INSTITUTO FEDERAL
PARANÁ

LABORATÓRIO DE SISTEMAS OPERACIONAIS

PROFª. M.Sc. JULIANA HOFFMANN QUINONEZ BENACCHIO

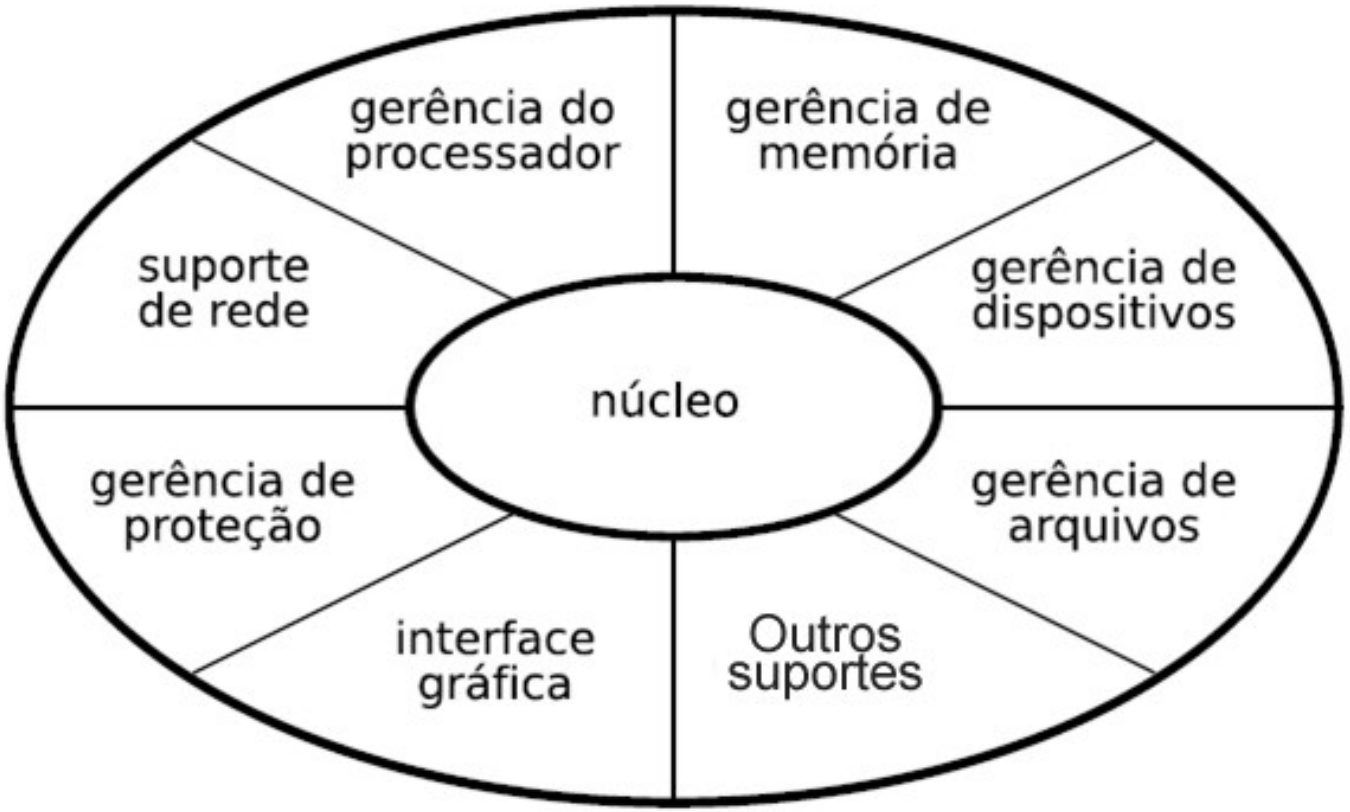
Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- Como gerenciador de recursos, um sistema operacional é composto de vários módulos com funcionalidades distintas.
- Cada módulo é responsável por gerenciar uma particularidade do sistema

Sistema Operacional

INSTITUTO FEDERAL
PARANÁ



Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- **Gerência do processador:** visa distribuir a capacidade de processamento (uso de CPU) de forma justa.

- **Gerência de memória:** tem como função fornecer, a cada aplicação, um espaço próprio de memória, independente e isolado das demais aplicações.

Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- **Gerência de dispositivos:** cada periférico do computador possui suas peculiaridades; logo, temos vários dispositivos diferentes, mas com problemas comuns. Pen-drives, discos IDE e SCSI são dispositivos diferentes, em essência iguais, já que basta um endereço ou área de locação para que um determinado dado seja buscado. Logo é possível criar uma abstração única de acesso.

Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- **Gerência de arquivos:** construída sobre a gerência de dispositivos, possibilita criar abstrações de arquivos e diretórios.

- **Gerência de proteção:** políticas de acesso e uso do sistema operacional. Permite a definição de usuários, grupos de usuários e registro de recursos por usuários.

Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- **Interface gráfica:** a interação com o usuário se faz necessária, assim a maioria dos sistemas operacionais apresentam “telas”, nas quais pode-se informar ao sistema operacional qual a operação que ele deverá fazer.

Sistema Operacional

INSTITUTO FEDERAL
PARANÁ

- **Suporte de rede:** a comunicação em rede é, atualmente, essencial ao mundo dos computadores. Assim, o gerenciamento dessas comunicações se faz necessário e é realizado sob uma abstração do sistema operacional sobre os dispositivos físicos, como placas de redes ou modems.

Sistema Operacional

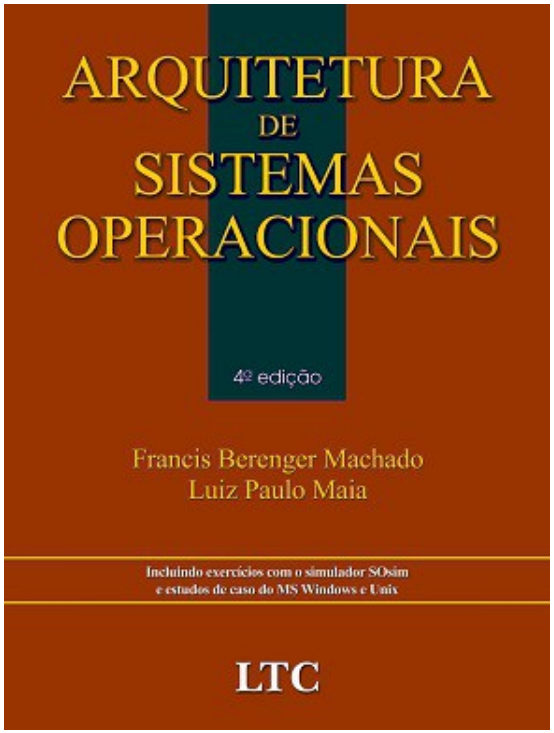
INSTITUTO FEDERAL
PARANÁ

- **Outros suportes:** há sistemas operacionais para os mais diversos usos. Sistemas de uso geral (que permitem ao usuário ouvir músicas, navegar na Internet, editar textos) normalmente têm mais recursos para gerência de multimídia Sistemas de uso específico (que possibilitam o controle de uma usina hidrelétrica, por exemplo) possuem outras características específicas, tais como tempo de resposta ou suporte a um hardware especial.

Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

Conteúdo retirado do livro



Arquitetura de Sistemas Operacionais
Francis Berenger Machado
Luiz Paulo Maia
4ª. edição
Editora LTC

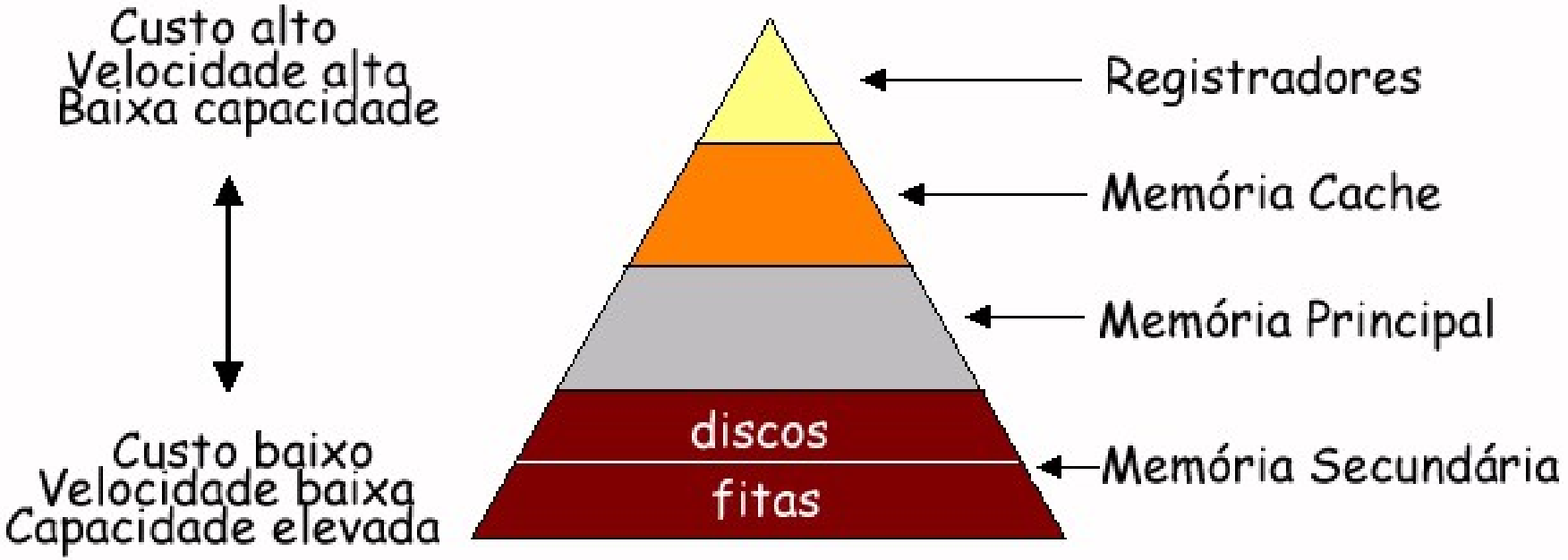
Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

- Historicamente, a memória principal sempre foi vista como um recurso escasso e caro.
- Mesmo atualmente, com a redução de custo e aumento da capacidade da memória principal, seu gerenciamento é um dos fatores mais importantes no projeto de SO.
- Uma das maiores preocupações sempre foi desenvolver SOs que não ocupassem muito espaço de memória e, ao mesmo tempo, otimizassem a utilização dos recursos computacionais.

