

# GERENCIAMENTO DE PROCESSOS

Sistema Monotarefa:



O tempo de (E/S) é muito maior que o de processamento.

O processador fica ocioso a maior parte do tempo.

# GERENCIAMENTO DE PROCESSOS

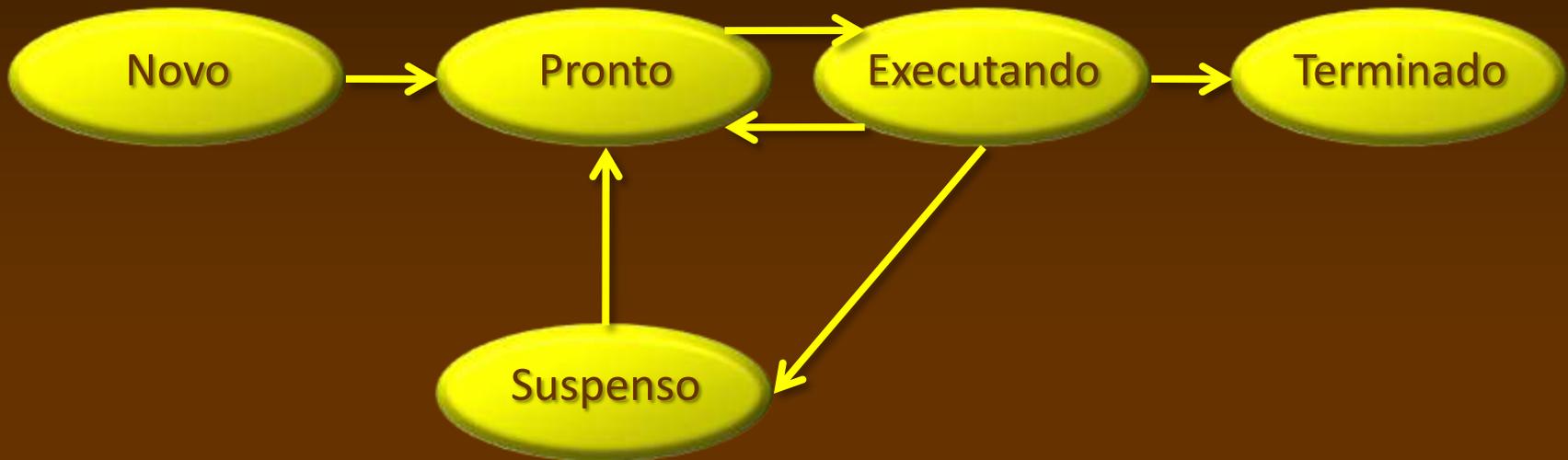
Sistema Multitarefa:



O tempo de espera de E/S pode ser aproveitado para outras tarefas.

Nem todas as tarefas têm a mesma necessidade de E/S ao mesmo tempo.

## ESTADOS DE UM PROCESSO

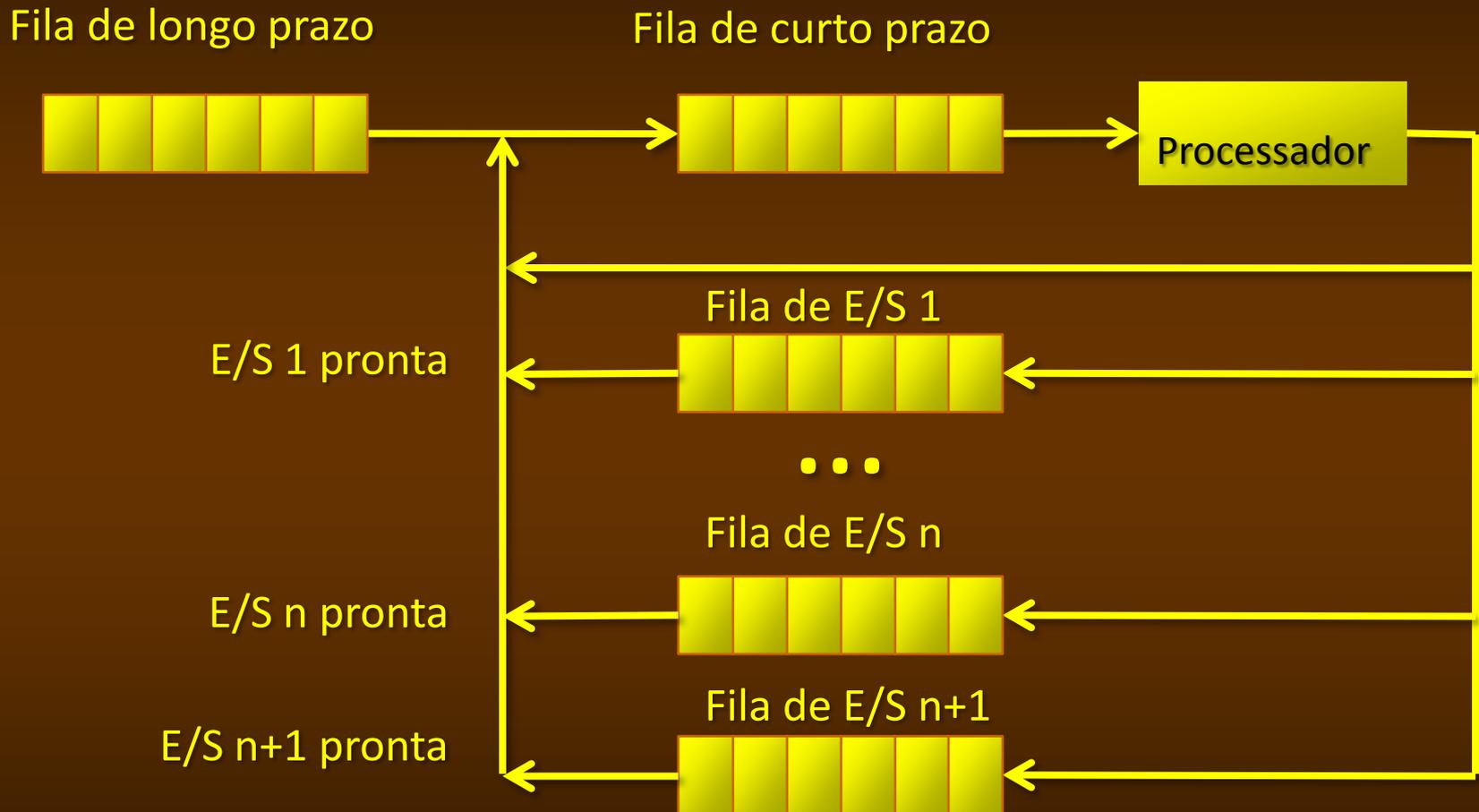


Um processo fica suspenso enquanto aguarda algum evento, p. ex. E/S.

