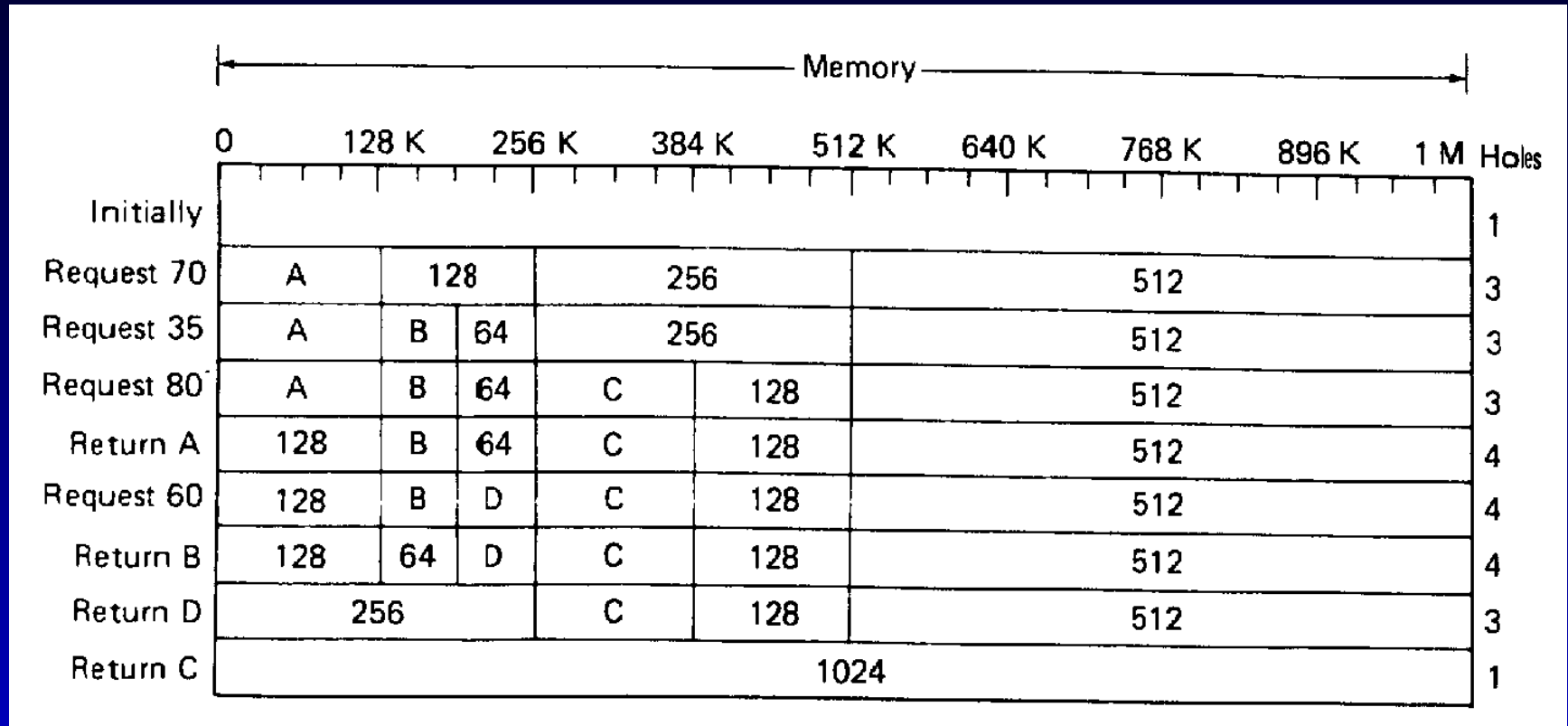


Métodos de Gerenciamento

- Buddy System
 - Utiliza o fato dos endereços serem binários
 - Existe uma lista de blocos livres de tamanho 1, 2, 4, 8 ... tamanho da memória
 - Ineficiente no uso da memória
 - Fragmentação interna



Buddy System





Relocação

- Em sistemas multitarefa, cada tarefa terá um endereço inicial diferente. Como o programador/compilador/linker saberá o endereço inicial (e endereços de subrotinas, dados, etc...) da tarefa?
 - Problema da relocação
 - Gerar apenas código com endereços relativos
 - Código relocável
 - Relocação durante a carga da tarefa na memória

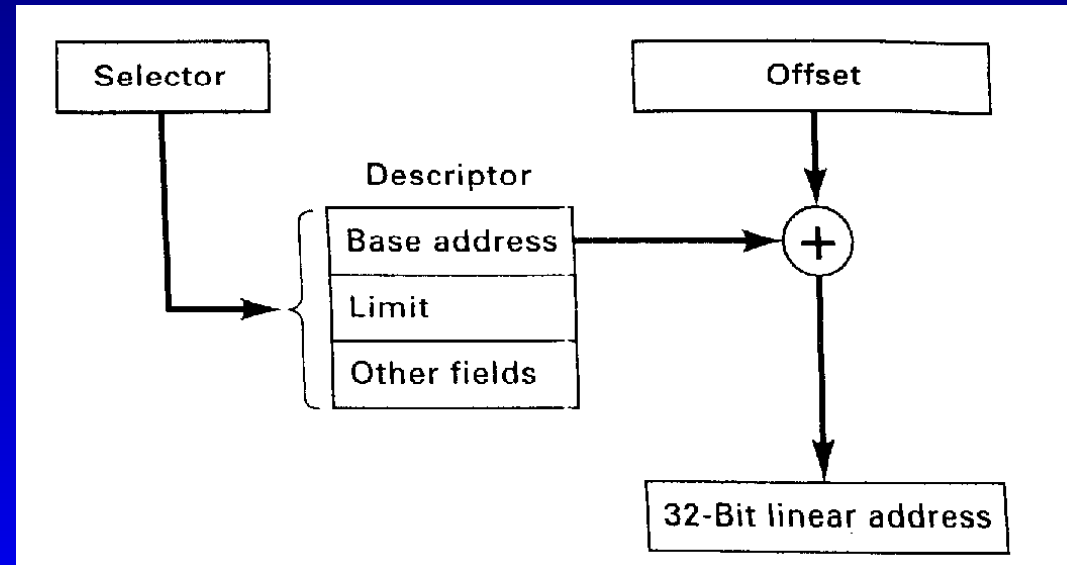


Segmentação

- Associa-se a cada área de memória um valor de base e um valor de limite
- base=endereço inicial
- limite=endereço máximo ou tamanho
- Todas as tarefas são codificadas assumindo que a sua área de memória começa em 0
- Quando a tarefa é selecionada para executar, base e limite são carregados em registradores especiais na CPU

