Engenharia de Computação

Arquitetura de Sistemas Operacionais

Gerência de Memória

Prof. Anderson Luiz Fernandes Perez

Prof. Martín Vigil

Conteúdo

- Introdução
- Modelo Computacional
- Endereços Reais e Virtuais
- Algoritmos de Gerenciamento de Memória

Introdução

- O Sistema Operacional é o responsável pelo gerenciamento da memória, ou seja, seu uso e otimização.
- Um processo ocupa uma porção de memória denominado *espaço de endereçamento do processo*.
- Um espaço de endereçamento de um processo é o conjunto de posições de memória que um programa executado por este processo pode referenciar.

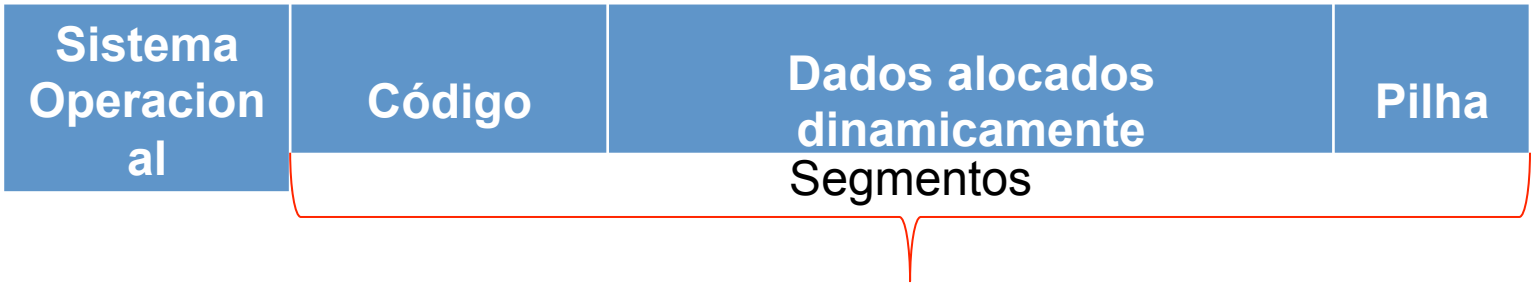
Introdução

- O espaço de endereçamento está associado ao processo, todas as informações acessadas ou guardadas neste espaço são acessíveis a partir do contexto do processo.
- O espaço de endereçamento organiza a memória de maneira a definir a área para uso do sistema operacional e a área para uso dos processos em execução.

Introdução

- Espaço de Endereçamento

- Espaço de Endereçamento Visto pelos Programadores



Introdução

- Existe uma clara noção de confinamento do processo ao seu espaço de endereçamento válido.
- O sistema operacional tem em cada instante um mapa preciso de quais posições de memória o programa pode acessar e de que forma.
- O confinamento garantido pelo SO é chamado de mecanismo de proteção de memória.

Modelo Computacional

- Os programas referenciam a memória para ler instruções e ler e escrever dados.
- Dados e instruções podem ter tamanho variável, desta forma uma operação de leitura de uma instrução ou a escrita de um dado transfere uma quantidade de informações da memória para a CPU.
- Os processadores estão organizados em palavras (múltiplos de bytes). Ex.: 4 bytes (32 bits), 8 bytes (32 bits).
- Os endereços de memória referenciam sempre bytes, indiferente da arquitetura do processador.

Modelo Computacional

- Um endereço de memória permite acessar um byte que conterá parte ou a totalidade do dado ou instrução que se quer acessar.

endereço -> valor

Ou

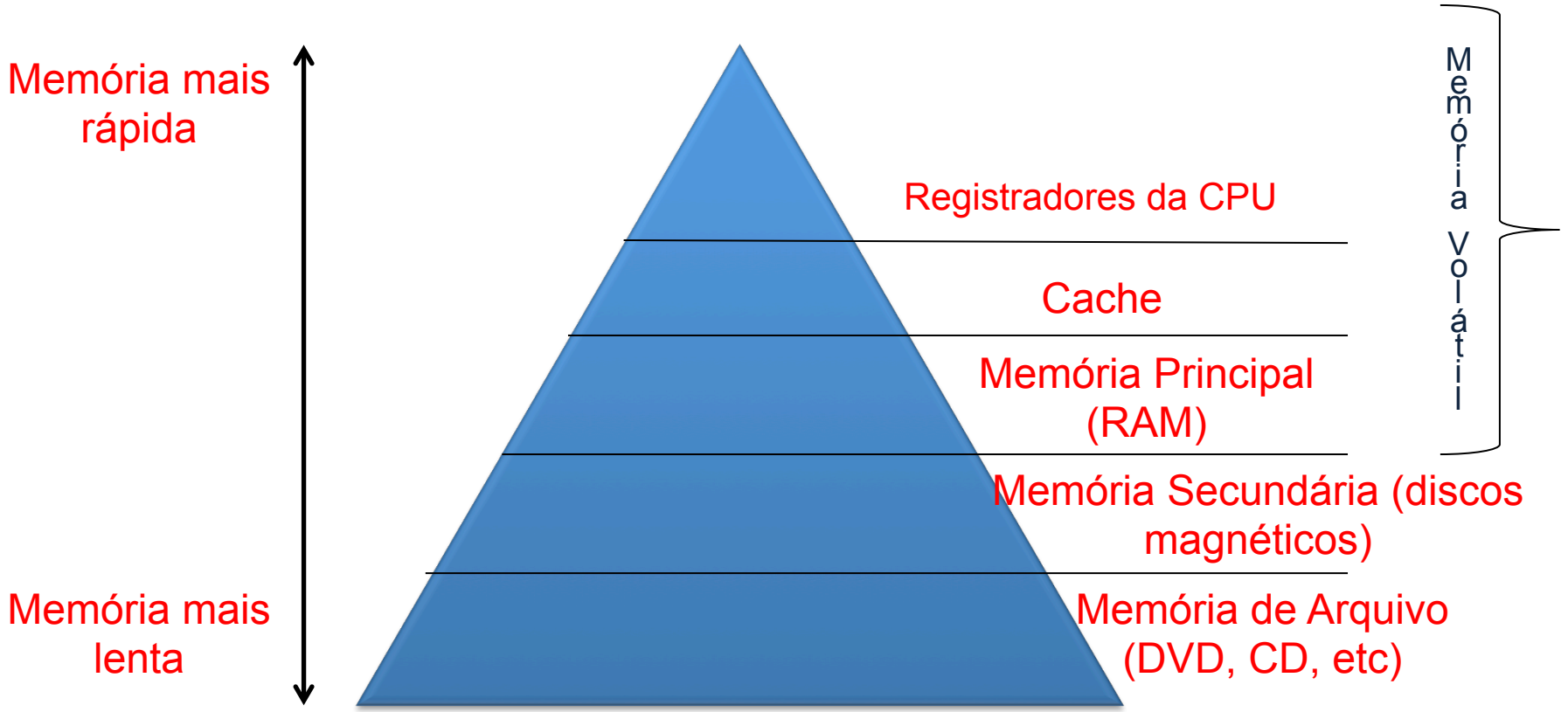
endereço virtual -> endereço real -> valor

Modelo Computacional

- Normalmente o sistema operacional disponibiliza um conjunto de chamadas de sistema para manipulação da memória.
- Chamadas de sistema para manipular memória:
  - Alocar (ex. malloc e calloc);
  - Liberar (ex. free);
  - Proteger;
  - Mapear;
  - Desmapear;
  - Associar;
  - Desassociar.

Modelo Computacional

- Hierarquia de Memória



Endereços Reais e Virtuais

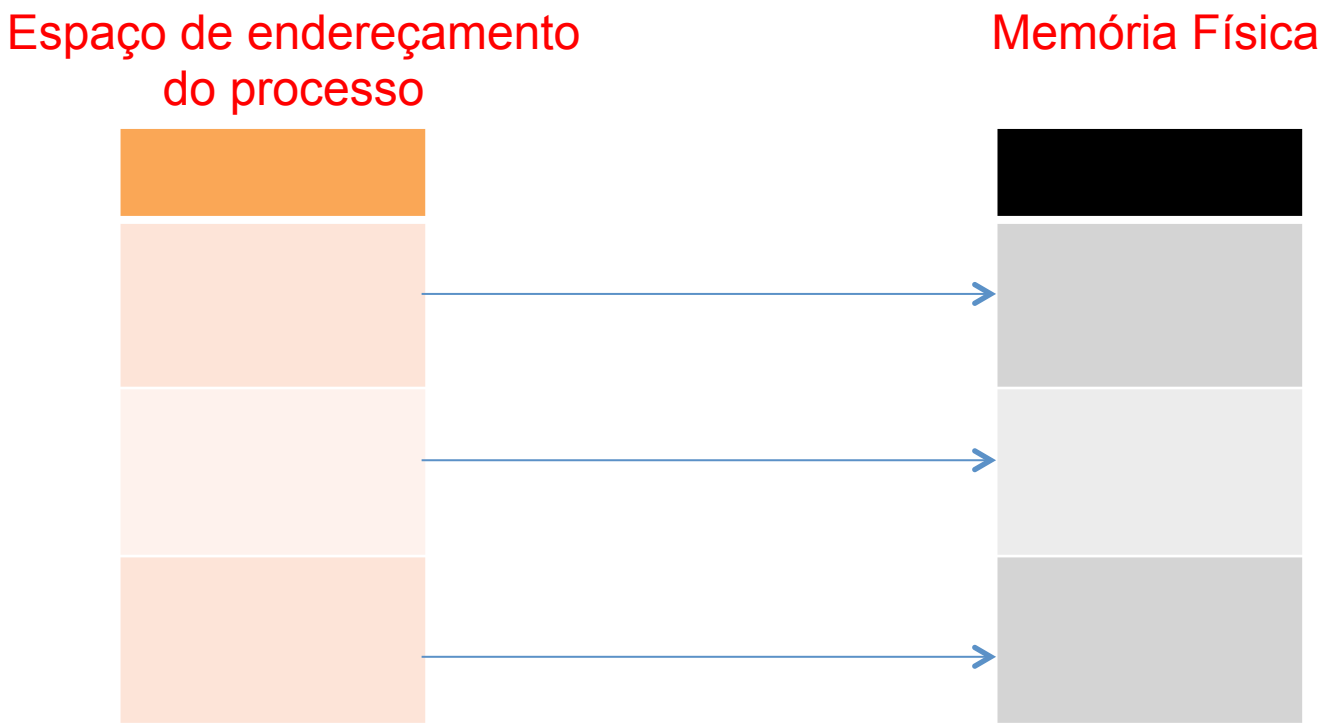
- Os computadores antigos suportavam apenas endereçamento real, ou seja, os endereços de memória acessado por um programa têm relação direta com os endereços de memória primária do computador.
- As desvantagens do endereçamento real são:
  - A dimensão do programa é limitada pela dimensão da **memória primária** do computador.
  - Um programa só pode funcionar para os **endereços físicos** para o qual foi escrito.
  - Dificultado em suportar a multiprogramação.