Hierarquia de Memórias

- As três características principais da memória custo, capacidade, tempo de acesso são conflitantes



Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

- Controlar as partes das memórias que estão em uso ou não;
- Alocar e desalocar memórias aos processos quando necessário;
- Gerenciar a troca entre memória principal e o disco quando a memória principal é muito pequena.

Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

- Geralmente, os programas são armazenadas em memórias secundárias, por serem meios não-voláteis, abundantes e de baixo custo.
- No entanto, o processador sempre executa instruções localizadas na memória principal, então o SO deve transferir dados da memória secundária para a principal com o objetivo de reduzir a quantidade de entrada e saída – I/O.

Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

- A gerência de memória deve tentar manter na memória principal o maior número possível de processos residentes, permitindo maximizar o compartilhamento do processador e demais recursos computacionais.
- Mesmo na ausência de espaço livre, o sistema deve permitir que novos processos sejam aceitos e executados.

Gerenciamento de Memória

INSTITUTO FEDERAL
PARANÁ

- Técnica de *swapping*, que é a transferência temporária de processos residentes na memória principal para a memória secundária.
- O sistema também deve permitir a execução de programas que sejam maiores que a memória física disponível, implementando técnicas de *overlay* e memória virtual.

Gerenciamento de Memória

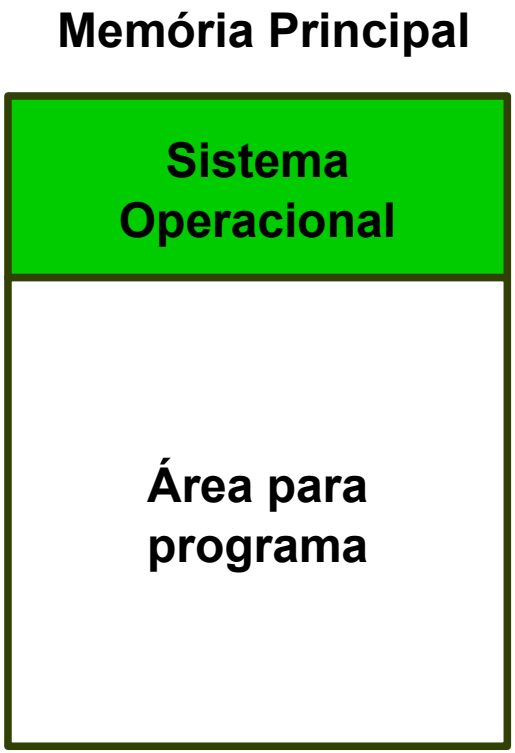
INSTITUTO FEDERAL
PARANÁ

- Portanto, a gerência de memória deve:
  - Manter na memória principal o maior número de processos residentes;
  - Permitir a execução de programas que sejam maiores que a memória física disponível;
  - Proteger áreas de memória ocupadas por cada processo, além da área de memória reservada ao Sistema Operacional.

Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

- A alocação contígua simples era implementada nos primeiros sistemas operacionais na qual a memória era dividida em duas áreas:
  - Uma para o SO
  - Outra para o programa do usuário



Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

- Dessa forma, o programador deve desenvolver suas aplicações preocupado, apenas, em não ultrapassar o espaço de memória disponível, ou seja, a diferença entre o tamanho total da memória principal e a área ocupada pelo SO.
- O problema é que neste modelo algumas vezes não existe espaço suficiente na memória para sua execução.

Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

- Nesse esquema, o usuário tem controle sobre toda a memória principal, podendo ter acesso a qualquer posição de memória, inclusive a área do sistema operacional.
- Para proteger o sistema desse tipo de acesso, que pode ser intencional ou não, alguns sistemas implementavam uma proteção que impedia o programador de obter acesso à área da memória ocupada pelo sistema operacional.

Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

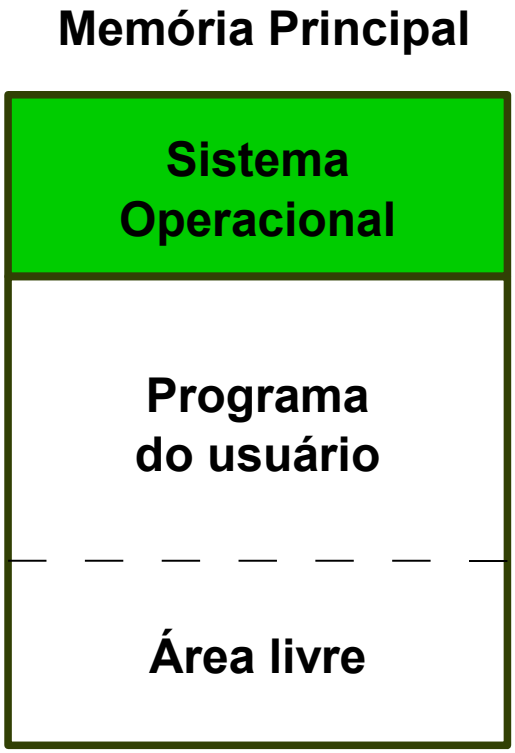
- Dessa forma, quando o programa fazia referência a um endereço da memória, o Sistema Operacional verificava se o endereço estava dentro dos limites permitidos.
- Caso não esteja, o programa é cancelado e uma mensagem de erro é gerada, indicando que houve uma violação no acesso à memória principal.

Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

- As desvantagens da alocação contígua simples é que somente um usuário faz uso dos recursos e na maior parte dos casos sempre existirá espaço de memória livre.

Problema da Subutilização da memória principal



Alocação Contígua Simples

INSTITUTO FEDERAL
PARANÁ

- Assim, não é possível permitir a utilização eficiente da UCP e da memória principal, pois apenas um processo pode utilizar esses recursos.
- A princípio os programas são limitados ao tamanho da memória disponível.

Técnica de *Overlay*

INSTITUTO FEDERAL
PARANÁ

- Na alocação contígua simples, todos os programas estão limitados ao tamanho da área de memória principal disponível para o usuário.
- Uma solução encontrada para o problema é dividir o programa em módulos, de forma que seja possível a execução independente de cada módulo, utilizando uma mesma área de memória.

Técnica de *Overlay*

INSTITUTO FEDERAL
PARANÁ

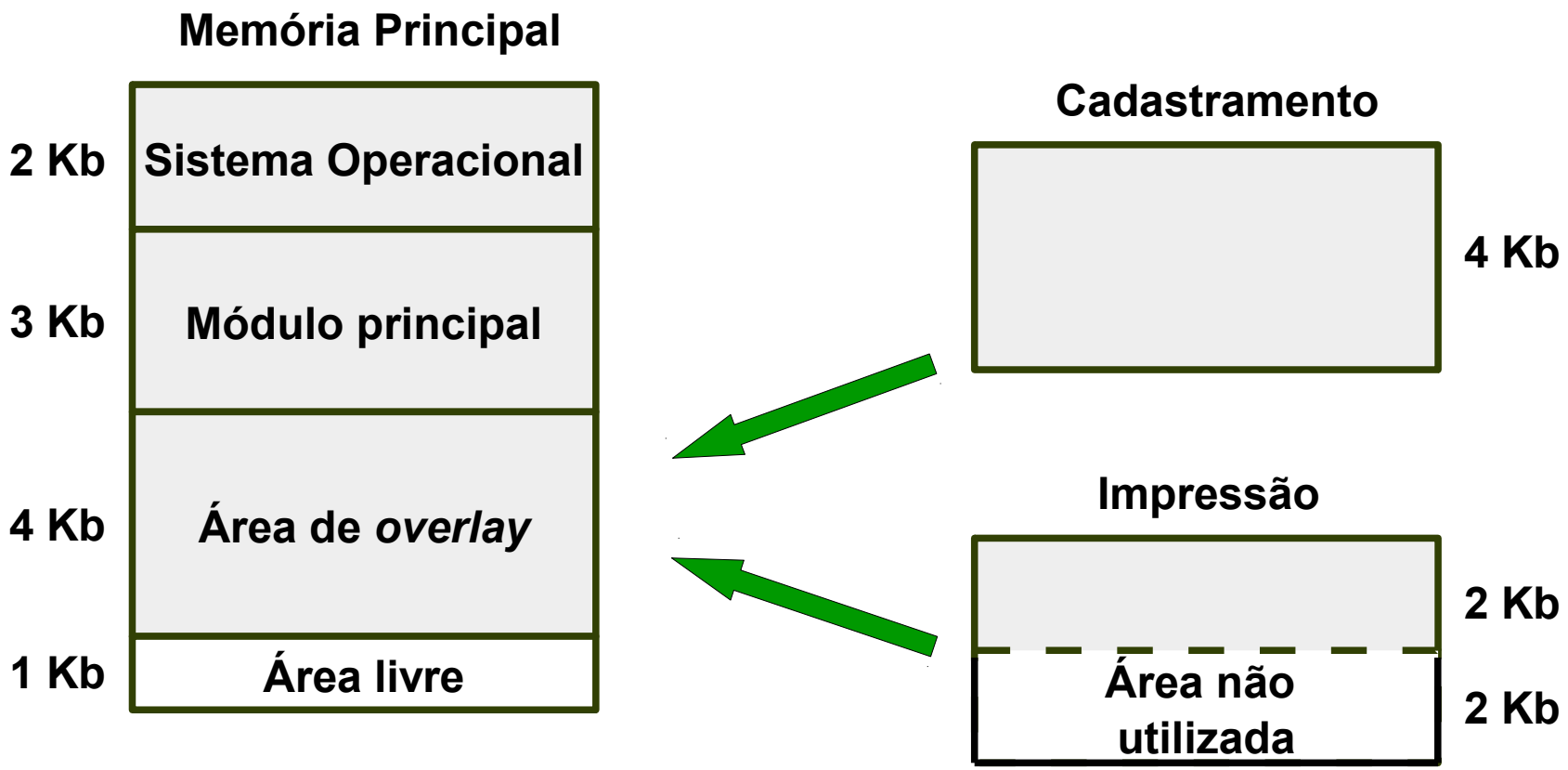
- Considere um programa que tenha três módulos:
  - Principal
  - Cadastramento
  - Impressão
- Sendo os módulos de cadastramento e de impressão independentes.