Segmentação

- Endereço final = Offset + base
 - Soma feita por hardware a cada acesso à memória
- Endereços são verificados quanto ao limite
 - Uma tarefa não pode acessar a área das outras





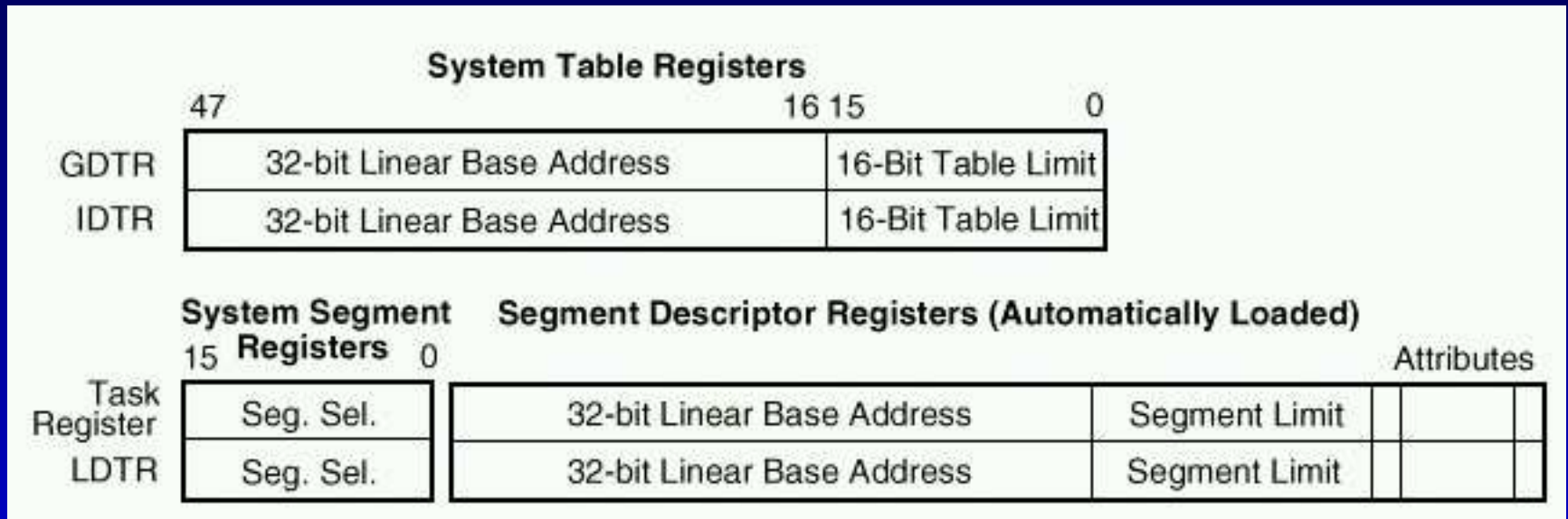
Segmentação

- O seletor e descritor de segmento são controlados pelo sistema operacional
- O offset é controlado pelo programa de aplicação
- Proteção
 - Usualmente o descritor de segmentos tem alguns bits associados que são utilizados para determinar as características de proteção do segmento
 - Permissões de leitura, escrita, execução
- Cache de descritor de segmento
 - Evita dois acessos a memória
 - Alterado apenas quando o registrador de seletor de segmento é alterado

