

## Gestão de Memória

### Parte I – Mecanismos

## Espaço de Endereçamento

- Conjunto de posições de memória que um processo pode referenciar
  - Para ler, escrever ou executar
- E se referenciar outras posições de memória?
  - HW de gestão de memória desencadeia exceção
  - Tratada pelo SO (tipicamente termina processo)

## Hierarquia de Memória

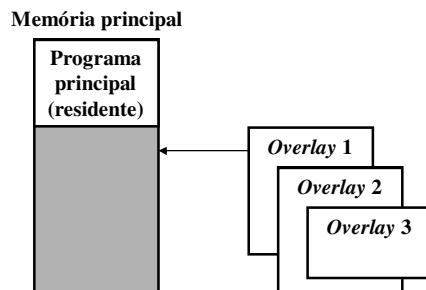
- Memória principal (física ou primária):
  - acesso aleatório
  - tempo de acesso reduzido
  - custo elevado → reduzida dimensão
  - informação volátil
  - RAM + caches [ + registos ]
- Memórias secundárias (ou de disco):
  - acesso aleatório (por blocos)
  - tempo de acesso elevado
  - custo reduzido → mais abundante
  - informação persistente

## Endereços Reais vs. Virtuais

- Endereçamento Real
  - Endereço indicado no programa é aquele que é acedido na memória principal
  - Usados pelos sistemas iniciais monoprogramados
  - Limitações?
    - Dimensão dos programas?
    - Multiprogramação?
    - Capacidade de ser executado em diferentes máquinas?

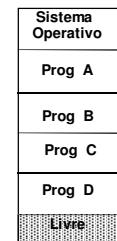
## Mecanismo de Sobreposição (Overlay)

- Possibilidade de executar programas com dimensão superior à memória principal em sistemas com endereçamento real
- Os overlays continuam a ter de ser dimensionados para a memória física disponível.
- Programador indica explicitamente quando deve ser carregado um overlay

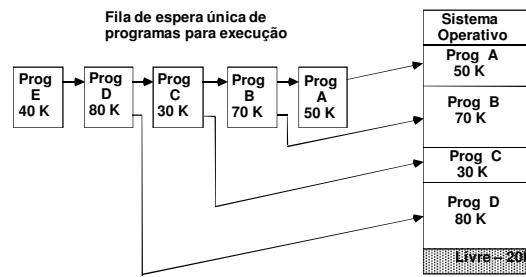


## Sistemas Multiprogramados com Endereçamento Real – Partições Fixas

- Memória dividida em partições de dimensão fixa
- Em cada partição é carregado um programa
- Programas recolocáveis
  - Endereçamento baseado
  - Registo base + deslocamento
- Vários programas em execução implica mecanismos de protecção
  - Registo limite → validado em cada acesso
- Fragmentação interna: dimensão dos programas não coincide exactamente com a dimensão das partições



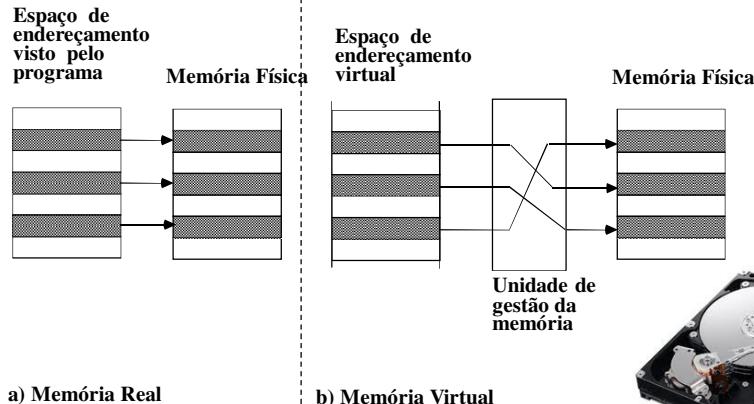
## Sistemas Multiprogramados com Endereçamento Real – Partições Variáveis



- Fragmentação externa: muitas partições de dimensão muito reduzida
- Quando um programa termina, a sua partição pode ser associada a outra (se for contígua), criando uma única de maior dimensão
- Implica a recompactação da memória → processamento suspenso

Como combinar todas estas técnicas numa única solução?

