Danton Cavalcanti Franco Junior
falecom@dantonjr.com.br

Gerenciamento de Memória

- Diversos processos dividem o processador, para que o chaveamento entre eles seja rápido eles devem estar na memória.
- O **gerente de memória** do sistema operacional deve prover mecanismos para oferecer segurança e eficiência.

Gerenciamento de Memória

- Nos sistemas multiprogramáveis é um controle extremamente crítico.
- A maior preocupação dos projetistas é desenvolver SOs que ocupem pouca memória e ao mesmo tempo otimizem sua utilização.

Gerenciamento de Memória

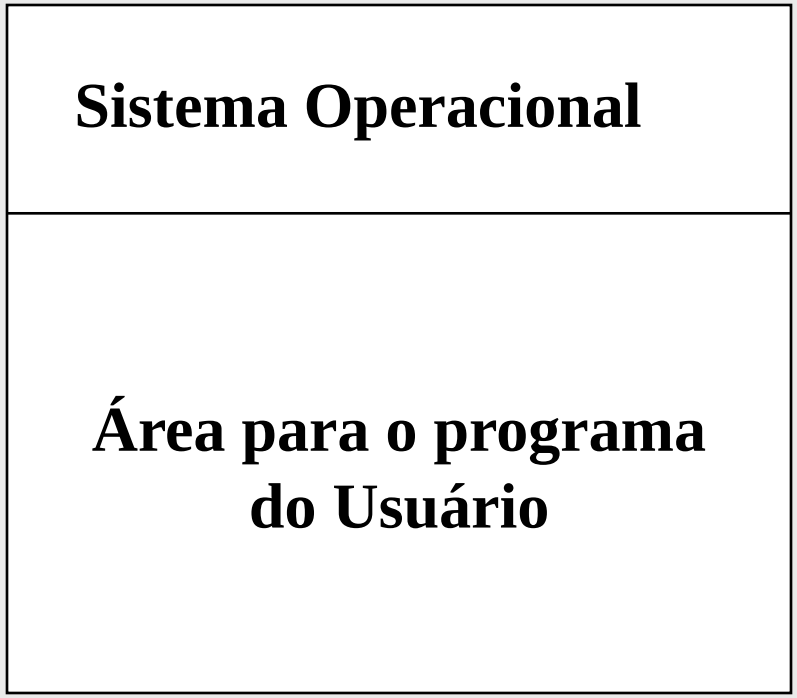
- **Não confundir:**
  - **Memória Principal:** Onde residem todos os programas e dados que serão utilizados pelo processador.
  - **Memória Secundária:** É um meio permanente, mais abundante e de baixo custo para armazenar os dados.

Gerenciamento de Memória

- **Alocação Contígua Simples:**
  - Foi implementada nos primeiros SO (ainda presente em sistemas monoprogramáveis).
  - A memória principal é dividida em 2 partes: uma para o SO e outra para os programas do usuário.
  - O programador não deve ultrapassar o espaço de memória disponível (o tamanho da memória principal menos o espaço ocupado pelo SO).

Memória Principal

- Alocação Contígua Simples

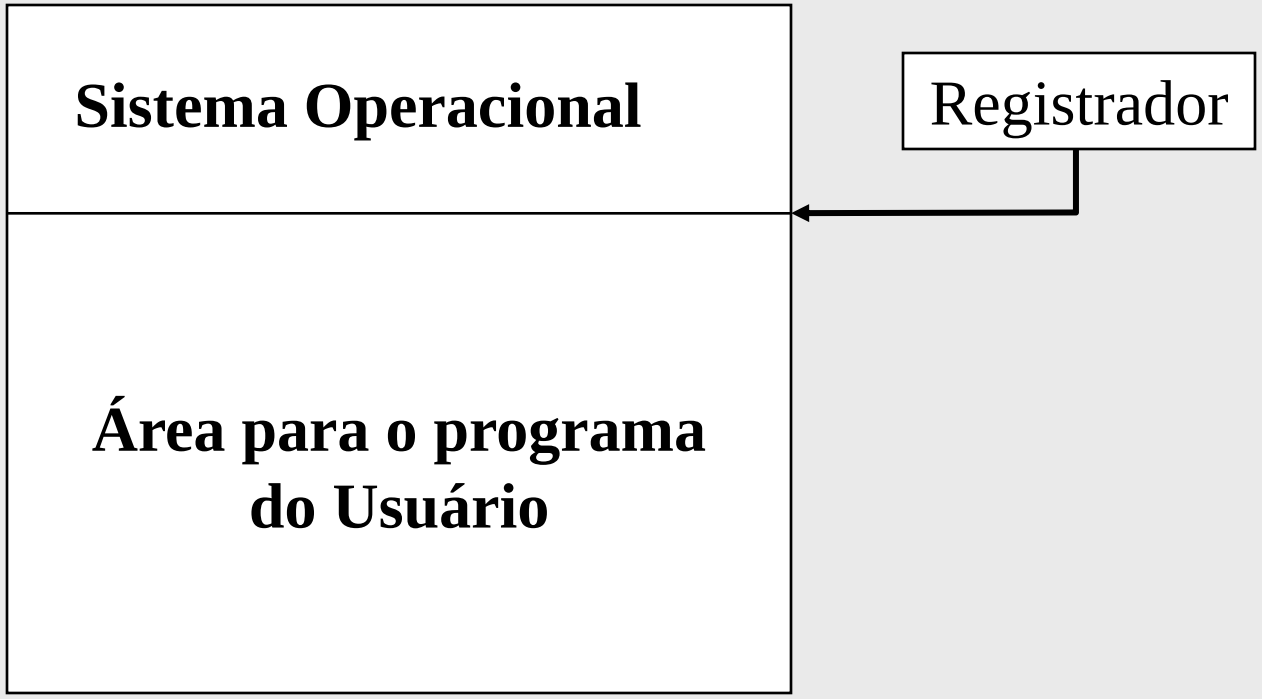


Gerenciamento de Memória

- **Alocação Contígua Simples:**
  - O usuário tem total acesso a memória, inclusive a área do SO.
  - É possível ao usuário destruir o SO.
  - Implementação de uma proteção através de um registrador.

Memória Principal

- Alocação Contígua Simples – proteção por registrador



Memória Principal

- **Alocação Contígua Simples:**
  - Há sempre o controle do endereço de memória.
  - **Access violation**: Quando o programa do usuário é cancelado por estar referenciando uma área de memória delimitada para o uso Sistema Operacional.