Segmentação no ix86

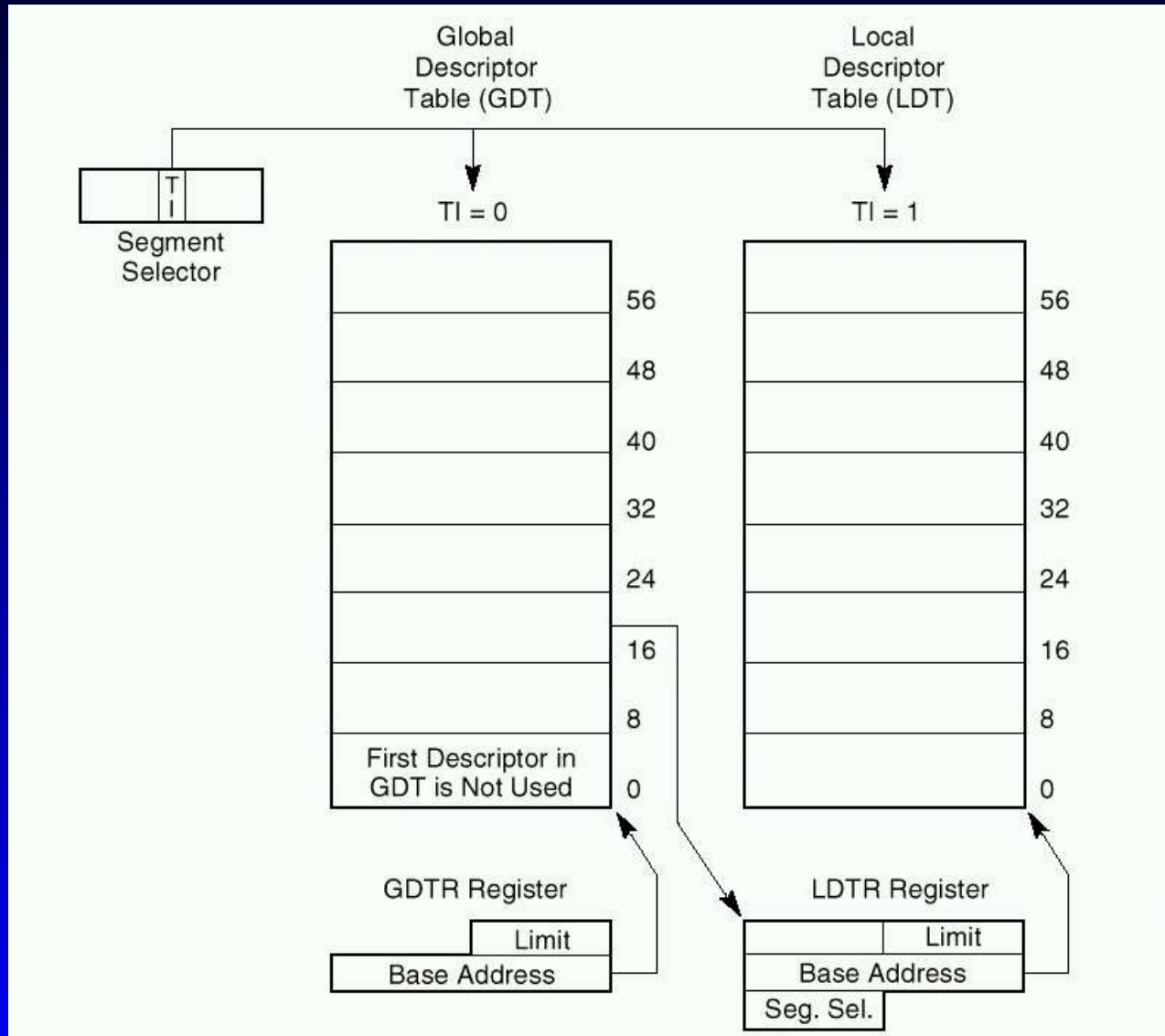


- Local Descriptor Table (LDT)
- Global Descriptor Table (GDT)

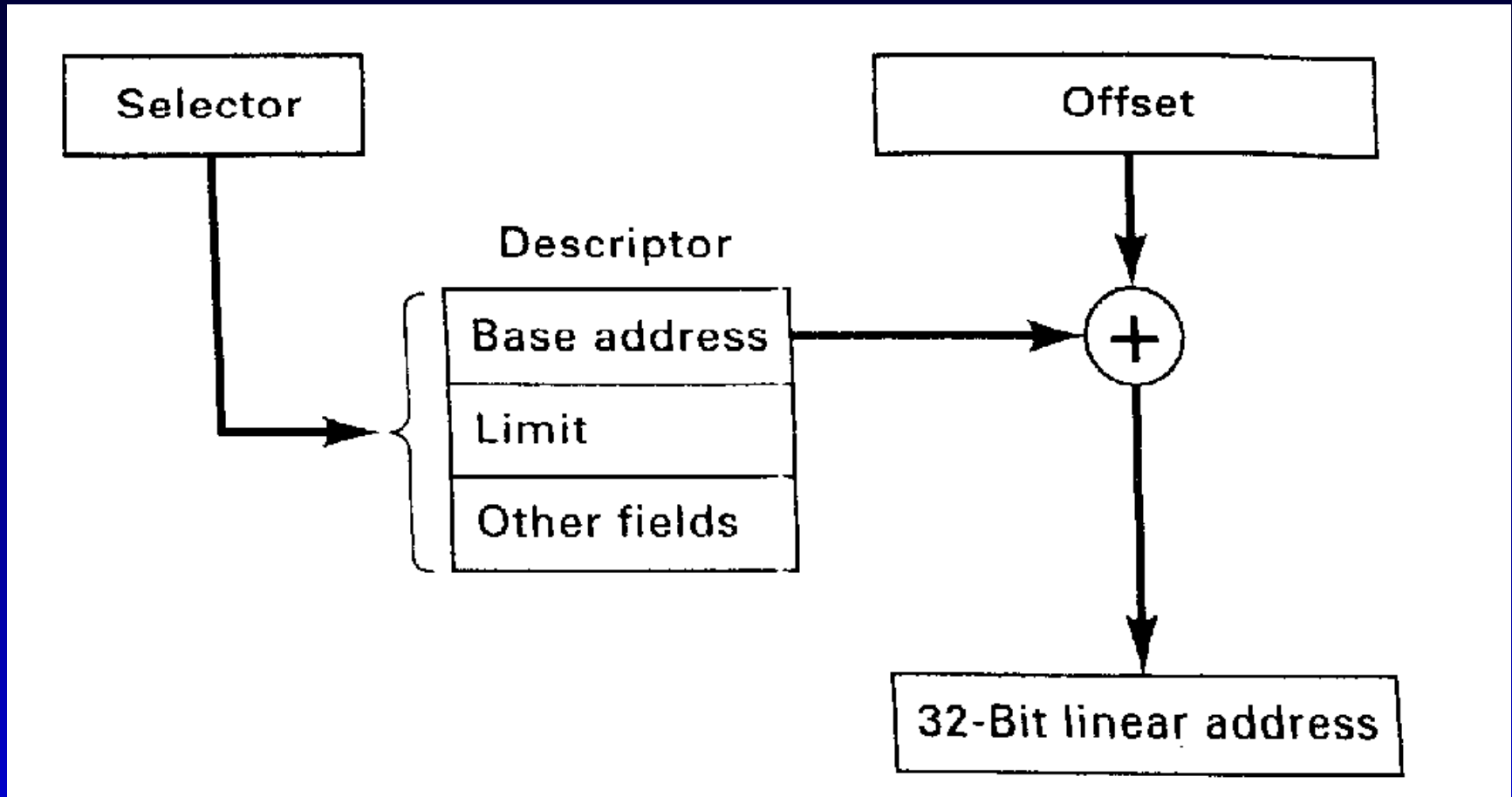




GLDT e LDT



Segmentação no ix86



Seletor de Segmento no ix86



Table Indicator

0 = GDT

1 = LDT

Requested Privilege Level (RPL)