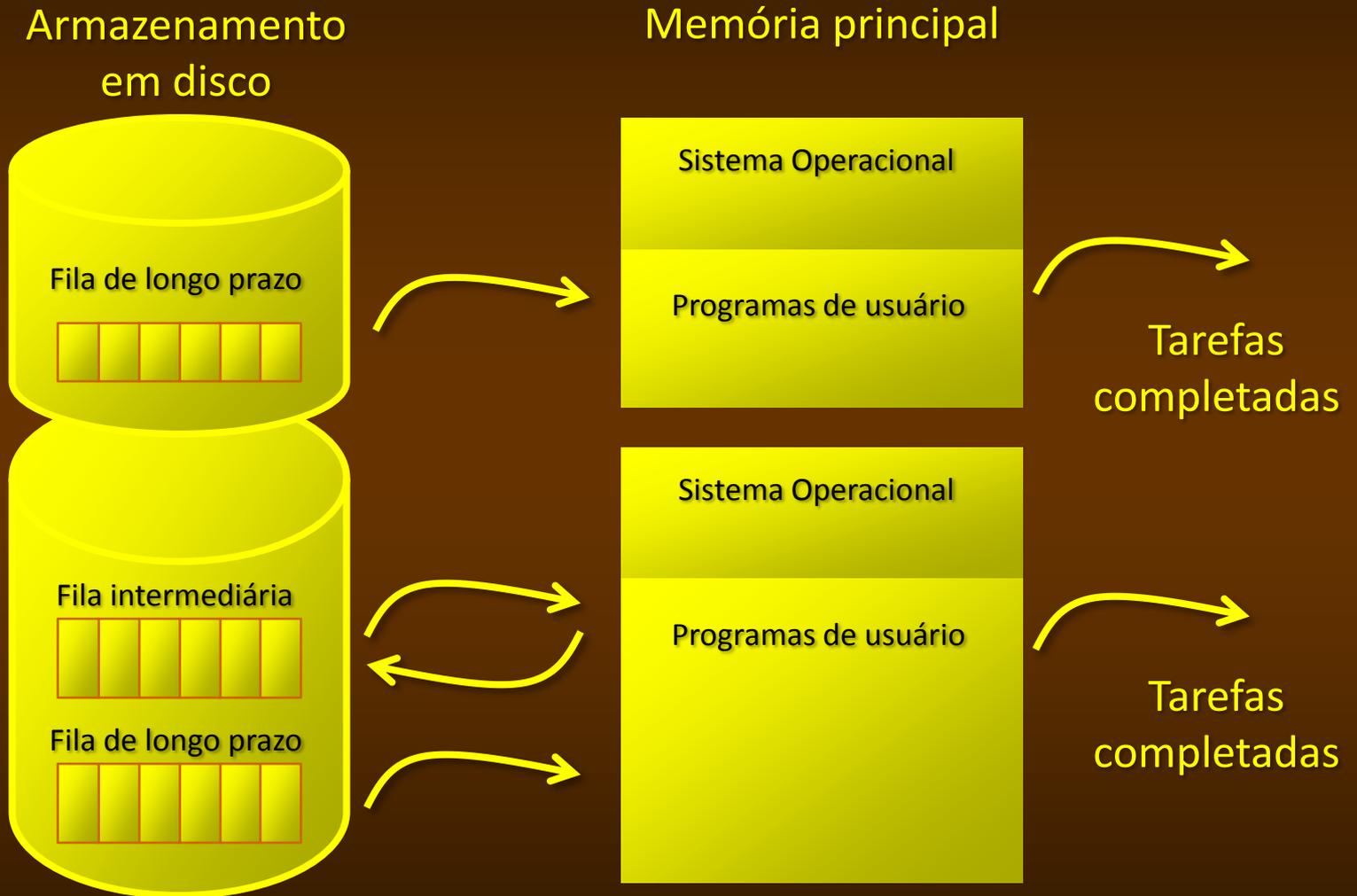
Os termos tarefa, job, processo, etc... Dependem muito do contexto.

Em um S.O. Windows, p. ex., um “job” é um conjunto de processos, que devem terminar juntos. Cada processo é constituído por um ou mais “threads”, que assumem os estados acima e são comutados pelo S.O.