Endereços Reais e Virtuais

- Os endereços gerados pelo programa (virtuais) são convertidos pelo processador, em tempo de execução, em endereços físicos (reais).
- No sistema de endereçamento virtual a CPU é dotada de um hardware especial chamado MMA (*Memory Access Translation*).

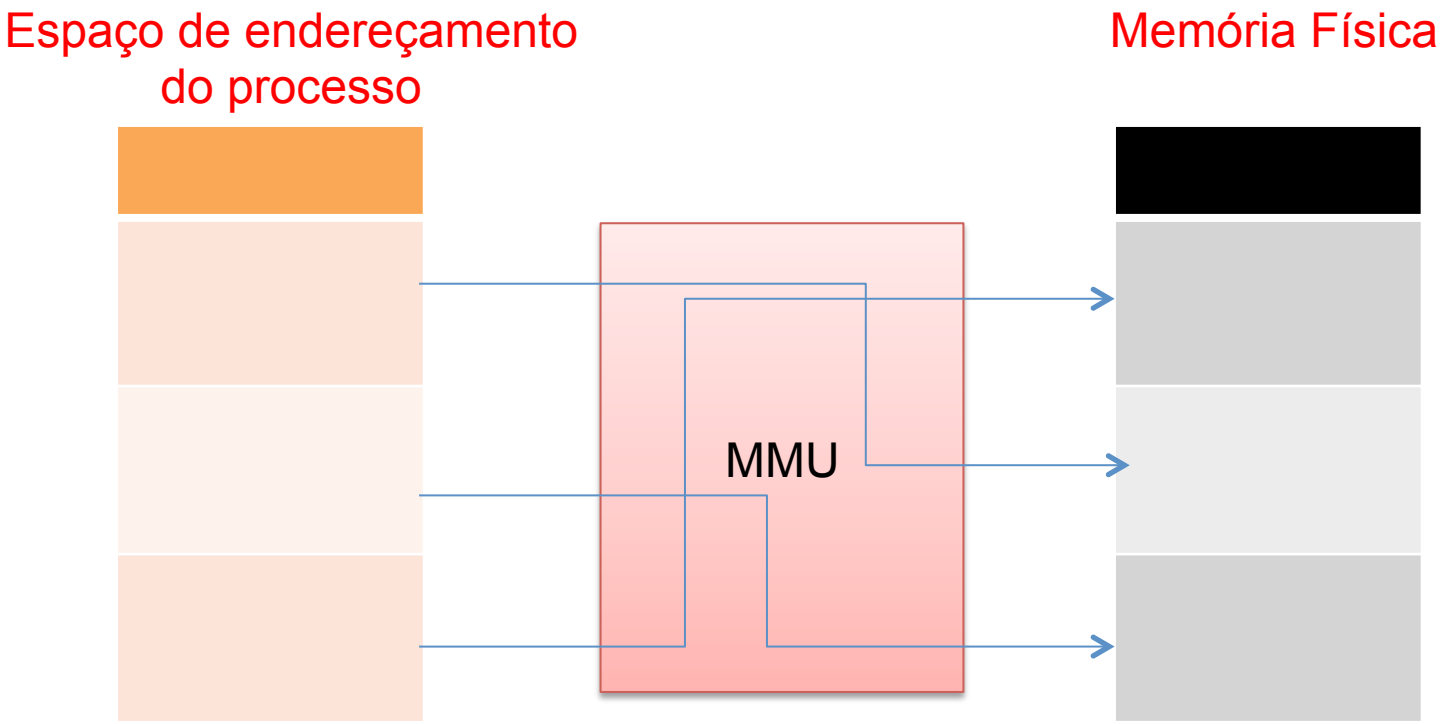
Endereços Reais e Virtuais

- Endereçamento Real (exemplo)



Endereços Reais e Virtuais

- Endereçamento Virtual (exemplo)



Endereços Reais e Virtuais

- Um endereço virtual não se refere nem a memória primária e nem a memória secundária, mas a um endereço lógico.
- O hardware de gerenciamento de memória (MMU) e o SO são responsáveis por traduzir/mapear o endereço virtual em endereço real na memória primária ou secundária.
- Caso o dado a ser acessado esteja em memória secundária, este é carregado na memória primária para ser acessado pelo processo.
- O uso de memória secundária permite alocar um programa que seja **maior** que a memória primária disponível.

Endereços Reais e Virtuais

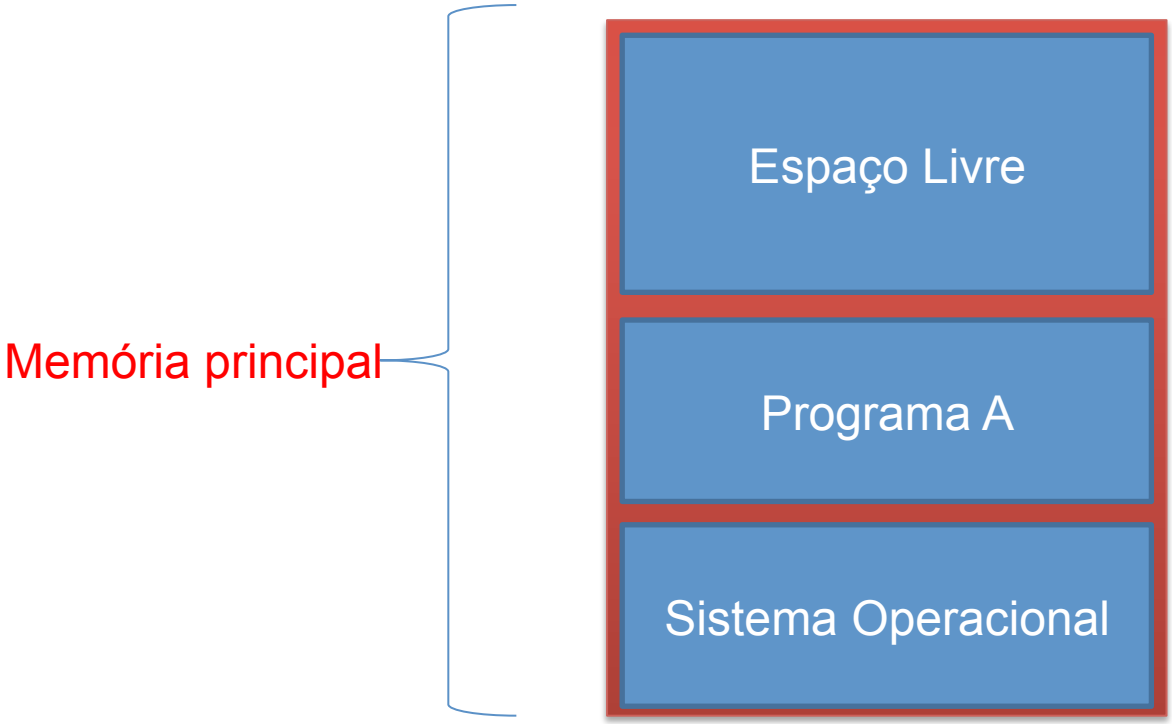
- Endereçamento Real
  - Apesar das desvantagens do modelo de endereçamento de memória real, este método é utilizado em alguns sistemas embarcados.
  - O endereçamento real pode ser aplicado em sistemas monoprogramados e sistemas multiprogramados.

Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Monoprogramados
    - O mapa de memória é basicamente composto de duas partes:
      - Uma ocupada pelo sistema operacional;
      - Outra ocupada pelo programa carregado na memória principal.
    - O problema de falta de espaço na memória principal para alocar um programa pode ser resolvido com o uso de overlays.
    - Um overlay é um rotina do programa que é carrega na memória somente quando for necessária.