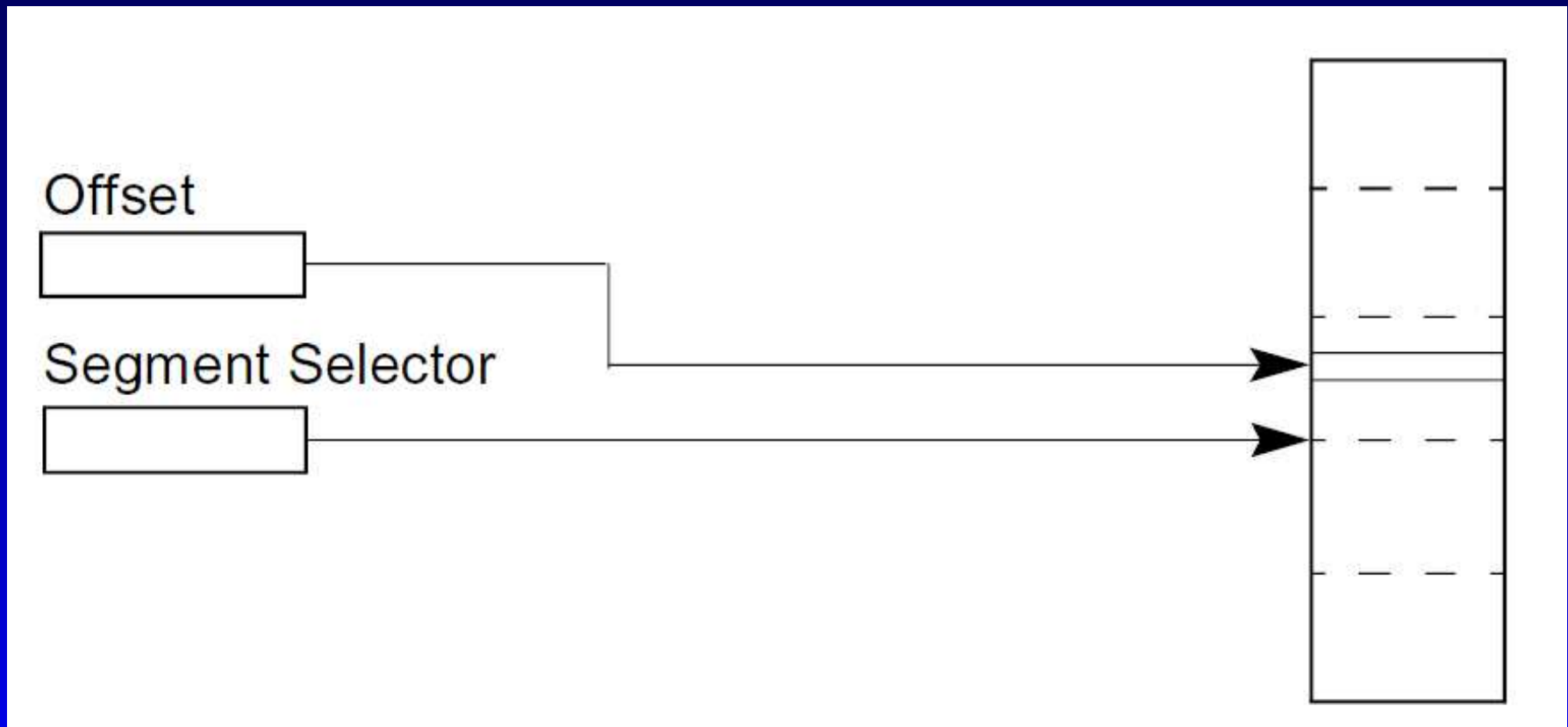


Segmentação no 8086/8088

- Processadores sem modo protegido
- Sem proteção, não tem sentido a existência de permissões e limite no segmento
 - O descritor de segmento resume-se à base
 - Como os registradores são de 16 bits decidiu-se limitar o início de segmentos apenas alinhados com "parágrafos"
- Segmentação mal implementada
 - O "tamanho máximo" do segmento é 64KB
 - O limite do segmento não é garantido
 - Na verdade é apenas um esquema para endereçar 1MB utilizando 2 registradores

Segmentação no 8086/8088

- Endereço físico = reg. de segmento * 16 + offset
- Endereço físico = reg. de segmento « 4 + offset



