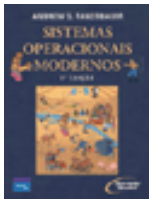
Capítulo 4

Gerenciamento de Memória

4.1 Gerenciamento básico de memória
4.2 Troca de processos
4.3 Memória virtual
4.4 Algoritmos de substituição de páginas
4.5 Modelagem de algoritmos de substituição de páginas
4.6 Questões de projeto para sistemas de paginação
4.7 Questões de implementação
4.8 Segmentação

Gerenciamento de Memória

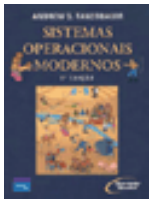


- Idealmente, o que todo programador deseja é dispor de uma memória que seja
  - Grande
  - Rápida
  - Não volátil
- Hierarquia de memórias
  - Pequena quantidade de memória rápida, de alto custo - cache
  - Quantidade considerável de memória principal de velocidade média, custo médio
  - Gigabytes de armazenamento em disco de velocidade e custo baixos
- O gerenciador de memória trata a hierarquia de memórias

Gerenciamento de Memória

- A memória é um dos recursos mais importantes, escassos e caros de de um computador, razão pela qual precisa ser gerenciada de uma maneira eficiente. Mesmo que os computadores atuais possuam muito mais memória que os computadores dos anos 1960, o tamanho dos programas cresce a uma taxa maior que a capacidade de memória.
- Na memória principal residem todos os programas e dados que serão executados e referenciados pelo CPU
- A memória secundária, podendo ser disco, fita, dentre outros, é um meio permanente, com mais capacidade e baixo custo, onde são armazenados programas e dados. Toda vez que desejamos executar um programa da memória secundária, este deve ser, obrigatoriamente carregado para a memória principal

Gerenciamento de Memória



- A parte do S.O. que gerencia a memória é denominado gerenciador (ou gerente) de memória, sua função é o de controlar quais partes da memória estão sendo usadas e quais não estão, de forma a alocar memória a processos quanto estes precisarem, liberar a memória quando estava sendo ocupada por um processo que terminou e tartar problemas entre memória principal e secundária. Quando a memória não for grande o suficiente, para guardar todos os processos utilizando técnicas como swapping, paginação e segmentação (utilizados na implementação da memória virtual
- Nos sistemas monoprogramáveis, o gerenciamento de memória é simples, já nos multiprogramáveis ela é muito complexa e se torna crítica, devido à necessidade de se manter o maior número de usuários possível utilizando a memória eficientemente, tornando sua gerência muito mais difícil
- Os sistemas de gerenciamento de memória podem ser divididos em duas categorias:
  - Os que não movem os processos entre memória principal e secundária
  - O que movem os processos entre memória principal e secundária (swapping, paginação e segmentação)

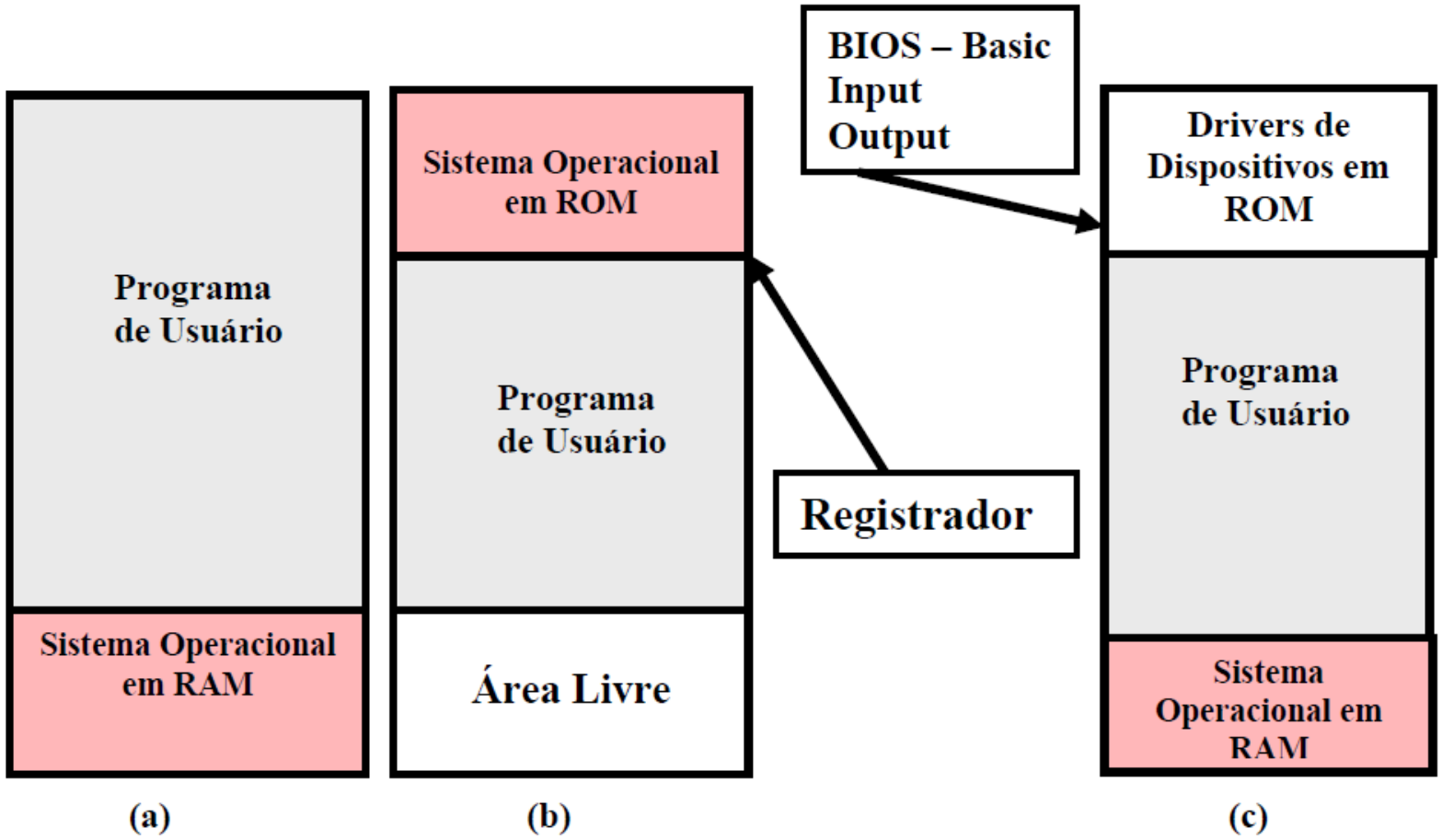
Gerenciamento de Memória Sem Swapping ou Paginação

- Esquema mais simples de alocação de memória a processos, usado até 1960, sendo o que apenas existe um único processo na memória em cada instante, sendo permitido que tal processo use toda a memória disponível, a monoprogramação
- O usuário carrega a memória principal com um processo vindo do disco ou da fita magnética e este assume o controle de todos os recursos da máquina
- Atualmente, nos computadores pessoais, os processos precisam de um driver para cada dispositivo de E/S e o código destes drivers deve estar presente na memória, junto com o código do processo

Gerenciamento de Memória Sem Swapping ou Paginação Alocação Contígua Simples

- Nesse tipo de gerenciamento de memória, a memória principal é dividida em duas partes:
  - Uma para o S.O.
  - Uma para o programa do usuário
- O programador deve desenvolver as aplicações preocupado, apenas, em não ultrapassar o espaço de memória disponível, ou seja, o tamanho total de memória principal menos o que está sendo ocupado pelo S.O. e pelo processo usuário
- Implementada nos primeiros Sistemas Operacionais e presente em S.O. monoprogramáveis
- O Sistema pode estar em:
  - (a) No início da memória RAM
  - (b) Na parte mais alta do sistema de endereçamento da memória em ROM
  - (c) Os drivers de dispositivo em ROM, conhecido com o nome de BIOS (Basic Input Output System) e o restante do S.O. em RAM

Gerenciamento de Memória Sem Swapping ou Paginação Alocação Contígua Simples



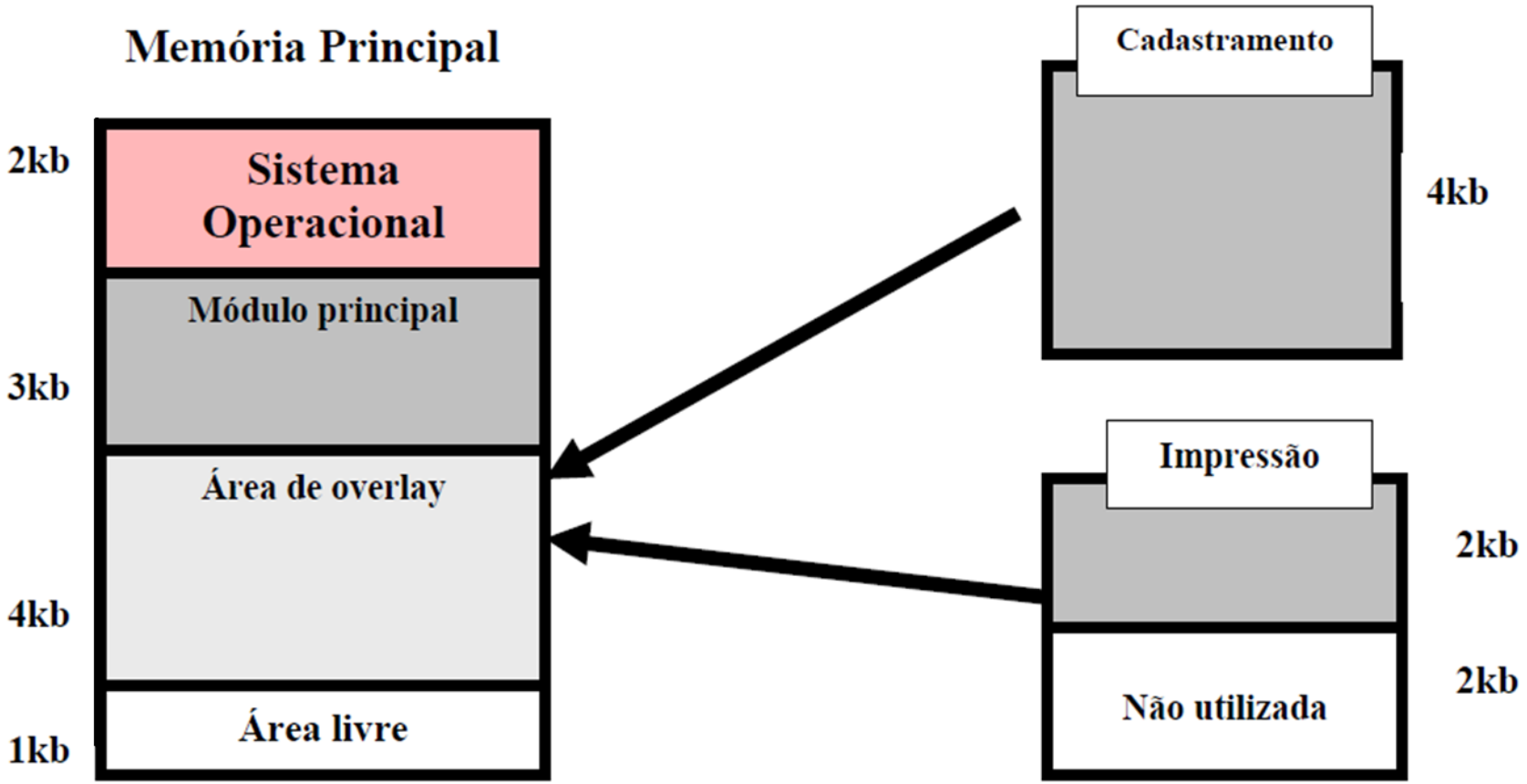
Gerenciamento de Memória Sem Swapping ou Paginação Alocação Contígua Simples