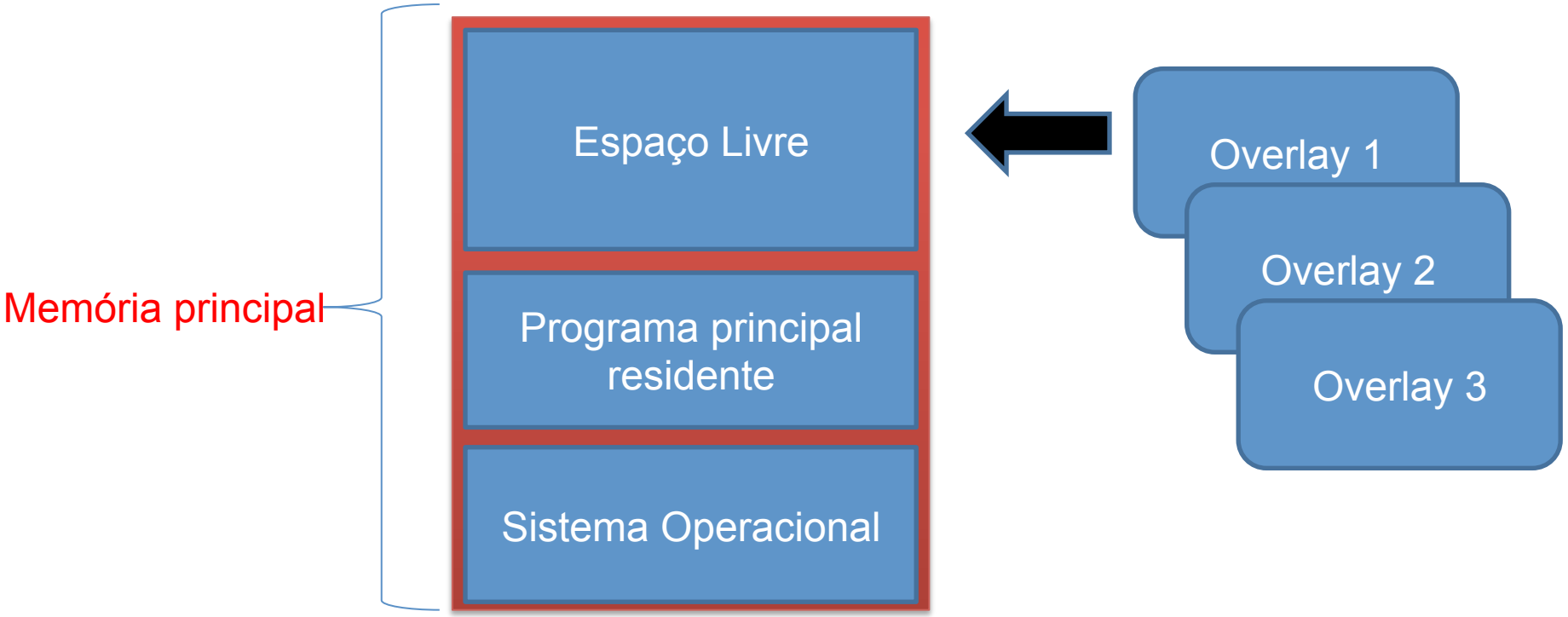
Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Monoprogramados (sem overlay)



Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Monoprogramados (com overlay)

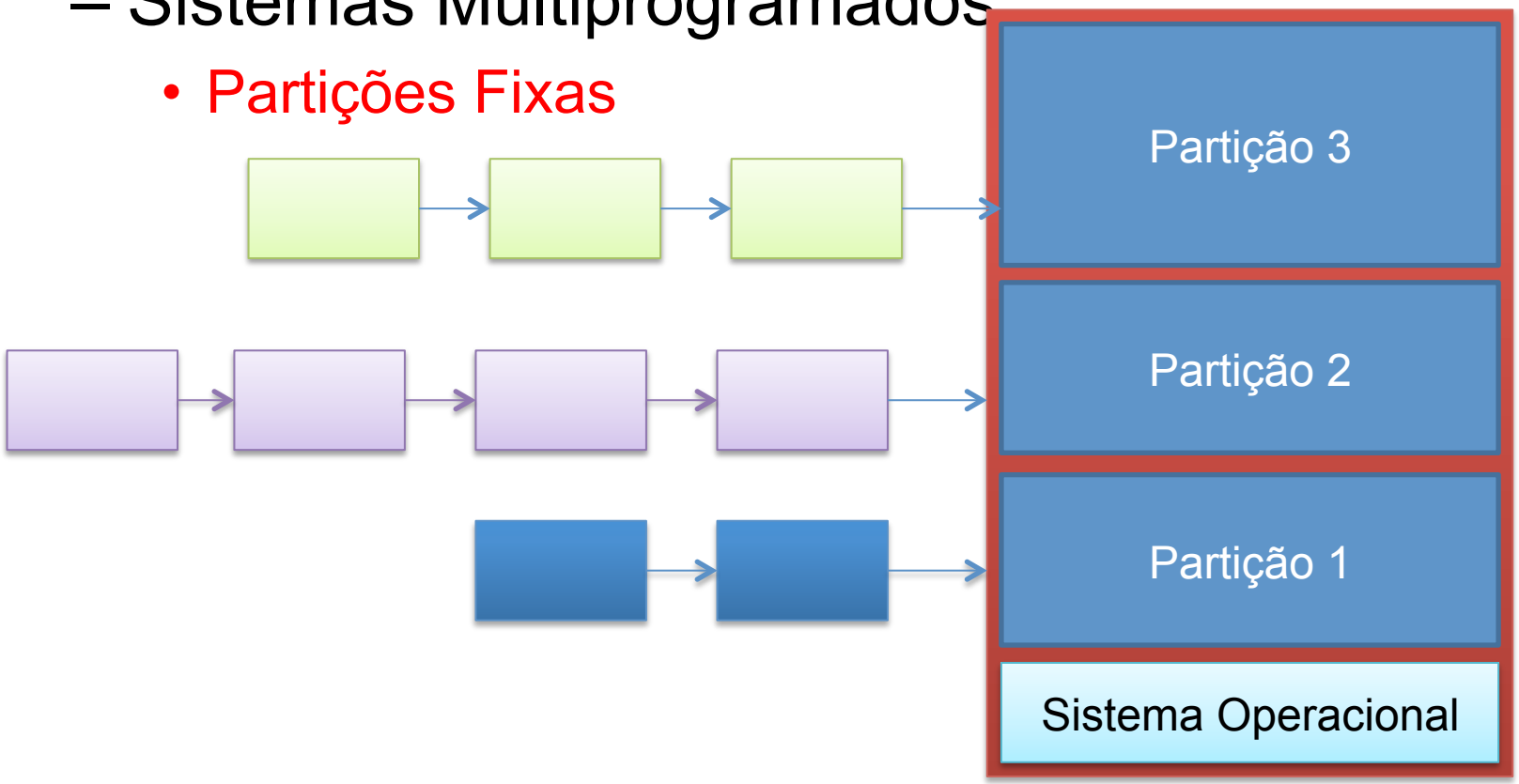


Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Os sistemas multiprogramados permitem que vários programas estejam alocados na memória principal.
    - Para suportar a multiprogramação a memória é dividida em partições que podem ser de tamanho fixo ou variável.
    - Quando um determinado programa for bloqueado em uma operação de entrada e saída de dados, outro programa, que está alocado em outra partição, é colocado em execução.
    - O grau de multiprogramação é dado pelo número de partições existentes no sistema.

Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Fixas



Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Fixas
      - Inicialmente a alocação de um programa a uma partição era feita diretamente, ou seja, os endereços gerados para os programas, correspondiam aos endereços reais da máquina.
      - Um forma de resolver o problema da geração de endereços foi a adoção da técnica de relocação.
      - Na relocação o compilador gera para todo o programa o endereço zero de memória, quando o programa for carregado em memória o SO decide em qual partição este será alocado, desta forma, adiciona um deslocamento ao endereço gerado pelo compilador.

Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Fixas
      - Uma outra opção para a alocação de programas era o uso de um registrador especial chamado de registrador base.
      - O registrador base era carregado com o endereço físico do início da partição.
      - O hardware somava o endereço do registrador base com o endereço do programa para obter o endereço real da alocação do programa.
      - Um outro registrador chamado de registrador de limite era utilizado para controlar o acesso a memória e impedir que programas acessassem partições de memória de outro programa.

Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Fixas pode gerar fragmentação interna.