Técnica de *Overlay*

INSTITUTO FEDERAL
PARANÁ

- A independência do código significa que quando um módulo estiver na memória para execução, o outro não precisa necessariamente estar presente.
- O módulo principal é comum aos dois módulos; logo deve permanecer na memória durante todo o tempo da execução do programa.

Técnica de *Overlay*



Técnica de *Overlay*

INSTITUTO FEDERAL
PARANÁ

- No exemplo, a memória é insuficiente para armazenar todo o programa, que totaliza 9kb.
- A técnica de *overlay* utiliza uma área de memória comum, onde os módulos de cadastramento e de impressão poderão compartilhar a mesma área de memória (área de *overlay*).
- Sempre que um dos dois módulos for referenciado pelo módulo principal, o módulo será carregado da memória secundária para a área de *overlay*.

Técnica de *Overlay*

INSTITUTO FEDERAL
PARANÁ

- A definição das áreas de *overlay* é função do programador, através de comandos específicos da linguagem de programação utilizada.
- O tamanho de uma área de *overlay* é estabelecido a partir do tamanho do maior módulo.
- A técnica de *overlay* tem a vantagem de permitir ao programador expandir os limites da memória principal.

Multiprogramação

INSTITUTO FEDERAL
PARANÁ

- Os sistemas operacionais evoluíram no sentido de proporcionar melhor aproveitamento dos recursos disponíveis.
- Nos sistemas monoprogramáveis, o processador permanece grande parte do tempo ocioso e a memória principal é subutilizada.

Multiprogramação

INSTITUTO FEDERAL
PARANÁ

- Os sistemas multiprogramáveis já são muito mais eficientes no uso do processador, necessitando, assim, que diversos programas estejam na memória principal ao mesmo tempo e que novas formas de gerência da memória sejam implementadas

Multiprogramação

INSTITUTO FEDERAL
PARANÁ

- Em geral a monoprogramação é utilizada em computadores pequenos com Sistema Operacional bastante simples. Contudo, com frequência é preciso permitir que vários processos executem ao mesmo tempo.
- Em sistemas de tempo compartilhado, ter vários processos na memória simultaneamente significa que quando um processo está bloqueado outro está usando o processador. Dessa forma, a multiprogramação aumenta a utilização da CPU.

Alocação Particionada Estática

INSTITUTO FEDERAL
PARANÁ

- Nos primeiros sistemas multiprogramáveis, a memória era dividida em pedaços de tamanho fixo, chamados **partições**.
- O tamanho das partições, estabelecido na fase de inicialização do sistema, era definido em função do tamanho dos programas que executariam no ambiente.

Alocação Particionada Estática

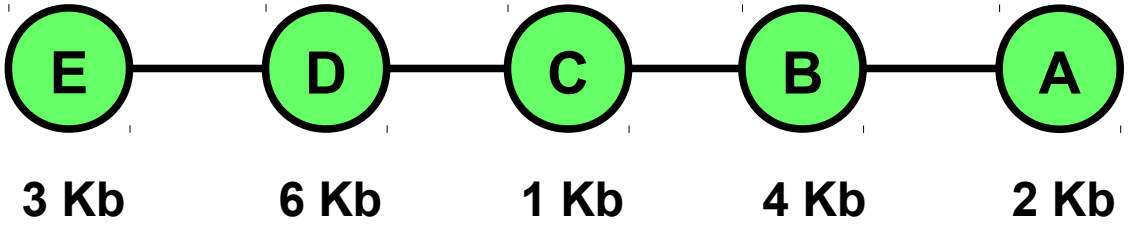
INSTITUTO FEDERAL
PARANÁ

Tabela de partições

Partição	Tamanho
1	2 Kb
2	5 Kb
3	8 Kb

Memória Principal

Programas a serem executados:



Alocação Particionada Estática

INSTITUTO FEDERAL
PARANÁ

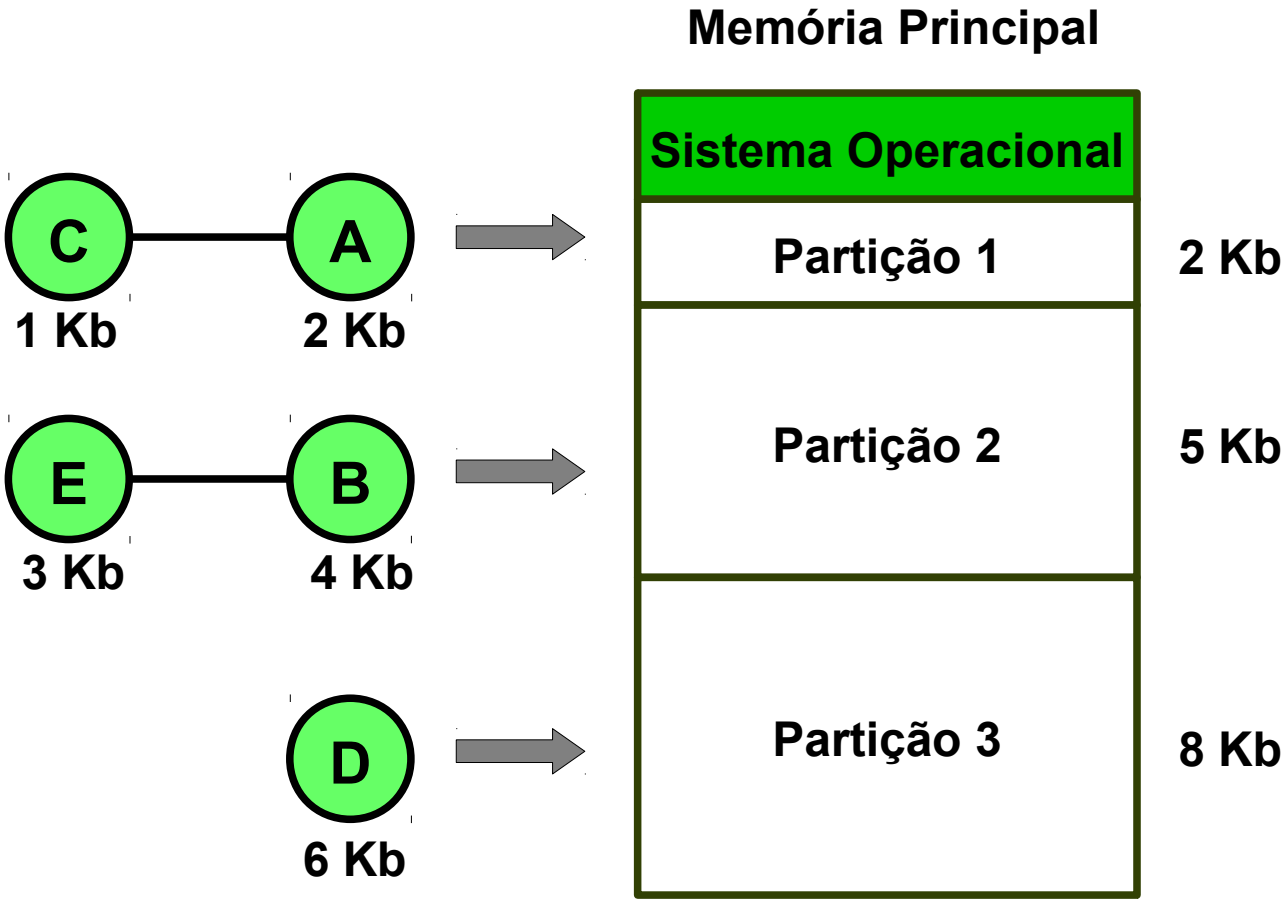
- Sempre que fosse necessária a alteração do tamanho de uma partição, o sistema deveria ser desativado e reinicializado com uma nova configuração.
- Inicialmente, os programas só podiam ser carregados e executados em apenas uma partição específica, mesmo se outras estivessem disponíveis.

Alocação Particionada Estática

INSTITUTO FEDERAL
PARANÁ

- Essa limitação se devia aos compiladores e montadores que geravam apenas código absoluto.
- No código absoluto todas as referências a endereços no programa são posições físicas na memória principal, ou seja, o programa só poderia ser carregado a partir do endereço de memória especificado no seu próprio código.

Alocação Particionada Estática Absoluta



Alocação Particionada Estática Absoluta

INSTITUTO FEDERAL
PARANÁ

- Se, por exemplo, os programas A e B estivessem sendo executados e a terceira partição estivesse livre, os programas C e E não poderiam ser processados.
- Com a evolução dos compiladores, montadores, *linkers* e *loaders*, o código gerado deixou de ser absoluto e passou a ser relocável.

Alocação Particionada Estática Relocável

- No código relocável, todas as referências a endereços no programa são relativas ao início do código, e não a endereços físicos de memória.
- Desta forma, os programas puderam ser executados a partir de qualquer partição.
- Quando o programa é carregado, o loader calcula todos os endereços a partir da posição inicial onde o programa foi alocado.

Alocação Particionada Estática Relocável

INSTITUTO FEDERAL
PARANÁ

- Caso os programas A e B terminassem, o programa E poderia ser executado em qualquer uma das duas partições.

