



Gerência de Memória

Prof. Edwar Saliba Júnior
Março de 2007



Memorial...

- **Programas em execução e dados acessados por ele devem estar na memória principal (ou pelo menos parte dele);**
- **Porém, a memória principal é pequena e cara;**
- **Processos em execução compartilham espaços diferentes de uma mesma memória principal;**
- **O gerenciamento da utilização dessa memória principal é uma das principais tarefas do S.O., sobretudo em sistemas multiprogramáveis.**



Mas o que é gerenciar memória?

- **Reduzir ao máximo a necessidade de acesso à memória secundária durante a execução de um processo;**
- **Maximizar o número de processos residentes na memória principal;**
- **Permitir a criação e execução de novos processos, mesmo que não haja mais espaço livre na memória principal (*swapping*: transferência temporária de processos para a memória secundária).**



Mas o que é gerenciar memória?

- **Permitir a execução de processos maiores que a memória principal (*overlay* e memória virtual);**
- **Garantir integridade da área ocupada por cada processo, inclusive pela execução do S.O.;**
- **Permitir o compartilhamento de dados entre processos de forma segura.**

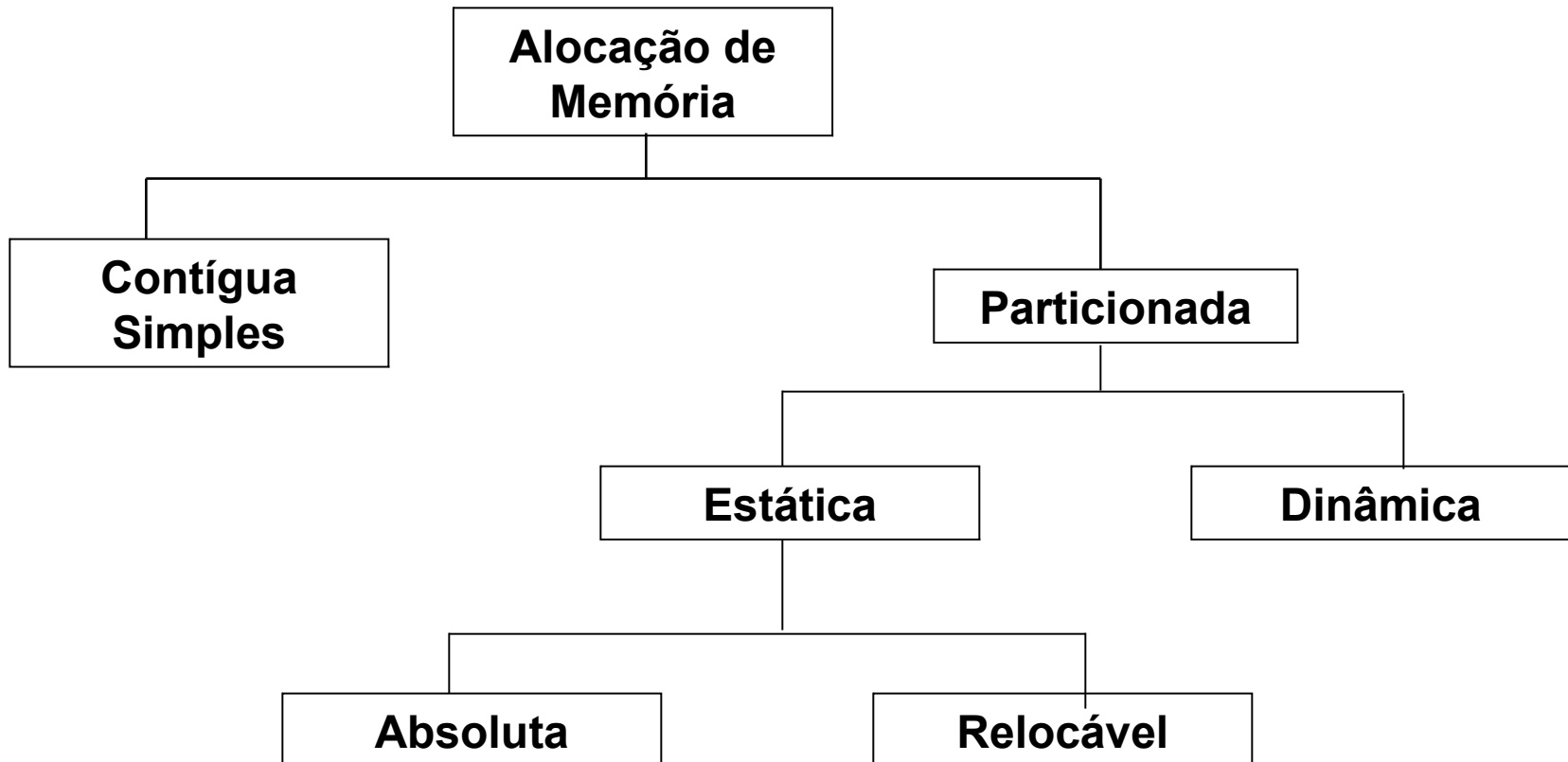


É bom saber:

- **Ciclo tradicional de execução de uma instrução:**
 - Busca instrução na memória;
 - Busca operandos na memória;
 - Executa a instrução e
 - Armazena o resultado na memória.



Tipos de Alocação





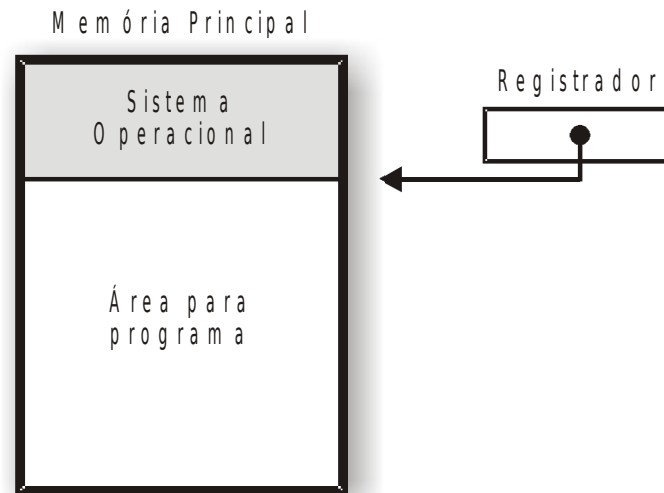
Alocação Contígua Simples

- **Divisão da memória em duas partes: uma para o S.O. e uma para processos do usuário;**
- **Necessidade de um registrador contendo limite de endereço da memória para proteção da área do sistema (registrador de fronteira);**
- **Sempre que o programa do usuário faz referência a um endereço de memória, ocorre a comparação com esse registrador para saber se o endereço acessado faz parte da área do usuário.**



Alocação Contígua Simples

- **Desvantagens:**
 - **Somente um processo de usuário a cada momento (sistemas monoprogramáveis);**
 - **Subutilização da memória principal.**





