

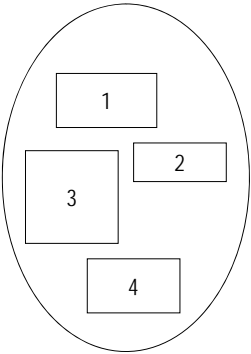
Sistemas Operacionais

Segmentação
Segmentação com paginação

Segmentação

- Leva em consideração a visão de programadores e compiladores
- Um programa é uma coleção de segmentos, tipicamente:
 - Código
 - Dados alocados estaticamente
 - Dados alocados dinamicamente
 - Pilha
- Um segmento pode ser uma unidade lógica
 - e.g: procedimentos (funções), bibliotecas, etc.
- Gerência de memória pode dar suporte diretamente ao conceito de segmentos

Esquema lógico da segmentação



Espaço de usuário



Espaço físico

Comparação entre paginação e segmentação

	Paginação	Segmentação
Usuário é consciente da técnica que está sendo usada?	Não	Sim
Nro. de espaços de endereçamento lógicos lineares?	1	n
O espaço lógico pode exceder a capacidade do endereço físico?	Sim (se usado memória virtual)	Sim (se usado memória virtual)
Faz distinção entre dados, pilhas e códigos?	Não (tudo são páginas)	Sim
Facilita acomodação de áreas com tamanhos variáveis?	Não	Sim
Principal objetivo	Simplificar a obtenção de um espaço de endereçamento lógico maior que o físico. (via memória virtual)	Permitir que programas e dados sejam divididos em diferentes unidades lógicas (facilitar proteção e compartilhamento).

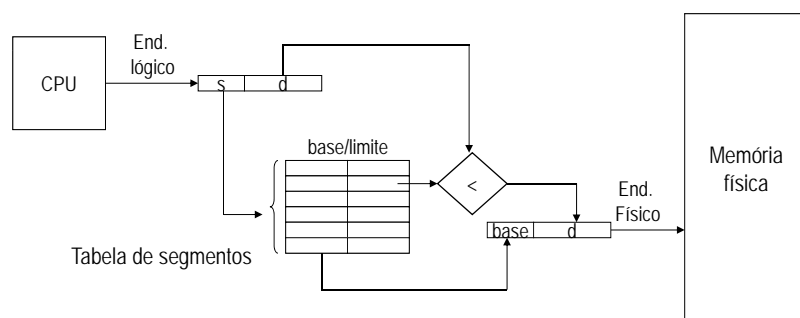
Endereço lógico em segmentação

- Endereço lógico é composto por duas partes:
 - Número de segmento
 - Deslocamento dentro do segmento
- Os segmentos não necessitam ter o mesmo tamanho
- Existe um tamanho máximo para um segmento
- Segmentação é similar a alocação particionada dinâmica

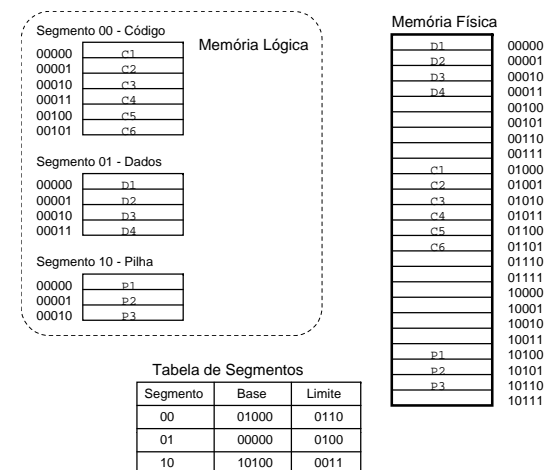
Tradução de endereço lógico em endereço físico

- Tradução é feita de forma similar a paginação (via tabela)
 - Tabela de segmentos
- Entrada na tabela de segmento:
 - Base: endereço inicial (físico) do segmento na memória
 - Limite: tamanho do segmento
 - Bits adicionais para controle de acesso, proteção e compartilhamento
- Necessidade de verificar a cada acesso se ele é válido
 - Hardware (comparador)

Esquema de tradução da segmentação



Exemplo de tradução de endereço lógico em físico



Aspectos de proteção e compartilhamento

- Os princípios já estudados para paginação continuam válidos para a segmentação
 - e.g.; bits de proteção (*rwX*), bit de validade, bits de compartilhamento, etc..
- Segmentação adiciona a possibilidade de compartilhar trechos da área de código
 - Facilita o procedimento de ligação (*linker*), pois uma biblioteca pode ter seu *entry point* sempre associado a informação segmento *s*, deslocamento *zero*.