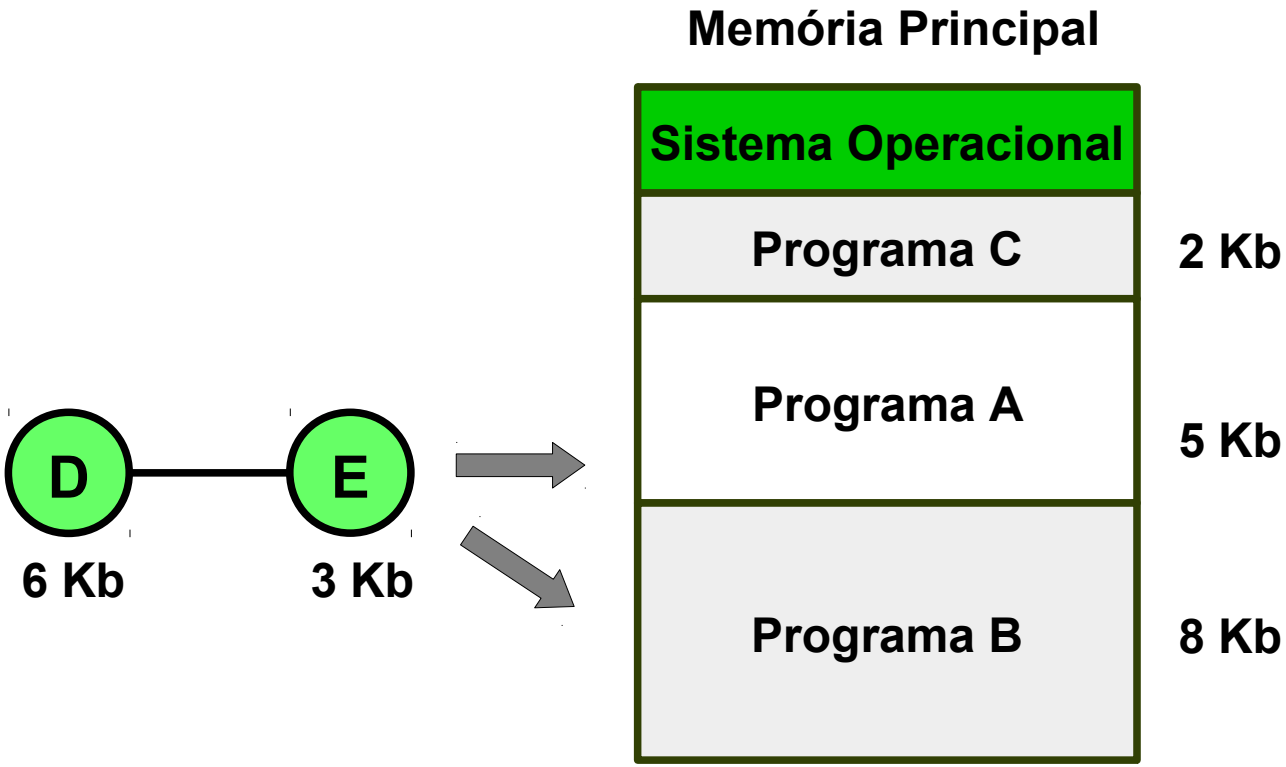


Tabela de Alocação de Partições

INSTITUTO FEDERAL
PARANÁ

- Para manter o controle sobre quais partições estão alocadas, a gerência de memória mantém uma tabela com o endereço inicial de cada partição, seu tamanho e se está em uso.
- Sempre que um programa é carregado para a memória, o sistema percorre a tabela, na tentativa de localizar uma partição livre onde o programa possa ser carregado.

Tabela de Alocação de Partições

INSTITUTO FEDERAL
PARANÁ

Partição	Tamanho	Livre
1	2 Kb	Não
2	5 Kb	Sim
3	8 Kb	Não

Tabela de Alocação de Partições

INSTITUTO FEDERAL
PARANÁ

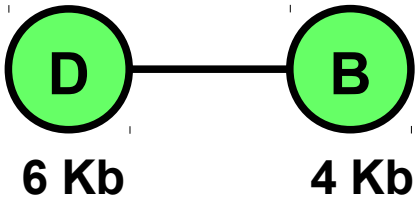
- Nesse esquema de alocação de memória a proteção baseia-se em dois registradores, que indicam os limites inferior e superior da partição onde o programa está sendo executado.
- Caso o programa tente acessar uma posição de memória fora dos limites definidos pelos registradores, ele é interrompido e uma mensagem de violação de acesso é gerada pelo sistema operacional.

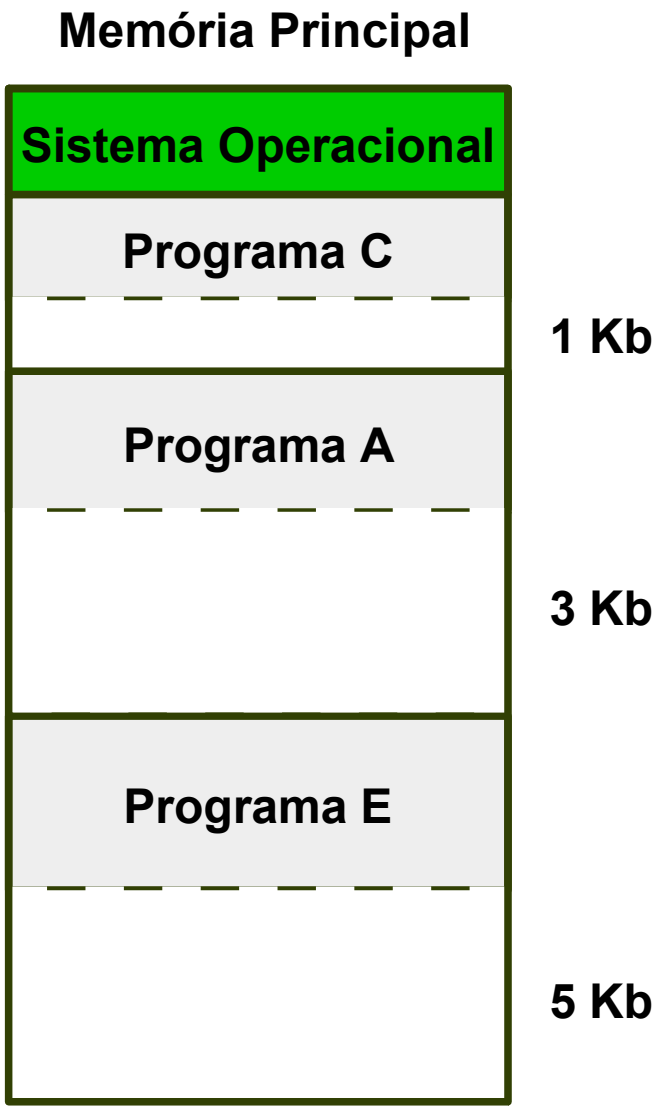
Fragmentação interna

INSTITUTO FEDERAL
PARANÁ

- Tanto nos sistemas de alocação absoluta quanto nos de alocação relocável os programas, normalmente, não preenchem totalmente as partições onde são carregados.
- Por exemplo, os programas C, A e E não ocupam integralmente o espaço das partições onde estão alocados, deixando 1Kb, 3Kb e 5 Kb de áreas livres, respectivamente.

Fragmentação interna





Alocação Particionada Dinâmica

INSTITUTO FEDERAL
PARANÁ

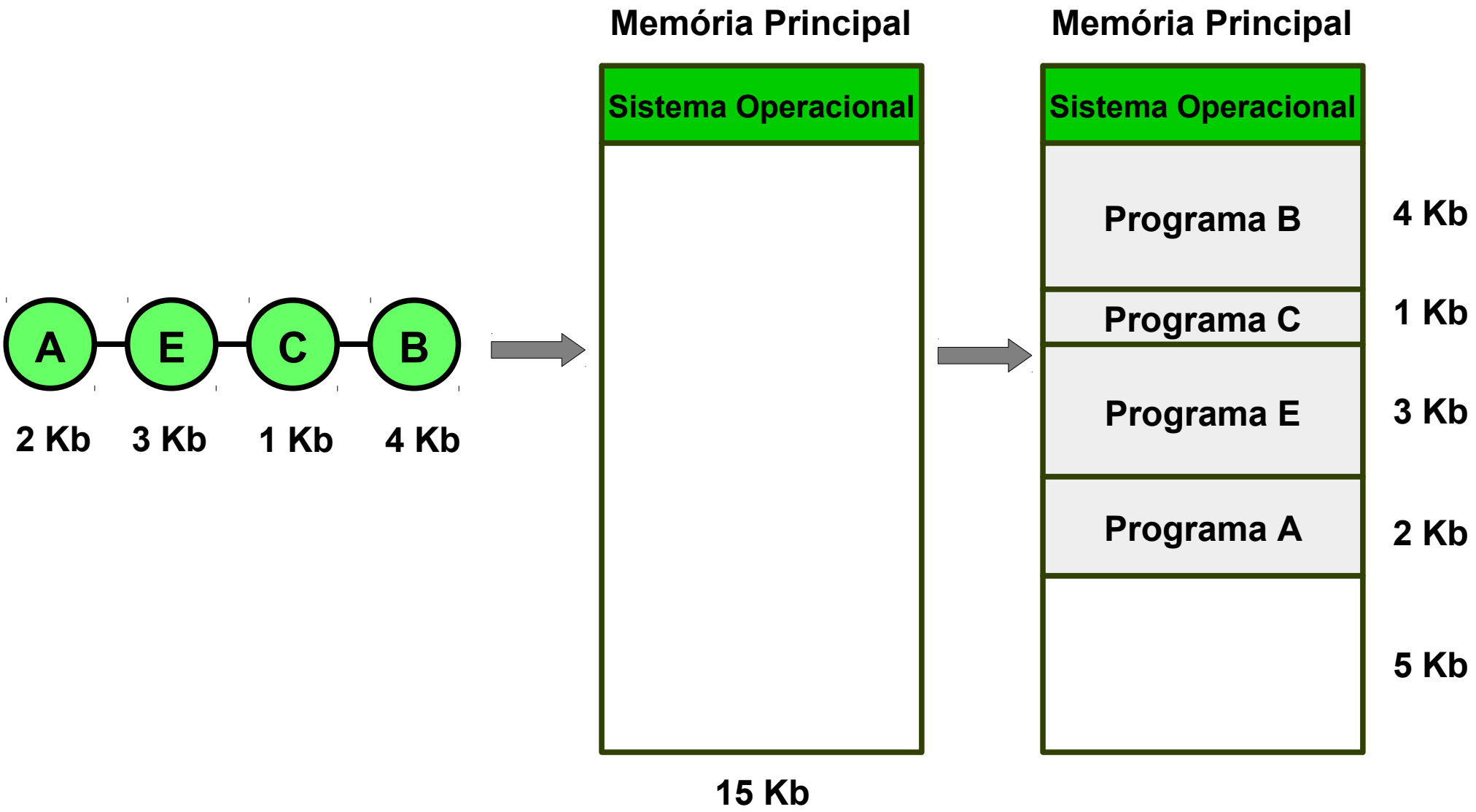
- A alocação particionada estática deixa evidente a necessidade de uma nova forma de gerência da memória principal, em que o problema da fragmentação interna fosse reduzido e, consequentemente, o grau de compartilhamento da memória aumentado.

Alocação Particionada Dinâmica

INSTITUTO FEDERAL
PARANÁ

- Na alocação particionada dinâmica, ou variável, foi eliminado o conceito de partições de tamanho fixo.
- Nesse esquema, cada programa utilizaria o espaço necessário, tornando essa área sua partição.