- Usuário tem controle sobre toda a memória principal, podendo acessar qualquer posição de memória, inclusive para alterar e destruir o S.O.
- Para protegê-lo desses ataques que podem ser conscientes ou não, alguns sistemas implementam proteção através de um registrador que delimita as áreas do S.O. e do usuário
- Sempre que um programa faz referência a um endereço de memória, o Sistema verifica se o endereço está nos seus limites e, caso não esteja, o programa do usuário é cancelado e uma mensagem de erro é enviada (violação de acesso – access violation)
- Apesar de fácil implementação e código reduzido, a alocação simples não permite a utilização eficiente do processador e da memória, pois apenas um usuário pode dispor desses recursos.
- Em relação à memória, caso o programa do usuário não a preencha totalmente, existirá um espaço de memória sem utilização

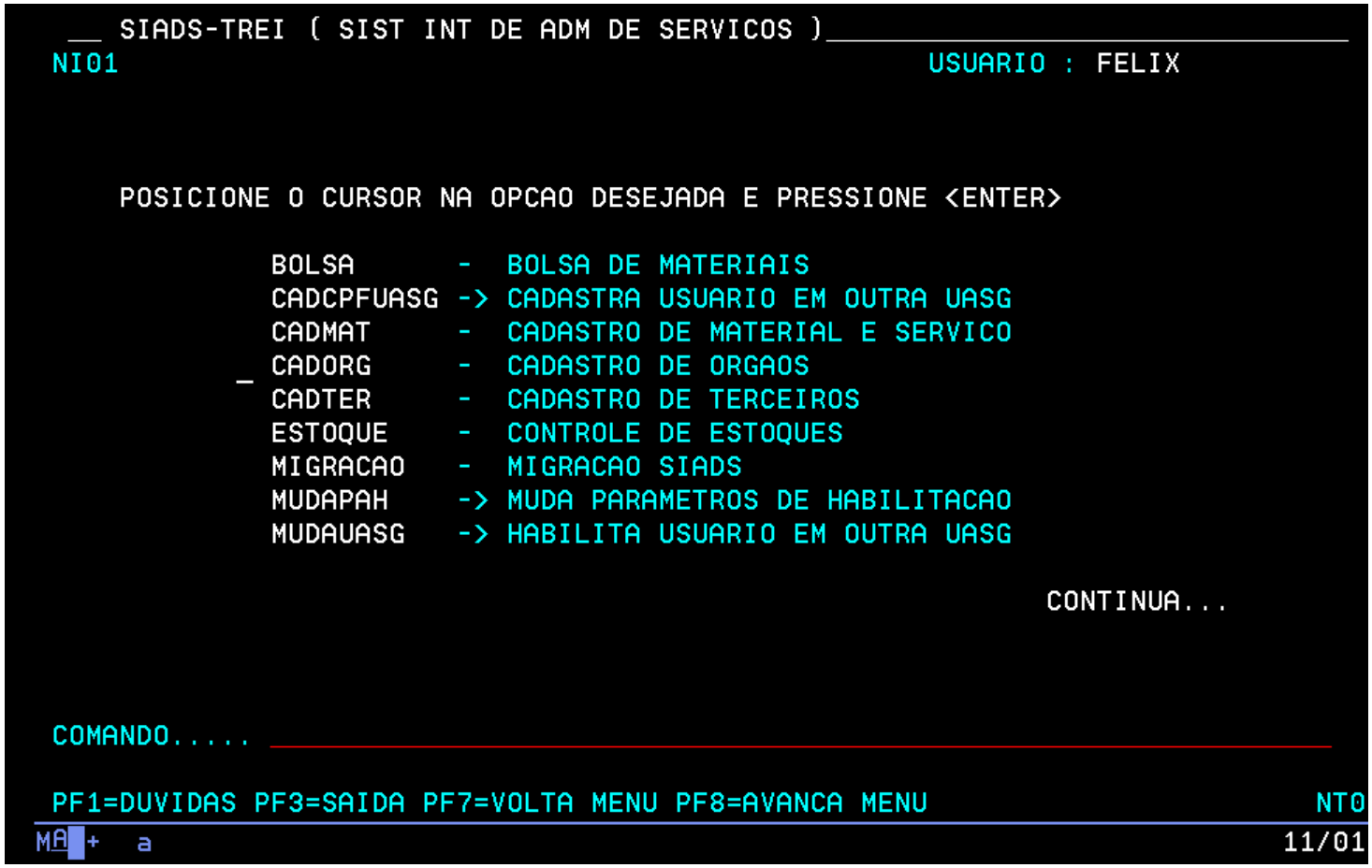
Gerenciamento de Memória Sem Swapping ou Paginação Técnica de Overlay

- A Técnica de Overlay (sobreposição) consiste em divider o programa em partes (módulos) de forma que os programas de usuários estejam limitados ao tamanho disponível da memória principal e de forma que pudessem executar independentemente esses módulos utilizando uma mesma área da memória
- Exemplo:
  - Programa com três módulos (Principal, Cadastramento e Impressão)
  - Cadastramento e Impressão são independentes
  - Quando um módulo estiver na memória, o outro não precisa estar
  - O módulo principal é comum a ambos e deve estar na memória o tempo todo
- Como a memória é insuficiente, a Técnica de Overlay utiliza uma área comum de memória
- Quando um modulo é referenciado pelo modulo principal, ele será carregado da memória secundária para a principal
- Se o modulo referenciado não está na memória, ele irá sobrepor o que está na área comum da memória, caso contrário, apenas é feita a referência
- A definição das áreas de Overlay é reponsabilidade do programador, através de commandos especiais da linguagem utilizada.
- O tamanho da área de Overlay é estabeldcido pelo tamanho do maior modulo do conjunto

Gerenciamento de Memória Sem Swapping ou Paginação Técnica de Overlay



Gerenciamento de Memória
Sem Swapping ou Paginação
Técnica de Overlay



Gerenciamento de Memória Sem Swapping ou Paginação Alocação Particionada Estática Absoluta

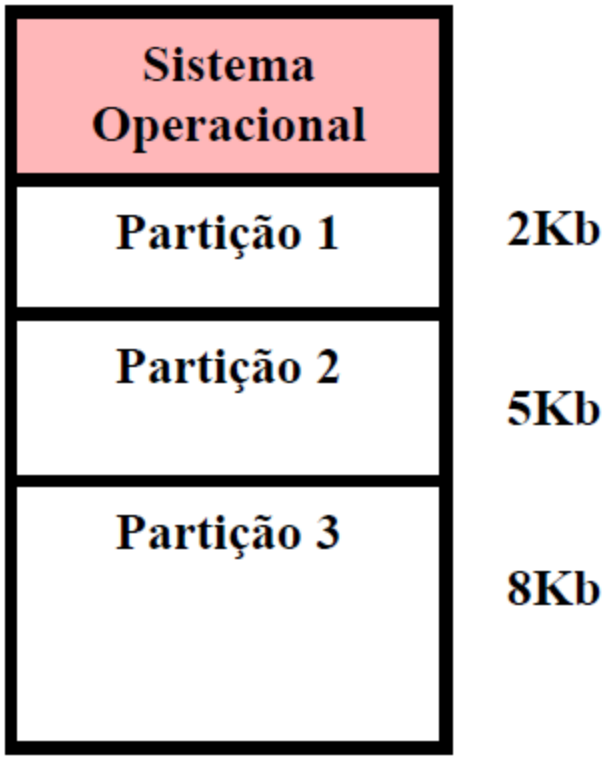
- Utilizada nos primeiros sistemas multiprogramáveis, a memória era dividida em pedaços de tamanho fixo, chamados partição.
- O tamanho das partições era estabelecido na fase de inicialização do sistema, em função do tamanho dos programas a serem executados
- Processos só poderia acessar partições específicas, de acordo com seu tamanho, mesmo que outras partições estivessem livres
- Para alteração de tamanho da partição, sempre que posse necessária, o sistema deveria ser desativado e reiniciado com uma nova configuração

Gerenciamento de Memória
Sem Swapping ou Paginação
Alocação Particionada Estática Absoluta