Memória Virtual

- Paginação
 - Falta de página
 - Page-table
- Endereço
 - Virtual
 - Físico

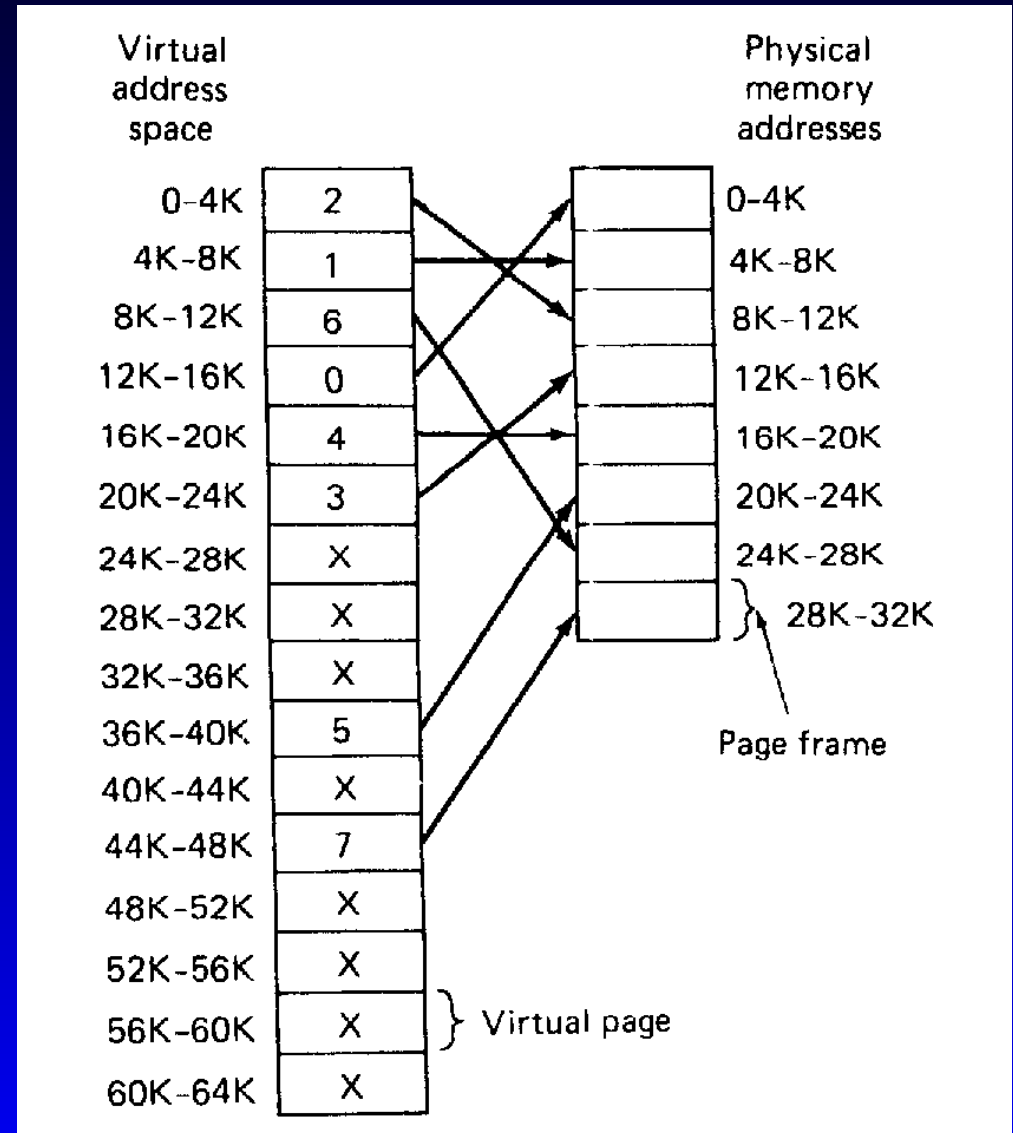
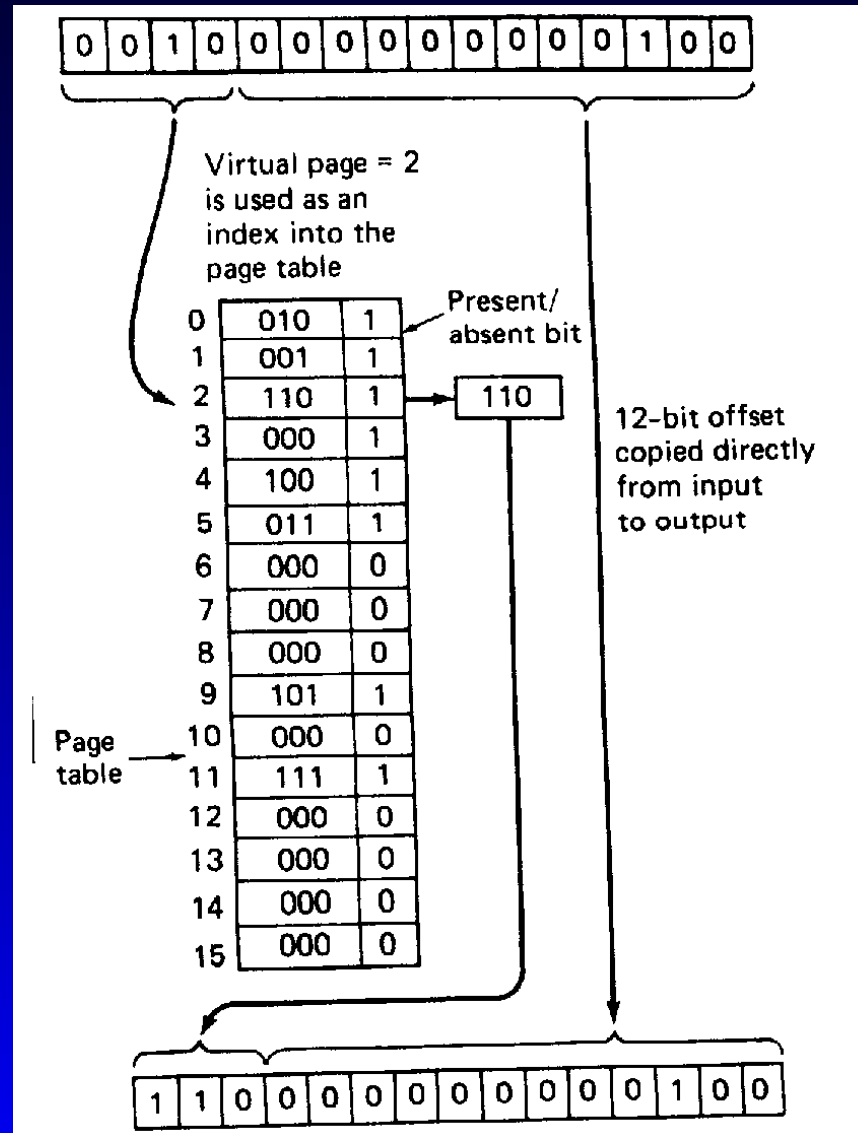