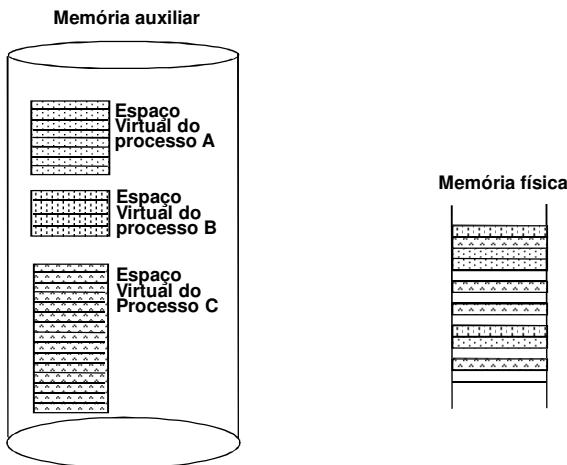
## Endereços Reais vs. Virtuais



## Endereçamento Virtual

- Espaço de endereçamento dos processos não linearmente relacionado com a memória física
- Endereços virtuais são sempre convertidos (pela UGM) para endereços reais
- Para minimizar a informação necessária à conversão, a memória virtual é logicamente dividida em blocos contíguos:
  - Endereço virtual = (bloco, deslocamento)
- **Dois tipos de blocos:**
  - Segmentos - dimensão variável.
  - Páginas - dimensão constante.
- Alguns blocos podem não residir em memória principal

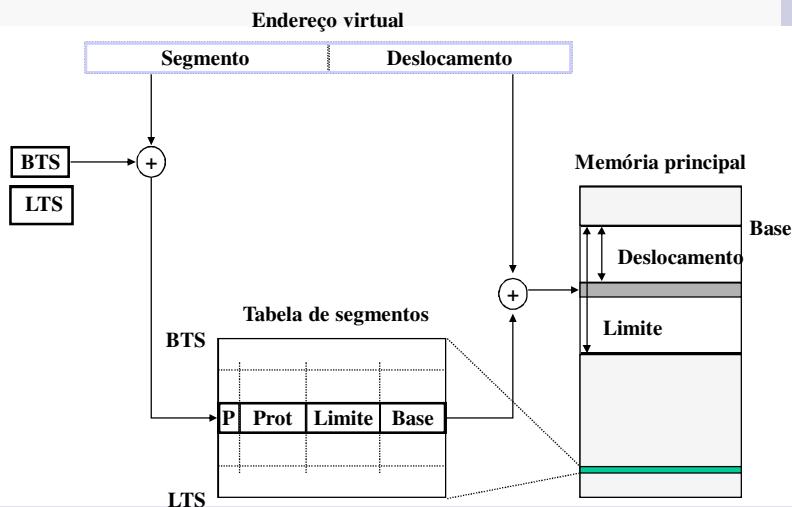
## Espaço de Endereçamento Virtual



## Segmentação

- Divisão dos programas em segmentos lógicos que reflectem a sua estrutura funcional:
  - rotinas, módulos, código, dados, pilha, etc.
  - a conversão de endereços virtuais é linear em cada segmento
  - o programador pode ter que se preocupar com a gestão de memória quando escreve um programa
- Gestão de memória suporta abstracções das linguagens de programação → Segmento é a unidade de:
  - carregamento em memória (eficiência)
  - protecção
- Dimensão dos segmentos é limitada pela arquitectura e não pode exceder a dimensão da memória principal

## Tradução de Endereços Virtuais em Memória Segmentada



## Memória Virtual Segmentada

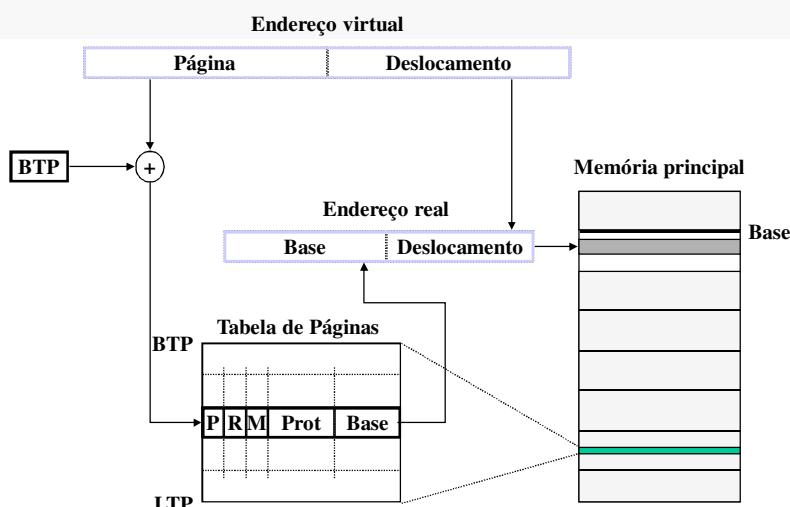
- Fragmentação: externa ou interna?
- Protecção:
  - verificação de limites de endereçamento intra-segments
  - verificação e limitação dos tipos de acesso ao segmento: leitura, escrita e execução
  - processos diferentes têm tabelas de segmentos diferentes: espaços de endereçamento disjuntos e inacessíveis a terceiros
- Partilha de memória entre processos:
  - basta colocar nas tabelas de segmentos dos processos em questão o endereço real do segmento a partilhar
  - os endereços virtuais usados para aceder ao segmento partilhado podem ser diferentes nos vários processos
  - a protecção dum segmento partilhado é definida para cada processo através da respectiva tabela de segmentos