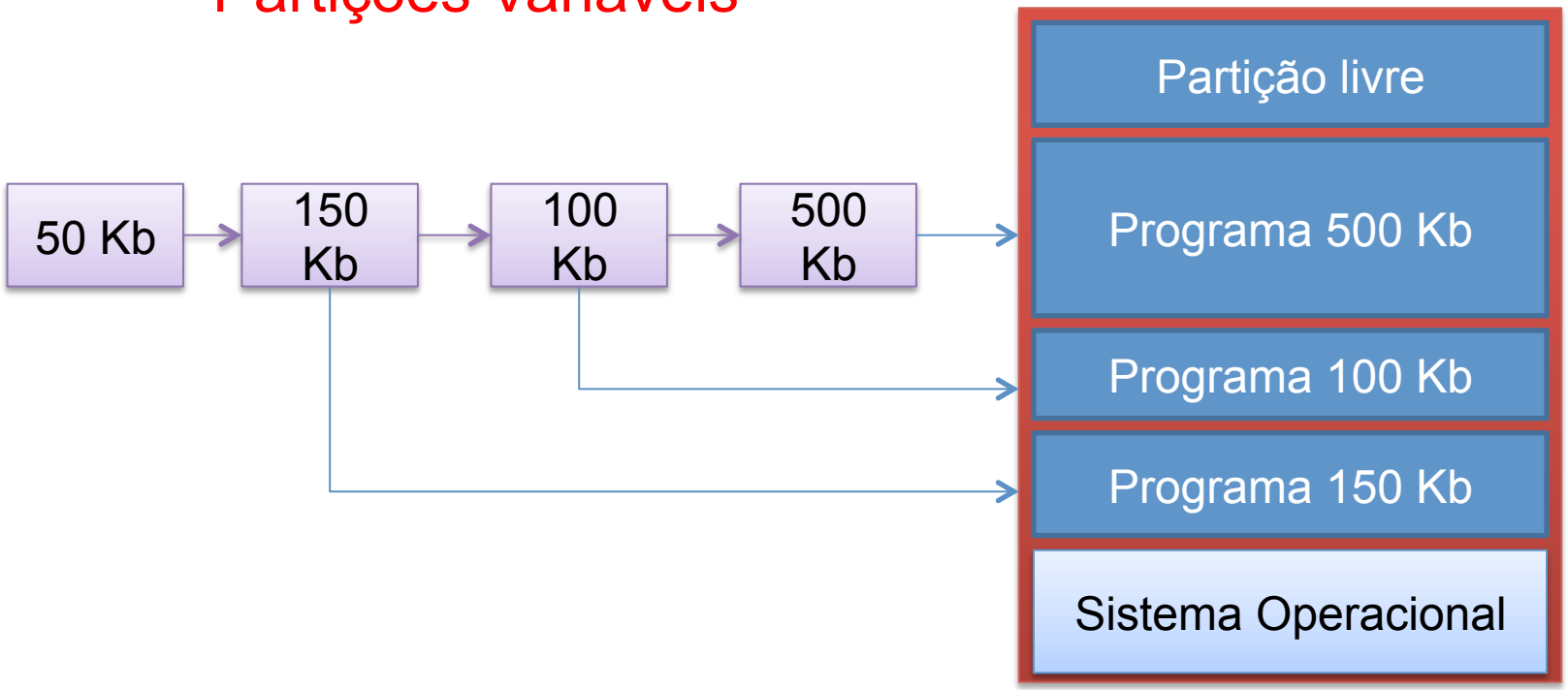
Partição

Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Variáveis
      - Na técnica de partições variáveis existe um espaço reservado para o SO e o restante da memória é considerado espaço livre.
      - A qualquer tempo, caso um processo demande memória, o espaço livre pode ser dividido e aproveitado para alocar programas.
      - Neste técnica um programa ocupa exatamente o quanto necessita da memória principal.

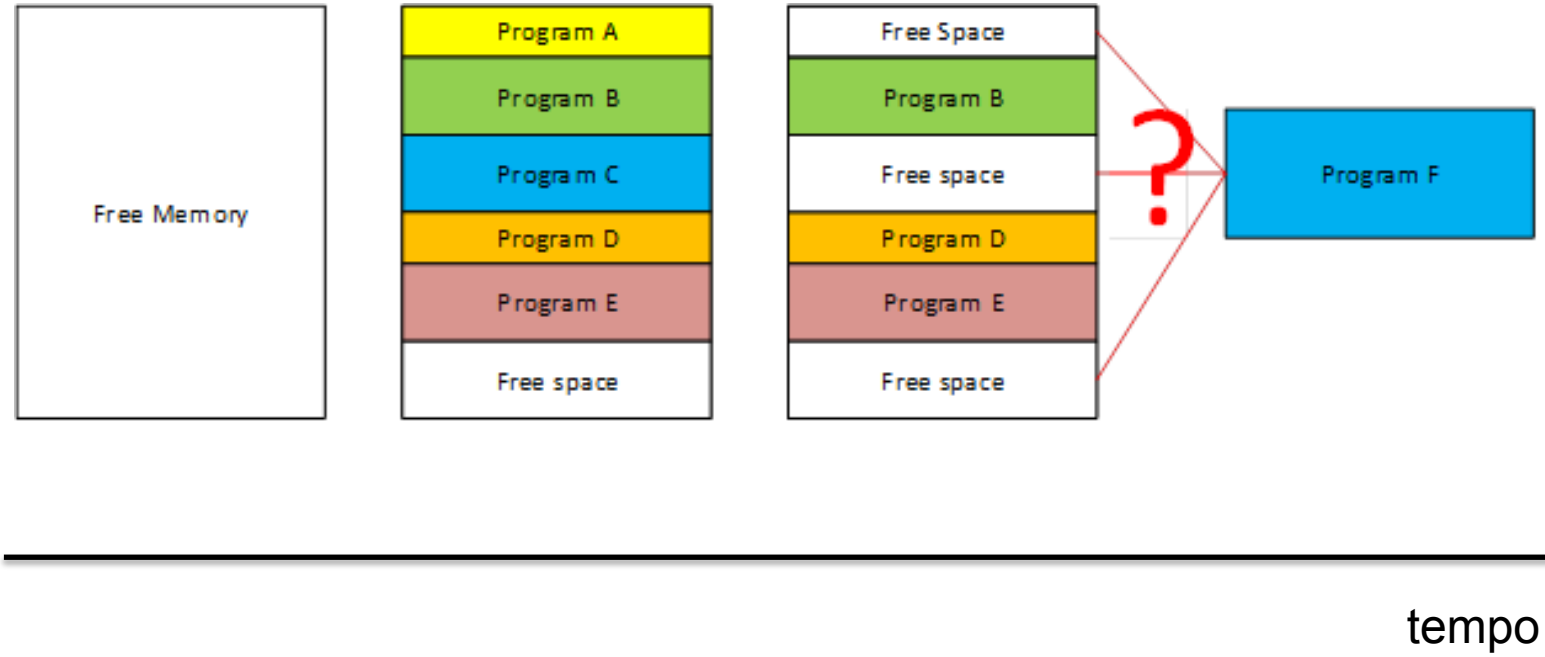
Endereços Reais e Virtuais

- Endereçamento Real
  - Sistemas Multiprogramados
    - Partições Variáveis



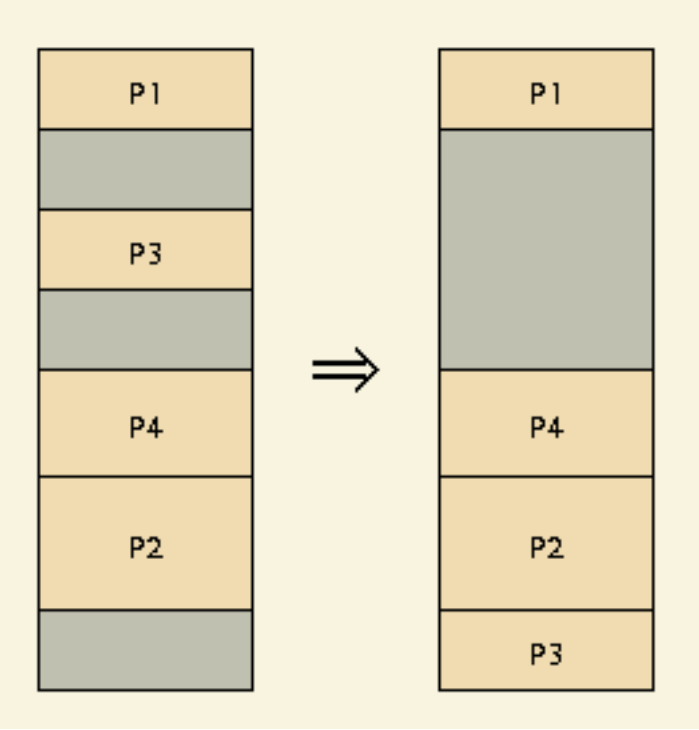
Endereços Reais e Virtuais

- Partições variáveis podem causar o problema de fragmentação externa



Endereços Reais e Virtuais

- O problema de fragmentação externa pode ser resolvido pela compactação de memória



Custo computacional significativo

Endereços Reais e Virtuais

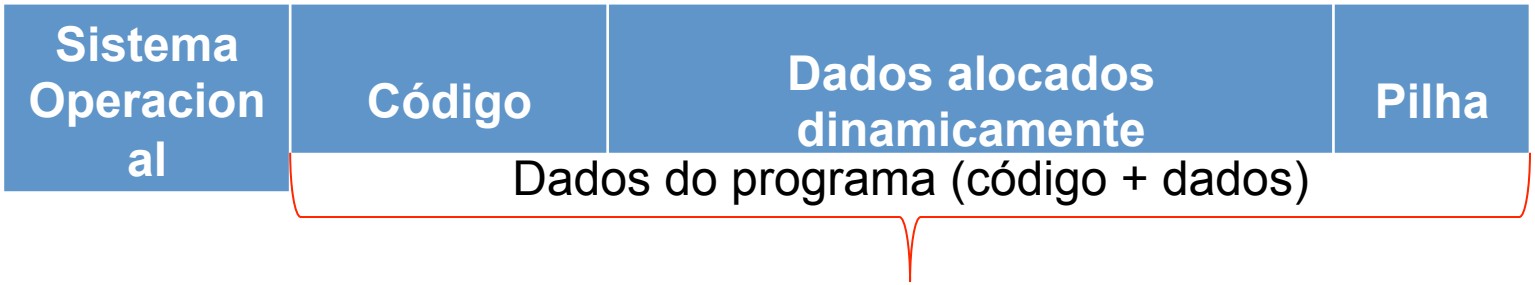
- Endereçamento Virtual
  - O sistema de endereçamento virtual desvincula os endereços gerados pelo programa dos endereços físicos acessados na memória principal.
  - O programador passa a ter uma espaço de endereçamento independente.
  - O espaço de endereçamento passa a ser maior do que o tamanho da memória principal.

Endereços Reais e Virtuais

- Endereçamento Virtual
  - A tradução de endereços virtuais em endereços reais é feito pelo tradutor de endereços (MMU – *Memory Management Unit*).
  - O mecanismo de tradução mantém uma tabela onde cada endereço virtual corresponde a um endereço real.
  - Para melhorar o desempenho e evitar que a tabela fique muito grande, ela indexa blocos, ou seja, o endereço de um bloco virtual corresponde a um endereço de um bloco real.
  - Um endereço passa a ter a forma bloco + deslocamento.