Memória Principal

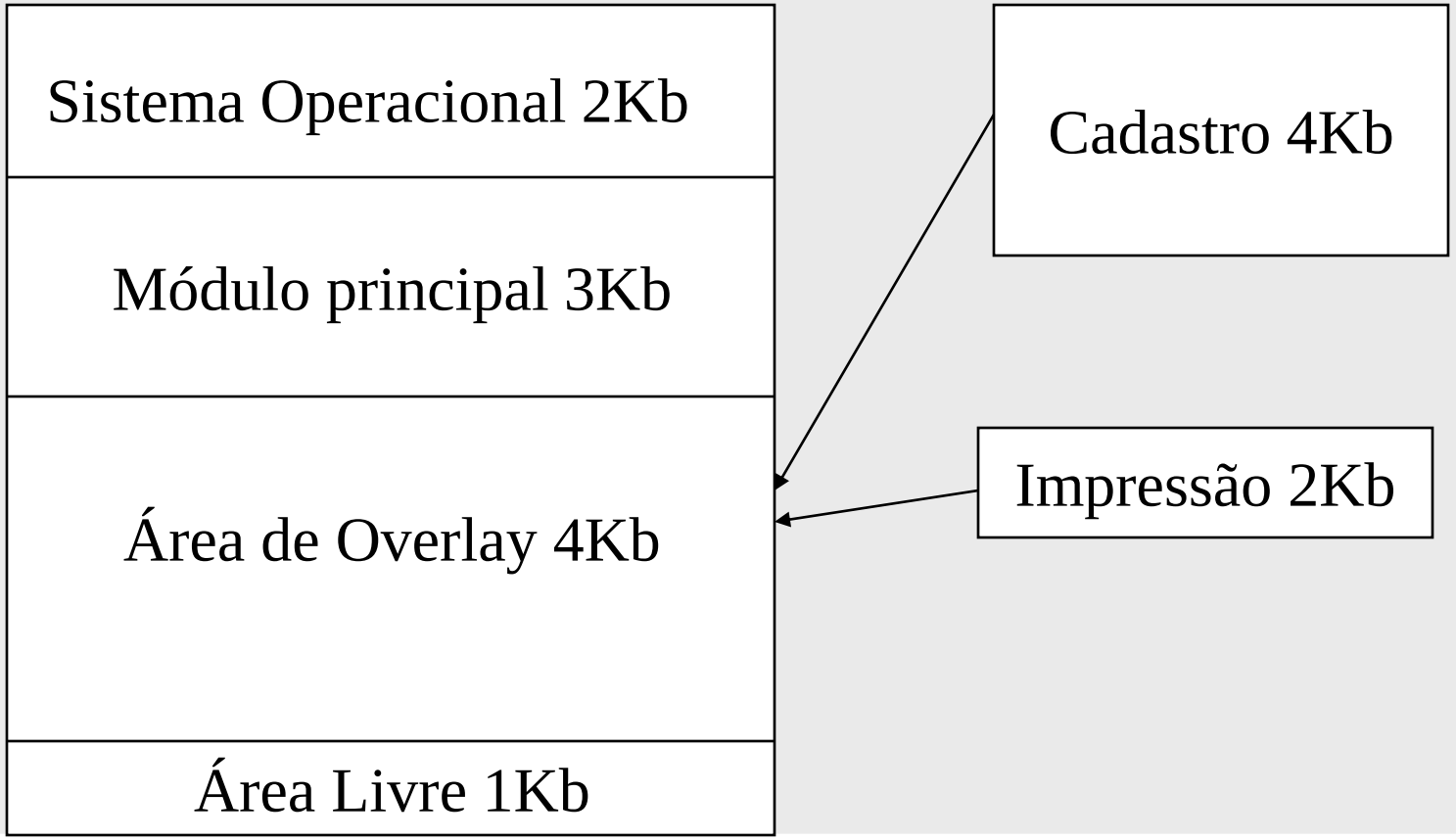
- **Alocação Contígua Simples:**
  - Problemas de limitação do acesso a memória.
  - Solução overlay (sobreposição).

Memória Principal

- **Técnica de Overlay:** Divide o programa em módulos que rodam independentes, utilizando uma mesma área de memória.
  - Quando um módulo estiver na memória o outro não precisa estar.
  - A área de overlay é definida pelo maior overlay disponível.
  - Cuidado apenas com a transferência disco/memória.

Memória Principal

- Overlay



Memória Principal

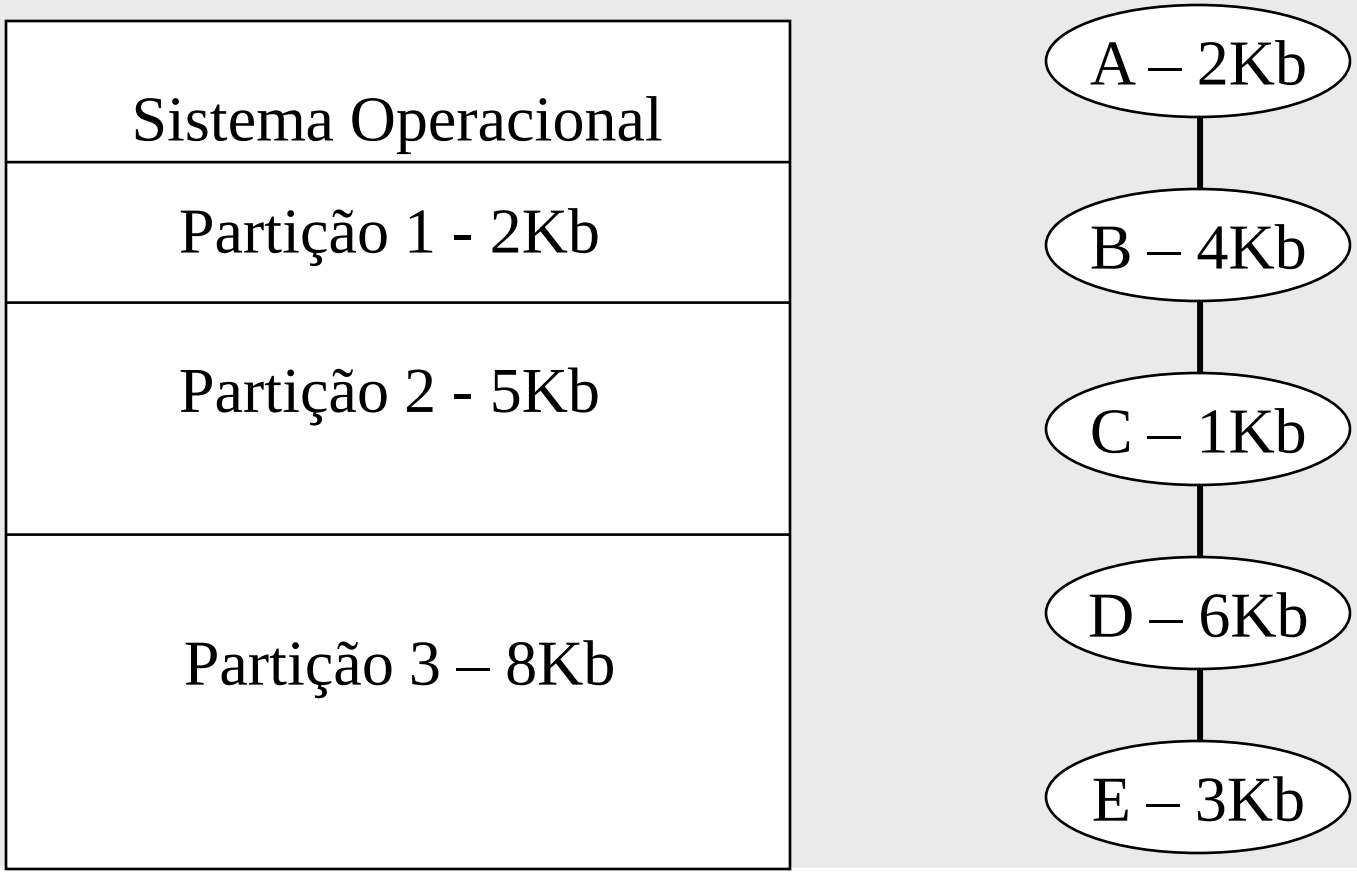
- **Alocação Particionada:**
  - Sistemas monoprogramáveis o processador fica ocioso, com isso o surgimento da multiprogramação.
  - Permitir que vários programas estejam na memória ao mesmo tempo.