Alocação Particionada Dinâmica



Alocação Particionada Dinâmica

INSTITUTO FEDERAL
PARANÁ

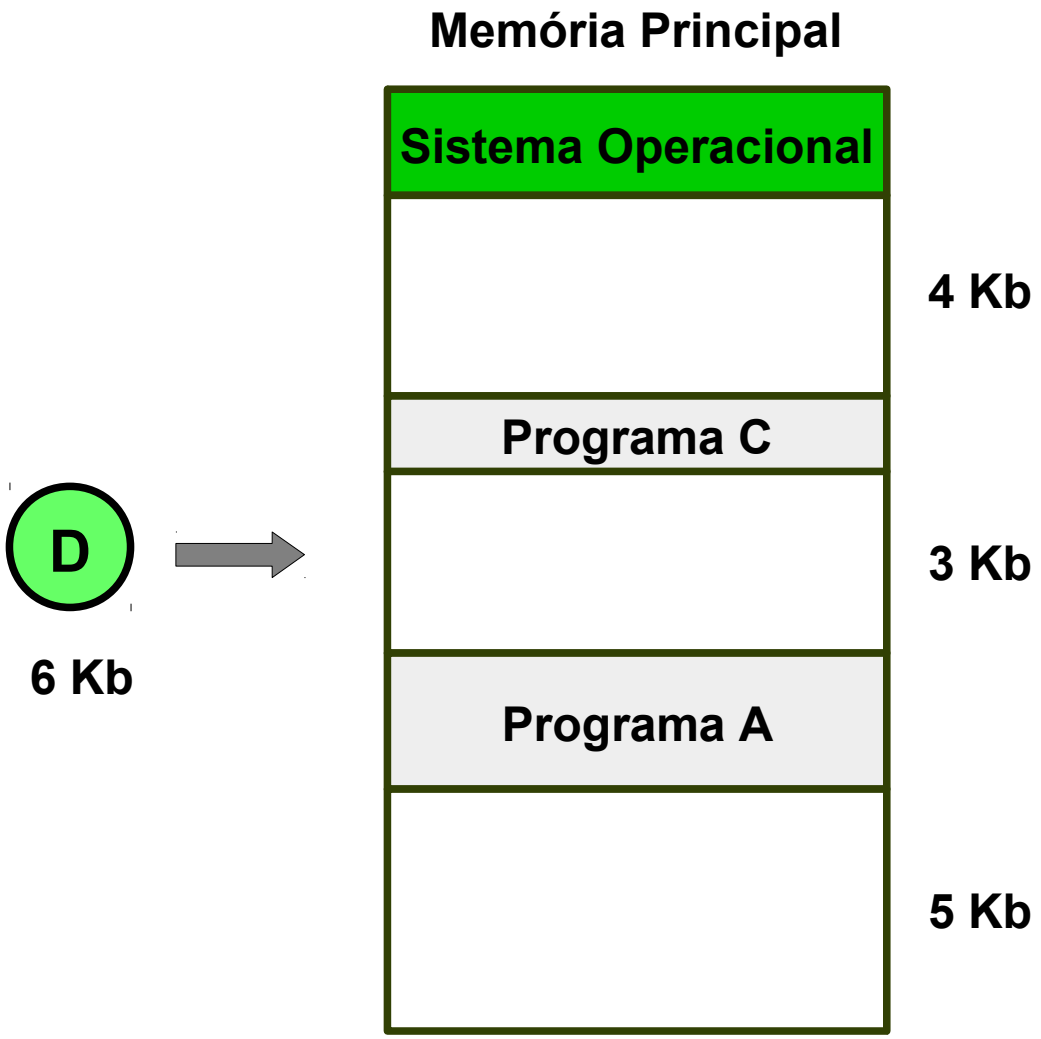
- Como os programas utilizam apenas o espaço de que necessitam, no esquema de alocação particionada dinâmica o problema da fragmentação interna não ocorre.
- Um diferente tipo de fragmentação começará a ocorrer, quando os programas forem terminando e deixando espaços cada vez menores na memória, não permitindo o ingresso de novos programas.

Fragmentação externa

INSTITUTO FEDERAL
PARANÁ

- Por exemplo, mesmo existindo 12 Kb livres de memória principal o programa D, que necessita de 6 Kb de espaço, não poderá ser carregado para execução, pois esse espaço não está disposto contiguamente.
- Esse tipo de problema é chamado de fragmentação externa.

Fragmentação externa

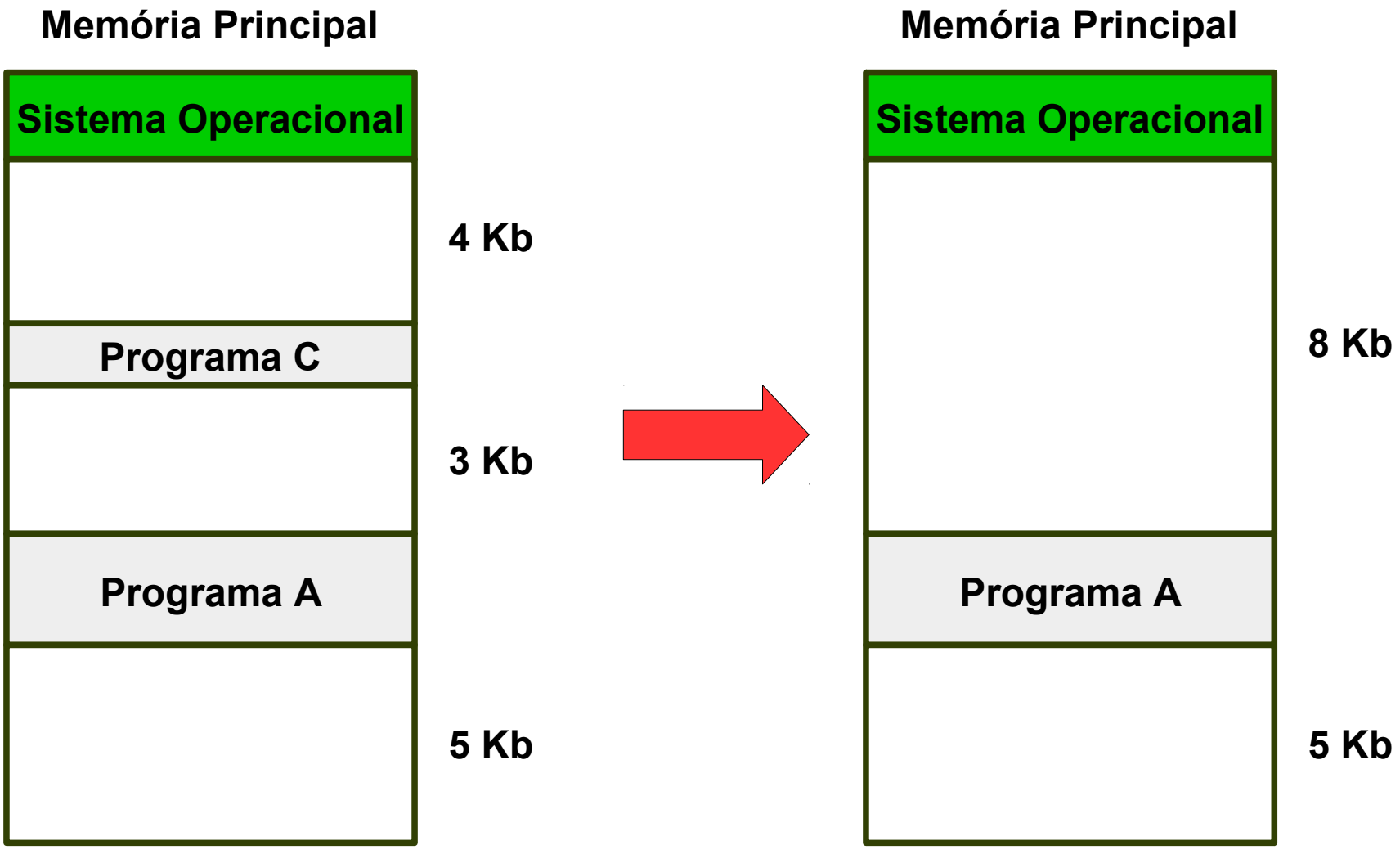


Fragmentação externa

INSTITUTO FEDERAL
PARANÁ

- Existem duas soluções para o problema da fragmentação externa da memória principal.
- Na primeira solução, conforme os programas terminam apenas os espaços livres adjacentes são reunidos, produzindo áreas livres de tamanho maior.