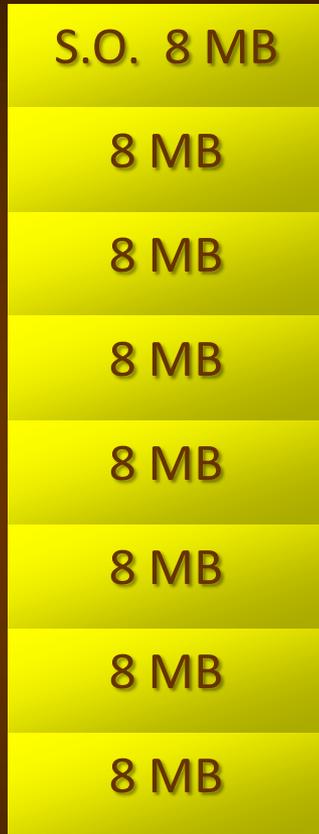
# ESCALONAMENTO DO PROCESSADOR



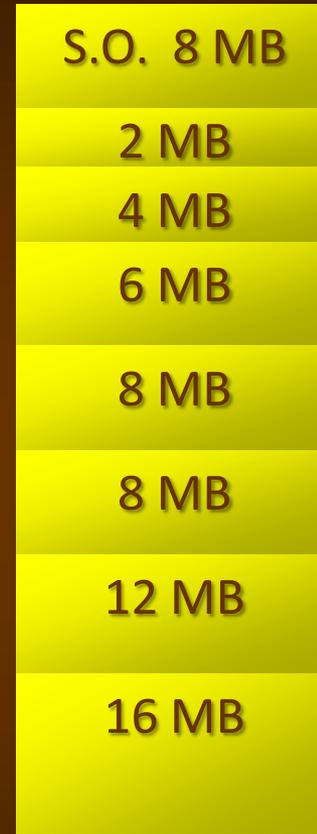
# ESCALONAMENTO DO PROCESSADOR



# PARTICIONAMENTO DA MEMÓRIA MULTITAREFA

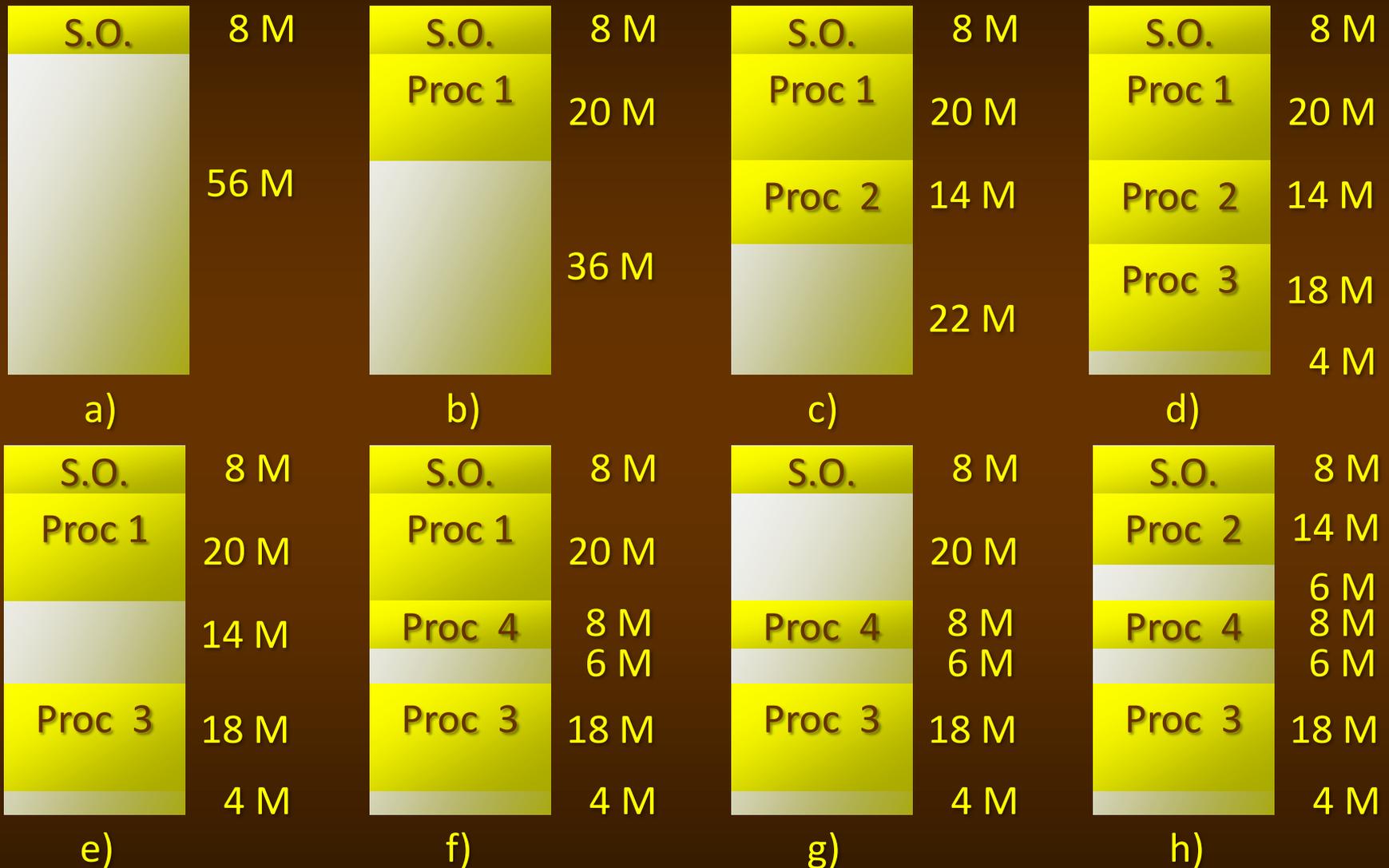


a) Particionamento fixo  
tamanho fixo



b) Particionamento fixo  
tamanho variável

# PROBLEMAS DO POSICIONAMENTO DINÂMICO



## PROCESSOS EM ENDEREÇOS VARIÁVEIS

A carga de processos em várias posições distintas da memória ao longo de sua execução, sem a intervenção do programador exige a diferenciação entre:

- ❑ ENDEREÇO LÓGICO – É aquele visto pelo programador. É o endereço referenciado ao início do programa. Se o programa for constituído de vários segmentos, é o endereço referenciado ao início do segmento.
- ❑ ENDEREÇO FÍSICO – É aquele efetivamente ocupado na memória. Não é conhecido em tempo de programação. É definido em tempo de execução pelo S.O.
- ❑ No caso da arquitetura IA-32, a localização (e privilégios) de cada segmento (código, dados, pilha) é definida por meio dos registradores de segmento (CS, DS, SS, etc. ) que são ajustados na carga do programa pelo S. O.

# OTIMIZAÇÃO DA MEMÓRIA VIA SEGMENTAÇÃO

A segmentação consegue reagrupar regiões livres para otimizar o uso da memória. Surgem porém diversas dificuldades:

- ❑ O processo de reagrupamento irá competir com os demais pela CPU.
- ❑ A cada reagrupamento – ou carga de um novo processo – as referências de acesso inter-segmentos (para programas com mais de um segmento) precisarão ser recomputadas. Não é possível manter os endereços anteriores.
- ❑ O tamanho e a quantidade de segmentos de cada processo estão sob o controle do programador e não do S.O.
- ❑ A implementação da memória virtual – armazenamento de partes do programa em disco – torna-se refém do tamanho dos segmentos.

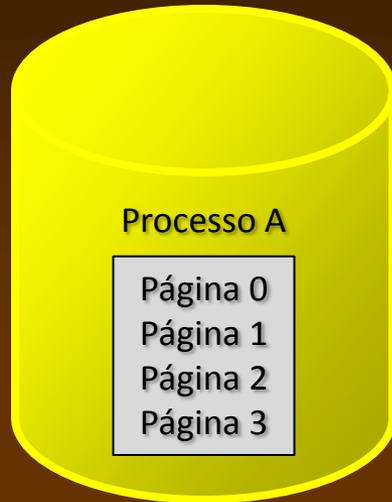
# OTIMIZAÇÃO DA MEMÓRIA VIA PAGINAÇÃO

A paginação visa minimizar ou eliminar os problemas citados anteriormente. Ela constitui-se em:

- ❑ Dividir a memória em pedaços pequenos de mesmo tamanho denominados “frames” de páginas.
- ❑ Dividir os processos em pedaços pequenos de mesmo tamanho denominados “páginas”.
- ❑ Alocar o número exigido de “frames” de página para cada processo.
- ❑ As páginas alocadas a cada processo não precisam ocupar um espaço contíguo de endereços.
- ❑ O S.O. mantém uma lista de “frames” livres e faz uso de tabelas para converter do endereço lógico para o físico.