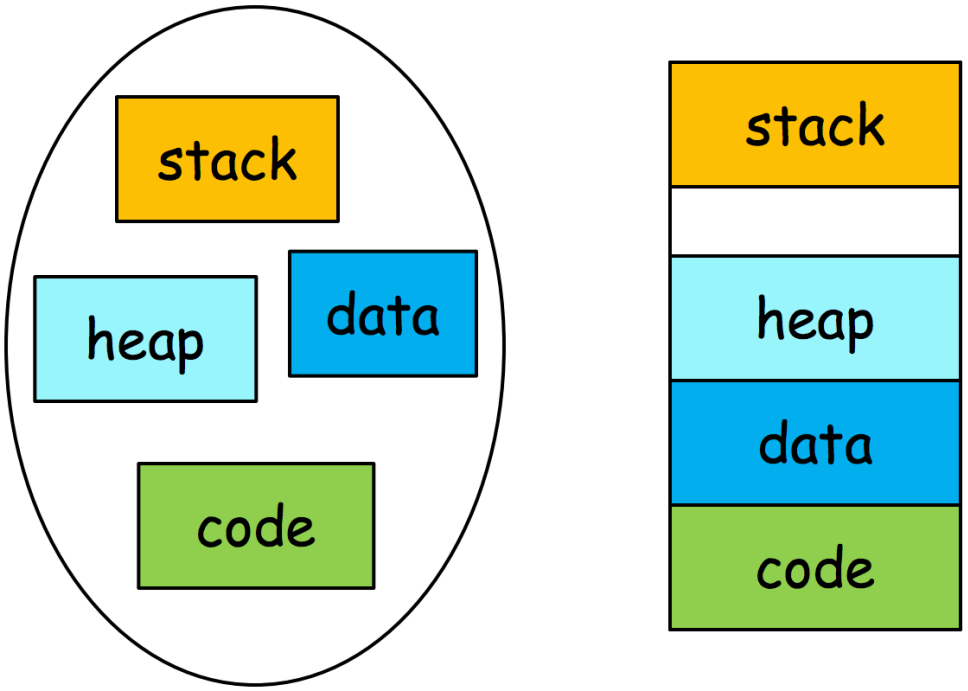
Endereços Reais e Virtuais

- Endereçamento Virtual
  - Visão da Memória pelo Programador



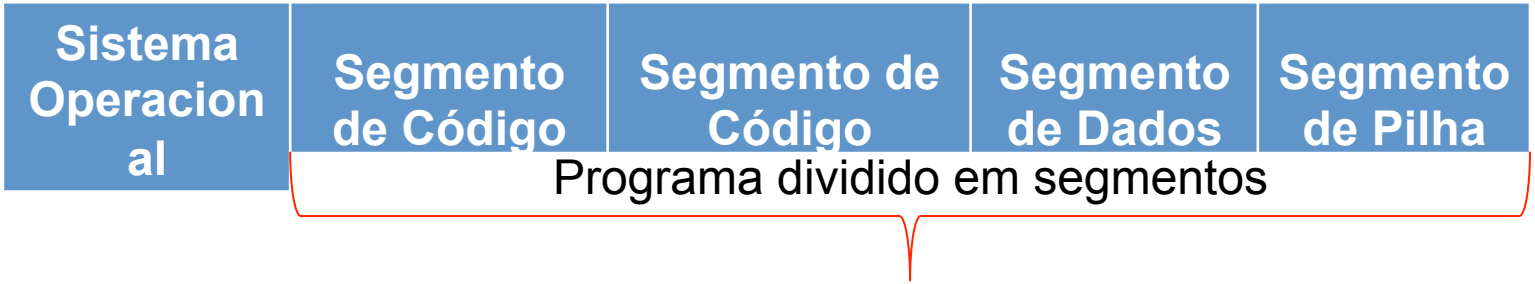
Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação



Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação



Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação
    - Os seguintes aspectos devem ser considerados na segmentação:
      - Carregamento em memória: o segmento é a unidade mínima a ser carregado em memória.
      - Proteção: impede um processo acessar dados do SO ou de outros processos
      - Eficiência: princípio da localidade, ou seja, ao ser acessado um endereço de um segmento existe uma grande probabilidade de que os próximos acessos sejam próximos a esse endereço.

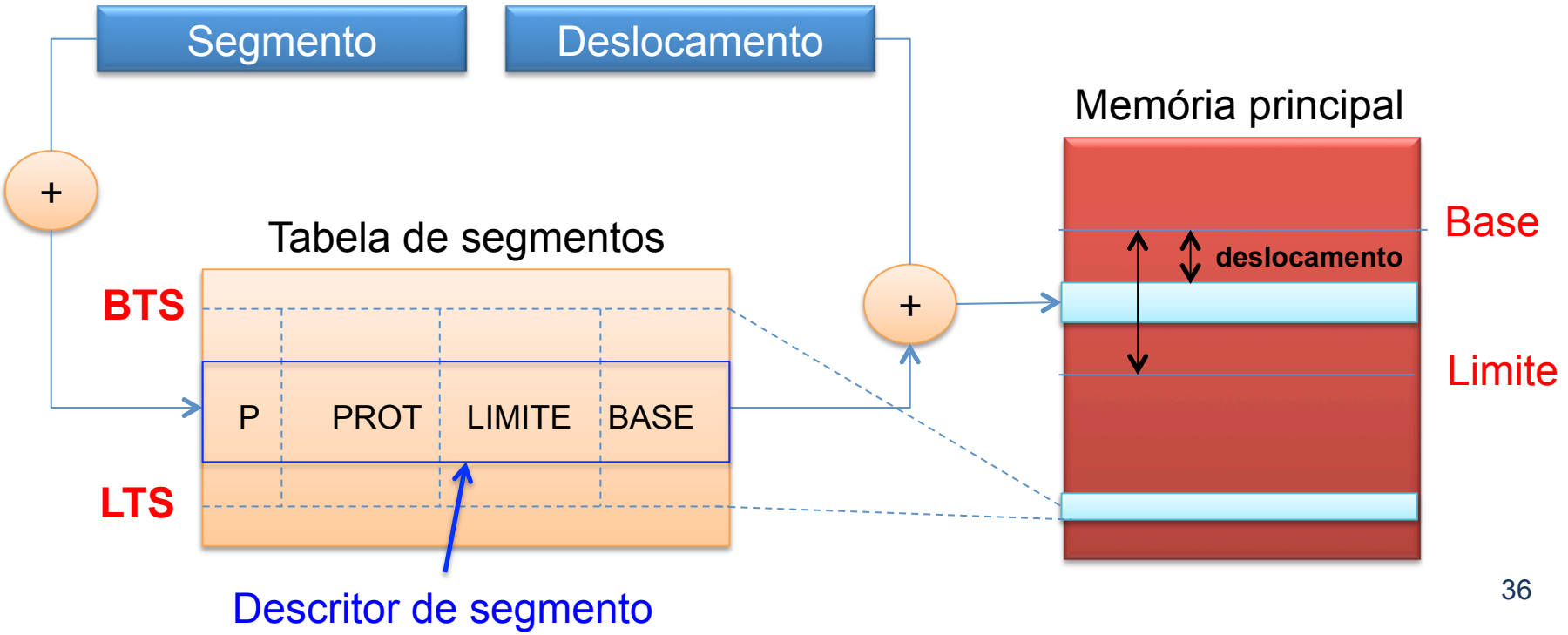
Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação
    - Endereço lógico

< #segmento, deslocamento>

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação
    - Tabela de Segmentos: end. lógico ➔ end. real



Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação
    - Tabela de Segmentos
      - Cada linha é um descritor de segmento.
      - Base: endereço físico onde começa um segmento.
      - Limite: endereço físico onde termina um segmento.
      - PROT: proteção (ex. escrita/leitura e nível de acesso necessário)
      - P: segmento está na memória principal
      - O registrador de BTS (Base da Tabela de Segmentos) e LTS (Limite da Tabela de Segmentos), contém respectivamente o endereço real do início e do fim da tabela.
      - Quando o programa gera um endereço lógico, o número do segmento é multiplicado pelo número de bytes do descritor e somado com o registrador BTS, obtendo-se a entrada na tabela de segmentos correspondente a este segmento.

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Vantagens da segmentação
    - Processos podem ser compartilhar segmentos (ex bibliotecas)
    - Mais fácil realocar segmento que processo inteiro
    - Reduz fragmentação interna
    - Proteção flexível (em nível de segmentos)

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Desvantagens da segmentação
    - Alocar segmentos com tamanhos diferentes
    - A segmentação gera fragmentação externa, uma vez que segmentos são alocados e liberados da memória.
    - A solução para o problema da fragmentação externa é recompactar a memória, ou seja, os segmentos são copiados para um dos extremos da memória, de forma a deixá-la livre no outro extremo.