Memória Principal

- **Alocação Particionada Estática:** A memória é dividida em pedaços de tamanhos fixos (partições).
  - Tamanho estabelecido na fase de inicialização.
  - Sempre que era necessário mudar o tamanho da partição o sistema deveria ser desativado e reinicializado.

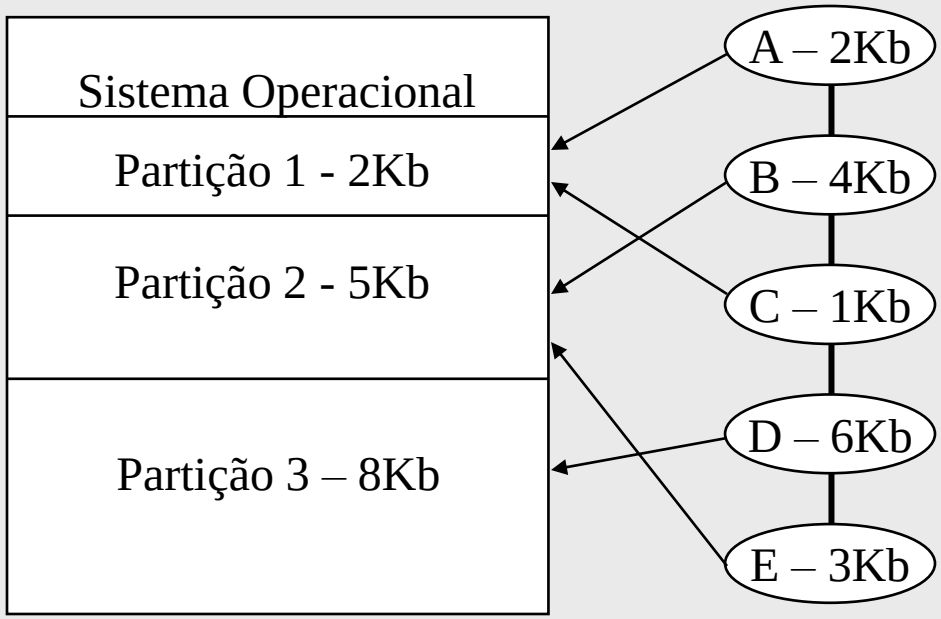
Memória Principal

- Alocação Particionada Estática



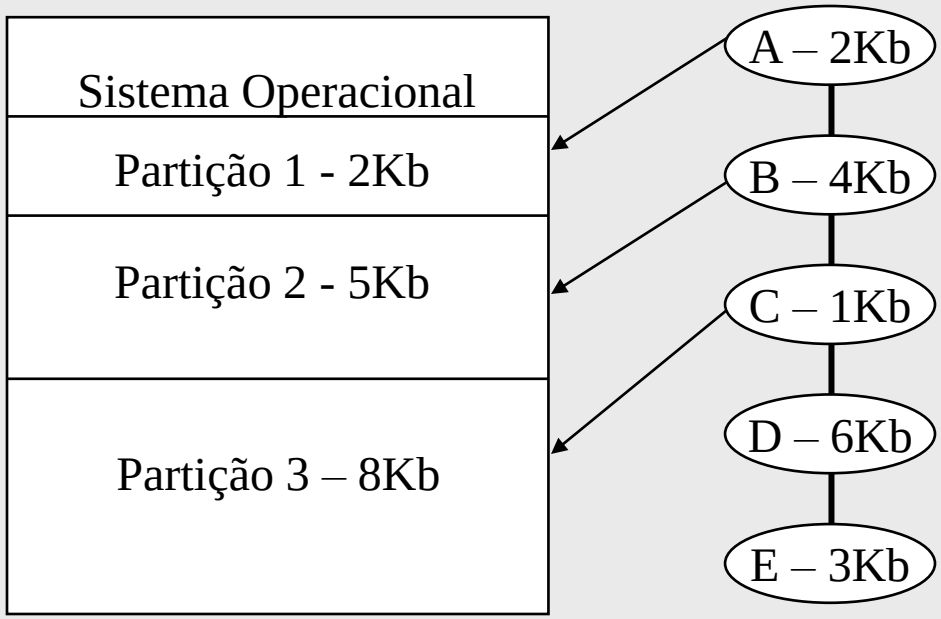
Memória Principal

- Alocação Particionada Estática Absoluta
  - Geração apenas de código absluto (compilador/montador)
  - Se A e B rodam, C espera.



Memória Principal

- Alocação Particionada Estática Relocável
  - Geração código relocável (compilador/montador)
  - Qualquer partição que suporte o programa permite que ele execute.