# ALOCAÇÃO DE “FRAMES” DE PÁGINA LIVRES

Memória Principal



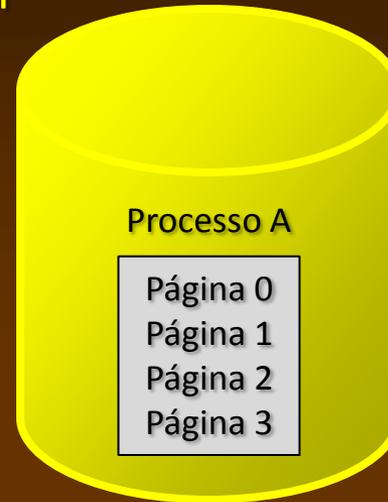
Lista de frames livres

13  
14  
15  
18  
20



antes

Memória Principal



Lista de frames livres  
20

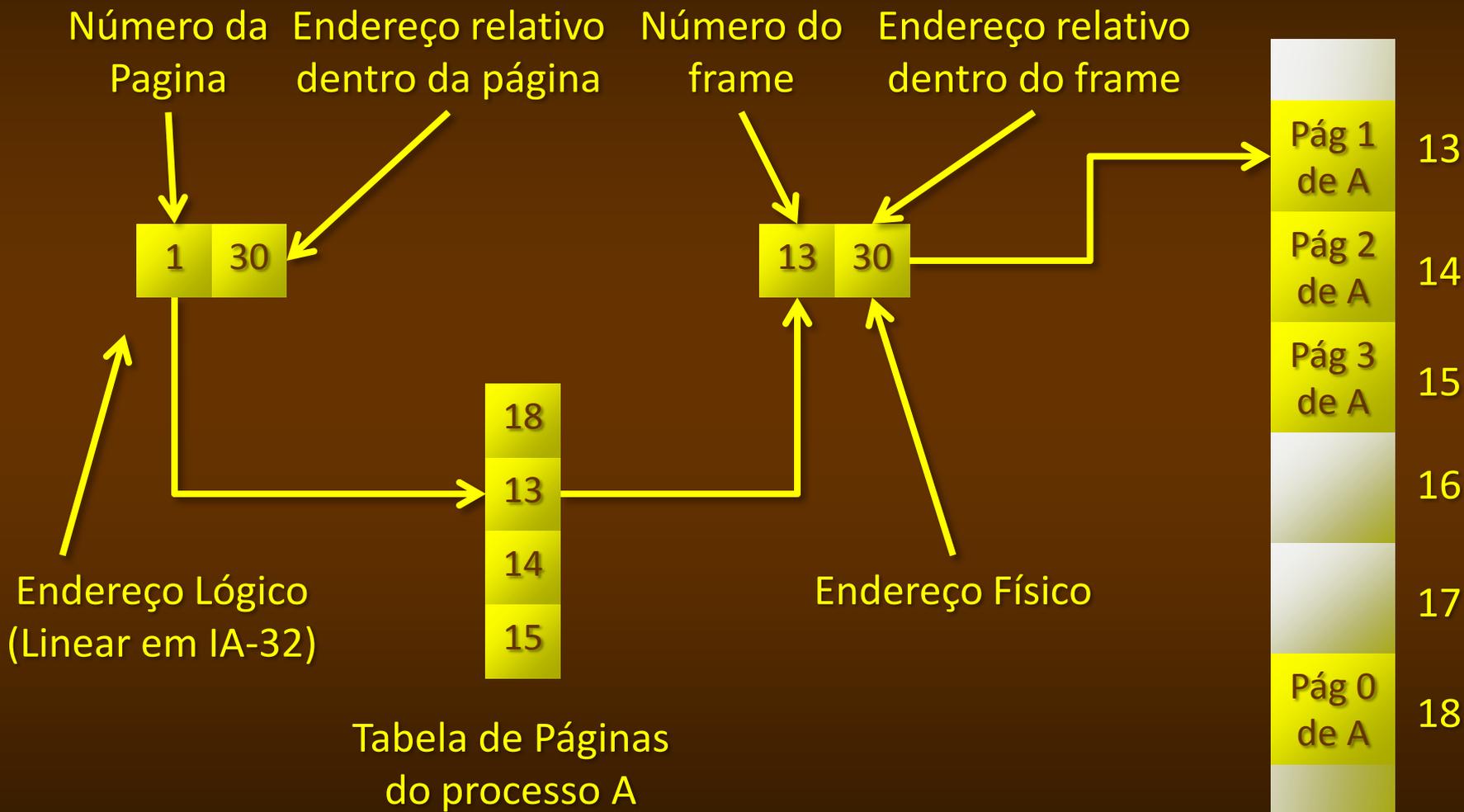
Tabela de Páginas do processo A

18  
13  
14  
15



depois

# ENDEREÇOS LÓGICOS E FÍSICOS – PAGINAÇÃO



# MEMÓRIA VIRTUAL

Paginação por demanda:

- Não é necessário ter todas as páginas de um processo simultaneamente na memória
- As páginas são trazidas para a memória pelo S.O. (possivelmente auxiliado pelo hw) conforme a necessidade.

Quando a página não está na memória (falta de página):

- S.O. deve carregar a página requisitada.
- Pode ser necessário retirar alguma página para abrir espaço.
- A página a ser retirada é selecionada com base na história recente: foi acessada? Foi modificada?

## “TRASHING”

Quando a memória é pouca e os processos muitos, pode ocorrer o fenômeno de “*trashing*”:

- ❑ O S.O. gasta todo seu tempo trocando páginas.
- ❑ Pouco, ou nenhum trabalho útil é realizado.
- ❑ A luz do HD fica acesa constantemente.

Possíveis soluções para esse problema:

- ❑ Bons algoritmos de substituição de página.
- ❑ Redução do número de processos em execução.
- ❑ Aumentar a quantidade de memória.

## ALGUMAS VANTAGENS DA PAGINAÇÃO

- ❑ Não há necessidade de se carregar um processo inteiro na memória.
- ❑ As páginas podem ser substituídas/atualizadas conforme a necessidade
- ❑ É possível executar processos muito maiores que a memória física total disponível.
- ❑ O usuário/programador enxerga sempre uma memória contígua e muito maior que aquela disponível – a chamada *memória virtual*.
- ❑ É possível acomodar processos que têm exigências específicas quanto à faixa e localização de endereços necessários. Por ex. os antigos programas DOS rodam apenas no primeiro 1 MB da memória.

# REFERÊNCIA BIBLIOGRÁFICA

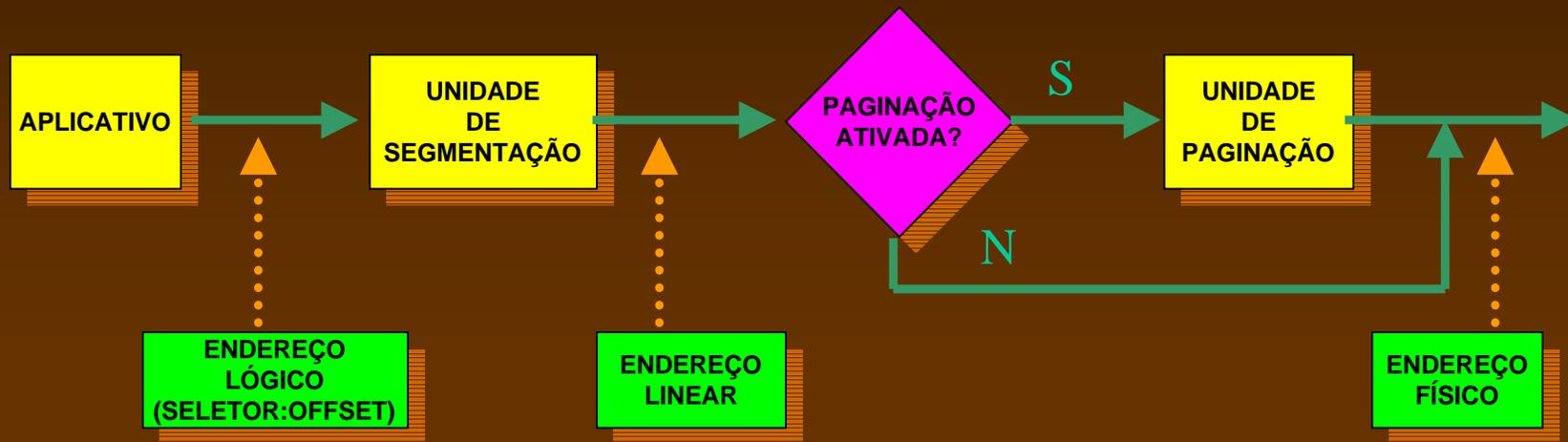
Este conjunto de transparências foi baseado no capítulo 8 do livro:

**William Stallings**  
**Arquitetura e Organização**  
**de Computadores**  
**8ª Edição**

**Capítulo 8**  
**Suporte do sistema**  
**operacional**



## ESQUEMA GERAL DE ENDEREÇAMENTO NA ARQUITETURA IA-32



### COMPOSIÇÃO DO ENDEREÇO LINEAR COM PAGINAÇÃO DE 4KB ATIVADA:



## COMPARAÇÃO ENTRE ALTERNATIVAS PARA TABELAS DE PAGINAÇÃO

COMPARE AS 3 POSSIBILIDADES SEGUINTE DE ORGANIZAÇÃO DAS TABELAS DE PAGINAÇÃO QUANTO À:

- A) CONSUMO DE MEMÓRIA PARA ARMAZENAMENTO DAS TABELAS.
- B) VELOCIDADE DE ACESSO À MEMÓRIA FÍSICA.

**MÉTODO I)** - BUSCA SEQÜENCIAL EM TABELA GRANDE - CADA POSIÇÃO DA TABELA CONTÉM:

- 20 BITS DO ENDEREÇO DE ORIGEM (LINEAR)
- 20 BITS DO ENDEREÇO TRADUZIDO (FÍSICO) E
- 1 BIT DE PRESENÇA

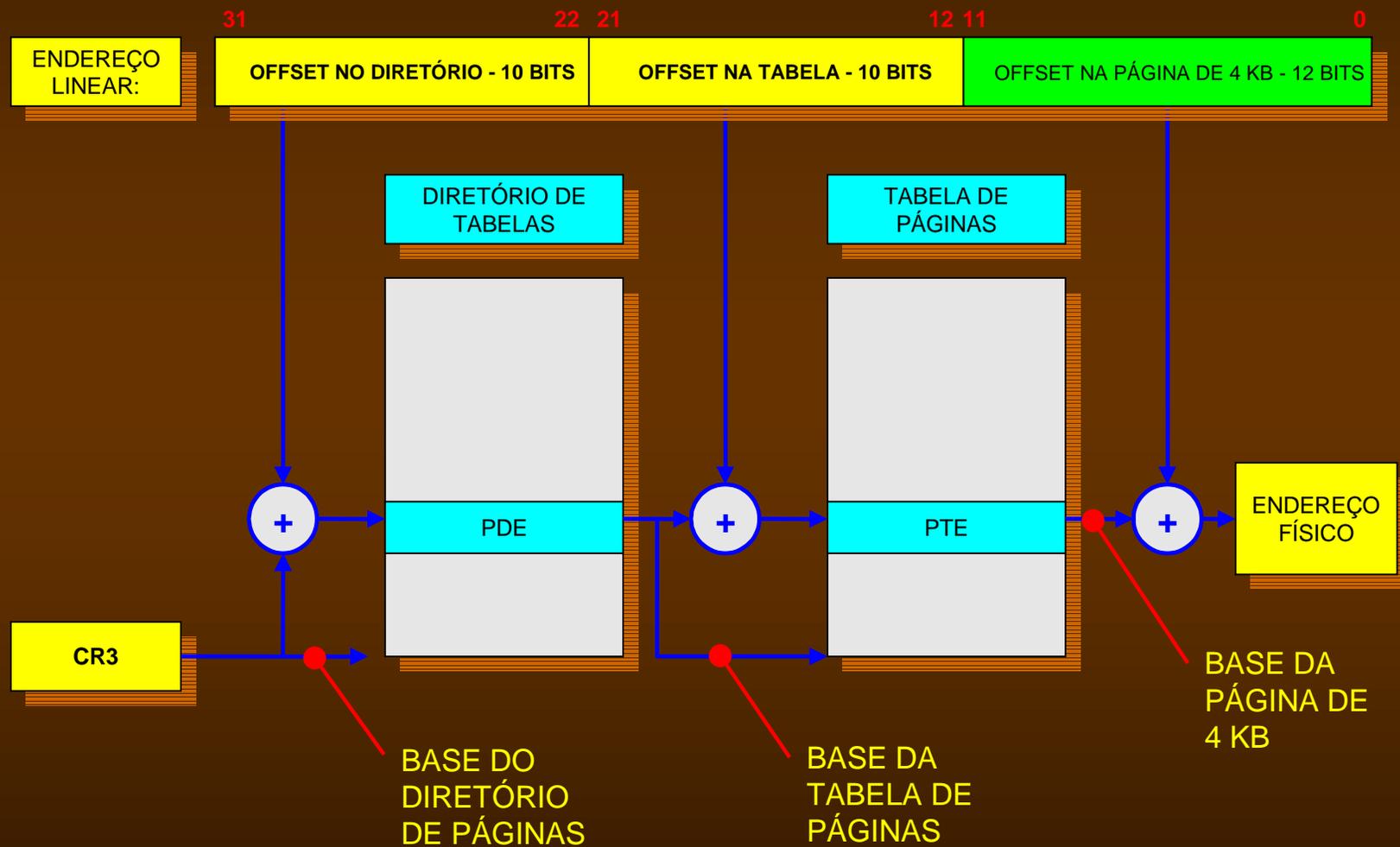
**MÉTODO II)** - BUSCA INDEXADA EM TABELA GRANDE

- OS 20 BITS DO ENDEREÇO DE ORIGEM (LINEAR) APONTAM PARA 1 MEGA POSIÇÕES DE UMA TABELA ONDE CADA POSIÇÃO CONTÉM:
- 20 BITS DO ENDEREÇO TRADUZIDO (FÍSICO) E
- 1 BIT DE PRESENÇA