Fragmentação externa

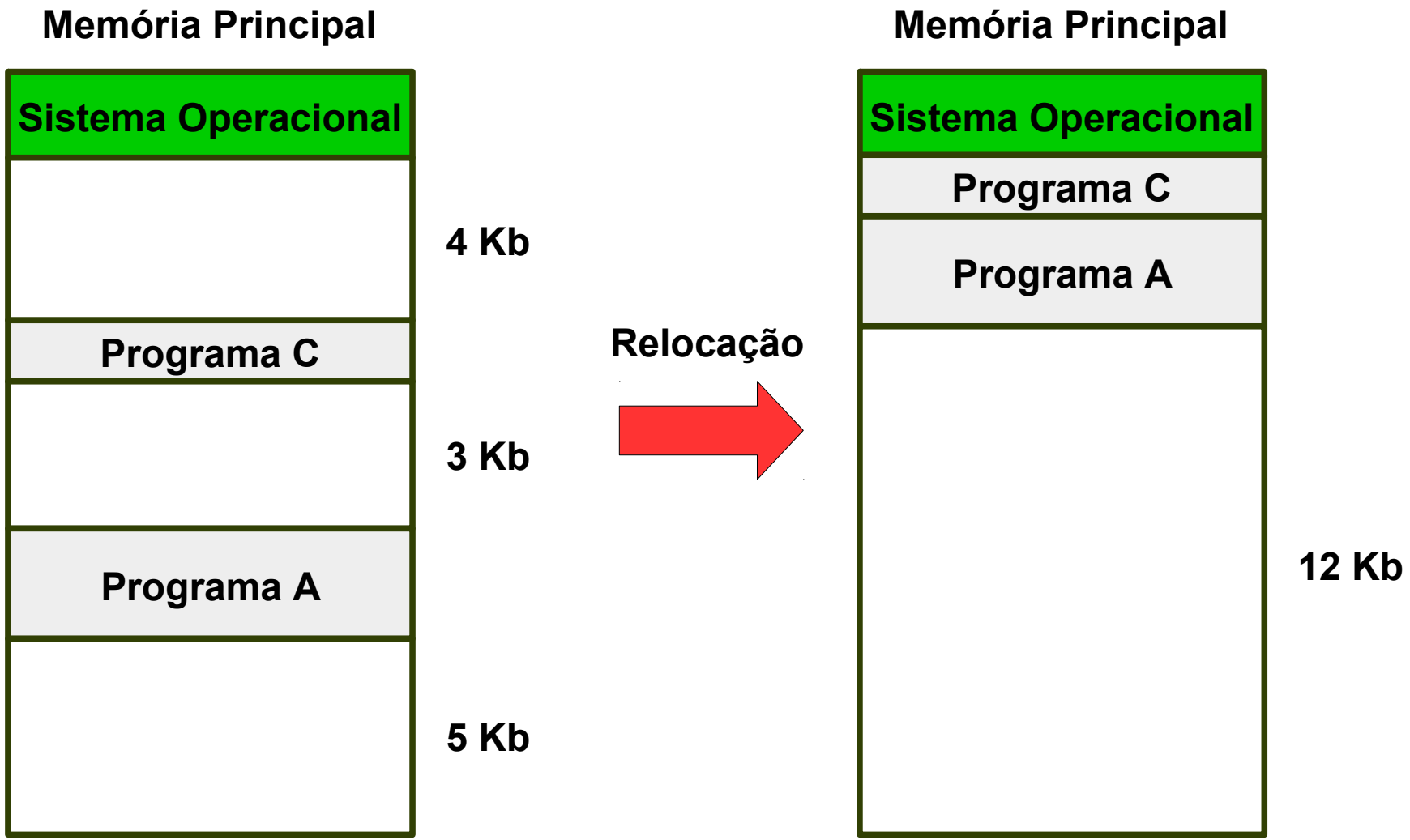


Fragmentação externa

INSTITUTO FEDERAL
PARANÁ

- A segunda solução envolve a relocação de todas as partições ocupadas, eliminando todos os espaços entre elas e criando uma única área livre contígua.
- Para que esse processo seja possível é necessário que o sistema tenha a capacidade de mover os diversos programas na memória principal, ou seja, realizar **relocação dinâmica**.

Fragmentação externa



Alocação Particionada Dinâmica com Relocação

INSTITUTO FEDERAL
PARANÁ

- Esse mecanismo de compactação, também conhecido como alocação particionada dinâmica com relocação, reduz em muito o problema da fragmentação, porém a complexidade do seu algoritmo e o consumo de recursos do sistema, como processador e área em disco, podem torná-lo inviável.

Estratégias de Alocação de Partição

INSTITUTO FEDERAL
PARANÁ

- Os sistemas operacionais implementam, basicamente, três estratégias para determinar em qual área livre um programa será carregado para execução.
- Essas estratégias tentam evitar ou diminuir o problema da fragmentação externa.

Estratégias de Alocação de Partição

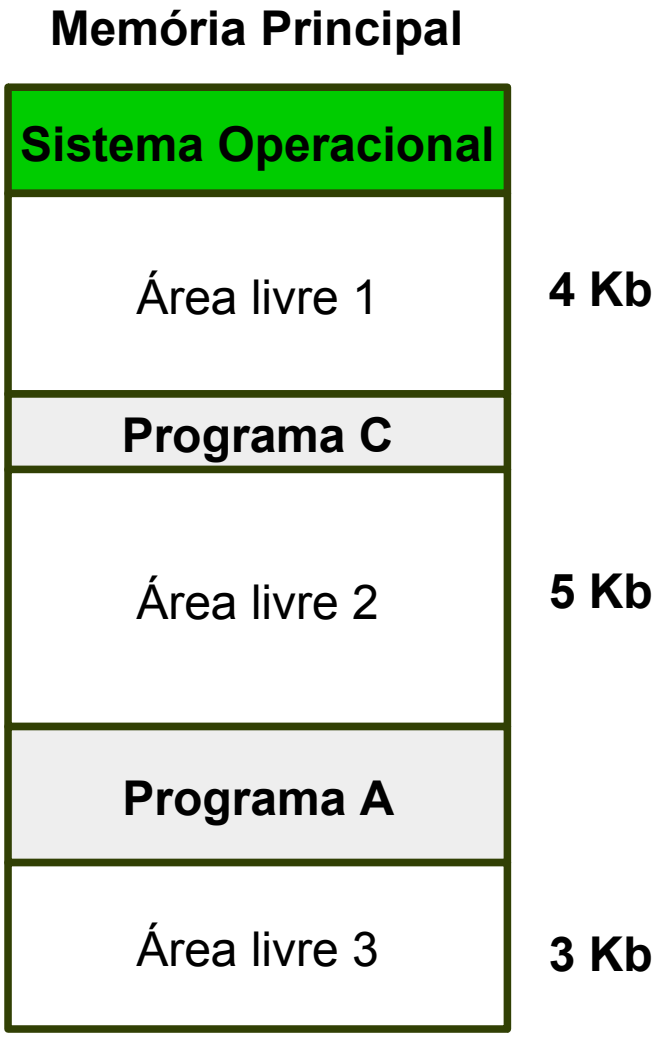
INSTITUTO FEDERAL
PARANÁ

- A melhor estratégia a ser adotada por um sistema depende de uma série de fatores, sendo o mais importante o tamanho dos programas processados no ambiente.
- Independentemente do algoritmo utilizado, o sistema possui uma lista de áreas livres, com o endereço e tamanho de cada área.

Estratégias de Alocação de Partição

INSTITUTO FEDERAL
PARANÁ

Áreas livres	Tamanho
1	4 Kb
2	5 Kb
3	3 Kb

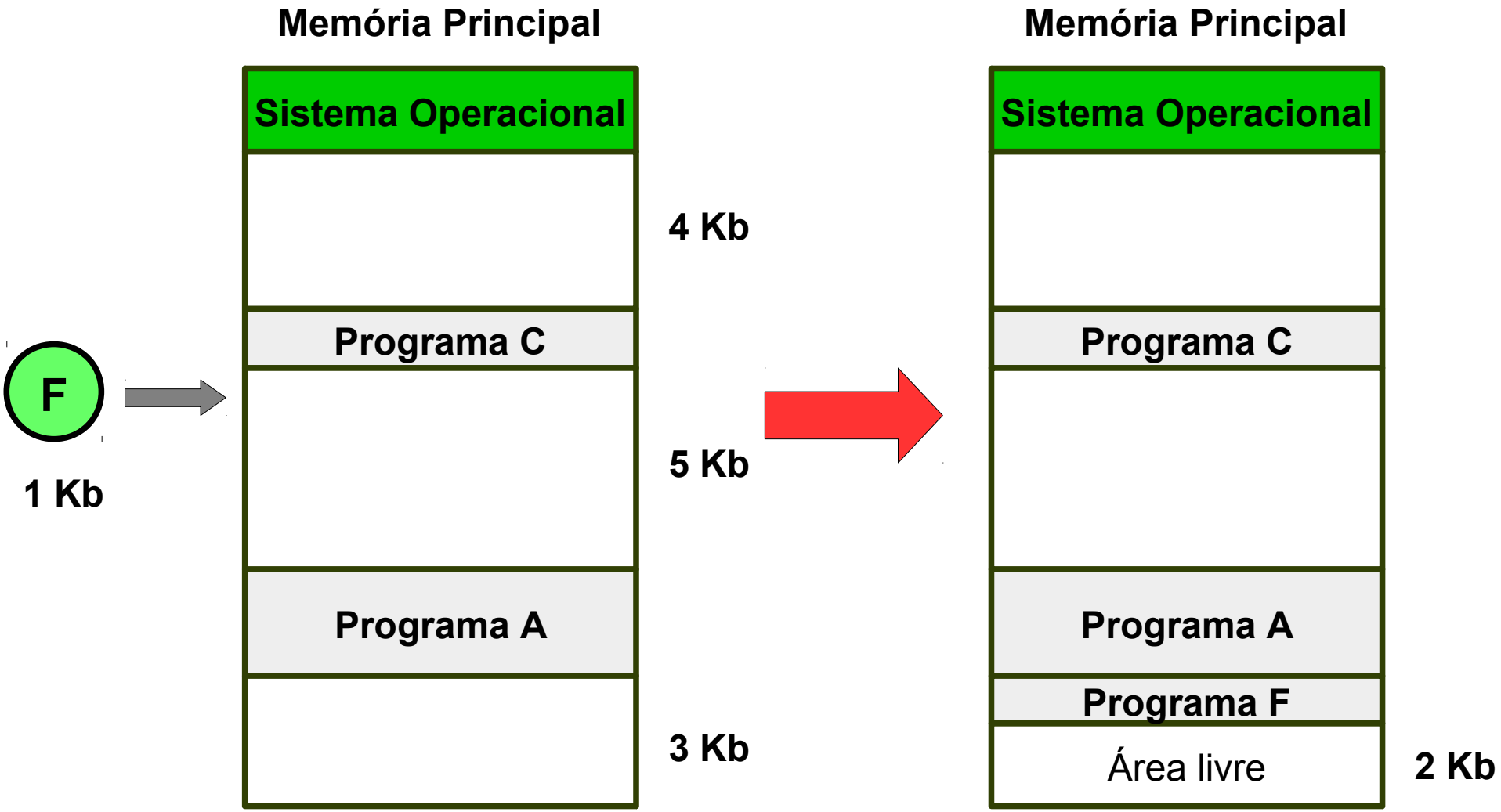


Estratégia *Best-fit*

INSTITUTO FEDERAL
PARANÁ

- Na estratégia *best-fit*, a melhor partição é escolhida, ou seja, aquela em que o programa deixa o menor espaço sem utilização.
- Nesse algoritmo, a lista de áreas livres está ordenada por tamanho, diminuindo o tempo de busca por uma área desocupada.