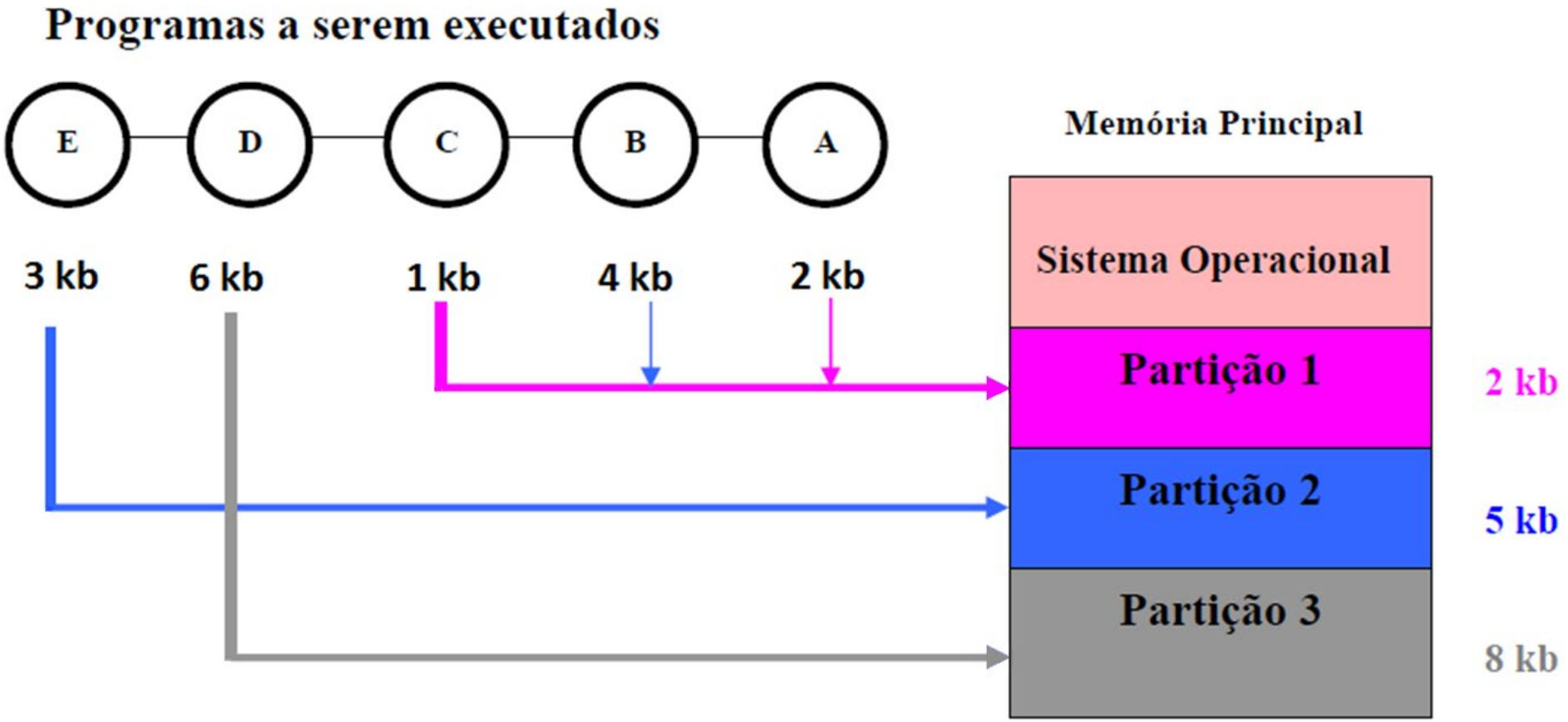
Tabela de partições

Partição	Tamanho
1	2 kb
2	5 kb
3	8 kb

Memória Principal



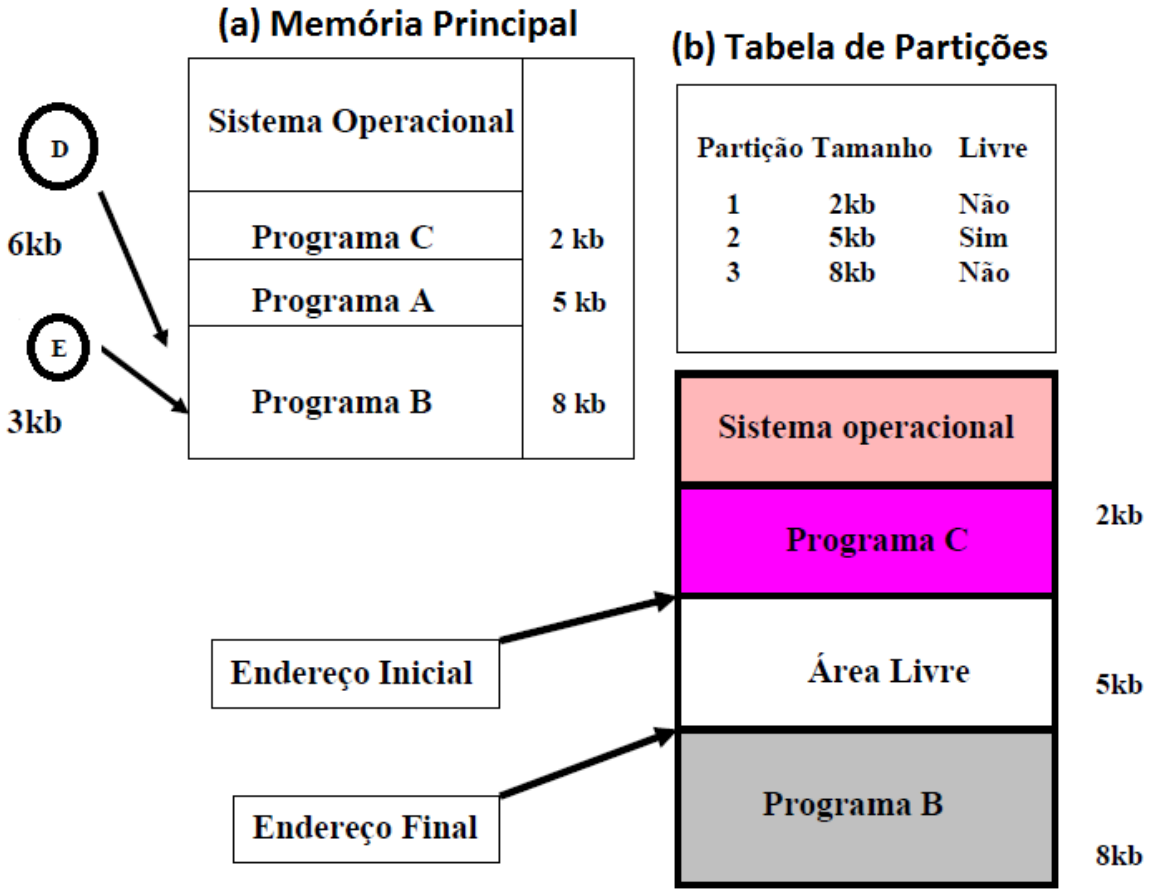
Gerenciamento de Memória
Sem Swapping ou Paginação
Alocação Particionada Estática Absoluta



Gerenciamento de Memória
Sem Swapping ou Paginação
Alocação Particionada Estática Relocável

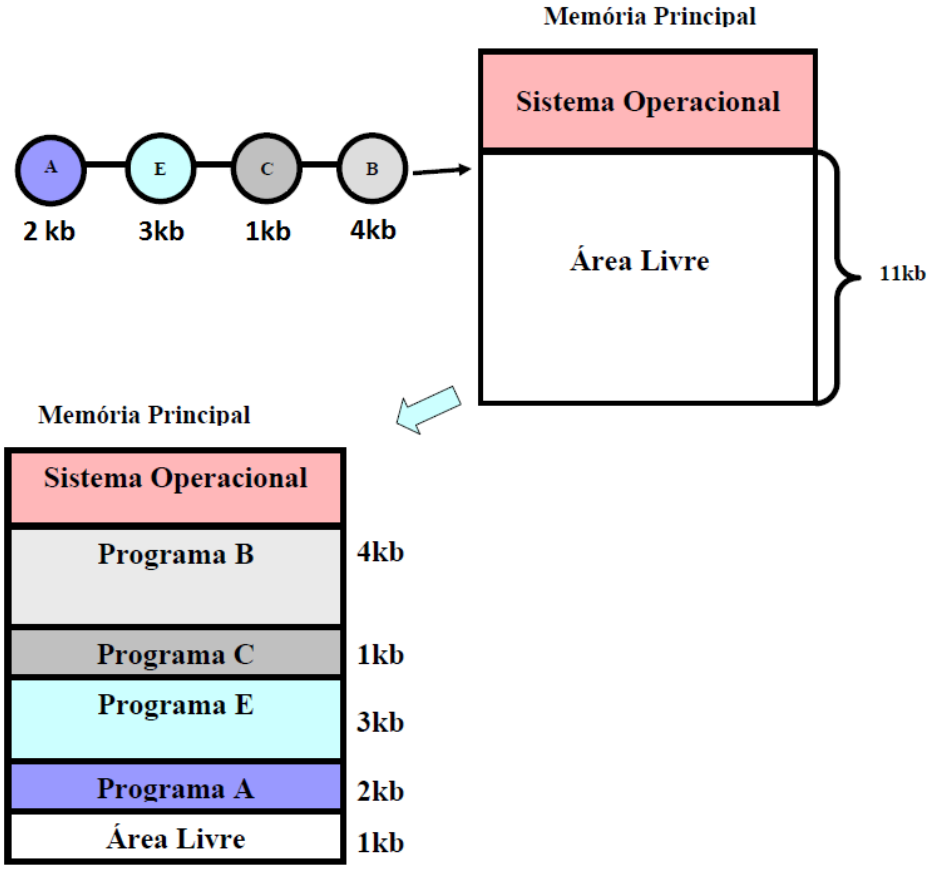
- Os compiladores links e loaders, nesta fase, geram código relocável e os programas podem ser carregados em qualquer partição
- A partir de então, independente da partição que estiver livre, o programa pode utilizar, desde que seu tamanho seja compatível com o tamanho livre
- Para manter o controle sobre quais partições estavam alocadas ou não, os sistemas possuiam uma tabela, chamada Tabela de Partições, delimitando a partição, seu tamanho e se está em uso
- Sempre que o sistema precisava alocar um programa, ele percorria a Tabela de Partições
- Esquema baseado em dois registradores que indicam os limites inferior e superiores da partição onde o programa está sendo executado
- Caso o programa tente acessar uma área fora dos limites definidos pelos registradores, ele é interrompido e uma mensagem de erro é enviada