Implementação da segmentação

- O sistema operacional deve manter
 - Áreas de memória livre e seu respectivo tamanho
 - Mapeamento de segmentos lógicos a segmentos físicos
 - Tabela de segmentos
- Dois problemas (paginação tem os mesmos)
 - Que tipo de estrutura de dados utilizar ?
 - Onde armazenar essas estruturas ?

Informação de áreas livres

- Lista encadeada:
 - Análogo a gerência de memória com alocação particionada dinâmica
- Cada elemento possui endereço inicial da área e seu tamanho
- Método de procura:
 - First fit, best fit, worst fit

Desvantagem da segmentação

- A segmentação provoca fragmentação externa quando segmentos começam a liberar memória
- Mesmos problemas da alocação particionada dinâmica com as mesmas soluções:
 - Concatenação de segmentos adjacentes
 - Compactação da memória

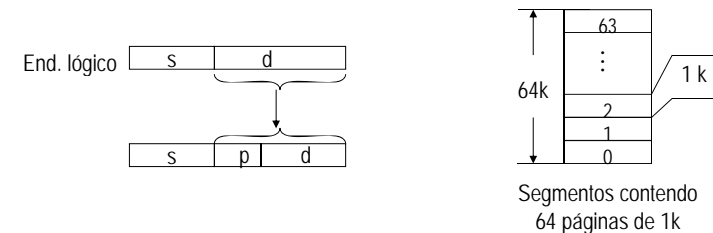
Solução para fragmentação externa

- A paginação é a solução natural para a fragmentação externa
- Analisar o problema sob dois pontos extremos:
 - Um processo é um único segmento
 - Cada byte é um segmento
 - Sem fragmentação externa, nem interna
 - Não viável pelos *overheads* envolvidos
 - Similar a página de 1 byte
- Solução: meio termo entre os extremos
 - Fazer um segmento ser composto por um número fixo (e reduzido) de bytes
 - Equivale a ter o segmento dividido internamente em blocos

13

Segmentação com paginação

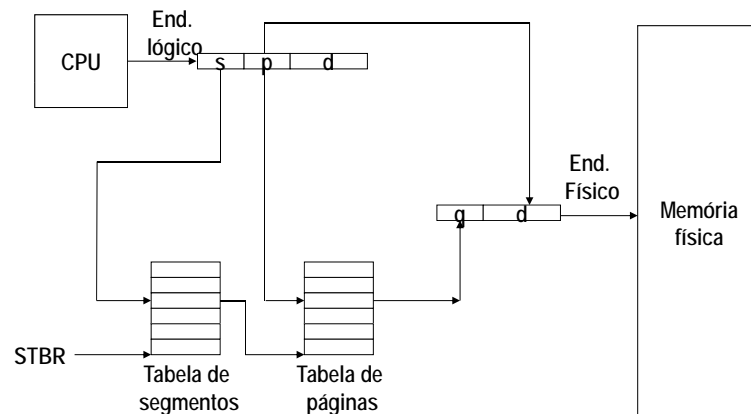
- Recuperar as vantagens dos dois métodos em relação a fragmentação:
 - Fragmentação interna: paginação apresenta, segmentação não
 - Fragmentação externa: segmentação apresenta, paginação não
- Solução se traduz em paginar segmentos