Memória Principal

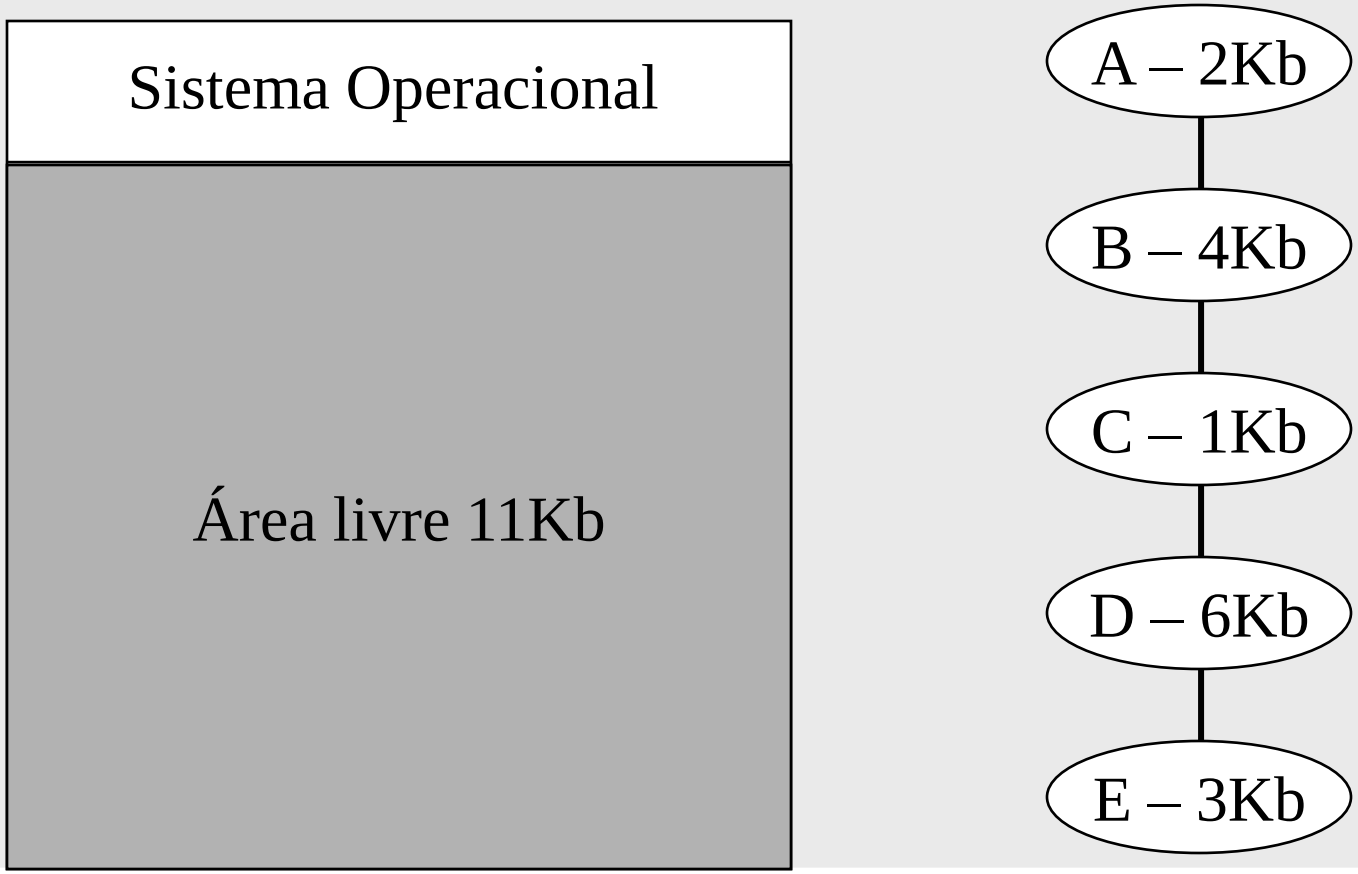
- **Alocação Particionada Estática**
  - A proteção baseia-se em dois registradores (inferior e superior).
  - Há uma tabela indicando a partição, seu tamanho e se está ou não disponível.
  - Programas geralmente não ocupam toda a partição.
  - Programas grandes esperam partições que o suportem (mesmo partições adjacentes somadas que suportem seu tamanho não são usadas = **fragmentação** – espaços perdidos).

Memória Principal

- **Alocação Particionada Dinâmica (variável):** A memória é dividida em pedaços de tamanhos variáveis.
  - Eliminado o conceito de partição fixa.
  - Cada programa utiliza o espaço que necessita para executar.
  - Ocorrência de fragmentação na saída (programas vão deixando espaços cada vez menores a medida que vão terminando, impedindo o ingresso de novos programas).