Gerenciamento de Memória
Sem Swapping ou Paginação
Alocação Particionada Estática Relocável



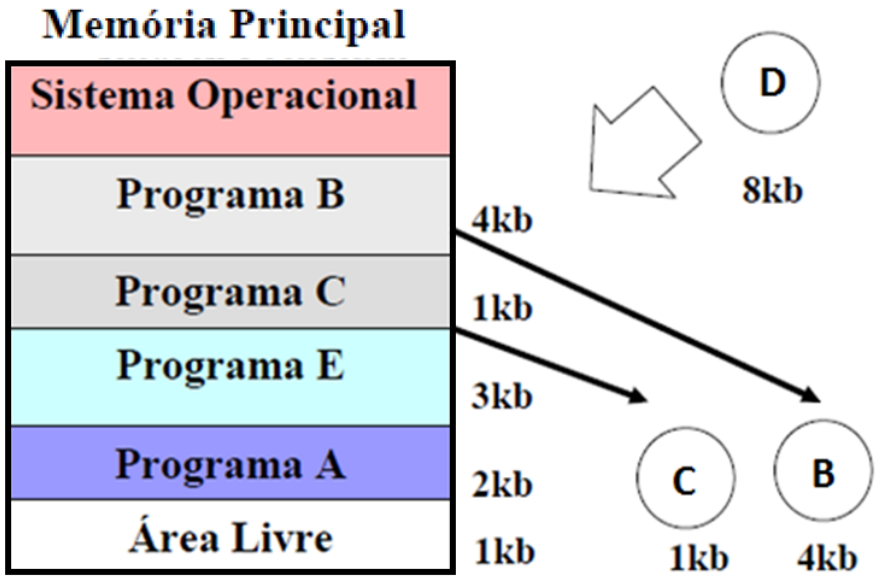
Gerenciamento de Memória Sem Swapping ou Paginação Alocação Particionada Dinâmica

- Na alocação particionada estática foi apresentado o problema da fragmentação (“buracos” criados nas partições quando os programas não ocupam a partição toda)
- Necessário um outro tipo de alocação como solução, a Alocação Particionada Dinâmica, onde foi eleminado o conceito de partições de tamanho fixo
- Cada programa utilizaria o espaço que necessitasse, pssando esse pedaço a ser sua partição
- Essa alocação diminui o problema da fragmentação para aumentar o uso compartilhado de memória



Gerenciamento de Memória Sem Swapping ou Paginação Alocação Particionada Dinâmica

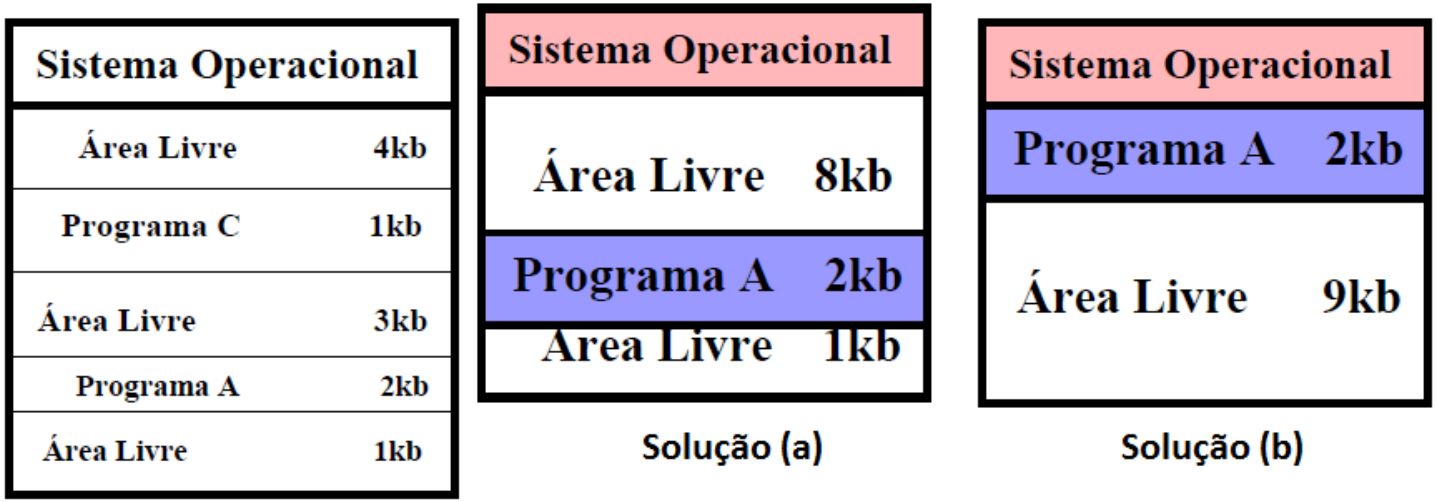
- Nos sistemas de alocação anteriores, os programas, normalmente, não preenchiam totalmente as partições onde eram carregados
- Se um programa for maior que a partição livre, ficará aguardando uma partição que o acomode, mesmo que duas partições adjacentes satisfaçam essa necessidade
- Esse problema é chamado de fragmentação, onde pedaços de memória ficam impedidos de serem utilizados por programas
- Fragmentação ocorre quando os programas vão finalizando deixando espaços cada vez menores na memória, não permitindo o ingresso de novos programas



Mesmo que B, C e E finalizem, o programa D não poderá ser carregado

Gerenciamento de Memória
Sem Swapping ou Paginação
Alocação Particionada Dinâmica

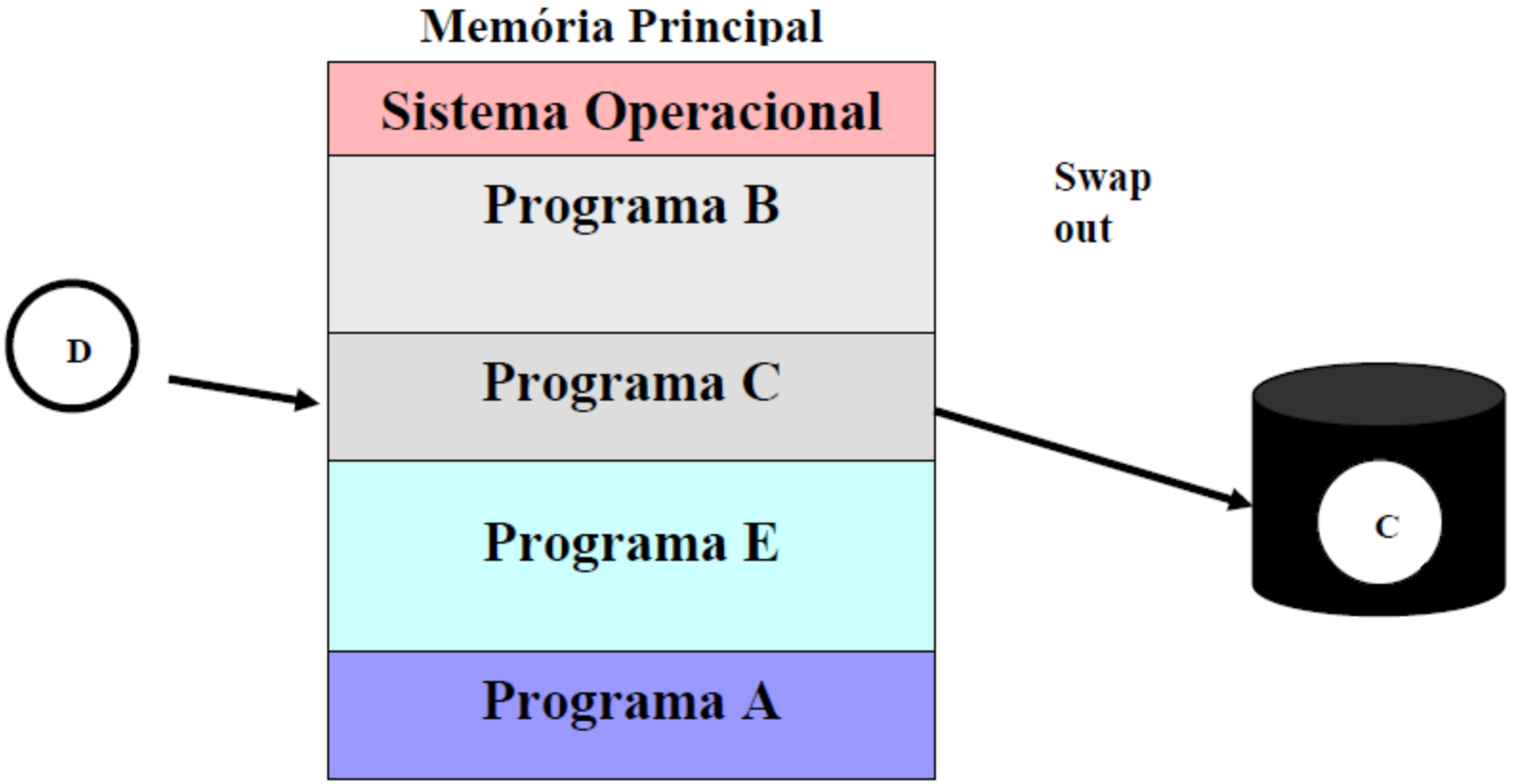
- Depois de já ter sido detectada a fragmentação da memória, existem duas soluções para o problema, chamado desfragmentação
  - (a) O sistema pode unir os espaços adjacentes, produzindo espaço de memória maior
  - (b) O sistema pode realocar todas as partições ocupadas, eliminado espaaços entre elas e criando uma única área livre contígua (compactação)
  - A complexidade do algoritmo de desfragmentação e o consumo de recursos do sistema, como processador e área do disco podem tornar esta técnica inviável