## Paginação

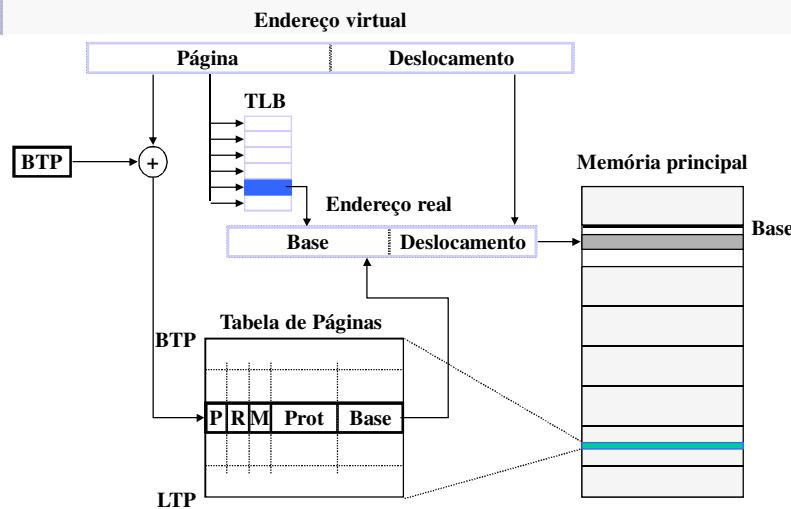
- Espaço de endereçamento virtual de dimensão superior à da memória principal:
  - o programador não “vê” a gestão de memória
- Na memória principal mantêm-se algumas páginas
  - Restantes carregadas de memória secundária quando necessário (falta de página)
- As instruções do processador têm de ser recomeçáveis
- A dimensão das páginas (constante) é normalmente muito menor que a da memória principal e influencia:
  - A fragmentação (externa ou interna?)
  - O número de faltas de páginas
  - Tempo da sua resolução (transferência)
  - A dimensão das tabelas de páginas e listas de páginas mantidas pelo sistema operativo

**Como?**

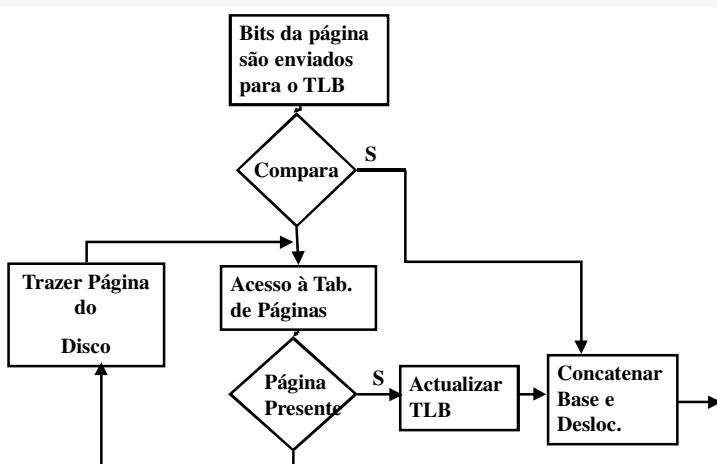
## Tradução de Endereços Virtuais em Memória Paginada



## Tabela de Tradução de Endereços (Translation Lookaside Buffer, TLB)



## Memória Virtual Paginada



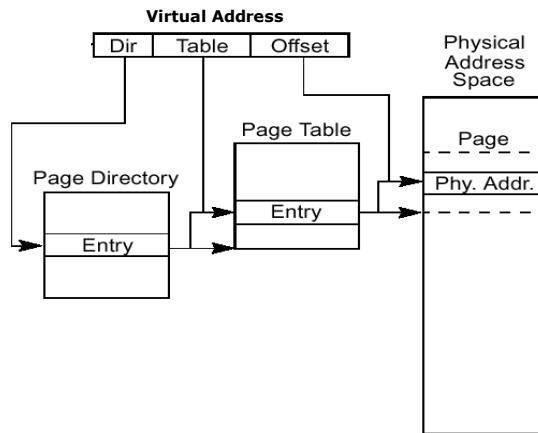
## Memória Virtual Paginada (cont.)

- **Protecção:**
  - Verificação dos tipos de acesso: leitura, escrita e execução.
  - Processos diferentes têm tabelas de páginas diferentes: espaços de endereçamento disjuntos e inacessíveis a terceiros
- **Partilha de memória entre processos:**
  - Semelhante ao usado para memória segmentada
  - Partilha de blocos lógicos: partilha de múltiplas páginas
  - Não é possível partilhar menos que uma página (versus arquitectura segmentada → partilha de uma divisão lógica do programa)

## Memória Virtual Paginada – Problema

- Qual a dimensão da tabela de páginas com endereços virtuais de 32 bits e páginas de 4 kBytes?
- (Ainda pior se fossem endereços de 64 bits...)

## Solução 1: Tabelas de Páginas Multi-Nível



**Vantagem?**

## Memória Segmentada/Paginada