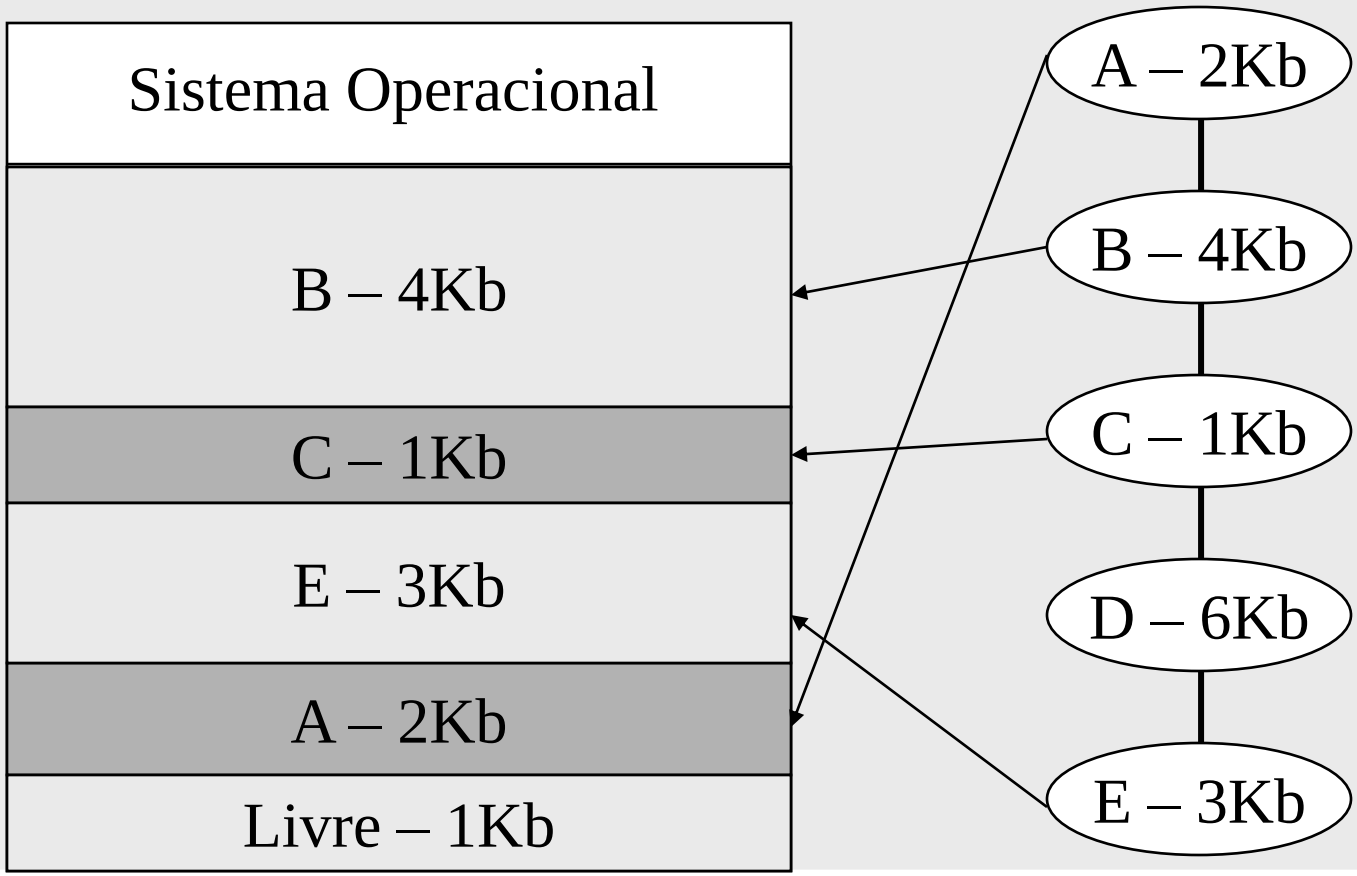
Memória Principal

- Alocação Particionada Estática



Memória Principal

- Alocação Particionada Estática

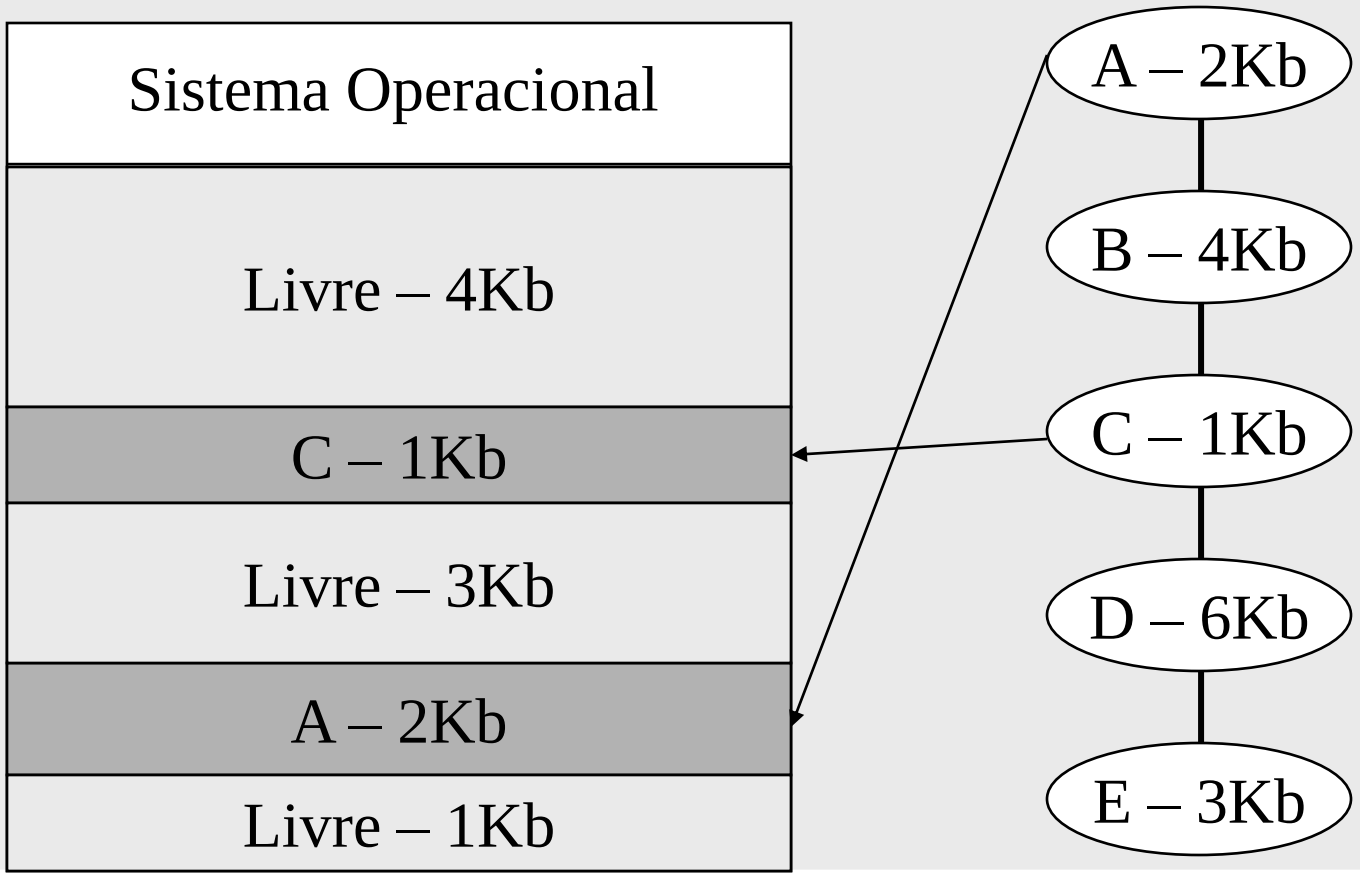


Memória Principal

- **Alocação Particionada Dinâmica**
  - Se B e E terminarem, D não pode rodar pois não há espaço livre (consecutivo).
  - Solução:
    - Juntar partes adjacentes a partir do momento que os programas terminam de executar.
    - Relocar os programas (Alocação Dinâmica com Relocação) – compactação. Consome muito recurso e é muito complexo.