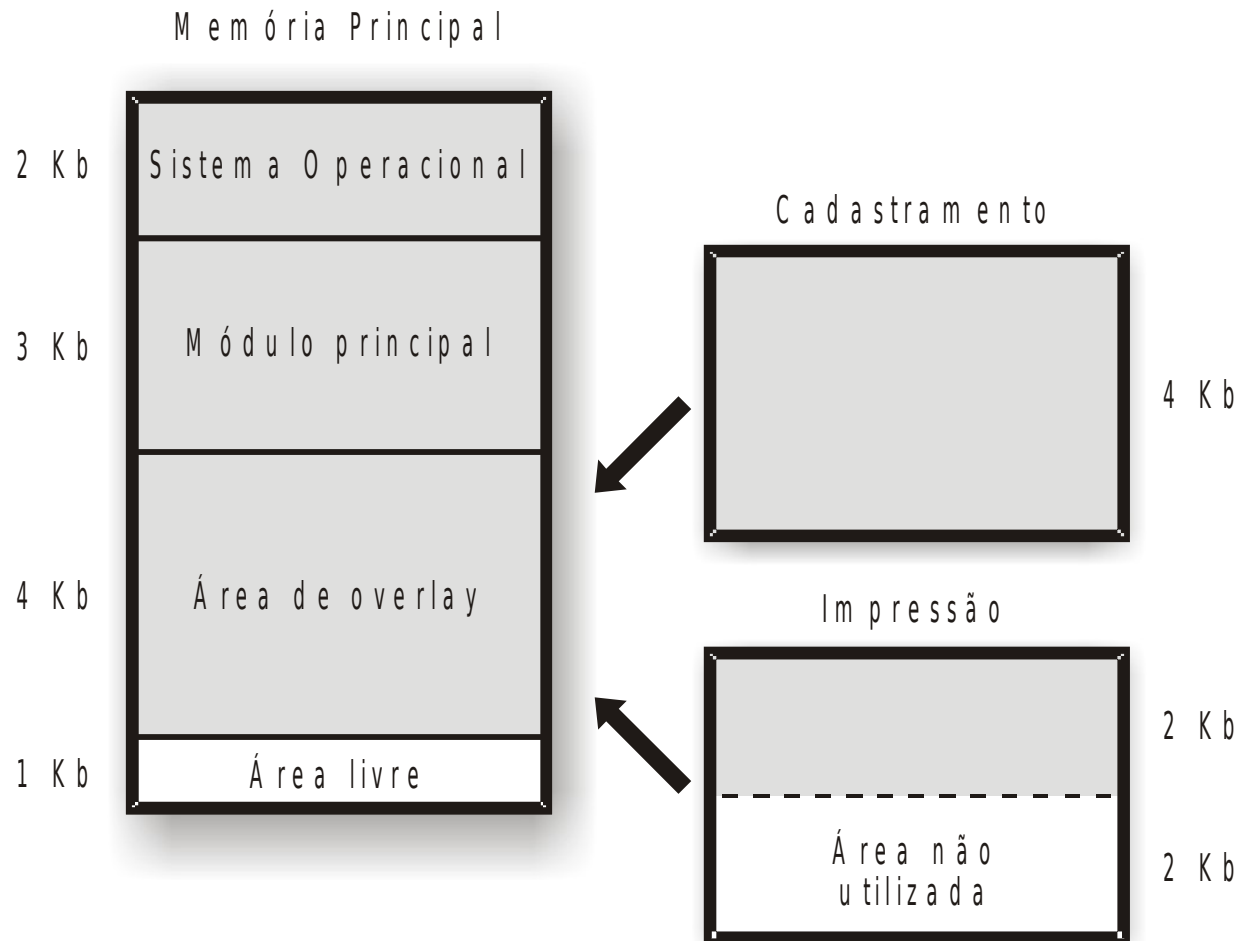
Alocação Contígua Simples

- **Overlay:**
 - A definição das áreas de *overlay* é função do programador (terá sempre o tamanho do maior módulo);
 - Risco: transferência excessiva de módulos entre memória principal e memória secundária;
 - Permite ao programador expandir os limites da memória principal;
 - Se o processo do usuário é maior que o tamanho da memória principal, ele é dividido em módulos;
 - A área da memória compartilhada por esses módulos é chamada de área de *overlay*.



Fundamentos de Sistemas Operacionais

Overlay





Alocação Particionada

- **Diversos processos na memória principal ao mesmo tempo;**
- **Subdivide-se em:**
 - **Estática e**
 - **Dinâmica.**



Alocação Particionada Estática

- **Estática:**
 - Divisão da área do usuário em partes fixas;
 - Os programas, muitas vezes, não ocupam toda a partição onde estão sendo executados (problema da fragmentação interna);
 - Tamanho das partições definido na inicialização do sistema;
 - Utilização da tabela de partições.



Alocação Particionada Estática

- **Pode ser:**
 - **Absoluta:**
 - Um programa só pode ser carregado em uma partição específica, ou seja, a partir de um endereço de memória especificado no próprio código;
 - **Relocável:**
 - As referências à memória são relativas: posição inicial da partição + deslocamento;
 - Na tabela de partições é informado, também, a posição inicial de cada partição e *flag* indicando se a partição está em uso;
 - Dois registradores para indicar limites permitidos para referência à memória, pelo programa em execução.