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - A ideia da paginação é oferecer ao programador um espaço de endereçamento (virtual) contíguo.
    - A memória física é dividida em quadros de tamanho fixo
    - Memória lógica é dividida em páginas
    - Tamanho da página = tamanho do quadro definido pelo hardware
    - #páginas ≥ #quadros

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Endereço lógico

< #página|deslocamento >

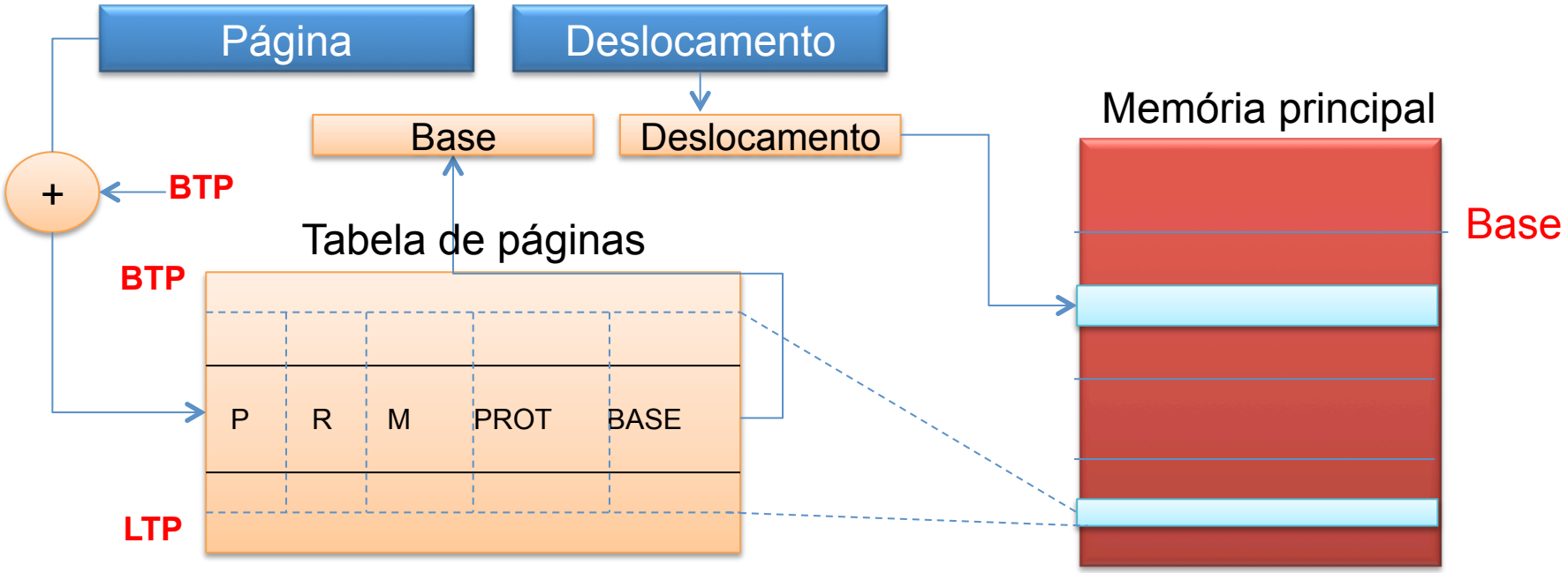
Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Exemplo de endereço lógico
      - 1024=2^{10} bytes de memória física
      - Quadros/páginas de 8=2^3 bytes

#página	deslocamente
3 bits	10-3 bits

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Tabela de Páginas



Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Tabela de Páginas
      - O mecanismo de acesso as páginas é semelhante ao utilizado na segmentação.
      - Existe uma tabela de páginas composta pelo descritor de cada página, chamado PTE *(Page Table Entry).*
      - Cada descritor contém o endereço físico inicial de cada página (BASE), informação de proteção (PROT) e informação de estado relativa à pagina em questão (bits P, R e M).
      - Os estados podem ser:
        - » P: indica se a página está carregada na memória primária.
        - » R: indica se a página foi acessada.
        - » M: indica se a página foi acessada e modificada.

Endereços Reais e Virtuais

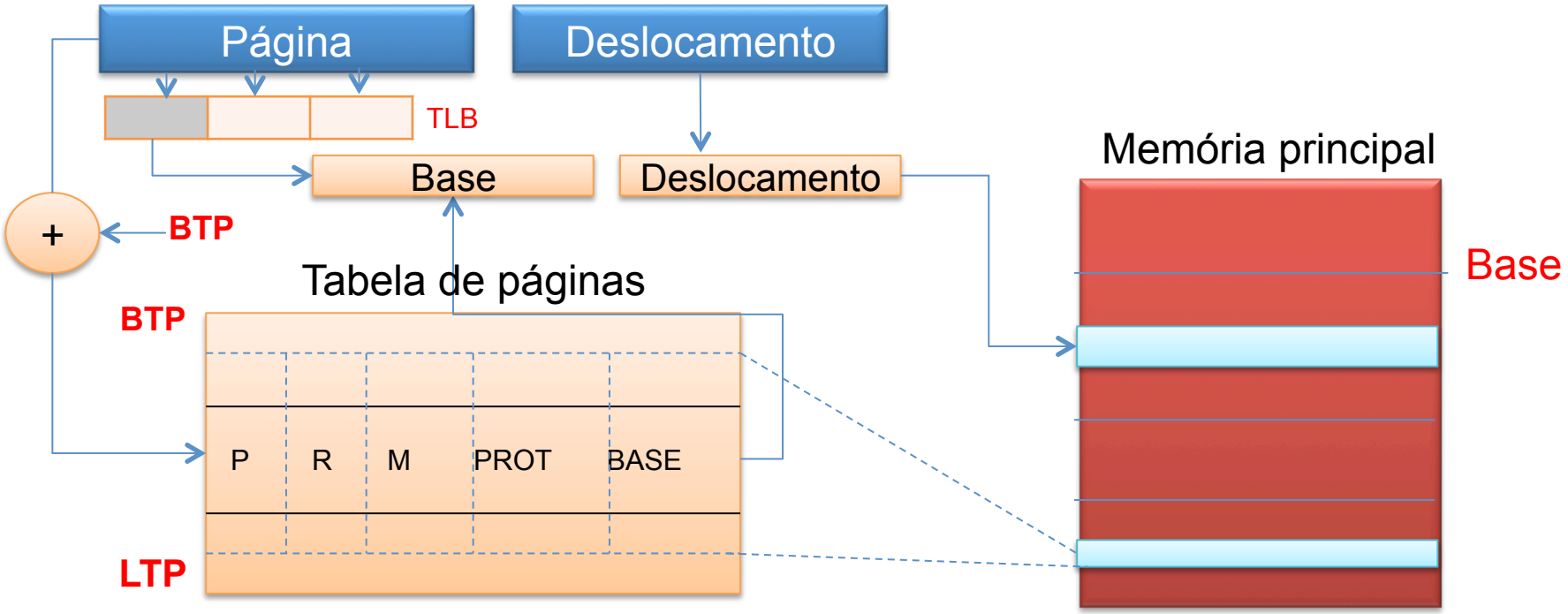
- Endereçamento Virtual
  - Paginação
    - Tabela de Páginas
      - As páginas possuem o mesmo tamanho, logo não é necessário guardar informações sobre a dimensão da página.
      - Os registradores BTP (Base da Tabela de Páginas) e LTP (Limite da Tabela de Páginas), contém respectivamente o endereço real de início da tabela de páginas e a dimensão desta.
      - Quando um processo gera um endereço, o número da página é comparado com o registrador LTP, se for inferior, é multiplicado pelo tamanho em bytes da PTE e somado com o registrador BTP.
      - A dimensão, ou seja, o tamanho de uma página está definido na arquitetura de hardware, podendo variar de 4 KB até 4 MB.

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Tabela de Páginas
      - Para aumentar o desempenho da tabela de páginas é utilizado uma memória associativa de acesso muito rápido, esta memória é chamada TLB *(Translation Lookaside Buffer).*
      - A TLB guarda os descritores da *n* últimas páginas acessadas pelo programa.
      - Quando o programa gera um endereço, a memória associativa consulta simultaneamente todas as posições, verificando se existe uma entrada cujo endereço da página seja igual ao endereço gerado pelo programa.
      - Se a página for encontrada na TLB seu endereço é colocado na saída, caso contrário é indicado ao hardware de memória que a página não foi encontrada.
      - A inserção de descritores na TLB se dá em ordem FIFO.

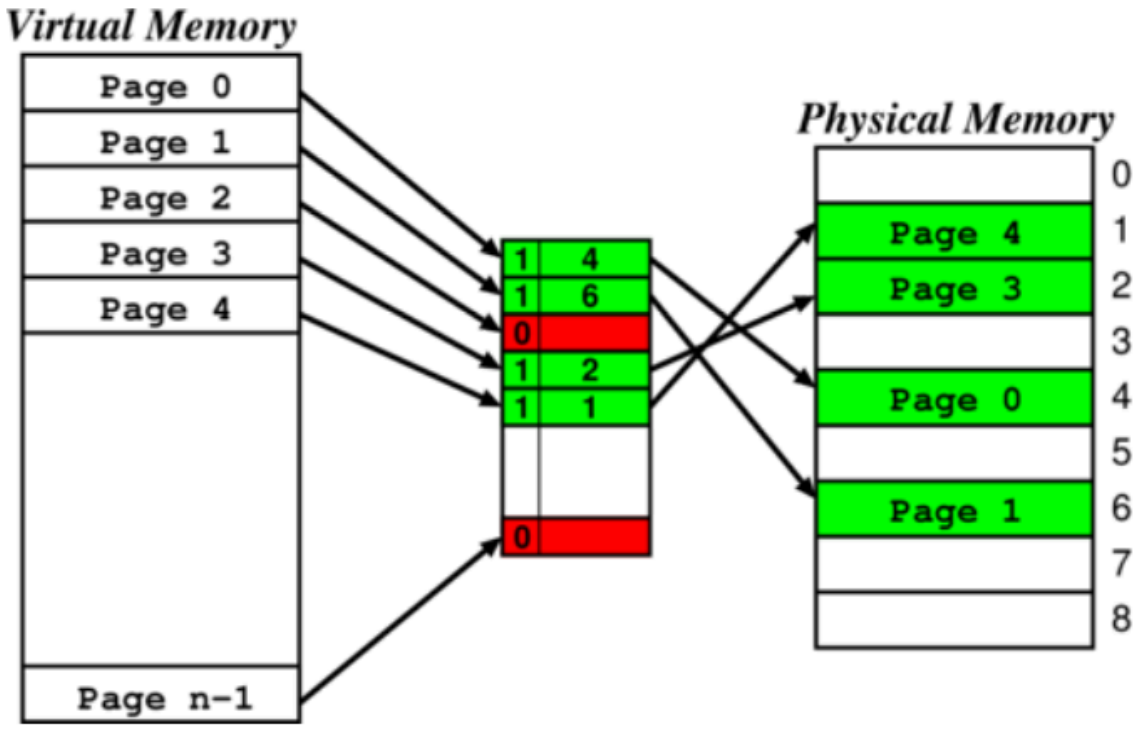
Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação
    - Tabela de Páginas com TLB



