

# Sistemas Operacionais

## Gerenciamento de Memória

Norton Trevisan Roman  
Marcelo Morandini  
Jó Ueyama

Apostila baseada nos trabalhos de Kalinka Castelo Branco, Antônio Carlos Sementille, Paula Prata e nas transparências fornecidas no site de compra do livro "Sistemas Operacionais Modernos"

# Troca de Processos

- Com um sistema em lote, é simples e eficiente organizar a memória em partições fixas.
- Em sistemas de tempo compartilhado:
  - Pode não existir memória suficiente para conter todos os processos ativos
  - Os processos excedentes são mantidos em disco e trazidos dinamicamente para a memória a fim se serem executados

# Troca de Processos

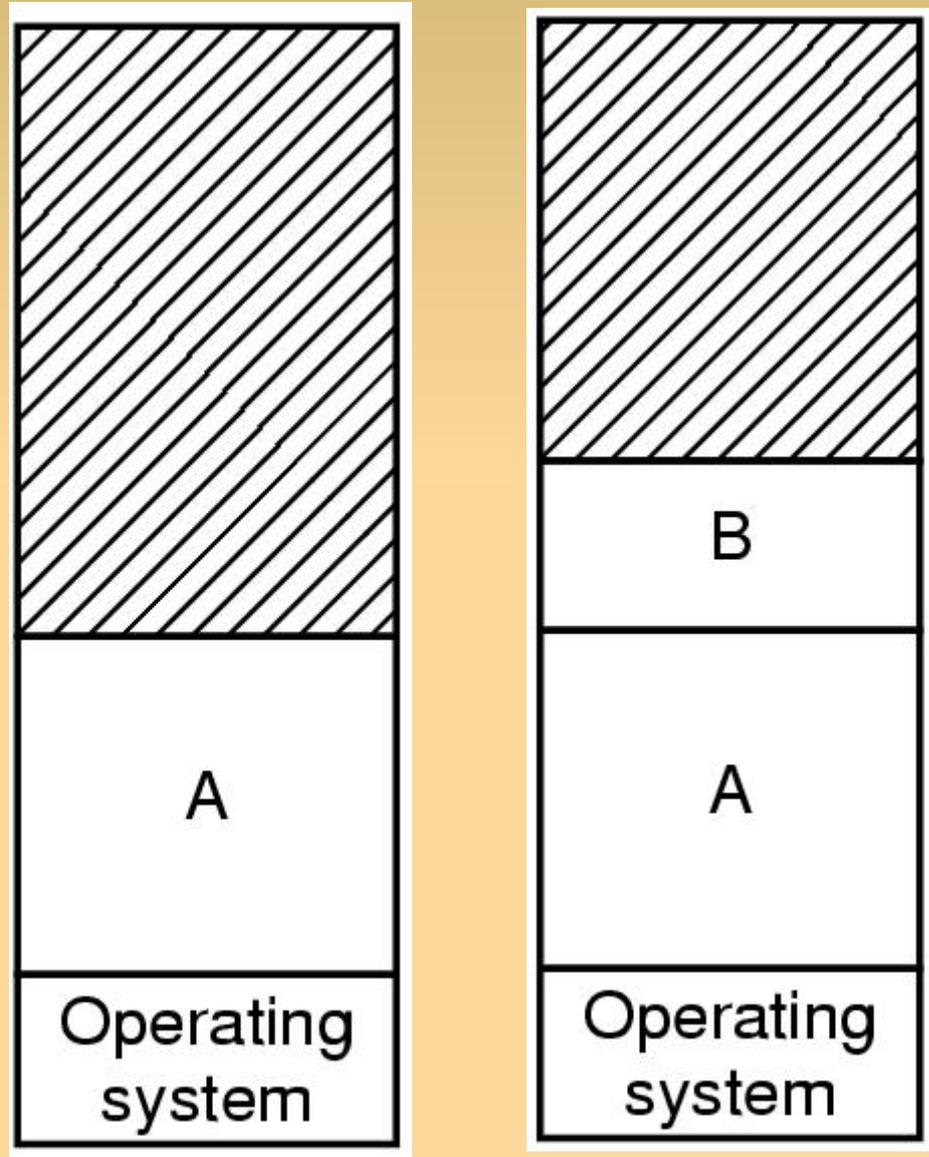
- Existem 2 processos gerais que podem ser usados:
  - A troca de processos (swapping):
    - Forma mais simples
    - Consiste em trazer totalmente cada processo para a memória, executá-lo durante um tempo e, então, devolvê-lo ao disco
  - Memória Virtual:
    - Permite que programas possam ser executados mesmo que estejam parcialmente carregados na memória principal.