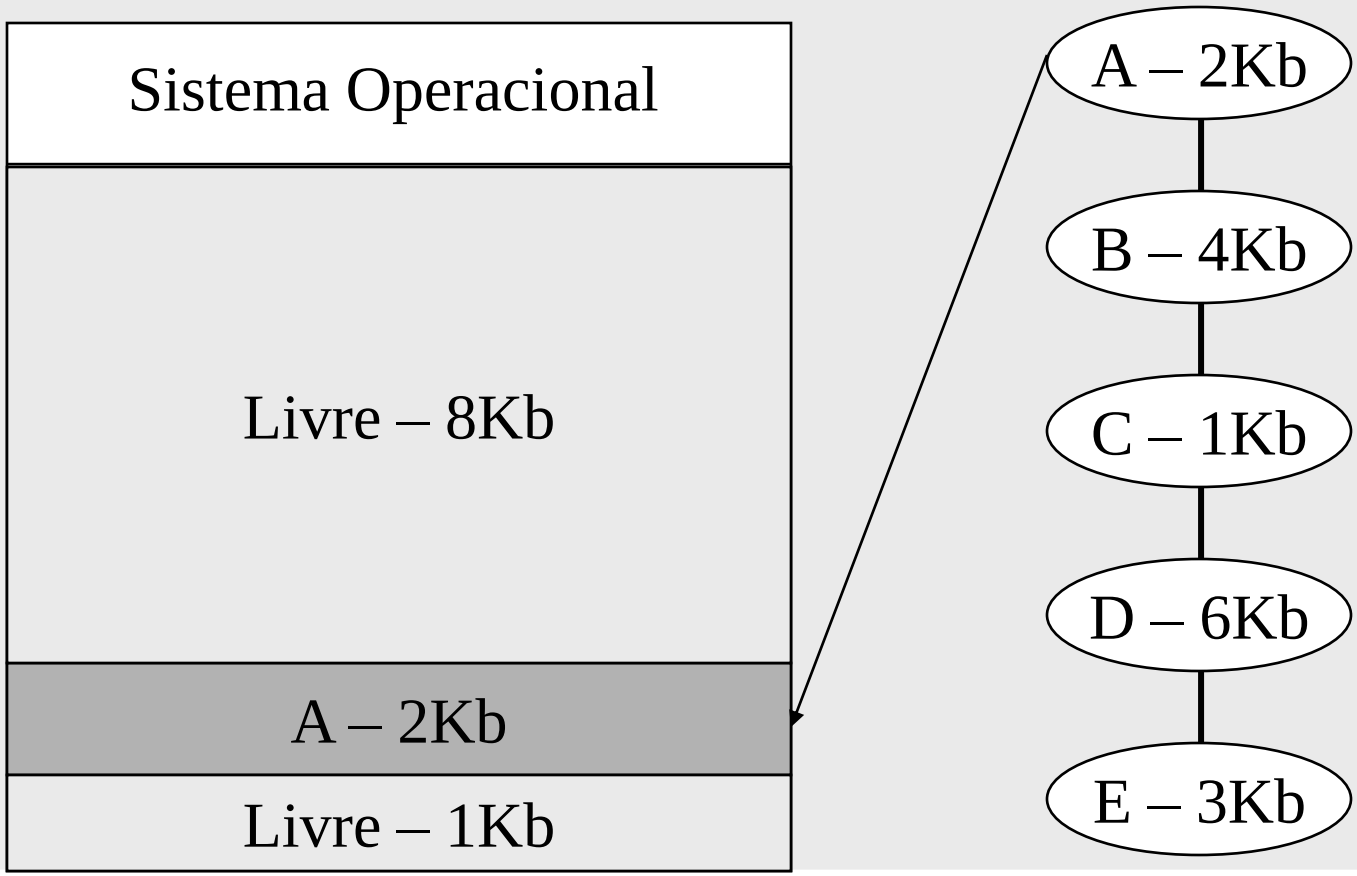
Memória Principal

- Alocação Particionada Estática



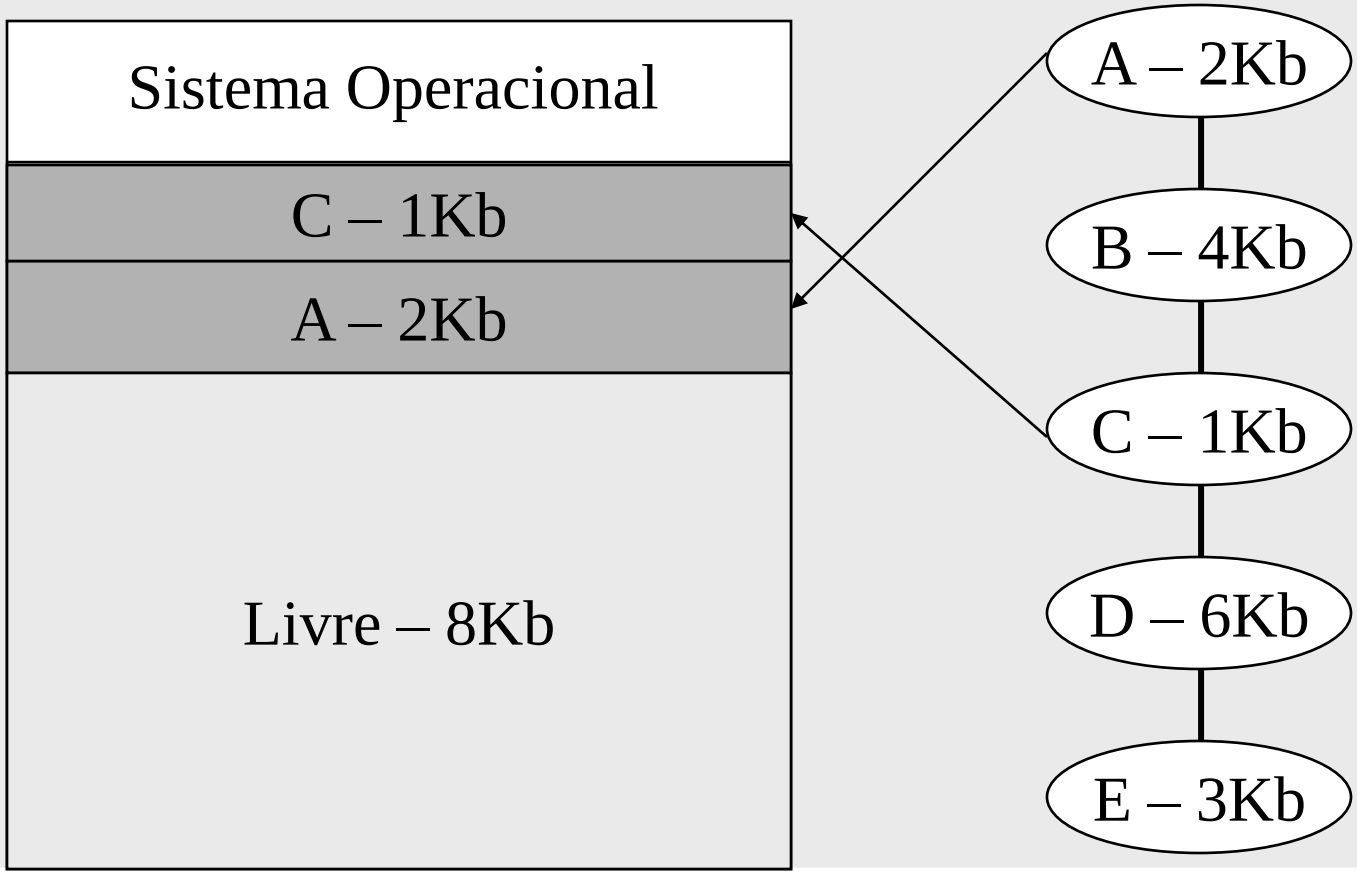
Memória Principal

- Alocação Particionada Estática (partes adjacentes)