Endereços Reais e Virtuais

- Endereçamento Virtual
  - Paginação



Endereços Reais e Virtuais

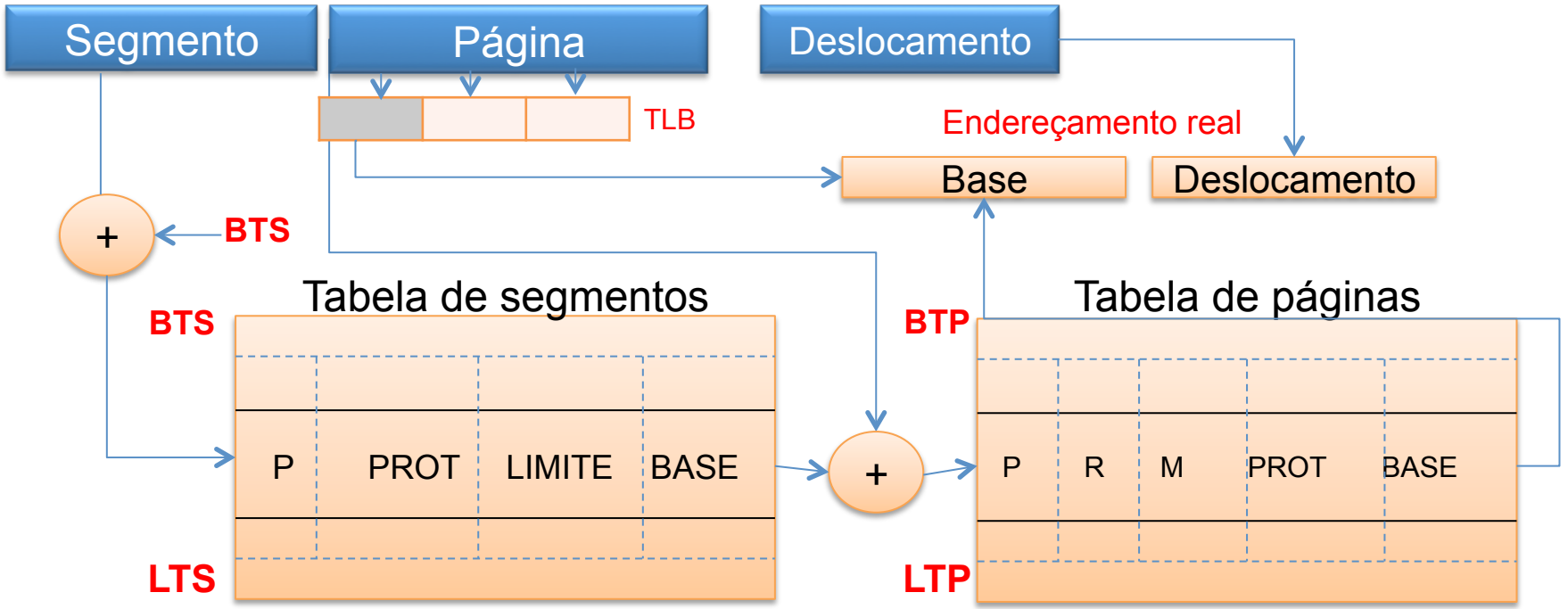
- Endereçamento Virtual
  - Vantagens da Paginação
    - Sem fragmentação externa
    - Programador trabalha apenas com end lógico
    - Permite criar algoritmos para trocar páginas de acordo com comportamento dos processos
  - Desvantagens
    - Fragmentação interna

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação Paginada
    - O mecanismo de alocação de memória baseado na metodologia segmentação paginada procura obter as vantagens da cada técnica (paginação e segmentação).
    - Um endereço de memória é formado por: segmento + + página + deslocamento.
    - O programa é dividido em segmentos lógicos e cada segmento é dividido em páginas.
    - O número do segmento é utilizado para obter o endereço físico do início da tabela de páginas do segmento, sendo a tradução do par página-deslocamento feita como no sistema de paginação.

Endereços Reais e Virtuais

- Endereçamento Virtual
  - Segmentação Paginada
    - Tabela de Segmentos/Páginas com TLB



Algoritmos de Gerenciamento de Memória

- Os algoritmos de gerenciamento de memória são necessários para:
  - Decidir onde se deve alocar um bloco de memória (segmento ou página);
  - Quando transferir um bloco da memória secundária para a memória secundária e vice-versa;
  - Qual bloco retirar da memória quando não existir mais espaço livre na memória primária.

Algoritmos de Gerenciamento de Memória

- Os algoritmos de gerenciamento de memória dependem fundamentalmente do Sistema Operacional, não importando o tipo de hardware da máquina.
- O SO deve executar as seguinte ações com relação a memória:
  - Alocação: alocar um bloco de memória de um determinado processo na memória principal;
  - Transferência: transferir blocos da memória principal para a secundária e vice-versa;
  - Substituição: decidir qual bloco será retirado da memória principal.

Algoritmos de Gerenciamento de Memória

- A operação de alocação de memória é realizada quando:
  - Um processo é criado ou finalizado;
  - Um processo solicita mais espaço de memória (alocação dinâmica. Ex. função *malloc()* linguagem C) e em virtude das chamadas de subrotinas (pilha de execução);

Algoritmos de Gerenciamento de Memória

- A operação de transferência é realizada quando:
  - Um processo necessita de um determinado segmento ou página que não foi alocado na memória principal.
  - Os algoritmos de transferência podem atuar de três maneiras distintas:
    1. A pedido (*on request*): o programa ou o SO invoca uma chamada de sistema que permite carregar outro bloco.
    2. Por necessidade (*on demand*): um programa necessita de um bloco de memória que ainda não foi carregado na memória (exceção por falta de bloco (segmento ou página).
    3. Por antecipação (*pre-fetching*): o bloco é carregado na memória principal antes de ser solicitado (acessado).

Algoritmos de Gerenciamento de Memória

- A operação de substituição é realizada quando:
  - O sistema operacional ao receber um pedido de alocação avalia se existe espaço livre disponível, se não existir ele deve substituir o bloco alocado na memória principal pelo bloco que está sendo solicitado.
  - O número de blocos livres em memória principal é delimitado por dois valores:
    - Mínimo (*low water mark*), e
    - Máximo (*high water mark*).

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A alocação de segmentos é mais difícil do que a alocação de páginas.
  - Os segmentos tem tamanho variado e devem ser alocados de forma contínua na memória principal.
  - O SO controle os blocos livres em memória mantendo uma lista que armazena o endereço do bloco e o seu tamanho.

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A busca por blocos livres na memória principal para a alocação de segmentos pode ser realizada com uma das seguintes abordagens (1/5):
    - **Best-Fit**: procura na lista de blocos livres um bloco cujo tamanho seja maior ou igual ao tamanho do bloco que se quer alocar. Este algoritmo pode gerar fragmentação externa.

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A busca por blocos livres na memória principal para a alocação de segmentos pode ser realizada com uma das seguintes abordagens (2/5):
    - **Worst-Fit**: procura na lista de blocos livres o que tem maior tamanho. A ideia é fazer com que o fragmento restante da alocação seja suficientemente grande para poder ser alocado a outro processo. A fila de blocos livres é ordenada em ordem decrescente de tamanho, ou seja, o bloco de maior tamanho será o primeiro da lista. A desvantagem é que aloca primeiro os blocos de maior tamanho.

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A busca por blocos livres na memória principal para a alocação de segmentos pode ser realizada com uma das seguintes abordagens (3/5):
    - **First-Fit**: escolhe o primeiro bloco livre de tamanho suficiente para satisfazer o pedido de alocação de memória. Na há a necessidade de se manter a lista de blocos livres ordenada.

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A busca por blocos livres na memória principal para a alocação de segmentos pode ser realizada com uma das seguintes abordagens (4/5):
    - **Next-Fit**: é uma variação do *first-fit* que visa pesquisar a lista de blocos livres a partir do ponto onde terminou a última pesquisa. Evite o acumulo de blocos pequenos nos extremos da memória.

Algoritmos de Gerenciamento de Memória

- Alocação de Segmentos
  - A busca por blocos livres na memória principal para a alocação de segmentos pode ser realizada com uma das seguintes abordagens (5/5):
    - **Buddy**: organiza a memória em blocos de tamanho b^n, b, n . Normalmente b = 2. Para satisfazer um pedido de tamanho T, o algoritmo percorre a lista de blocos livres à procura de um bloco de dimensão 2^k tal que $2^{k-1} < T < 2^k$. Se não for encontrado, a lista é percorrida à procura de um bloco de tamanho 2^{k+i} , i < 0, que será dividido em duas partes iguais de tamanho 2^{k+i-1} que serão chamados de buddies. Um desses buddies ainda será subdividido quantas vezes forem necessária até se obter um bloco de tamanho 2^k.

Algoritmos de Gerenciamento de Memória

- Transferência de Segmentos
  - Quando um processo é criado é possível que não haja espaço na memória primária para conter os seus segmentos.
  - O SO precisará transferir um ou mais segmentos de outro processo para a memória secundária.
  - Os segmentos que não cabem na memória ficam em uma área na memória secundária chamada de área de transferência (*swap*).