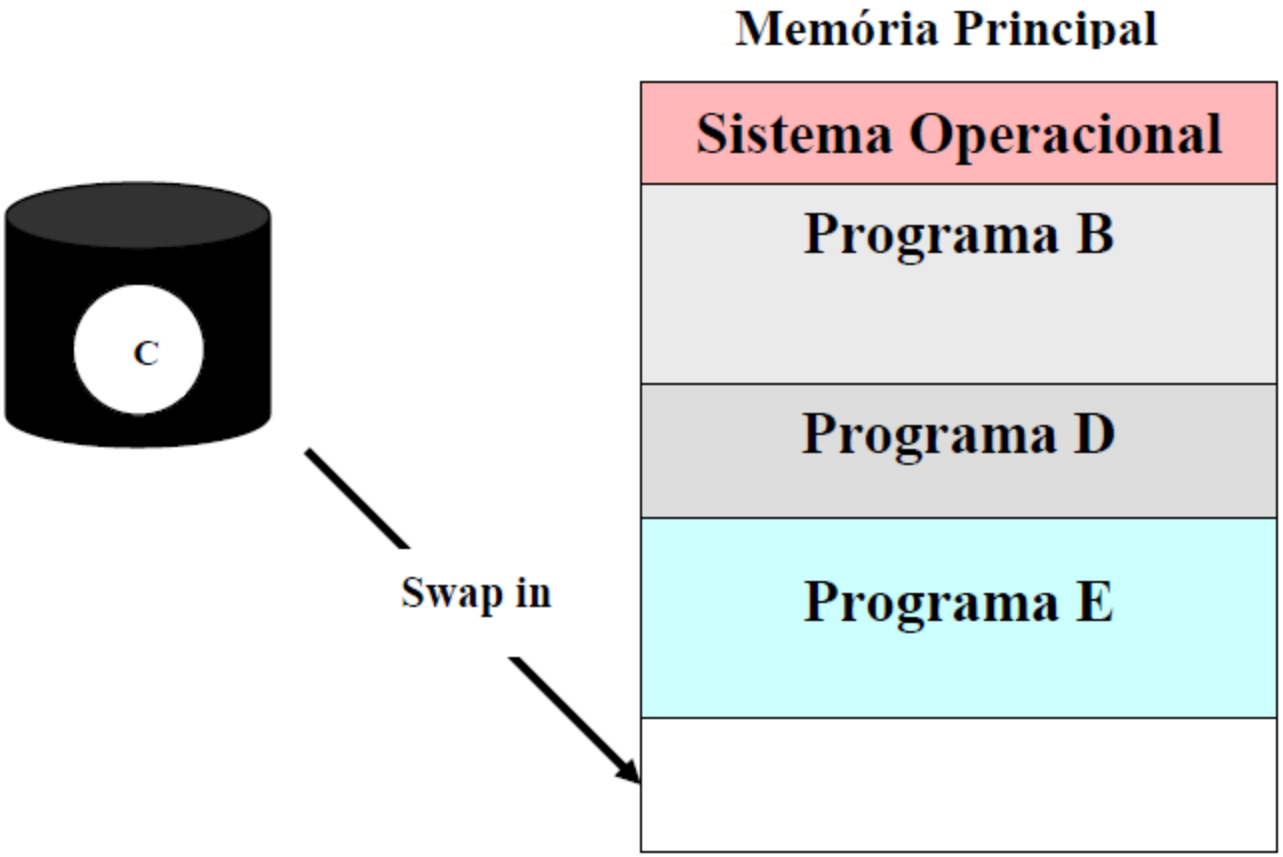
Gerenciamento de Memória Com Swapping

- Mesmo tendo um aumento na eficiência da multiprogramação e no gerenciamento de memória, muitas vezes um programa não podia ser executado por falta de uma partição livre disponível
- Em todas as técnicas anteriores, o programa permanecia na memória principal até o final de sua execução, inclusive nos momentos em que espeava um outro evento, como uma operação de leitura ou gravação em periféricos
- Swapping é uma técnica aplicada ao gerenciamento de memória para programas que aguardam memória livre para serem processados
- Tenta resolver o problema da insuficiência de memória para todos os usuários
- Nessa situação, o sistema escolhe um programa residente, que é levado da memória principal para a memória secundária (Swap out), retornando posteriormente para a memória principal (Swap in), como se nada tivesse ocorrido

Gerenciamento de Memória Com Swapping



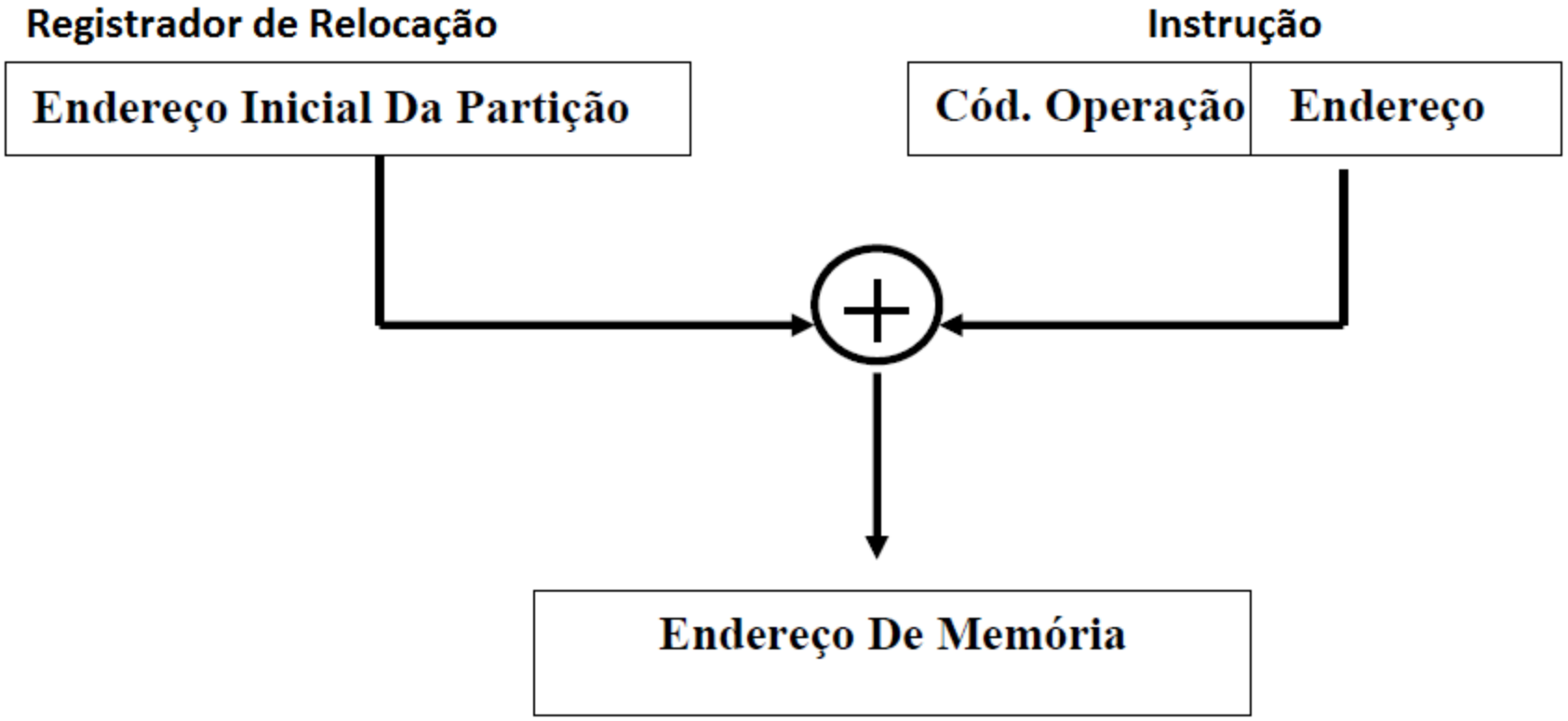
Gerenciamento de Memória Com Swapping



Gerenciamento de Memória Com Swapping

- Um dos problemas gerados pelo swapping é a realocação dos programas
  - Carregador (loader) permite que um programa seja colocado em qualquer lugar da memória
  - Realocação, normalmente ocorre no momento do carregamento
  - Se o programa sai e volta, ele será relocado a cada carregamento
  - Mecanismo é ineficiente pelo tempo gasto para o carregamento
  - Poderia se aguardar o espaço de memória anterior ficar livre
  - Melhor solução é implantar no hardware mecanismo para relocar durante a execução do programa, denominado relocação dinâmica
- Relocação dinâmica é realizada através de um registrador denominado registrador de relocação.
- Quando o programa é carregado na memória o registrador recebe o endereço inicial da região de memória que o programa utilizará
- Toda vez que houver uma referência a algum endereço, o endereço contido na instrução será somado ao endereço armazenado no registrador, gerando assim o endereço físico
- Relocação dinâmica é essencial para implementação de sistemas multiprogramáveis
- Swapping permitiu maior compartilhamento de memória e aproveitamento maior da CPU
- Eficiente para sistemas com poucos usuários competindo por memória e em ambientes com aplicações pequenas
- Maior problema é o elevado custo das operações de E/S (Swap In / Swap Out)

Gerenciamento de Memória Com Swapping



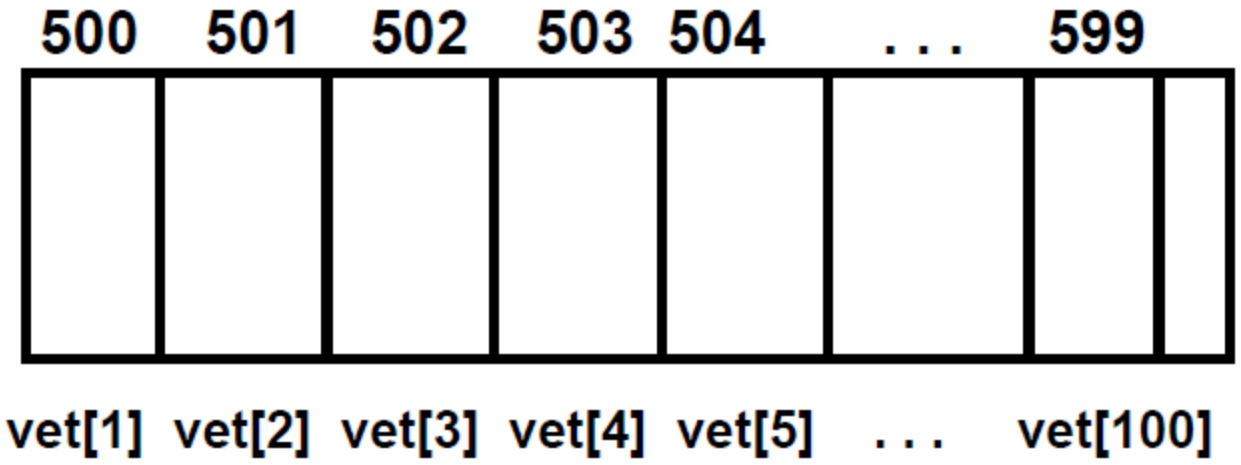
Gerenciamento de Memória
Com Swapping
Memória Virtual