# Swapping

- Chaveamento de processos inteiros entre a memória principal e o disco;
- Swap-out
  - Da memória para uma região especial do disco, chamada “área de swap”
- Swap-in
  - Do disco pra memória
- Pode ser utilizado tanto com partições fixas quanto com partições variáveis;

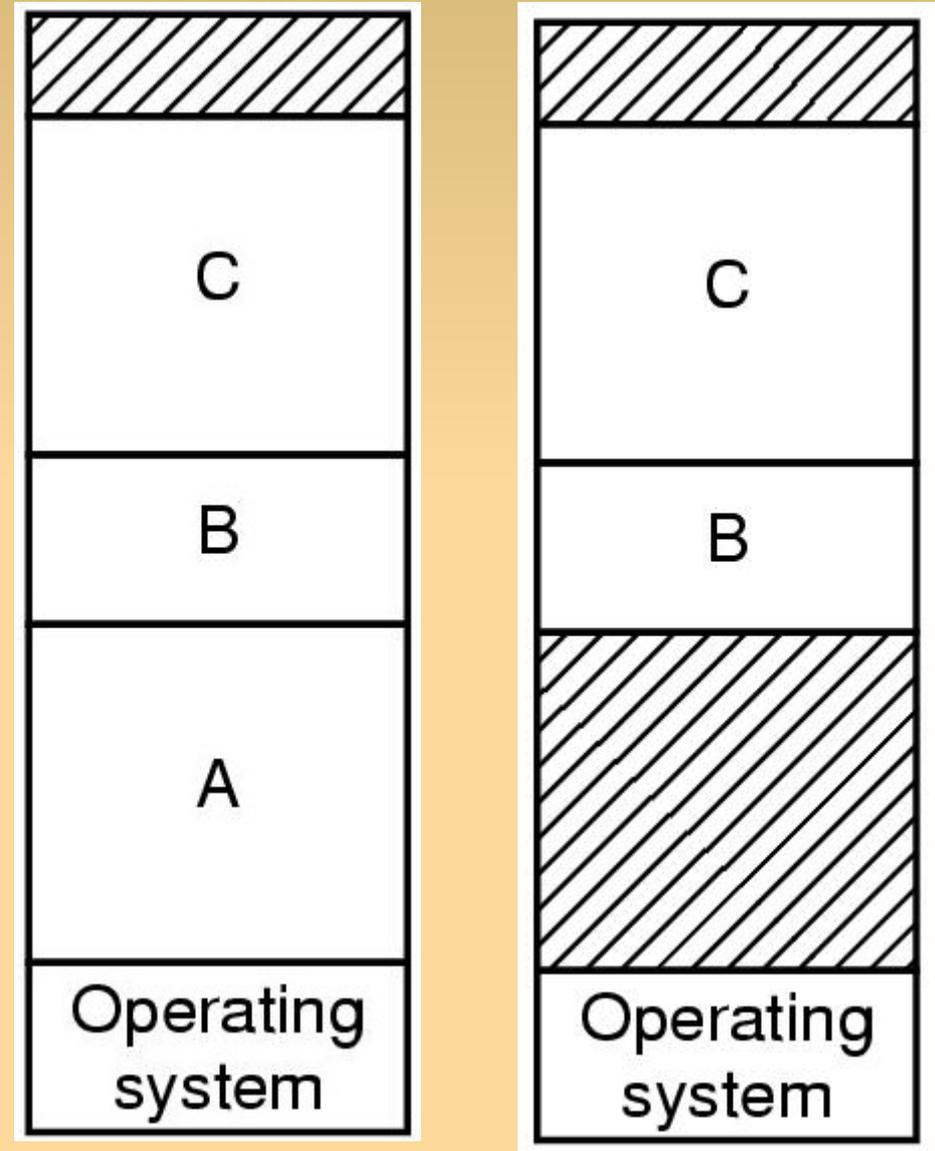
# Swapping

- Operação:
  - Inicialmente, somente o processo A está na memória
  - Então B ou é criado ou trazido de volta (swap in) à memória