Memória Principal

- Alocação Particionada Estática (relocação)

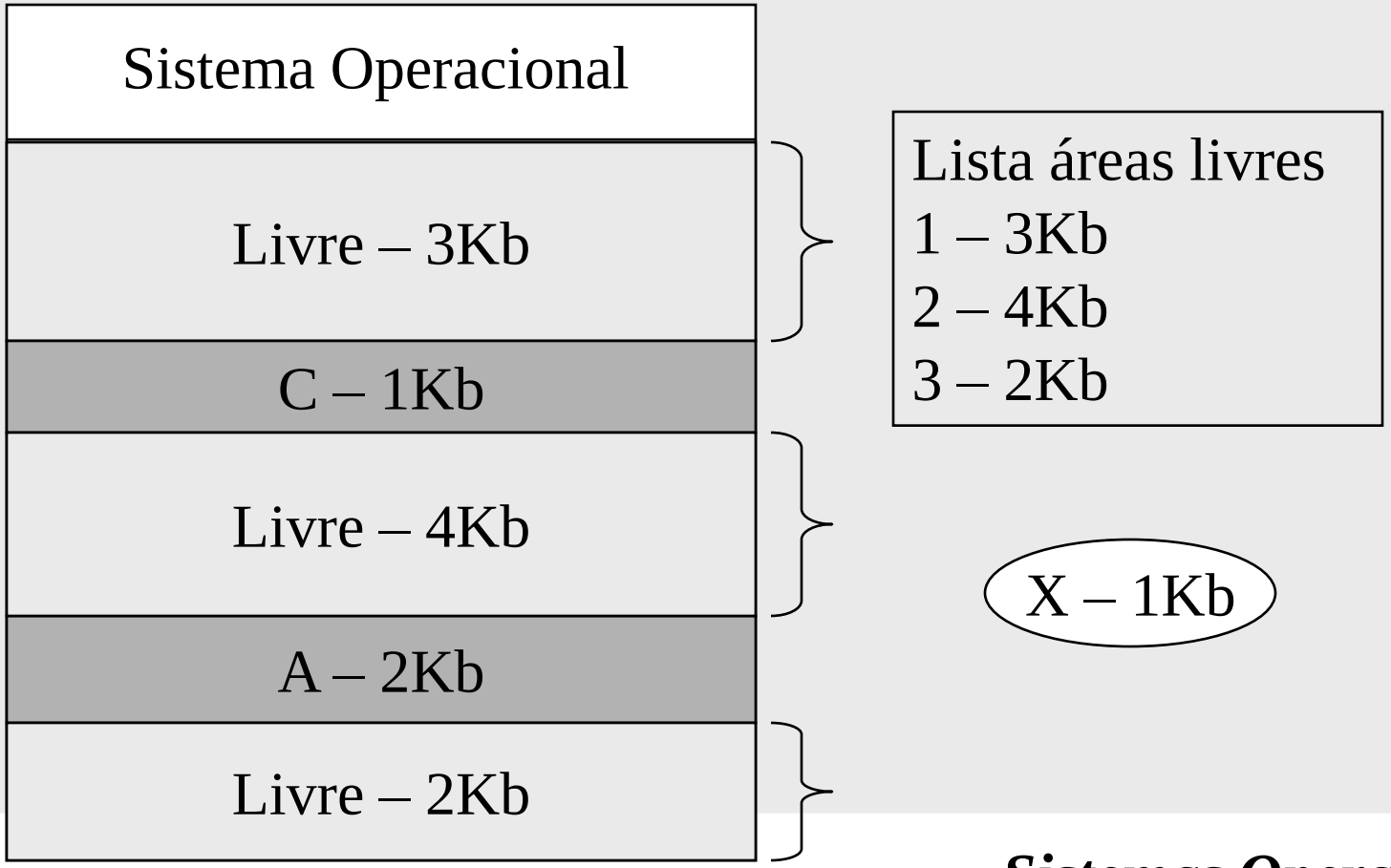


Memória Principal

- **Estratégias para Escolha da Partição**
  - Devem principalmente evitar ou diminuir o problema da fragmentação.
  - O fator mais importante na escolha da estratégia é o tamanho do programa.
  - Basicamente implementam-se três estratégias: Best-fit, Worst-fit e First-fit.

Memória Principal

- Independente do mecanismo, sempre há uma lista indicando as partições livres.