**MÉTODO III)**

- O ESPAÇO DE ENDEREÇAMENTO DE 1 MEGA PÁGINAS DE 4 KB CADA É DIVIDIDO EM 1024 (1 K) GRUPOS DE TABELAS, SENDO QUE CADA GRUPO CONTÉM 1024 (1 K) TABELAS.
- USAR UMA TABELA MESTRE (DIRETÓRIO DE TABELAS) COM ATÉ 1024 POSIÇÕES (NEM TODAS PRECISAM EXISTIR) COM CADA POSIÇÃO CONTENDO OS 20 BITS MAIS SIGNIFICATIVOS DO ENDEREÇO BASE DE UMA TABELA E
- 1 BIT DE PRESENÇA
- CONTEÚDO DAS DEMAIS TABELAS (APONTADAS PELAS ENTRADAS NO DIRETÓRIO) COMO NO MÉTODO II)

## DETERMINAÇÃO DO ENDEREÇO FÍSICO A PARTIR DO LINEAR (PÁGINAS DE 4 KB)



## CONTEÚDO DAS PTE'S E PDE'S

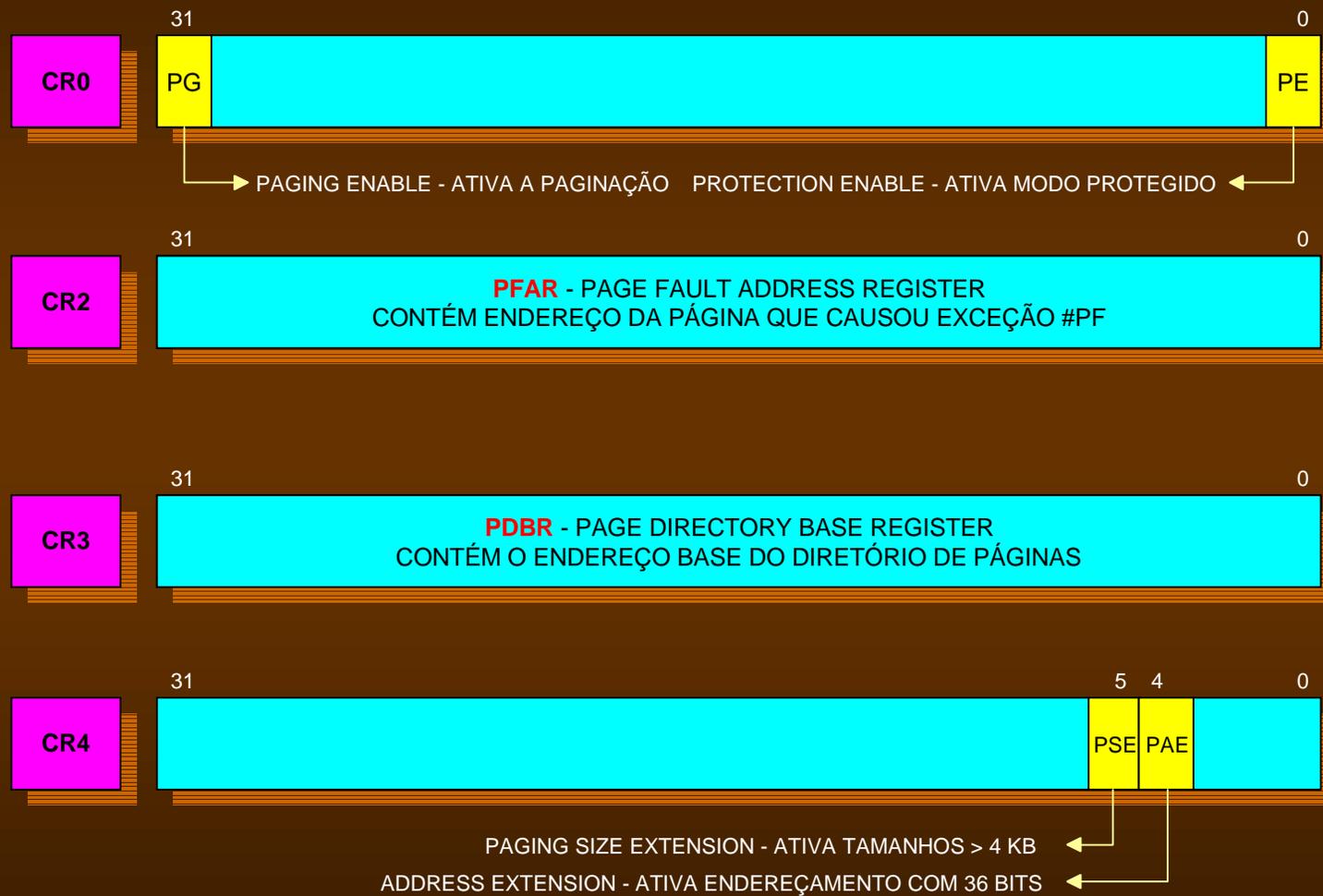
QUANDO A TABELA OU PÁGINA ESTÃO PRESENTES NA MEMÓRIA (P=1):



QUANDO A TABELA OU PÁGINA NÃO ESTÁ PRESENTE NA MEMÓRIA: (P=0)



## REGISTRADORES ENVOLVIDOS NO CONTROLE DA PAGINAÇÃO



## REPAGINAÇÃO DA MEMÓRIA DE UM PC PARA UTILIZAÇÃO DE UMB'S

### EXERCÍCIO

**OBJETIVO:** REPAGINAR 16 MB DE MEMÓRIA DA SEGUINTE FORMA:

- ÁREA ENTRE C8000 h e EFFFF h DEVE SER REALOCADA PARA 110 000 h e \_\_\_\_\_ (DETERMINAR)
- DEMAIS POSIÇÕES DE MEMÓRIA MANTÊM SEU ENDEREÇO **LINEAR = ENDEREÇO FÍSICO**, EXCETO AS POSIÇÕES 110 000 h - \_\_\_\_\_ QUE DEVERÃO APONTAR PARA C8000 h - EFFFF h.
- BASE DO DIRETÓRIO DE PÁGINAS DEVERÁ SER 2000 h (FÍSICO)
- TABELAS DE PÁGINAS DEVERÃO SER POSICIONADAS A PARTIR DE 3000 h (FÍSICO)