- Memória virtual é uma técnica onde memória principal e memória secundária são combinadas, dando ao usuário a ilusão de existir uma memória muito maior que a memória principal
- Baseado em desvincular o endereçamento feito pelos programas dos endereços físicos da memória principal
- Assim, programas e estruturas de dados deixam de estar limitados ao tamanho da memória física disponível
- Minimiza problemas de fragmentação
- Assemelha-se a um vetor, como os existentes em linguagem de alto nível, pois, quando se faz referência a um componente do vetor, não se sabe em que posição de memória aquele dados está armazenado, sendo papel do compilador gerar instruções que implementam o mecanisco de encontrá-lo.
- Memória virtual utiliza conceito semelhantes, mas em relação dos endereços dos programas e aos seus dados

Gerenciamento de Memória Com Swapping Memória Virtual

- Um programa no ambiente de memória virtual não faz referência a endereços físicos de memória (endereços reais), mas apenas a endereços virtuais
- No momento da execução de uma instrução, o endereço virtual é traduzido para um endereço físico, pois a CPU processa apenas posições da memória principal
- Tradução de endereço virtual para endereço físico é denominamo mapeamento
- O espaço de endereçamento virtual não tem relação direta com os endereços no espaço real, sendo que um programa pode fazer referências a endereços virtyuais que estejam fora dos limites do espaço real, portanto, programas e estruturas de dados não estão mais limitados ao tamanho da memória física, mas apenas parte deles pode estar residente na memória, como extensão da memória principal
- Quando um programa é executado, apenas parte dele fica residente na memória principal, permanecendo o restante na memória secundária até o momento de ser referenciado
- Os programadores ignoram a existência de endereços virtuais quando do desenvolvimento de aplicações, cabendo ao compilador e aos linkers gerar código executável em função desses endereços e o S.O. cuida dos detalhes da execução

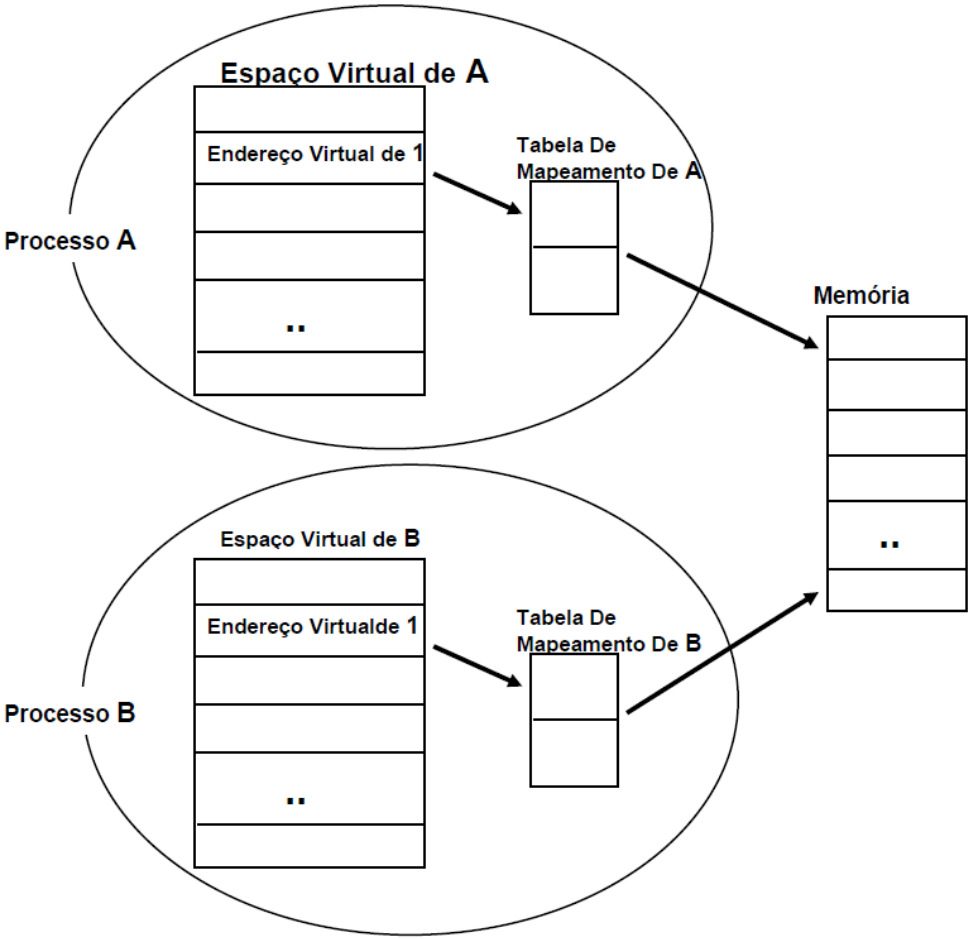
Gerenciamento de Memória Com Swapping Memória Virtual



Gerenciamento de Memória Com Swapping Mapeamento

- O mecanismo de mapeamento permite ao Sistema Operacional traduzir um endereço localizado no espaço virtual para um no espaço real, pois o programa executado em seu contexto precisa estar no espaço real para poder ser referenciado ou executado, logo, um programa não precisa estar em espaço contíguo na memória real para ser executado
- Cada processo tem o mesmo espaço de endereçamento virtual, como se possuísse sua propria memória virtual
- O mecanismo de tradução se encarrega, portanto, de manter tabelas de mapeamento exclusivas para cada processo, relacionando os endereços virtuais do processo às suas posições na memória física
- Quando um programa está em execução, o sistema, para realizar a tradução, utiliza a tabela de mapeamento do processo no qual o programa executa
  - Para outro processo, o sistema deve referenciar a tabela do novo processo
  - Realizado através de um registrador que indica a posição inicial da tabela de mapeamento corrente, onde, toda vez que há mudança de context, o registrador é atualizado com o endereço da nova tabela

Gerenciamento de Memória Com Swapping Mapeamento



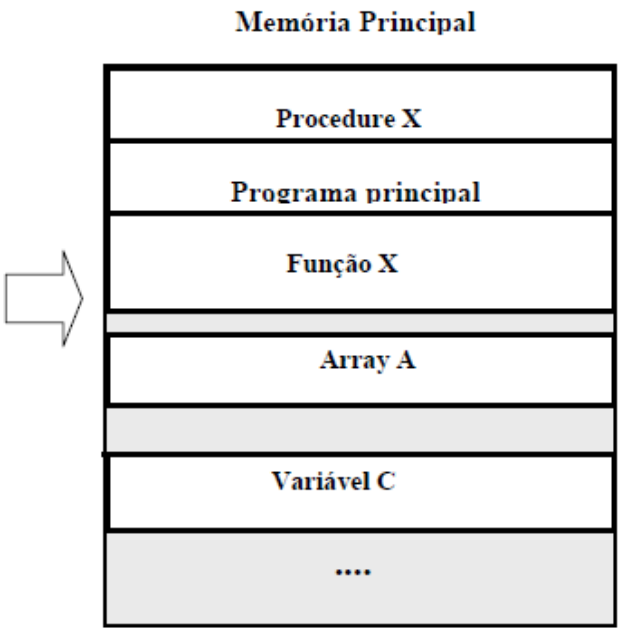
Gerenciamento de Memória Com Swapping Segmentação

- Técnica de gerenciamento de memória onde os programas são divididos, logicamente, em sub-rotinas e estruturas de dados e colocados em blocos de informações na memória
- Os blocos tem tamanhos diferentes e são chamados segmentos, cada um com seu próprio espaço de endereçamento

```
PROGRAM segmento
Var A : array ...
   C : integer;

PROCEDURE X;
.
.
END;

FUNCTION Y;
.
.
END;
BEGIN
.
.   } Programa principal
.
END.
```



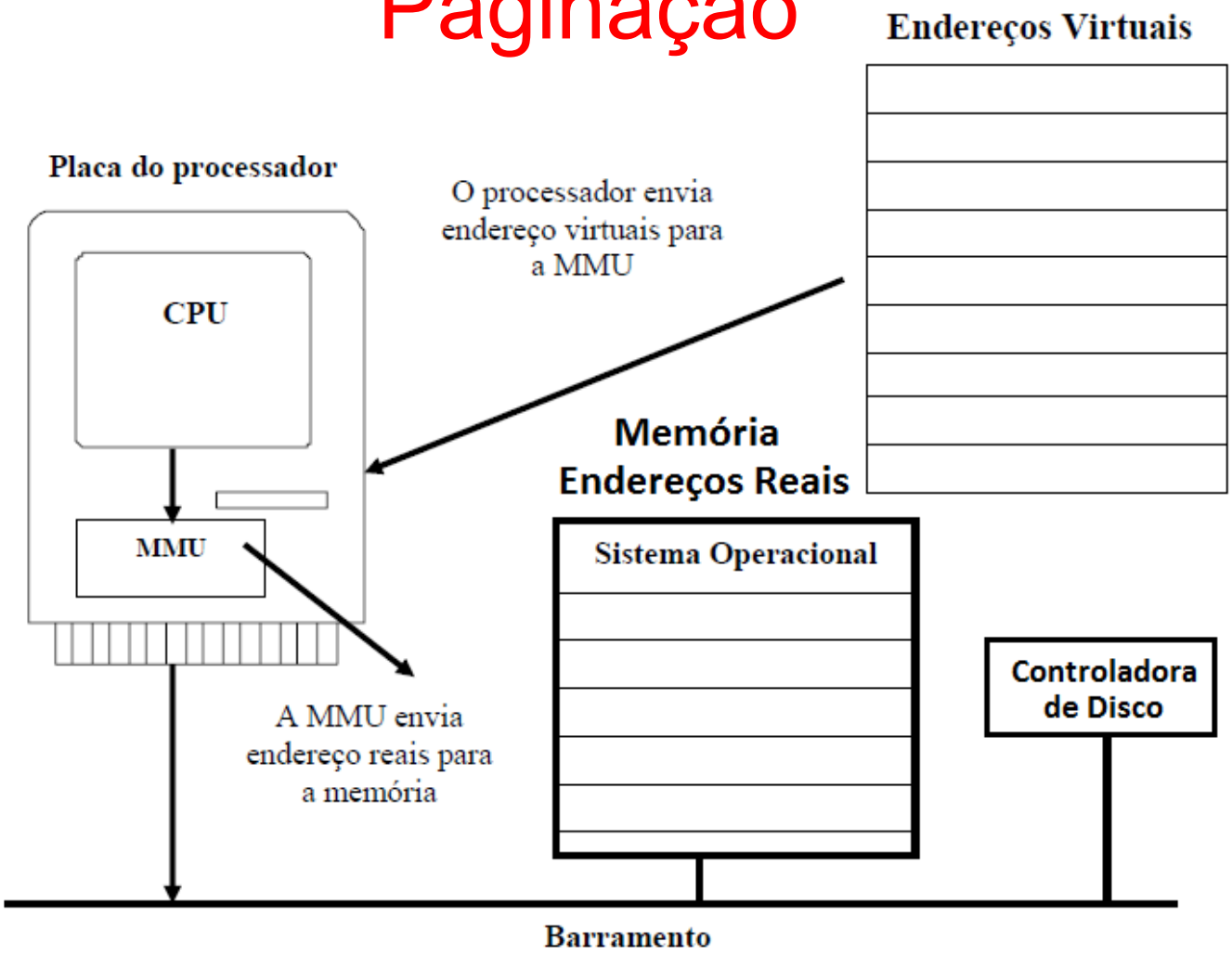
Gerenciamento de Memória Com Swapping Segmentação