Tabela de Páginas

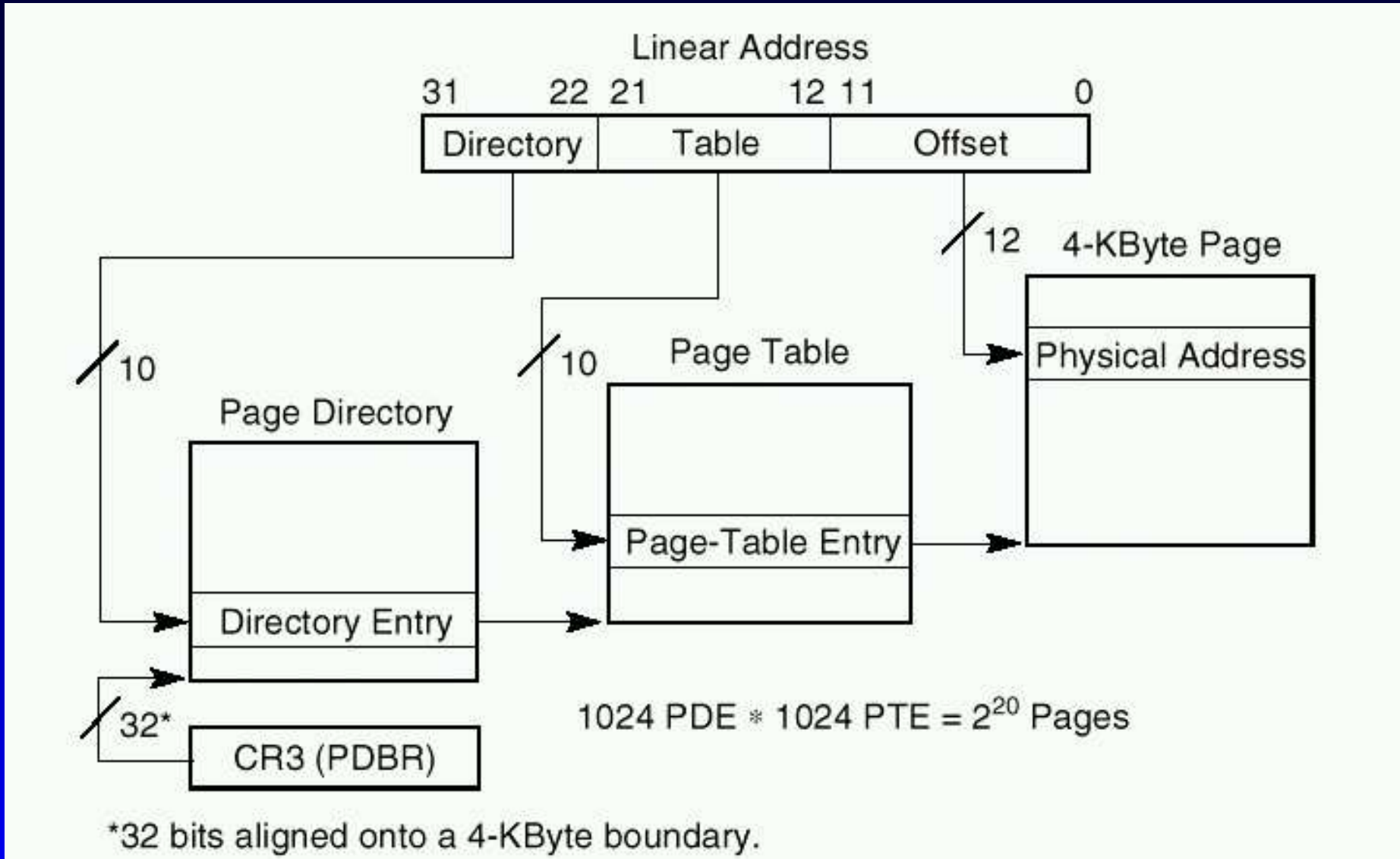




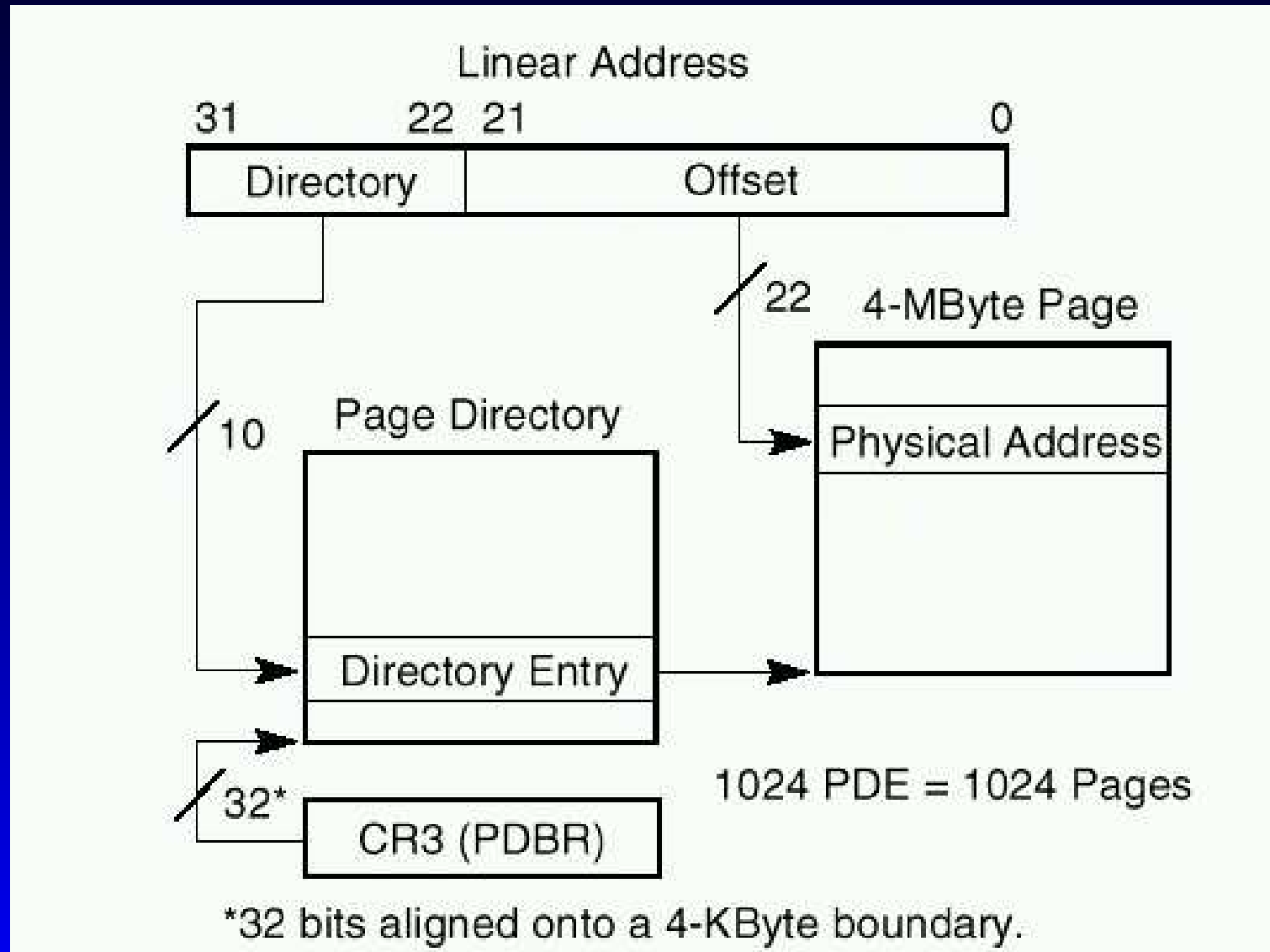
Hardware de Paginação

- Usualmente a tabela de páginas é mantida em memória e não em registradores
 - Registrador de base da tabela de páginas
 - Translation look-aside buffer (TLB)
 - Evita dois acessos a memória
 - Memória associativa
- Páginas também podem ter bits de proteção associados
 - Bit de presença/ausência