Algoritmos de Gerenciamento de Memória

- Transferência de Segmentos
  - Quando há a transferência de segmentos entre a memória primária e a secundária todo o segmento é transferido e não somente parte dele.
  - Alguns sistemas, quando há falta de memória principal, transferem todo o programa para a memória secundária, ao invés de somente alguns segmentos.
  - A transferência de um programa da memória principal para a memória secundária é chamada de swap, ou seja, o programa foi *swap out*.

Algoritmos de Gerenciamento de Memória

- Substituição de Segmentos
  - A substituição de segmentos é necessária quando não há mais espaço na memória primária para alocar novos segmentos, por isso o sistema precisa decidir qual ou quais segmentos deixarão a memória para dar espaço para o novo segmento.
  - Geralmente são utilizados três critérios para decidir quais segmentos serão transferidos para a memória secundária.

Algoritmos de Gerenciamento de Memória

- Substituição de Segmentos
  - Critérios para substituir segmentos:
    1. Estado e prioridade: geralmente processos em estado de aguardando ou menos prioritários são os primeiros candidatos a irem para a memória secundária.
    2. Tempo de permanência dos segmentos na memória primária: segmentos devem permanecer na memória principal por um determinado tempo, o suficiente para o processo poder usá-lo sem necessitar trocas com a memória secundária.
    3. Tamanho do segmento: o segmento escolhido para ser *swap out* deve ter tamanho suficiente para liberar uma determinada quantidade de memória suficiente para alocar os segmentos *swap in*.

Algoritmos de Gerenciamento de Memória

- Alocação de Páginas
  - Em um sistema baseado em paginação a alocação de página da memória é realizada em ordem FIFO.
  - As páginas livres são mantidas em uma lista, gerenciada, geralmente, conforme uma fila do tipo FIFO.
  - Na alocação retira-se a primeira página da lista e na liberação insere-se a página no fim da lista.

Algoritmos de Gerenciamento de Memória

- Transferência de Páginas
  - A transferência de páginas da memória secundária para a memória primária é por demanda.
  - Uma vez que um processo necessite de uma página ainda não alocada na memória principal, é gerado um erro por falta (exceção)de página, então o SO deve carregar a página faltante da memória secundária para a memória primária.
  - Toda a vez que uma página escrita tiver que deixar a memória, o sistema necessita gravar este página na memória secundária em uma área chamada de *área de paginação.*

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - O SO deve manter duas listas de páginas:
    1. A lista de páginas livres, ou seja, páginas que o sistema mantém uma cópia na memória secundária. Essas páginas podem ser utilizadas a qualquer momento, ou seja, substituídas por outras página do processo.
    2. A lista de páginas livres modificadas, ou seja, página que foram escritas pelo processo. Neste caso, antes da página ser substituída por outra, esta deve ser copiada para a memória secundária.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Quando ocorre um erro por falta de páginas, o SO retira a primeira página da lista de páginas livres.
  - Se o número de páginas livres ficar abaixo de um valor mínimo (*low water mark*), o SO deve acordar (colocar em execução) o paginador.
  - O paginador irá liberar páginas utilizando algum critério (*algoritmo de substituição de página*) de liberação até que o número de páginas livres suba até um valor máximo (*high water mark*).

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Periodicamente a lista de páginas livres modificadas é escrita na memória secundária, então as páginas desta lista passam para a lista de páginas livres.
  - A escolha da página vítima, ou seja, a página que será substituída por outra, deve ser baseada em algum critério, de preferência que não prejudique a execução do programa, ou seja, que não degrade seu desempenho em virtude de uma alta demanda de paginação.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (1/6)
    - LRU (*Least Recently Used* – Menos Recentemente Usada)
      - Retira-se da memória principal a página não utilizada recentemente.
      - A marcação da página vítima é feita por base em um contador de acessos baseado nos bit de leitura (R) e no bit de modificação (M), ou seja, quanto menos a página for acessada maior será a chance de ser substituída.
      - O paginador irá incrementar um contador marcando quão velha é a página (contador de idade).
      - Quando o sistema necessita substituir páginas, a escolha se dará baseado no contador de idade, quanto mais velha maior a chance de ser substituída.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (2/6)
    - NRU (*Not Recently Used* – Não Usada Recentemente)
      - Esse algoritmo escolhe como vítima a página que não tenha sido acessada recentemente.
      - O processo paginador percorre regularmente a tabela de páginas, analisa os bits R e M e volta a colocar o bit R em zero.
      - As páginas agrupam-se em quatro grupos:
        - » Grupo 0: (R = 0, M = 0), não referenciado, não modificada.
        - » Grupo 1: (R = 0, M = 1), não referenciado, modificada.
        - » Grupo 2: (R = 1, M = 0), referenciado, não modificada

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (2/6)
    - NRU (*Not Recently Used* – Não Usada Recentemente)
      - A página vítima é escolhida a partir da busca pelos grupos em ordem crescente de identificação dos grupos, sendo que a página escolhida é a que tiver o menor identificador.
      - O NRU considera apenas duas situações:
        1. Página nova – que tem o bit R com o valor 1;
        2. Página velha – que tem o bit R com o valor 0.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (3/6)
    - FIFO (*First In, First Out* – Primeiro a Entrar, Primeiro a Sair)
      - Esse algoritmo tem o funcionamento bastante simples.
      - É baseado em uma lista do tipo FIFO, onde a página a ser substituída ocupa a primeira posição da lista.
      - Toda página nova, que está sendo carregada na memória principal, é inserida no fim da lista.
      - Tende a degradar o desempenho de execução dos processos, uma vez que a página a ser substituída pode estar sendo constantemente utilizada.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (4/6)
    - Segunda Chance
      - O algoritmo da segunda chance é uma melhoria do algoritmo FIFO.
      - Permite alterar a posição de uma página muito antiga na lista que tenha sido referenciada (usada).
      - Se a página escolhida tiver o valor do bit R em 1, muda-se o valor para zero e então escolhe-se outra página.
      - Se a página escolhida tiver o valor do bit R em zero, efetua a substituição da página.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (5/6)
    - Relógio
      - Este algoritmo e uma melhoria do algoritmo segunda chance e visa manter as páginas em uma lista circular.
      - Um ponteiro referencia a página mais antiga presente na lista. Solução análoga a um ponteiro de um relógio.
      - Quando é necessário subsistir páginas, o algoritmo verifica o bit R da página apontada pelo ponteiro, se o valor do bit for zero a página será substituída.
      - Caso o valor bit R da página apontada seja 1, o seu valor é mudado para zero e o ponteiro avança uma posição na lista.
      - Quando uma página for substituída o ponteiro avança uma posição.

Algoritmos de Gerenciamento de Memória

- Substituição de Páginas
  - Algoritmos de Substituição de Páginas (6/6)
    - Espaço de Trabalho (*Working Set*)
      - O espaço de trabalho em um processo em um determinado espaço de tempo é definido como sendo um conjunto de páginas acessadas pelo processo nesse intervalo de tempo.
      - Em cada momento um processo deve ter na memória nem menos nem mais páginas que o seu espaço de trabalho.
      - Um processo só é colocado na memória se existir um número mínimo de páginas livres para evitar *trashing*.

Algoritmos de Gerenciamento de Memória

- Diagrama de Estados das Páginas
  - Uma página pode estar em um dos seguintes estados:
    - Livre: a página não está sendo usada, ou seja, está disponível para ser alocada. Todas as páginas inicialmente encontram-se neste estado.
    - Ocupada: página alocada pelo sistema operacional e que pode conter dados e código do programa.
    - Modificada: página que foi alterada e que depois deixou de ser utilizada.
    - Modificada em escrita: a página está sendo escrita na memória secundária. Operação que deverá ser realizada sempre que uma página liberada da memória e tiver sido escrita por um processo.

Algoritmos de Gerenciamento de Memória

- Diagrama de Estados das Páginas

Algoritmos de Gerenciamento de Memória

- Comparação entre a Segmentação e a Paginação
  - Segmentação (vantagens)
    - Adapta-se á estrutura lógica do programa;
    - Permite a realização de sistemas simples sobre um hardware simples;
    - Permite realizar eficientemente as operações que agem sobre um segmento inteiro.

Algoritmos de Gerenciamento de Memória

- Comparação entre a Segmentação e a Paginação
  - Segmentação (desvantagens)
    - O programador tem de ter sempre algum conhecimento dos segmentos subjacentes;
    - Os algoritmos de gerenciamento de memória se tornam bastante complicados em sistemas complexos;
    - O tempo de transferência dos segmentos entre a memória principal e a memória secundária pode ser muito grande e inaceitável para segmentos muitos grandes;
    - O tamanho (dimensão) dos segmentos é limitada.

Algoritmos de Gerenciamento de Memória

- Comparação entre a Segmentação e a Paginação
  - Paginação (vantagens)
    - O programador não tem de se preocupar com o gerenciamento de memória;
    - Os algoritmos de alocação, substituição e transferência são mais simples e eficiente;
    - O tempo de leitura de uma página da memória secundária é razoavelmente pequeno.

Algoritmos de Gerenciamento de Memória

- Comparação entre a Segmentação e a Paginação
  - Paginação (desvantagens)
    - O hardware é mais complexo para otimizar a tradução e tratar as faltas de páginas do que com memória segmentada;
    - Operações sobre segmentos lógicos são mais complexas e menos elegantes, pois têm de ser realizados sobre um conjunto de páginas;
    - O tratamento das faltas de páginas representa uma sobrecarga adicional de processamento.