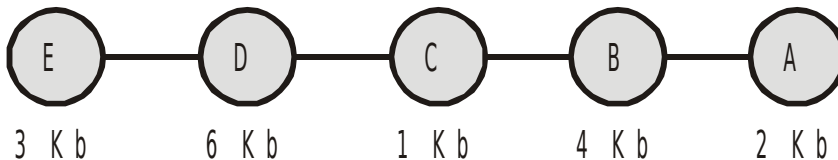


Alocação Particionada Estática

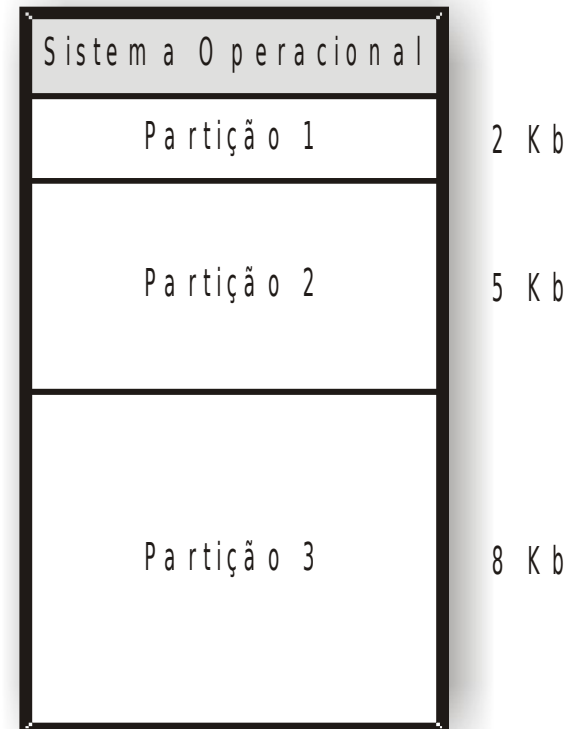
Tabela de partições

Partição	Tamanho
1	2 Kb
2	5 Kb
3	8 Kb

Programas a serem executados:

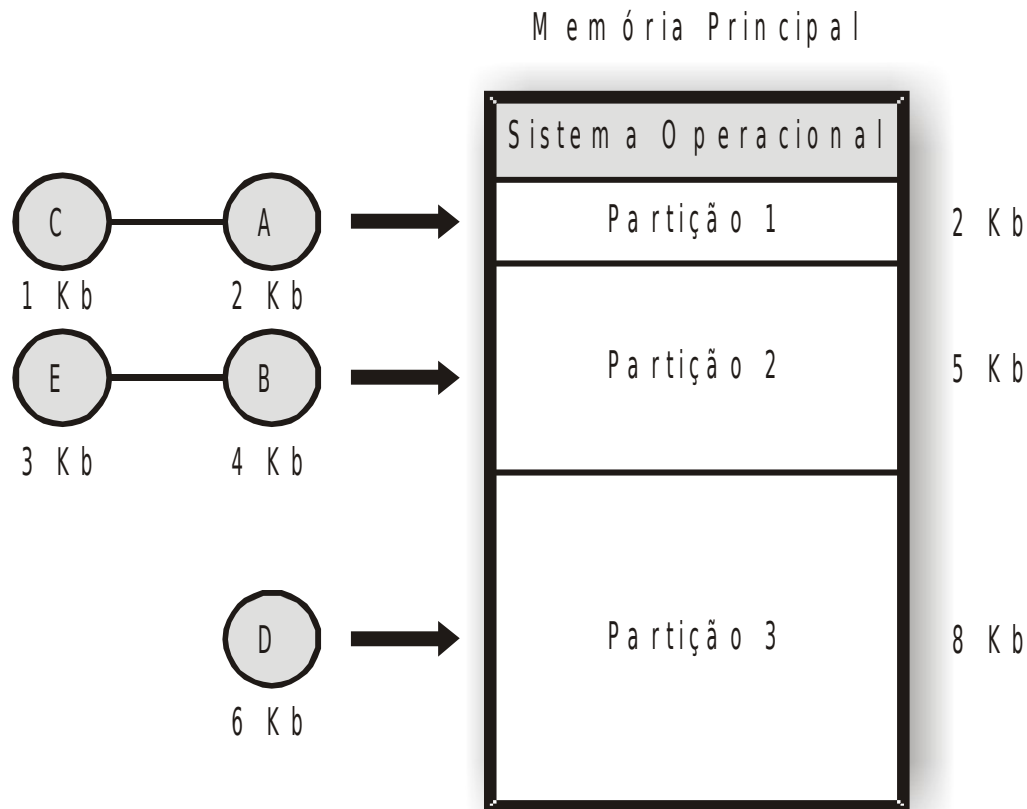


Memória Principal



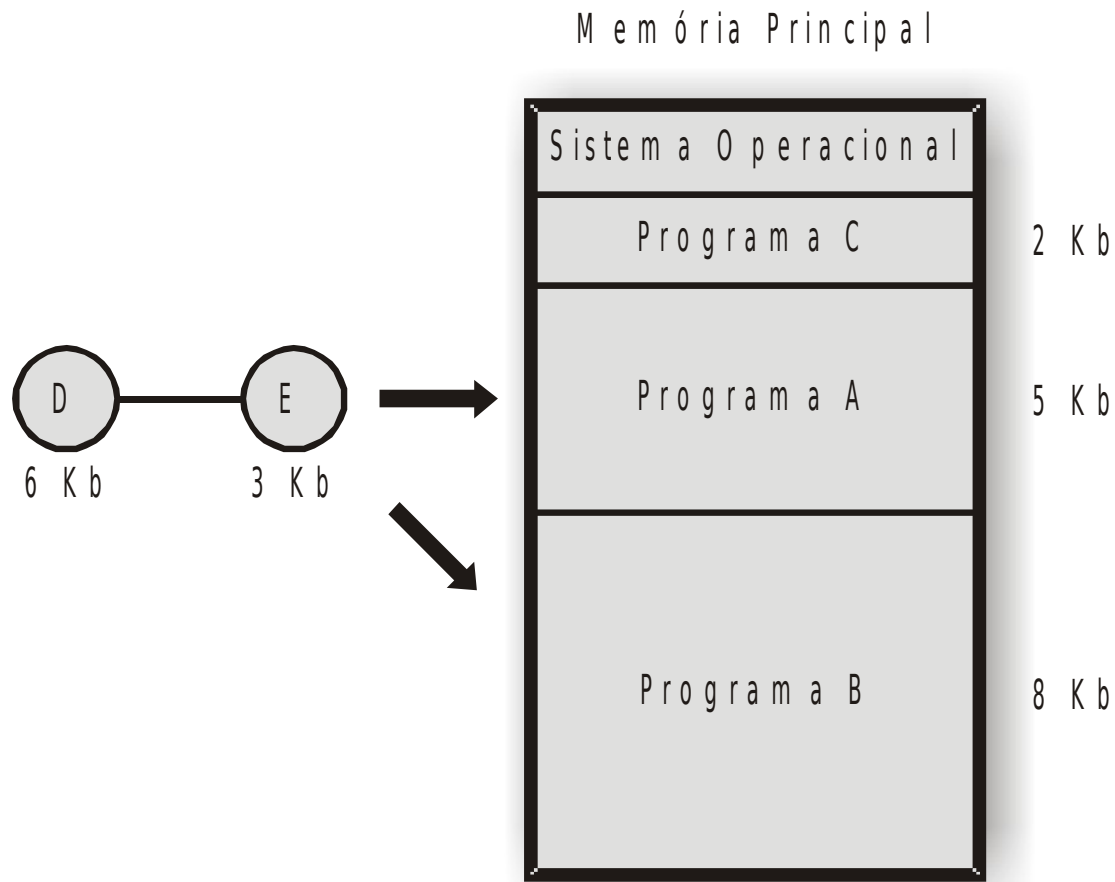


Alocação Particionada Estática Absoluta





Alocação Particionada Estática Relocável



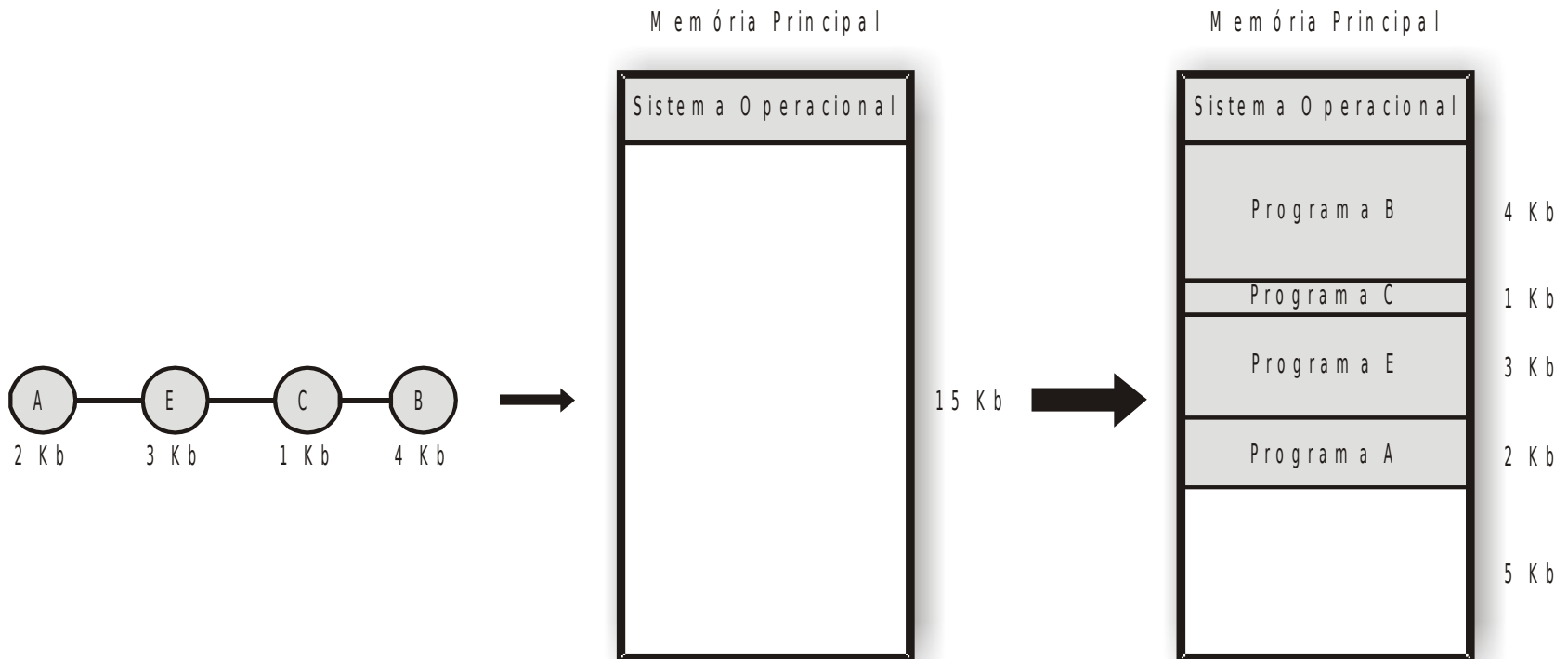


Alocação Particionada Dinâmica

- **Objetivo:** reduzir o problema da fragmentação interna;
- **Cada programa utiliza somente o espaço necessário;**
- **Problema:** fragmentação externa (várias partições livres pequenas);
- **Possíveis soluções:**
 - Reunir os espaços livres adjacentes;
 - Reunir as partições ocupadas, criando uma partição livre única;
 - Algoritmo complexo -> consumo de recursos do sistema;
 - Recebe o nome de alocação particionada dinâmica com relocação.