- A diferença entre segmentação e paginação é que a segmentação faz divisões lógicas e a paginação faz divisões em blocos de tamanho fixo não vinculado com a estrutura do programa
- O mapeamento é feito utilizando, além do endereço do segment na memória física cada entrada na tabela de segmentos possui informaçãões sobre o tamanho do segment e se ele está ou nao na memória principal
- Se as aplicações não estiverem divididas em módulos, grandes pedaços de código estarão na memória desnecessariamente, não permitindo que outros usuários também utilizem a memória
- O problema da fragmentação também ocorre nesse modelo, quando as áreas livres são tão pequenas, que não acomodam nenhum segmento que necessite ser carregado

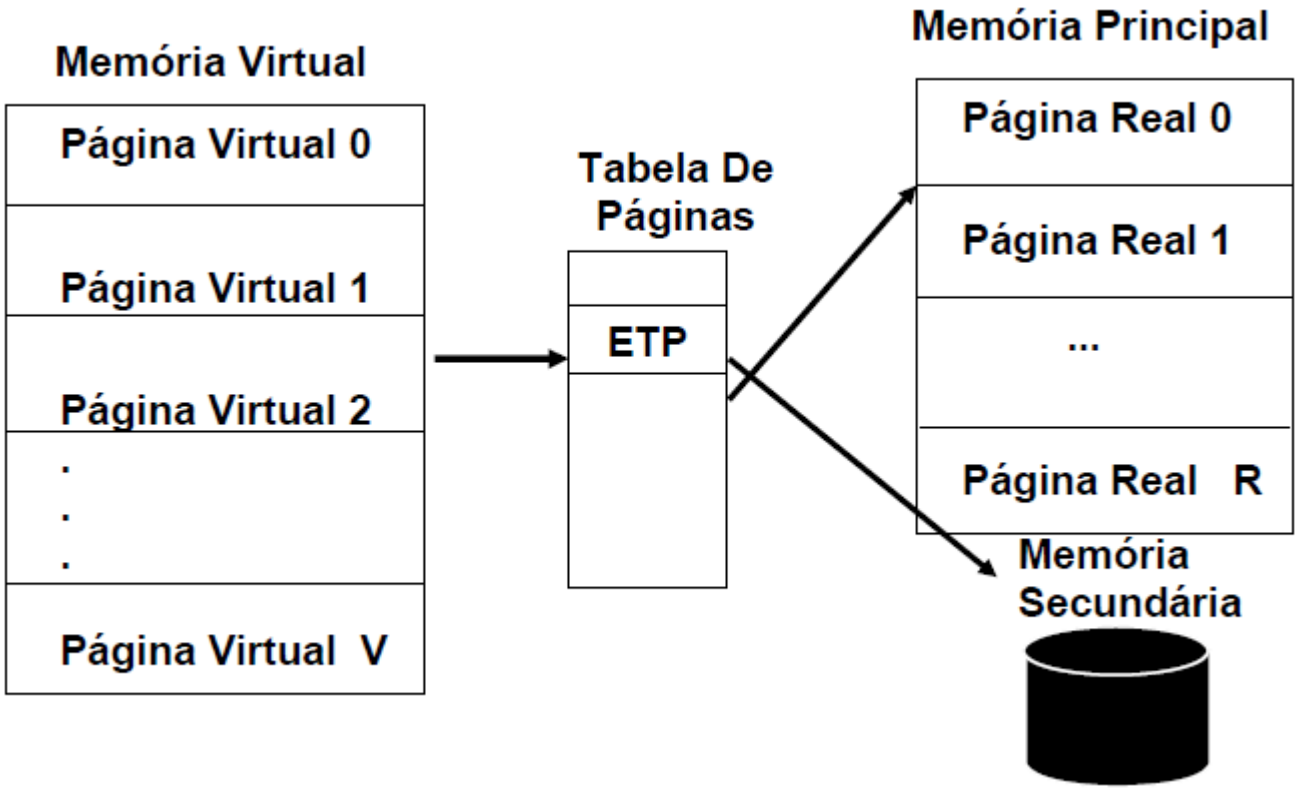
Gerenciamento de Memória Com Swapping Paginação

- Técnica de gerenciamento de memória onde o espaço de endereçamento virtual e o de endereçamento real são divididos em blocos de mesmo tamanho, chamados páginas
- No espaço virtual são denominadas páginas virtuais, enquanto as do espaço real são chamadas de páginas reais ou frames (quadros)
- Endereços gerados por um programa em execução são chamados de endereços vrtuais e formam o espaço de endereçamento virtual
  - Em computadores sem memória virtual o endereço virtual é colocado diretamente no barramento de memória, fazendo com que o endereço físico da memória, equivalente a ele, seja lido ou escrito
  - Quando a memória virtual é usada, os endereços virtuais não vão direto para o barramento da memória, eles são encaminhados à unidade de gerenciamento de memória (MMU), podendo estar dentro da CPU ou ser um chip (ou conjunto de chips) que mapeiam os endereços virtuais em endereços físicos da memória
- Quando um programa é executado, as páginas virtuais são transferidas da memória secundária para a memória principal e colocadas em frames
- Sempre que o programa fizer referência a um endereço virtual, o mecanismo de mapeamento localizará, na tabela de mapeamento, o endereço físico do frame