Sistemas Operacionais

14

Esquema de tradução da segmentação c/ paginação



Sistemas Operacionais

15

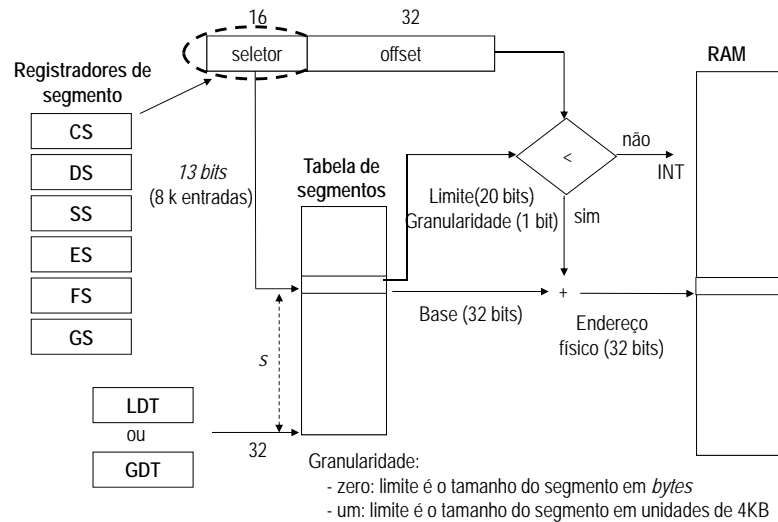
Estudo de caso: arquitetura Intel

- Segmentação com paginação
- Endereço lógico é 48 bits:
 - 2 bits são usados para proteção (RPL)
 - 1 bit é empregado para identificar a tabela (Local - LDT ou Global - GTD)
 - 13 bits identificam um nro de segmento em duas tabelas (LDT e GDT)
 - 32 bits servem como deslocamento dentro de cada segmento
- Instruções referenciam endereços (acesso) em apenas 32 bits
 - O valor do seletor é colocado em registradores específicos
 - Registradores de segmento (CS, DS, ES, SS, FS e GS)
 - Qualquer acesso a memória consiste sempre como referência a um registrador de segmento e a um deslocamento

Sistemas Operacionais

16

Tradução endereço lógico (segmentação pura)



Sistemas Operacionais

17

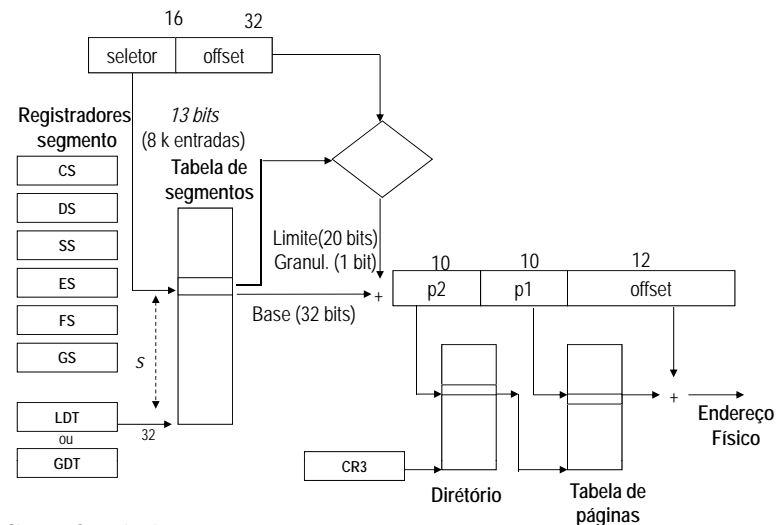
Paginação no Intel: segmentação paginada

- Uso opcional no 80386
 - Desativada no momento do *reset* do computador
- Características gerais
 - Páginas de 4 Kbytes
 - Espaço de endereçamento linear e espaço de endereçamento físico de 4 Gbytes (2^{32})
 - Máximo de 1 Megapáginas
- Tabela de páginas considera dois níveis
 - Registrador (CR3) aponta para o diretório de tabela de páginas
- Diretório de tabela de páginas
 - 1 K entradas de 4 bytes
 - Cada entrada aponta para uma tabela de páginas

Sistemas Operacionais

18

Tradução endereço lógico (segmentação paginada)



Sistemas Operacionais

19

Leituras complementares

- A. Tanenbaum. *Sistemas Operacionais Modernos* (3ª edição), Pearson Brasil, 2010.
 - Capítulo 3: seção 3.7
- A. Silberchatz, P. Galvin; *Sistemas Operacionais*. (7ª edição). Campus, 2008.
 - Capítulo 8 (seções 8.6 e 8.7)
- R. Oliveira, A. Carissimi, S. Toscani; *Sistemas Operacionais*. Editora Bookman 4ª edição, 2010
 - Capítulo 6 e capítulo 7 (seção 7.5)

Sistemas Operacionais

20