Estratégia *Best-fit*



Estratégia *Best-fit*

INSTITUTO FEDERAL
PARANÁ

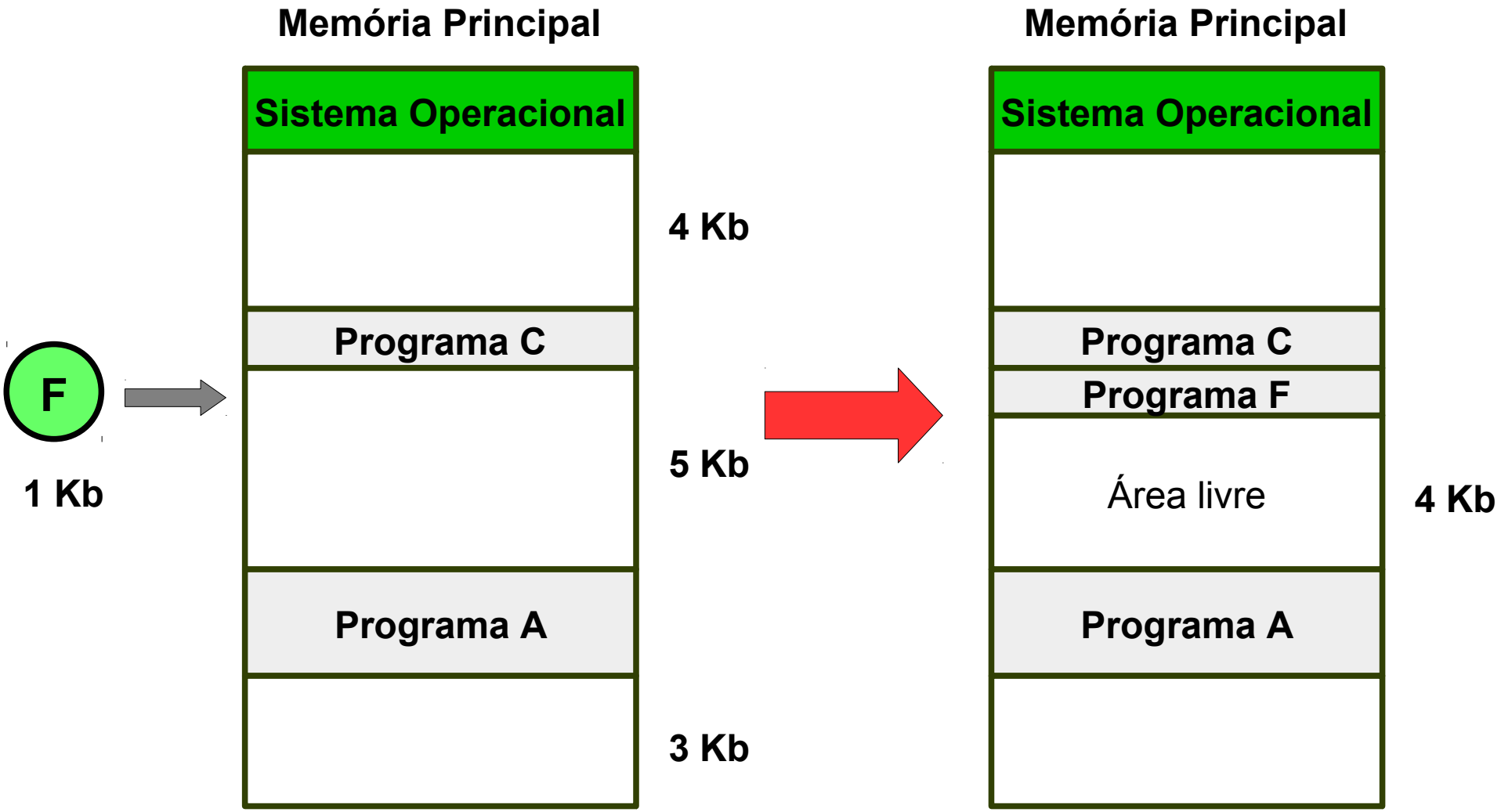
- Uma grande desvantagem desse método é consequência do próprio algoritmo.
- Como é alocada a partição que deixa a menor área livre, a tendência é que cada vez mais a memória fique com pequenas áreas não-contíguas, aumentando o problema da fragmentação.

Estratégia *Worst-fit*

INSTITUTO FEDERAL
PARANÁ

- Na estratégia *worst-fit*, a pior partição é escolhida, ou seja, aquela em que o programa deixa o maior espaço sem utilização.
- Apesar de utilizar as maiores partições, a técnica de *worst-fit* deixa espaços livres maiores que permitem a um maior número de programas utilizar a memória, diminuindo o problema da fragmentação.

Estratégia *Worst-fit*



Estratégia *First-fit*

INSTITUTO FEDERAL
PARANÁ

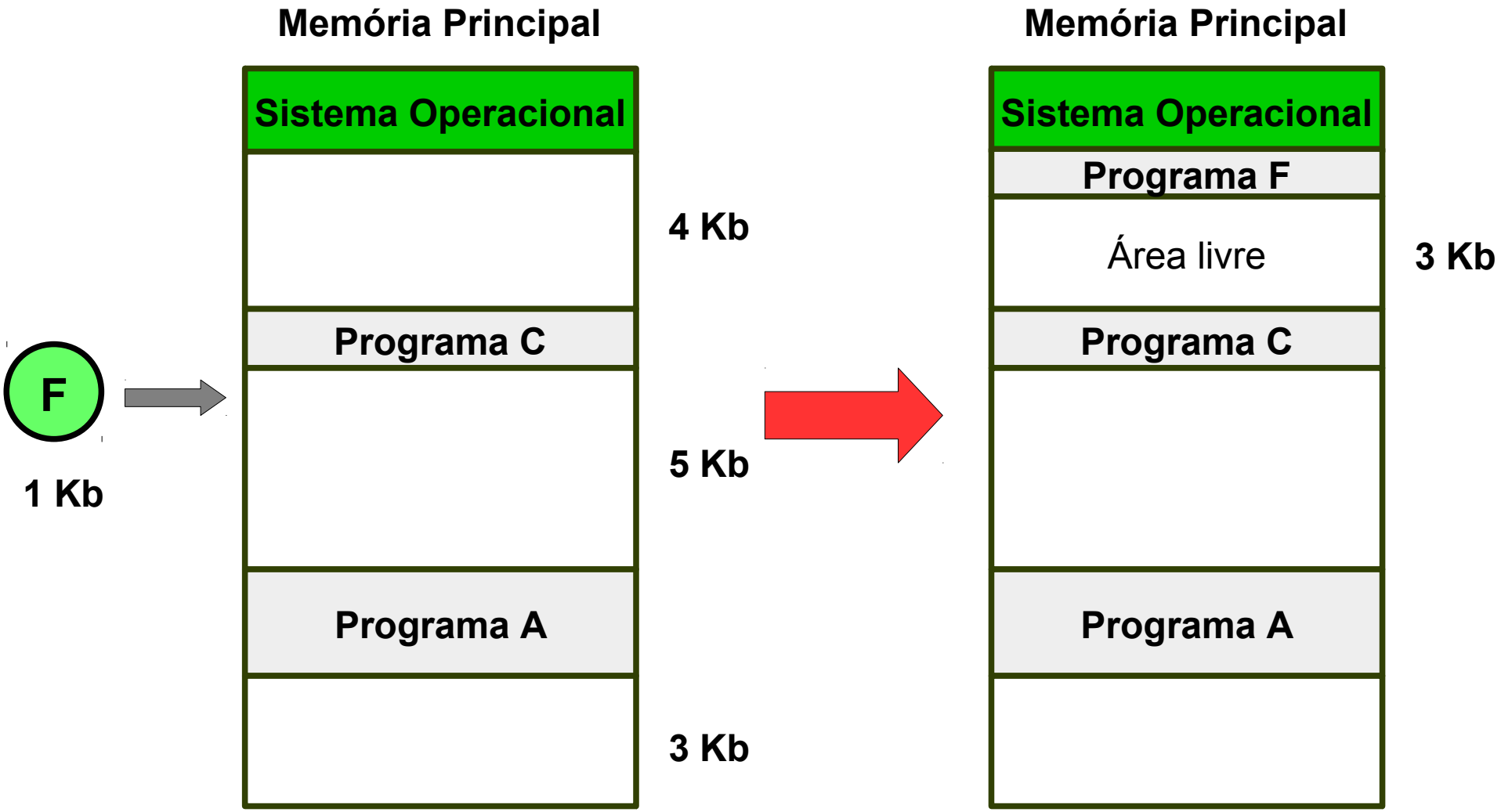
- Na estratégia *first-fit*, a primeira partição livre de tamanho suficiente para carregar o programa é escolhida.
- Nesse algoritmo, a lista de áreas livres está ordenada por endereços crescentemente.
- Como o método tenta primeiro utilizar as áreas livres de endereços mais baixos, existe uma grande chance de se obter uma grande partição livre nos endereços de memória mais alto.

Estratégia *First-fit*

INSTITUTO FEDERAL
PARANÁ

- Na estratégia *first-fit*, a primeira partição livre de tamanho suficiente para carregar o programa é escolhida.
- Nesse algoritmo, a lista de áreas livres está ordenada por endereços crescentemente.

Estratégia *First-fit*



Estratégia *First-fit*

INSTITUTO FEDERAL
PARANÁ

- Como o método tenta primeiro utilizar as áreas livres de endereços mais baixos, existe uma grande chance de se obter uma grande partição livre nos endereços de memória mais alto.
- Das três estratégias, a *first-fit* é a mais rápida, consumindo menos recursos do sistema.

Swapping

INSTITUTO FEDERAL
PARANÁ

- Em sistemas operacionais que operam em lotes é muito efetivo utilizar partições fixas
- Cada *job* é carregado em uma partição quando chega ao começo da fila e permanece na memória até que termine.

Swapping

INSTITUTO FEDERAL
PARANÁ

- Com sistemas de compartilhamento de tempo ou computadores gráficos pessoais esta estratégia não pode ser utilizada, pois muitas vezes não há memória suficiente para armazenar todos os processos ativos.
- Dessa forma, os processos em excesso são mantidos no disco e trazidos de lá para execução dinamicamente.

Swapping

INSTITUTO FEDERAL
PARANÁ

- Em todos os esquemas apresentados anteriormente, um processo permanecia na memória principal até o final da sua execução, inclusive nos momentos em que esperava por um evento, como uma operação de leitura ou gravação.