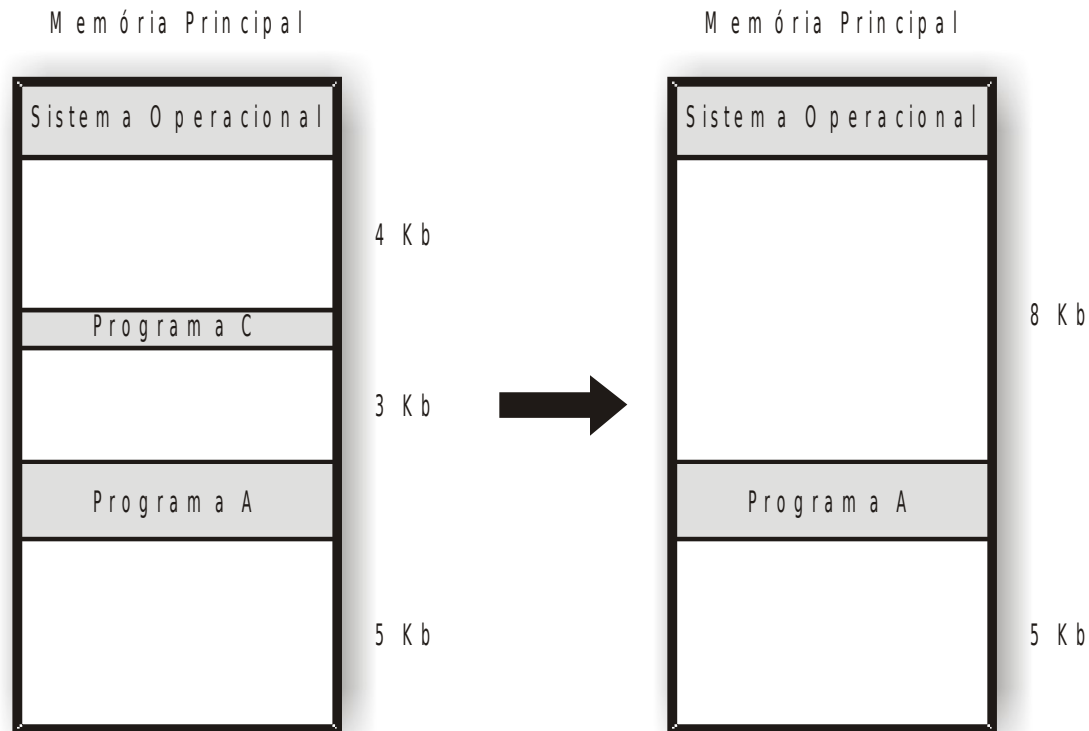


Alocação Particionada Dinâmica



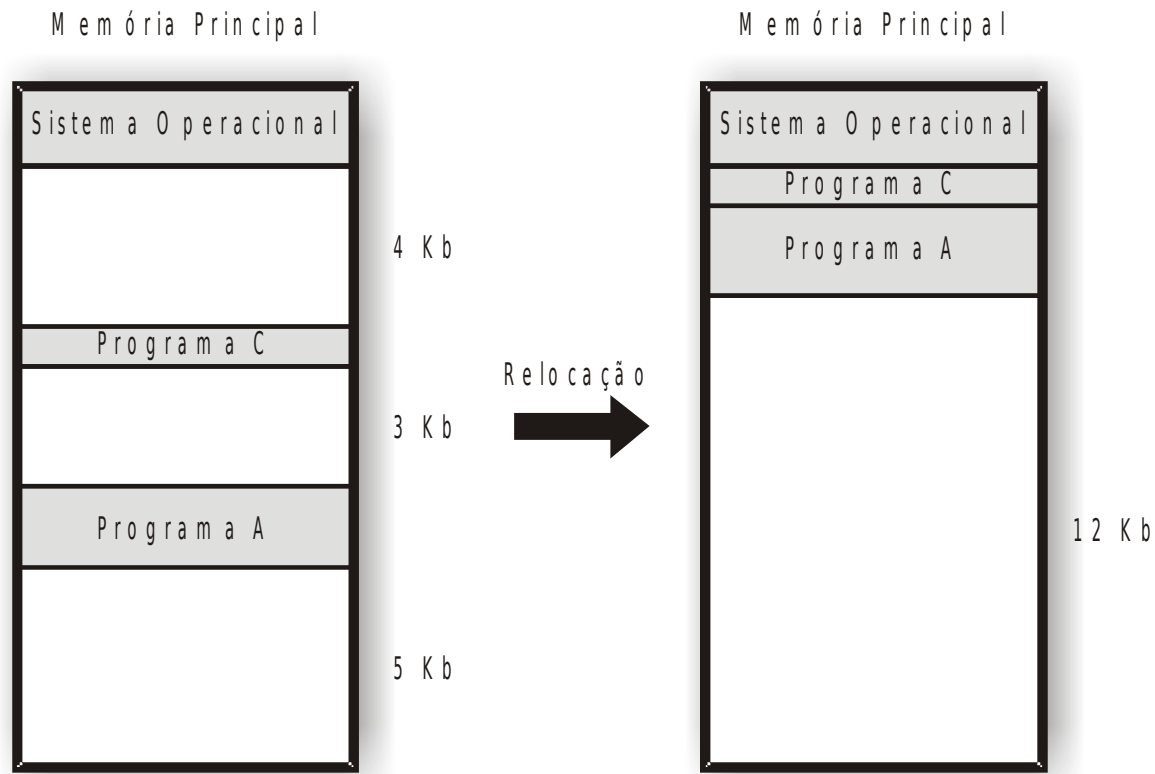


Alocação Particionada Dinâmica: 1ª Solução





Alocação Particionada Dinâmica: 2ª Solução





Selecionando uma partição para o programa

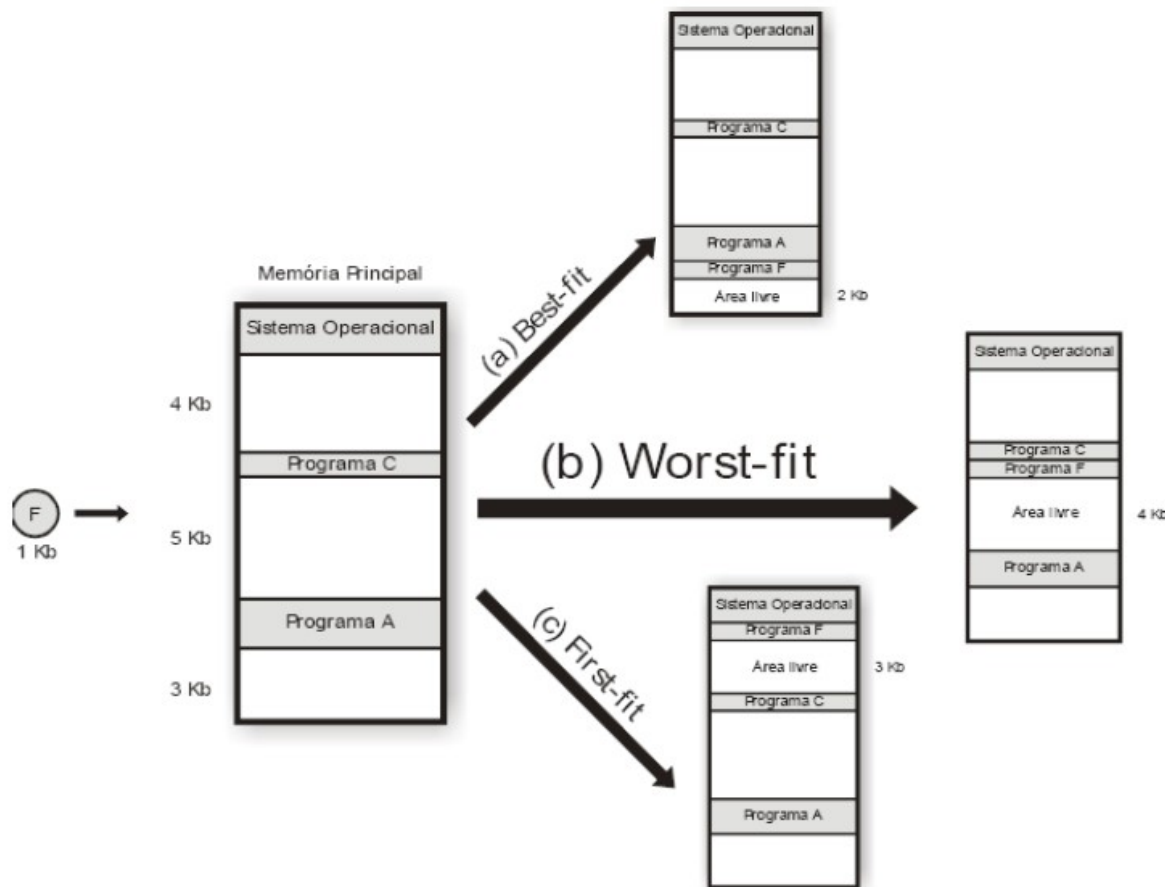
Esta seleção baseia-se em uma lista de áreas livres:

- **Best-fit:** é escolhida a partição livre que resultar em menor espaço sem utilização;
 - Desvantagem: surgimento de pequenas áreas não contíguas;
- **Worst-fit:** é escolhida a partição que resultar em maior espaço livre;
- **First-fit:** é escolhida a primeira partição livre com tamanho suficiente;
 - Estratégia mais simples (menor consumo de recursos).



Fundamentos de Sistemas Operacionais

Exemplo





Fundamentos de Sistemas Operacionais

Swapping

- Técnica utilizada para execução de processos quando não existir partição livre com tamanho suficiente;
- Retira temporariamente da memória principal processos que não estão sendo executados (em espera ou pronto), colocando-os em disco;
- Deve ser retirado da memória principal o processo com menor chance de ser executado (menor prioridade);
- É necessário que o *loader* faça a relocação a cada *swap in*;
- Problema: elevado custo das operações de I/O;
- A técnica de *swapping* é utilizada no gerenciamento da memória virtual.



Bibliografia

- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**, 3^a Ed., Rio de Janeiro: LTC Editora, 2002.
- SILVA, Guilherme Baião S. *Slides da disciplina de Sistemas Operacionais de Arquitetura Fechada*. Faculdade INED, 2005.



Overlay

- Seção de um programa projetada para residir em um dispositivo de armazenamento designado, como um disco, e para ser carregada na memória quando necessário, em geral substituindo um ou mais *overlays* que já estejam na memória. O uso de *overlays* permite que programas grandes sejam armazenados em uma área limitada da memória, o que resulta, entretanto, em uma redução da velocidade de execução. (Fonte: <http://www.netpedia.com.br/MostraTermo.php?TermID=4883> acesso em: 17 abr. 2007)
- Na computação gráfica, posicionar uma imagem gráfica sobre outra. Na área de vídeo, posicionar uma imagem gráfica gerada em um computador sobre sinais de vídeo, ao vivo ou gravados. (Fonte: <http://www.netpedia.com.br/MostraTermo.php?TermID=4884> acesso em: 17 abr. 2007)