Gerenciamento de Memória Com Swapping Paginação

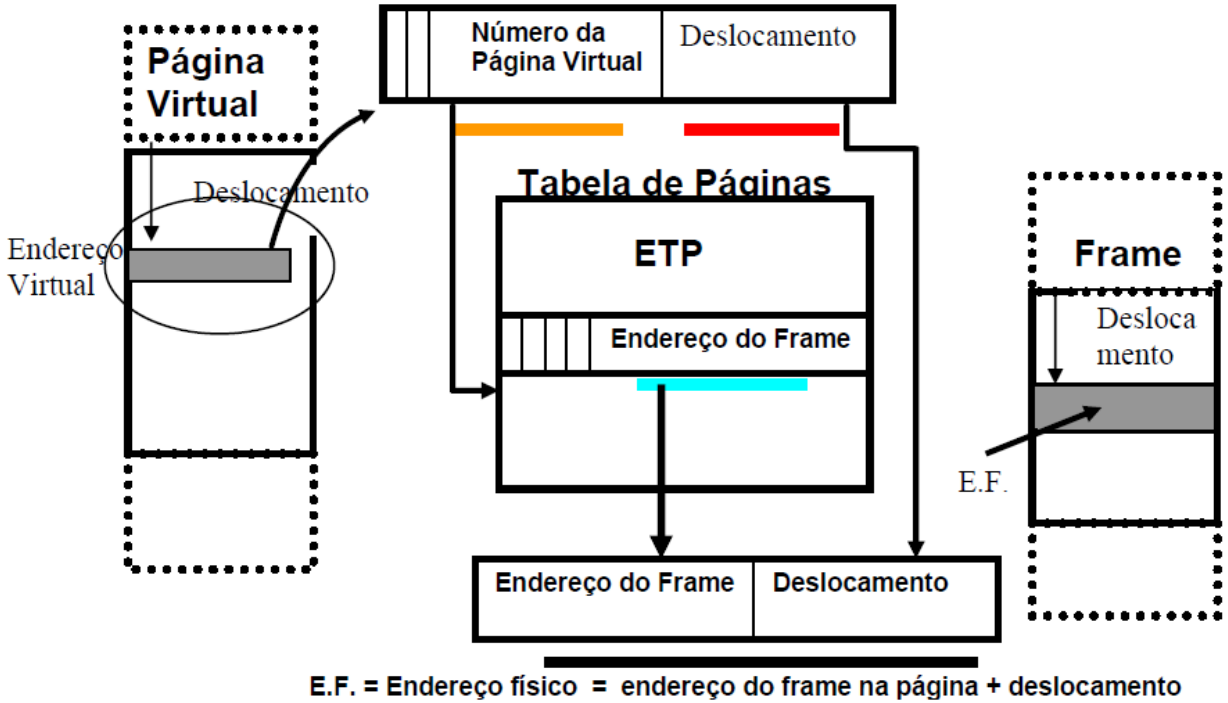


Gerenciamento de Memória Com Swapping Paginação



Gerenciamento de Memória Com Swapping Paginação

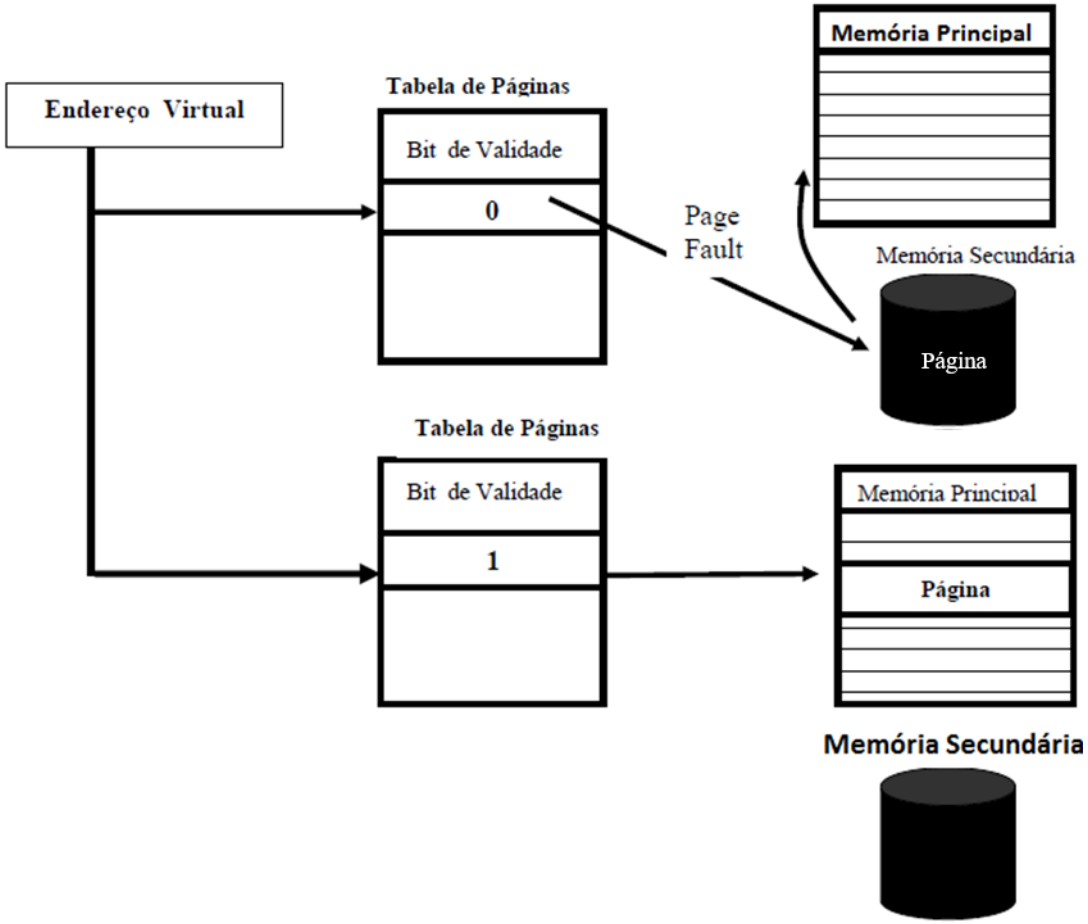
- O endereço virtual é formado pelo número da página virtual e um deslocamento dentro da página.
- O endereço físico é calculado somando-se o endereço do frame localizado na tabela de páginas com o deslocamento contido no endereço vitual



Gerenciamento de Memória Com Swapping Paginação

- **Tipos de Paginação**
- A transferência das páginas de um processo da memória secundária para a principal pode ser:
  - Quando referenciada – Paginação por demanda (Demanding paging):
    - Apenas as partes do programa realmente referenciadas são trazidas para a memória principal
  - Modo especulativo – Paginação antecipada (Antecipatory paging):
    - Pode reduzir o *overhead* na execução de programas. Se o sistema "errar na previsão" recursos desnecessários terão sido gastos
- **Bit de validade**
  - Além da informação sobre a localização da página virtual, a entrada na tabela de páginas possui a informação se a página que contém o endereço referenciado está ou não na memória principal por meio de um bit de validade
    - Bit 0 – página virtual não está na memória principal
    - Bit 1 – Página está localizada na memória principal
  - O sistema sempre verifica se a página que contém o endereço referenciado está ou não na mamória principal
    - Caso não esteja, deve-se fazer a transferência da página da memória secundária para a memória principal.

Gerenciamento de Memória Com Swapping Paginação

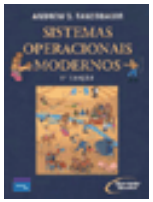


Gerenciamento de Memória Com Swapping Paginação – Relocação de Páginas

- Quando uma falta de páginas (*page fault*) ocorre, o S.O. tem que escolher uma página para remover da memória a fim de dar lugar à que será trazida da memória secundária
  - Se a página a ser removida foi modificada enquanto estava na memória, ela deve ser reescrita na memória secundária para manter a cópia atualizada
  - Se não tiver sido alterada, não é necessária a reescrita
  - A página a ser lida é superposta à retirada
- Apesar de ser possível fazer uma escolha aleatória para dar lugar à página em demanda, o desempenho do sistema será melhorado caso seja esolhida uma página menos usada (referenciada)
  - Se uma página muito usada é removida, ela possivelmente terá de ser trazida de volta em breve, resultado em trabalho adicional
  - Algoritmos eficientes minimizam esse esforço
- O sistema de gerenciamento de memória deve:
  - Decidir que páginas remover quando o conjunto de páginas que devem permanecer na memória, do processo, está no limite e novas páginas são refenciadas
  - Estratégia para páginas em memória que foram modificadas
- Para liberar um quadro (página), o S.O. deve fazer o Swap out da página da memória principal e o Swap in da nova página a ser referenciada
- Quando um processo é criado, as páginas vão para a memória secundária, sendo carregadas para a memória principal conforme as referências acontecem

Gerenciamento de Memória Com Swapping

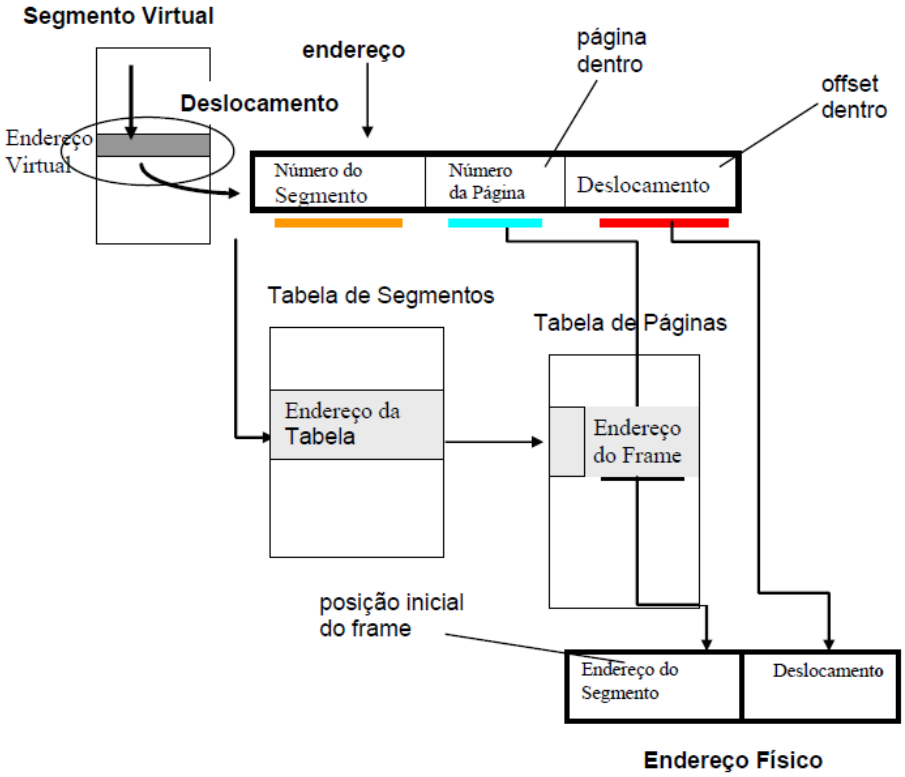
Paginação – Algoritmos de Troca de Páginas



- Aleatória (Random)
  - Escolhe uma página qualque do working set (conjunto de páginas do processo)
  - Consome poucos recursos
  - Raramente utilizada
- First-in-first-out (FIFO)
  - A página primeiro utilizada sairá do working set
  - Fila
  - Razoável ?
    - E em caso de loops (Páginas constantemente referenciadas) ?
    - E em caso de utilitários do S.O. (que não estão no kernel) que são constantemente utilizados ?
  - Pode acarretar muitas faltas de páginas
- Least recently used (LRU)
  - Página usada menos recentemente (Página há mais tempo sem ser referenciada)
  - Boa estratégia
  - Pode levar muito tempo na operação
  - Qual a definição de recente ?
- Not recently used (NRU)
  - Página não usada recentemente (Semelhente ao LRU)
  - Flag define a utilização da página, começando todas com 0
    - Conforme utilização, a flag é modificada
  - Precisa de tempo para definer as páginas utilizadas
- Least frequenly used
  - Página menos referenciada é selecionada
  - Mantido um Contador de referências
    - A de menor contagem é selecionada
  - Privilegia páginas bastante utilizadas
  - Parece boa estratégia, mas pode impor às últimas páginas a entrar no working set, o destino de serem escolhidas para sair

Gerenciamento de Memória
Com Swapping
Segmentação com Paginação

- Sistemas que implementam segmentação com paginação permitem a divisão lógica dos programas em segmentos e, por sua vez, cada segmento é dividido fisicamente em páginas



Gerenciamento de Memória Com Swapping Compartilhamento de Memória

- Em sistemas multiprogramáveis é comum usuários utilizarem certos programas simultaneamente (código reentrante), o que evita que várias cópias de um mesmo programa ocupem a memória desnecessariamente
- Exemplo:
  - Utilitários do sistema
  - Compiladores
  - Editores de Texto
  - Etc
- Em sistemas que implementam memória virtual é bastante simples o compartilhamento de código e dados entre vários processos. Para isso, basta que as entradas de tabelas de páginas apontem para as mesmas páginas na memória virtual
- É possível reduzir o número de programas na memória e aumentar o número de usuários compartilhando o mesmo recurso

Gerenciamento de Memória
Com Swapping
Compartilhamento de Memória