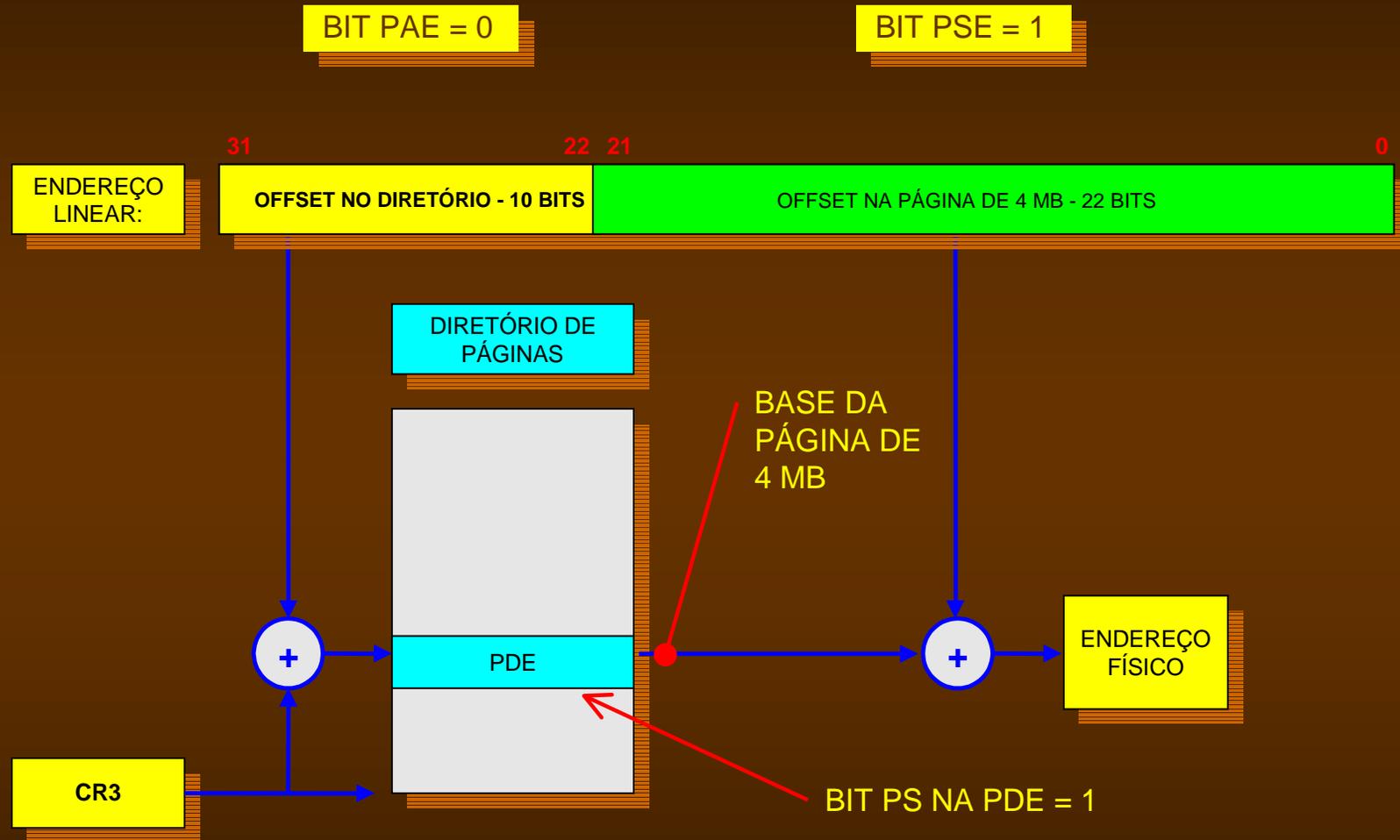
**DETERMINAR:**

- A QUANTIDADE DE PÁGINAS QUE SERÃO REPAGINADAS
- A QUANTIDADE DE TABELAS NECESSÁRIAS PARA REPAGINAR OS 16 MB
- OS ENDEREÇOS FÍSICOS INICIAL E FINAL DE CADA TABELA E O NÚMERO DE POSIÇÕES DE CADA UMA
- O CONTEÚDO DE CADA UMA DAS ENTRADAS DE CADA TABELA (ESQUEMA GERAL).

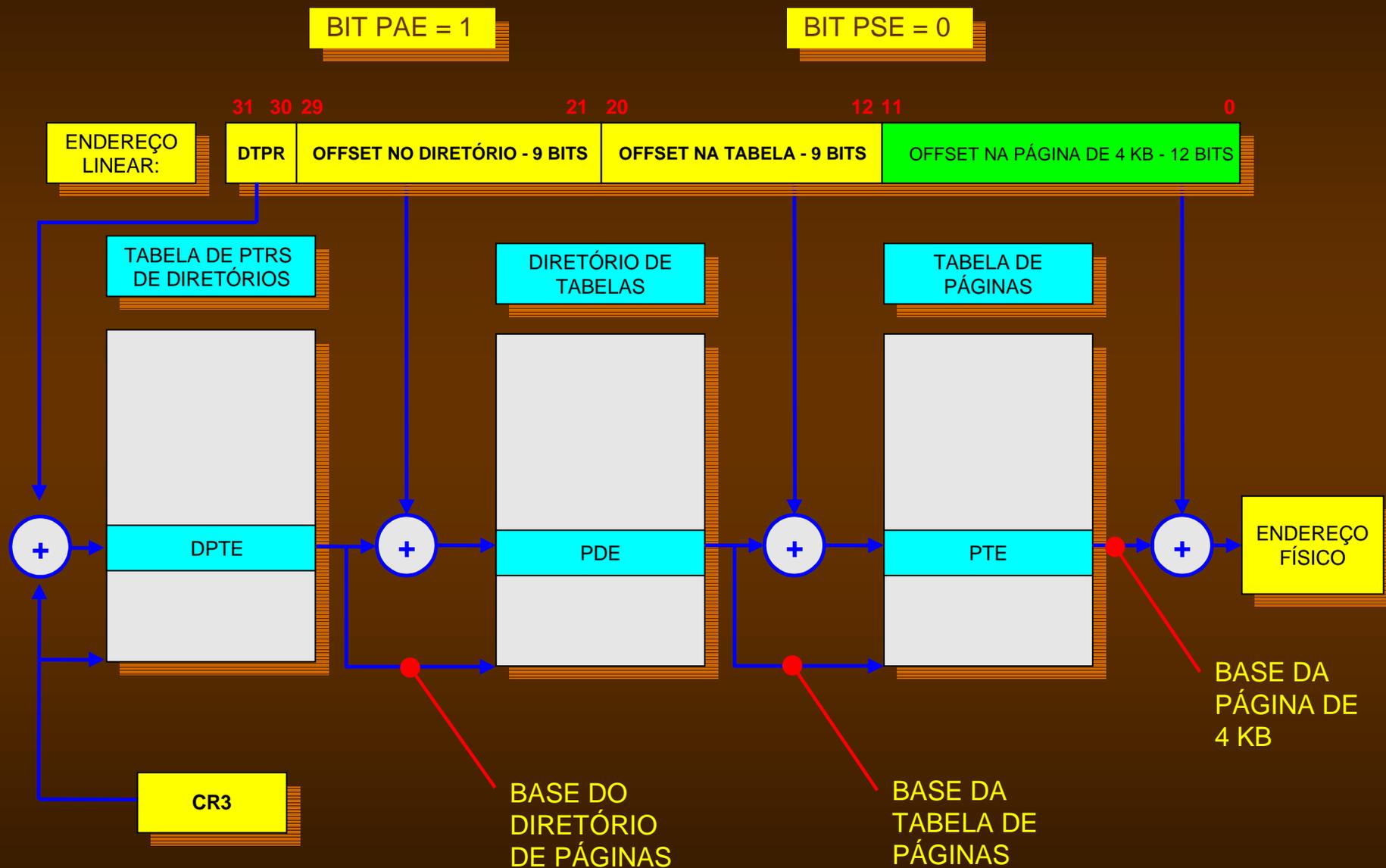
## ESQUEMAS ALTERNATIVOS PÓS-386 (TAMANHOS DE PÁGINAS E ENDEREÇAMENTO)