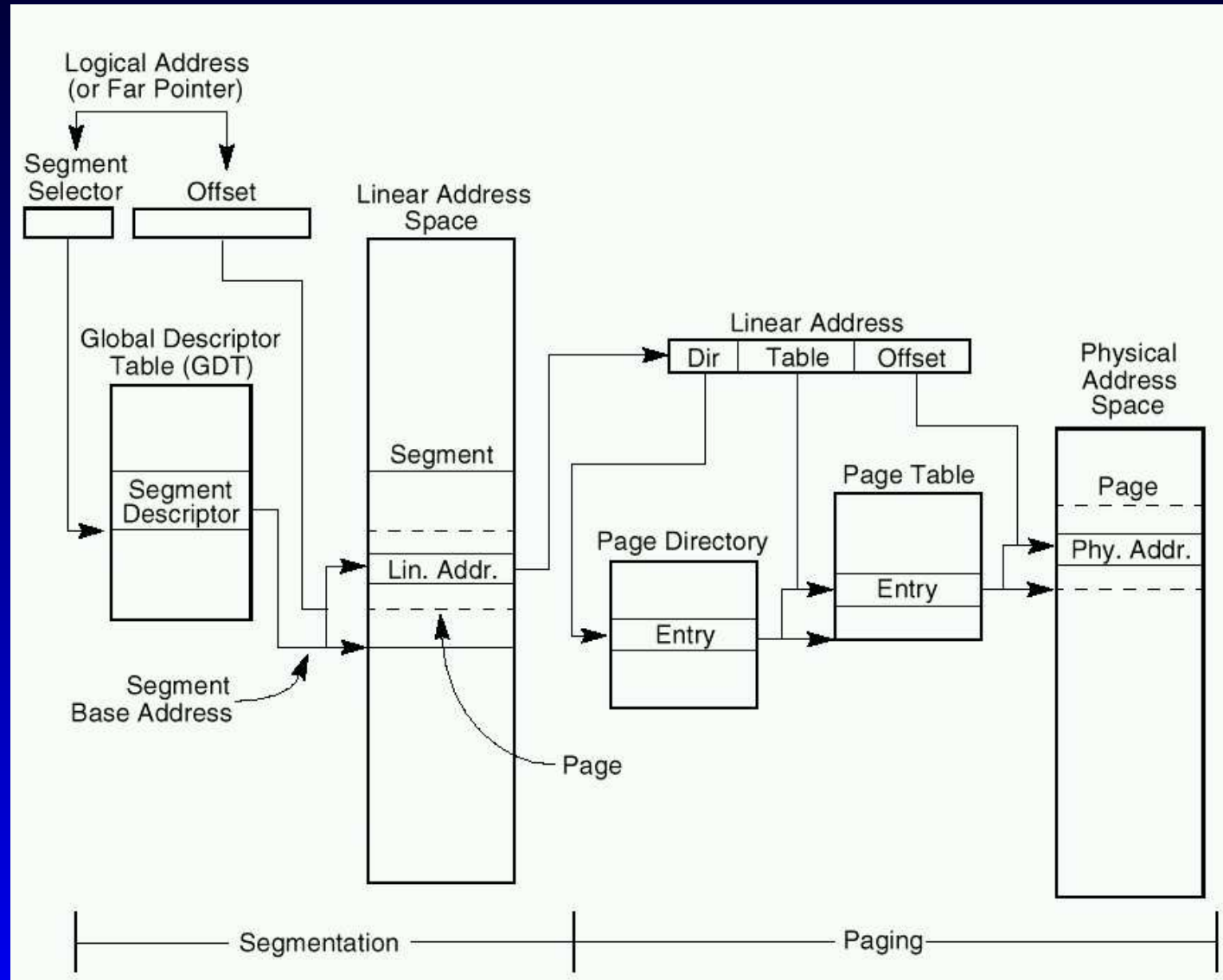
Páginas de 4KB no ix86



Páginas de 4MB no ix86



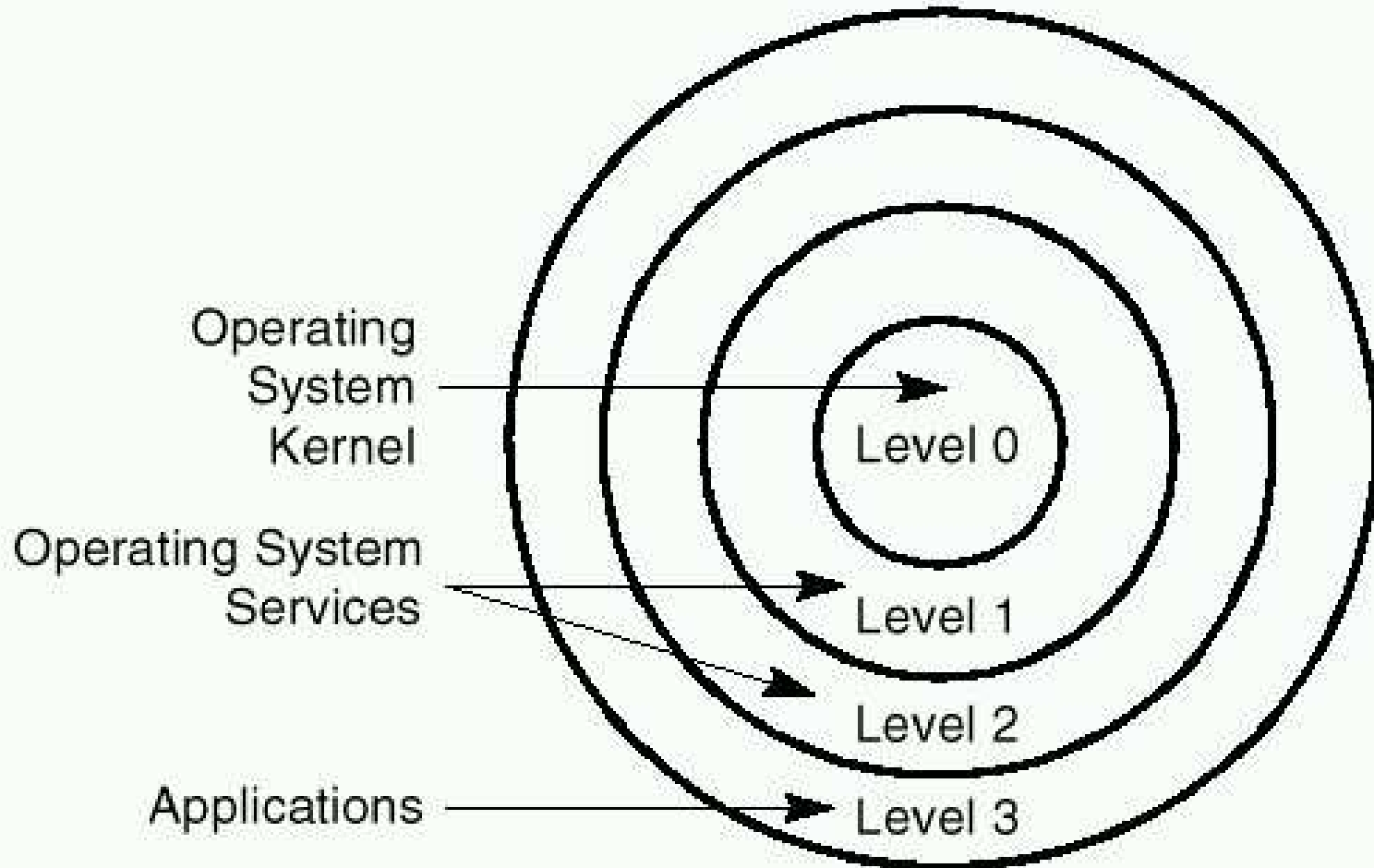
Segmentação e Paginação no ix86



Proteção de Memória no ix86



Protection Rings

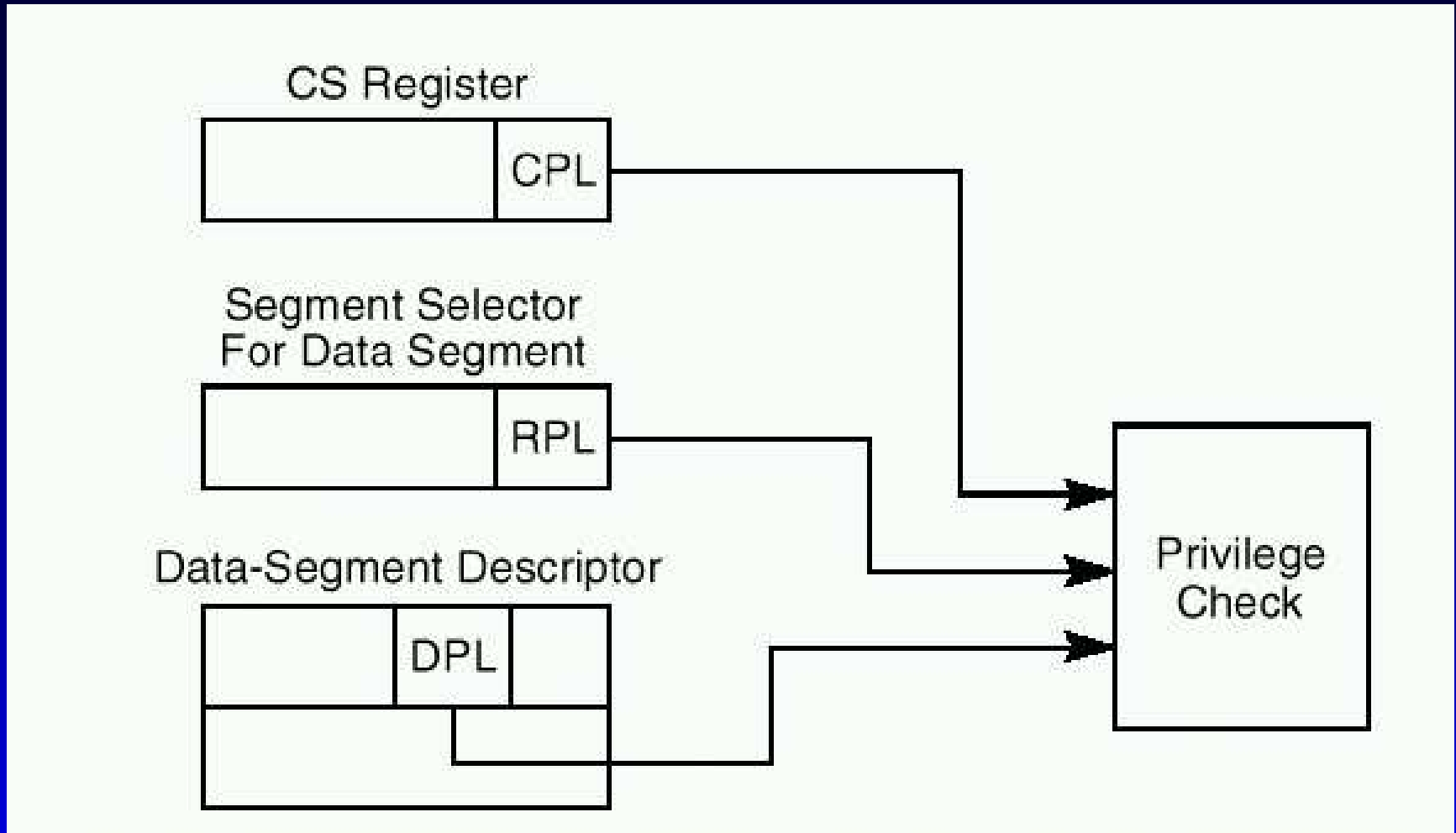




Proteção de I/O no ix86

- Existe um nível de privilégio de I/O
 - Apenas processos com nível de privilégio menor ou igual podem executar instruções de I/O
- Associado a cada processo pode existir um bitmap de permissões de I/O
 - Apenas as portas habilitadas podem ser acessadas
- Hardware Virtual
 - Virtualização de dispositivos

Verificação de Privilégios



I/O Permission Bit Map