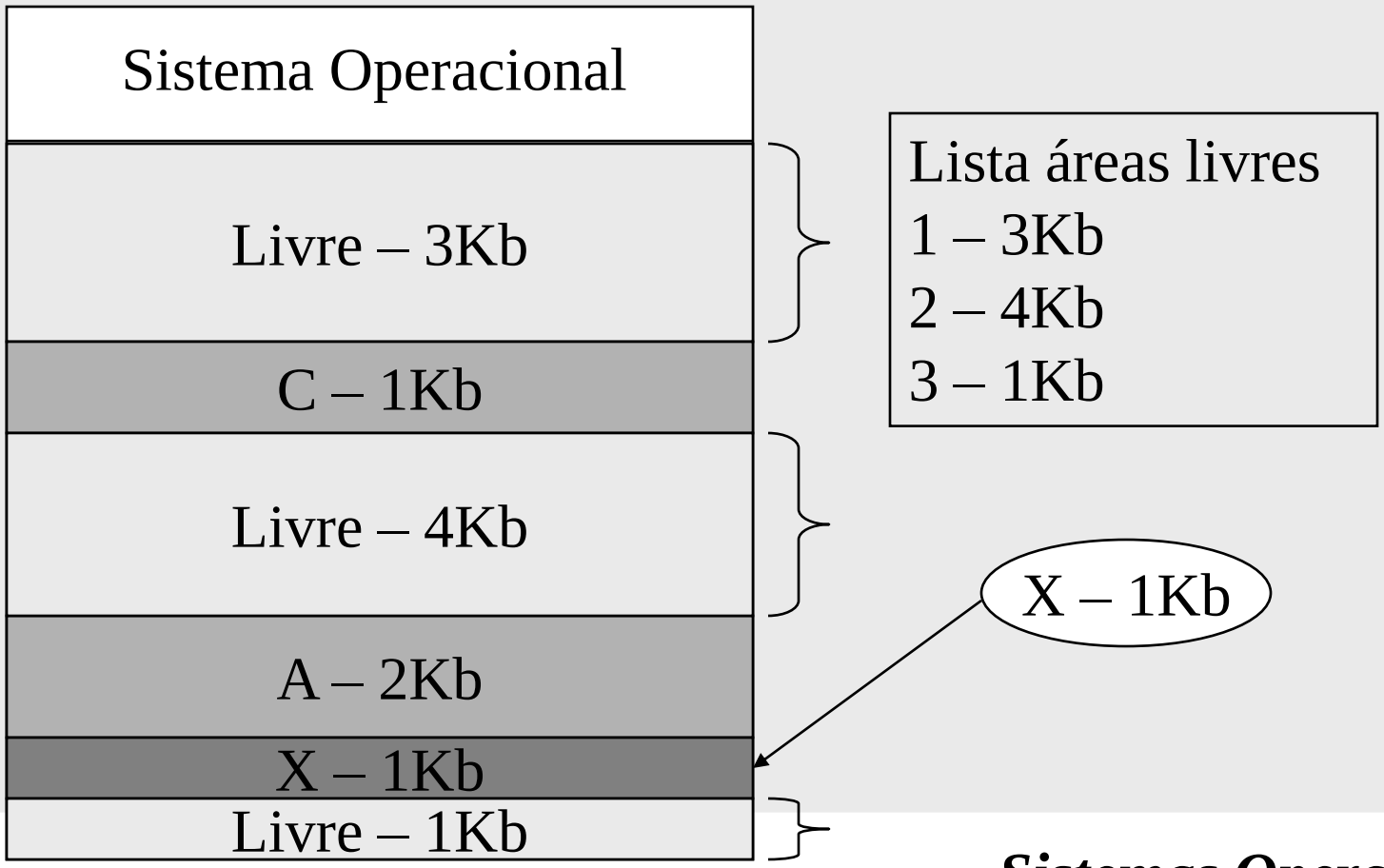


Memória Principal

- **Estratégia Best-fit:**
  - Escolhe a melhor partição, ou seja, aquela em que o programa deixa o menor espaço sem utilização.
  - A lista de partições livres é ordenada por tamanho.
  - Há uma forte tendência de que cada vez mais a memória fique com pequenas áreas não contíguas, aumentando o problema da fragmentação.

Memória Principal

- **Best-fit**

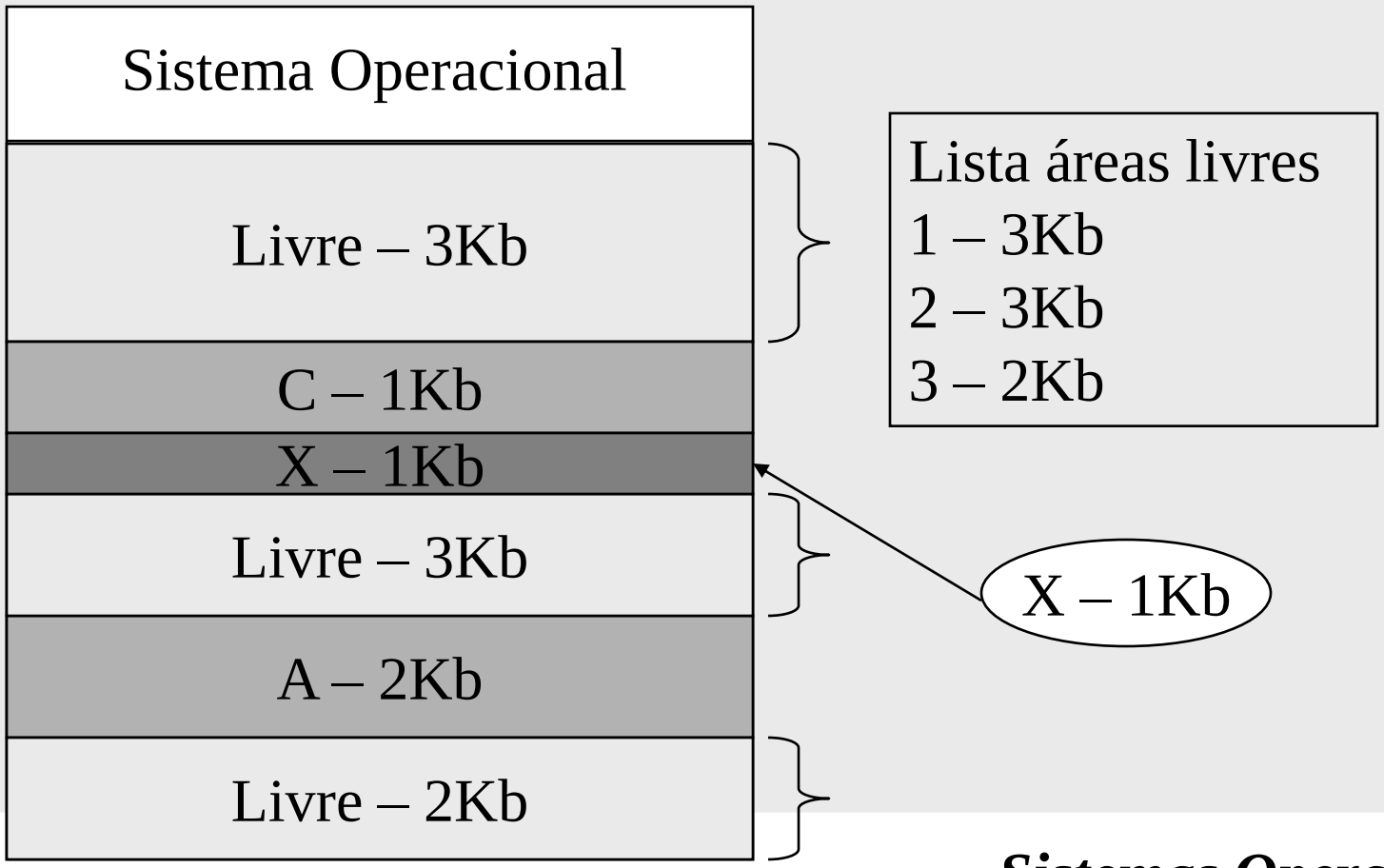


Memória Principal

- **Estratégia Worst-fit:**
  - Nesta estratégia, o mecanismo escolhe a pior partição, ou seja, aquela em que o programa deixa o maior espaço sem utilização.
  - Como conseqüência, deixa espaços livres maiores o que permite um maior número de programas utilizando a memória, diminuindo a fragmentação.

Memória Principal

- **Worst-fit**



Memória Principal

- **Estratégia First-fit:**
  - O mecanismo escolhe a primeira partição livre de tamanho suficiente para carregar o programa.
  - A lista é ordenada por endereços crescentes.
  - É a mais rápida.
  - Há uma grande chance de se obter uma grande partição livre nos endereços mais altos da memória.

Memória Principal

- **First-fit**