FLAG PG EM CR0	FLAG PAE EM CR4	FLAG PSE EM CR4	FLAG PS NA PDE	FLAG PSE-36 (CPUID)	TAMANHO DA PÁGINA	LARGURA DO ENDEREÇO FÍSICO EM BITS
0	X	X	X	X	-	PAGINAÇÃO DESATIVADA
1	0	0	X	X	4 KB	32 BITS
1	0	1	0	X	4 KB	32 BITS
1	0	1	1	0	4 MB	32 BITS
1	0	1	1	1	4 MB	36 BITS
1	1	X	0	X	4 KB	36 BITS
1	1	X	1	X	2 MB	36 BITS

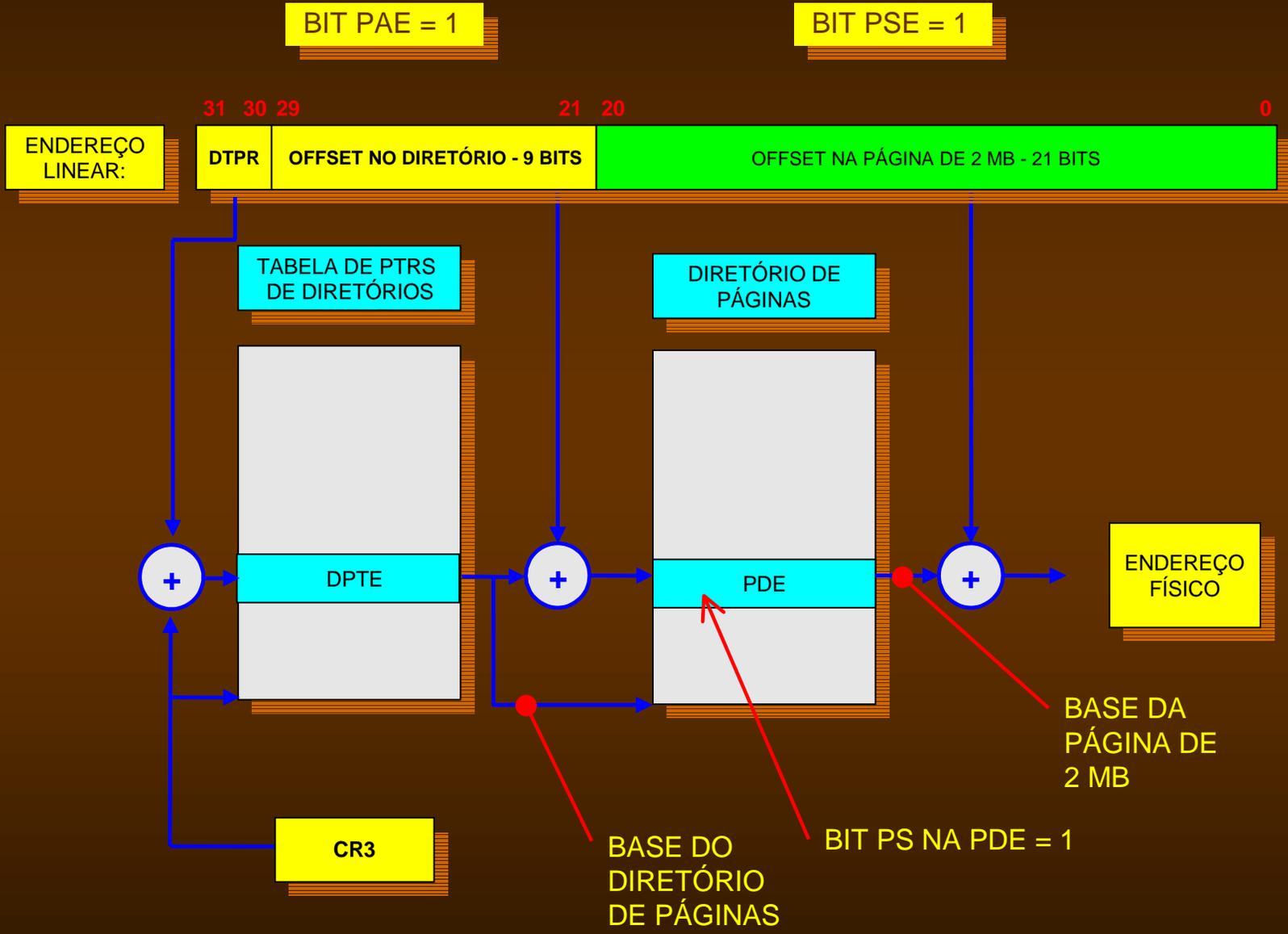
# DETERMINAÇÃO DO ENDEREÇO FÍSICO PARA PÁGINAS DE 4 MB E ENDEREÇAMENTO COM 32 BITS



# DETERMINAÇÃO DO ENDEREÇO FÍSICO PARA PÁGINAS DE 4 KB E ENDEREÇAMENTO COM 36 BITS



# DETERMINAÇÃO DO ENDEREÇO FÍSICO DE PÁGINAS DE 2 MB E ENDEREÇAMENTO COM 36 BITS



## TLB'S - TRANSLATION LOOKASIDE BUFFERS

**REALIZAR UM OU MAIS ACESSOS À MEMÓRIA EXTERNA DO COMPUTADOR PARA DETERMINAR O ENDEREÇO FÍSICO FINAL SEMPRE QUE A MEMÓRIA FOR ACESSADA É INEFICIENTE.**

**PARA EVITAR OS ATRASOS INCORRIDOS AO ACESSAR A MEMÓRIA EXTERNA A ARQUITETURA IA-32 PREVÊ UM PEQUENO BUFFER - INTERNO À CPU - ONDE SÃO ARMAZENADAS AS TRADUÇÕES MAIS RECENTES PARA REUTILIZAÇÃO.**

**O 386 E 486 MANTINHAM UM ÚNICO BUFFER (TLB) COMUM PARA DADOS E INSTRUÇÕES, CAPAZ DE ARMAZENAR AS 32 TRADUÇÕES MAIS RECENTES.**

**A LINHA PENTIUM PREVÊ TLB'S SEPARADOS PARA DADOS E INSTRUÇÕES ASSIM COMO PARA PÁGINAS DE CAPACIDADE DIFERENTE. A QUANTIDADE DE TRADUÇÕES ARMAZENADAS DEPENDE DO MODELO DE PROCESSADOR E PODE SER CONSULTADA ATRAVÉS DA INSTRUÇÃO CPUID.**