Swapping

INSTITUTO FEDERAL
PARANÁ

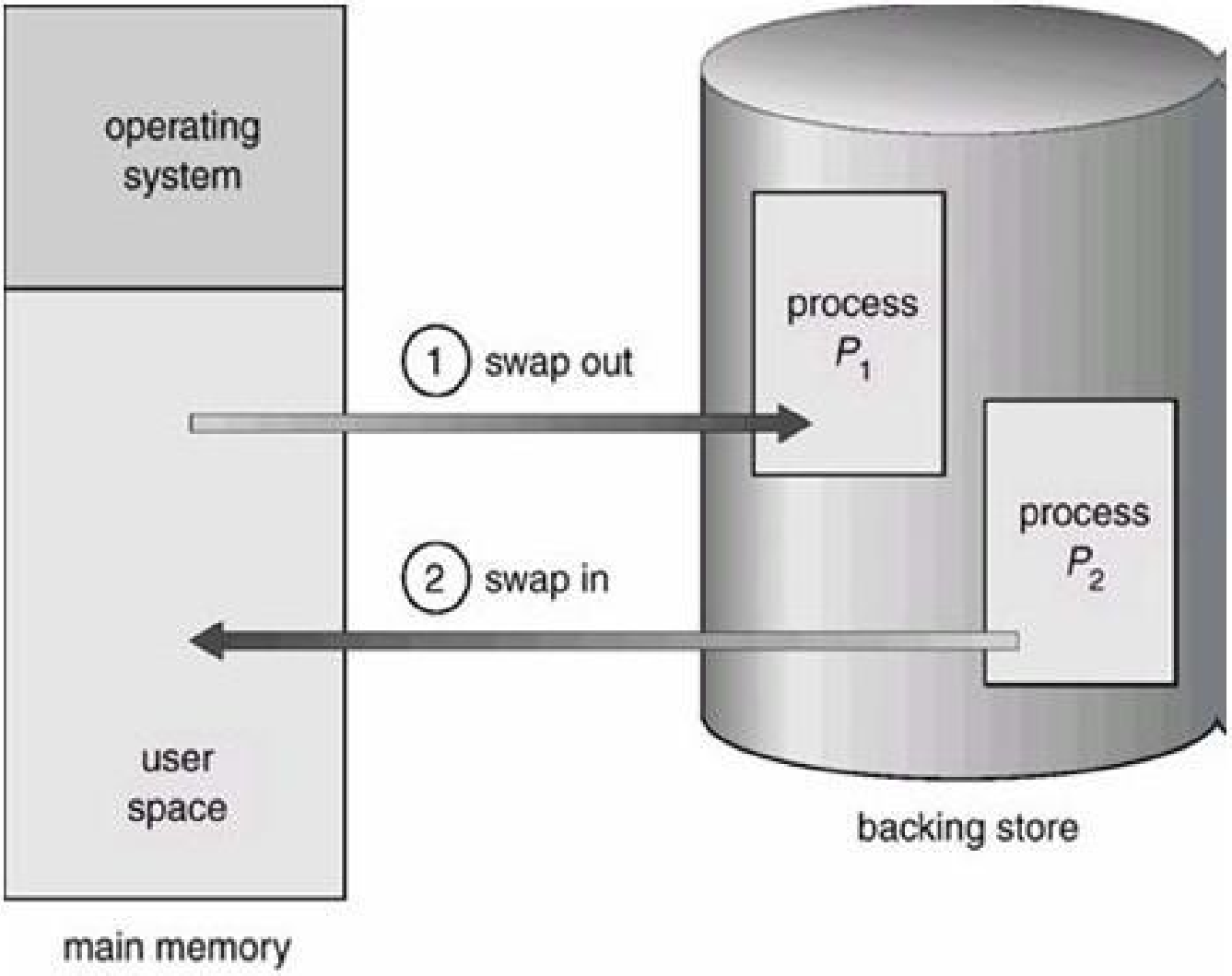
- O *swapping* foi introduzido para contornar o problema da insuficiência de memória principal.
- É aplicado à gerência de memória para programas que esperam por memória livre para serem executados.
- Nesse processo são executadas duas operações:
  - ***Swap out***
  - ***Swap in***

Swapping

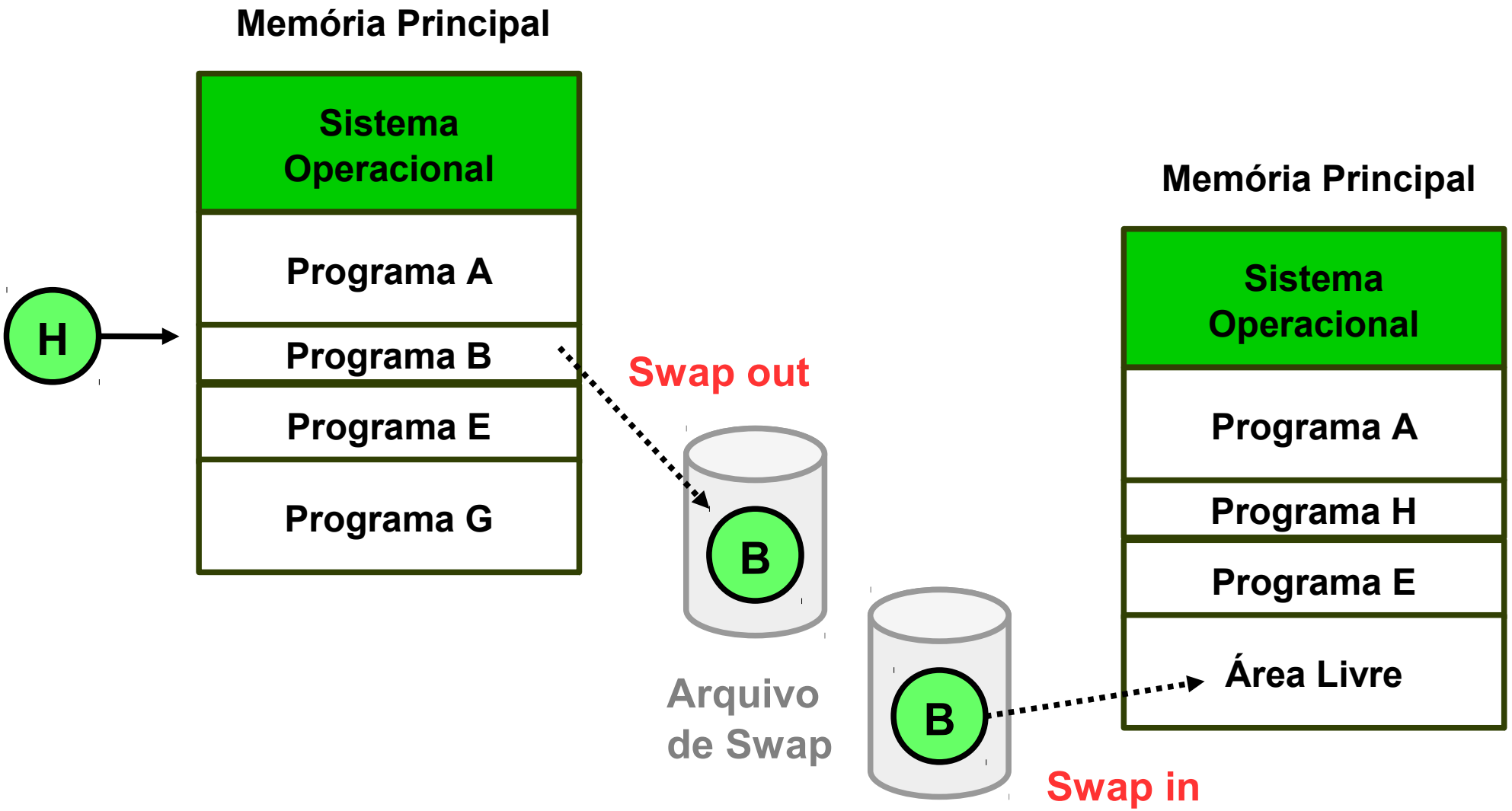
INSTITUTO FEDERAL
PARANÁ

- ***Swap Out:*** o Sistema Operacional escolhe um processo residente e o transfere para a memória secundária.
- ***Swap in:*** ocorre um processo inverso ao *swap out*. O processo é carregado de volta da memória secundária para a memória principal e o programa pode continuar sua execução como se nada tivesse acontecido.

Swapping


operating system
user space
main memory
1 swap out
2 swap in
process P_1
process P_2
backing store

Swapping


Memória Principal
Sistema Operacional
Programa A
Programa B
Programa E
Programa G
H
Swap out
B
Arquivo de Swap
B
Memória Principal
Sistema Operacional
Programa A
Programa H
Programa E
Área Livre
Swap in

Swapping

INSTITUTO FEDERAL
PARANÁ

- O algoritmo deve selecionar os processos com menos chances de serem executados.
- Para tanto, geralmente são escolhidos os processos que estão no estado de espera. Mas processos no estado de pronto também poderão ser selecionados.
- Os processos no estado de espera ou pronto quando fora da memória são ditos não-residentes (***outswapped***).

Swapping

INSTITUTO FEDERAL
PARANÁ

- Sempre que o Escalonador (*CPU scheduler*) decide executar um processo ele chama o *Dispatcher*.
- O *Dispatcher* verifica se o processo a ser executado está residente em memória, se não, ele remove (*swap out*) algum processo que esteja correntemente na memória principal e carrega (*swap in*) o processo escolhido para ser executado.
- Após isto o contexto do processo a ser executado é restaurado e o controle é passado ao processo que passa então a executar.

Swapping

INSTITUTO FEDERAL
PARANÁ

- Para que a técnica de *swapping* seja implementada é essencial que o sistema ofereça um *loader* que implementa a relocação dinâmica de programas.
- Um *loader* relocável que não ofereça esta facilidade permite que um programa seja colocado em qualquer posição de memória, porém a relocação é apenas realizada no momento do carregamento.

Swapping

INSTITUTO FEDERAL
PARANÁ

- No caso do swapping, um programa pode sair e voltar diversas vezes para a memória, sendo necessário que a relocação seja realizada pelo loader a cada carregamento.

Swapping

INSTITUTO FEDERAL
PARANÁ

- O conceito de *swapping* permite um maior compartilhamento da memória principal e, consequentemente, uma maior utilização dos recursos do sistema computacional.
- Seu maior problema é o elevado custo das operações de entrada/saída (***swap in/out***).

Swapping

INSTITUTO FEDERAL
PARANÁ

- Em momentos em que há pouca memória disponível, o sistema pode ficar quase que dedicados à realização de *swapping*, deixando de executar outras tarefas e impedindo a execução dos demais processos residentes.
- Esta situação é denominada ***trashing***, e é considerada um problema crítico na gerência de memória dos sistemas operacionais.

Swapping

INSTITUTO FEDERAL
PARANÁ

- Com a evolução dos sistemas operacionais, novos esquemas de gerência de memória passaram a incorporar a técnica de *swapping*, como a **gerência de memória virtual**.