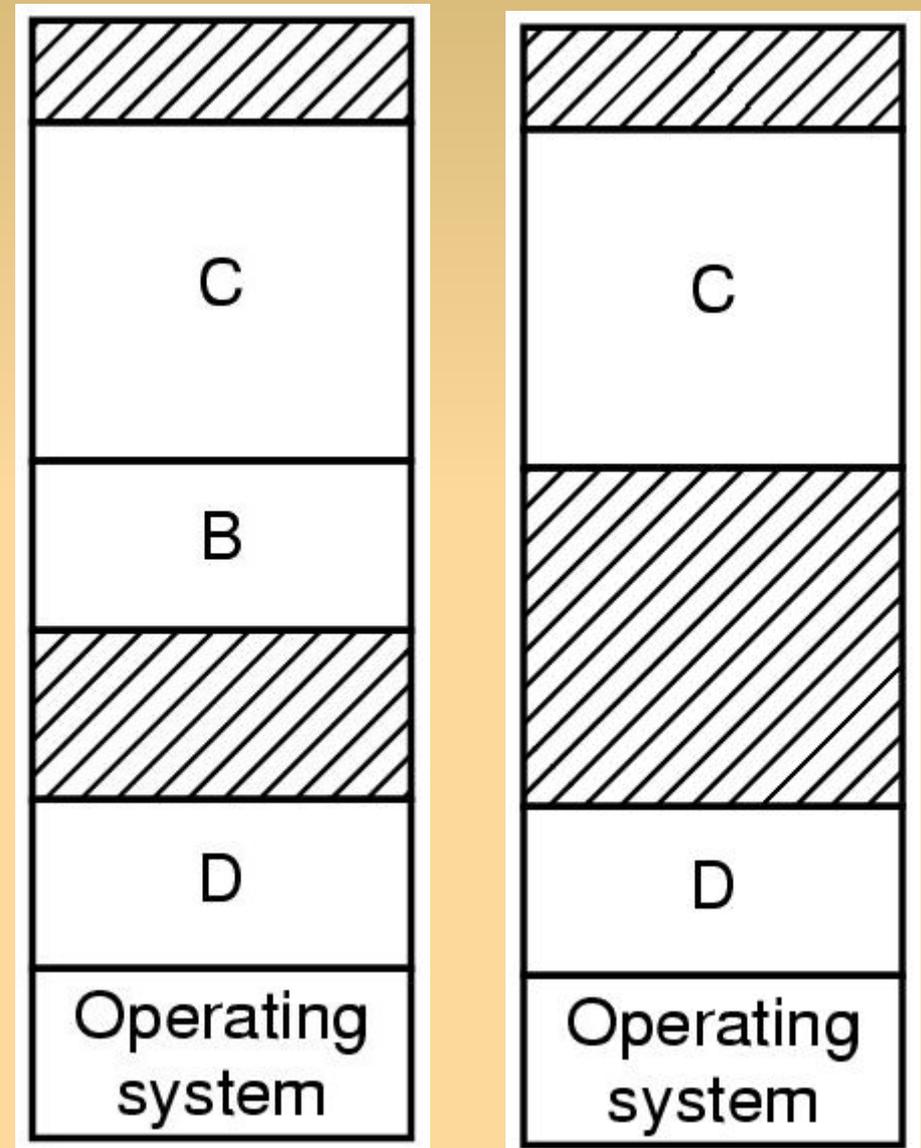
# Swapping

- Operação:
  - Em seguida, C é criado ou trazido de volta (swap in) à memória
  - A então é levado ao disco (swap out)



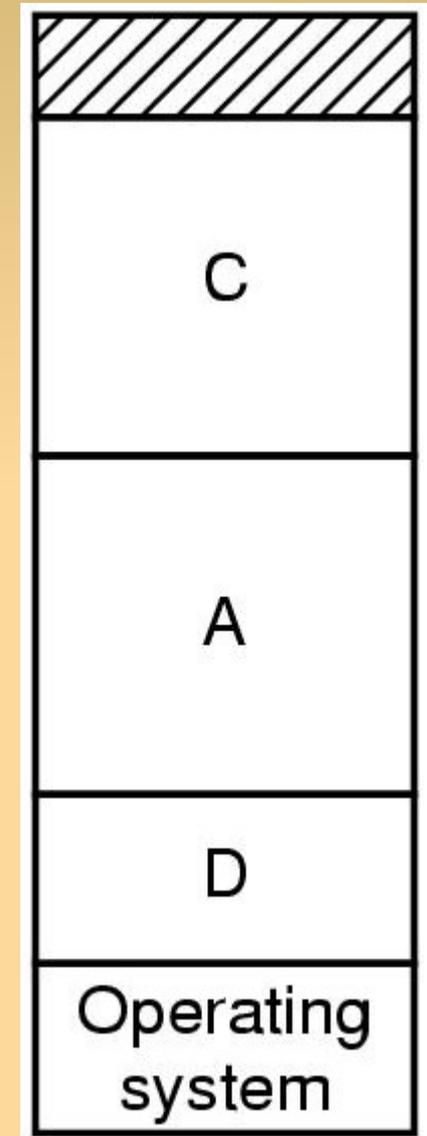
# Swapping

- Operação:
  - E D é trazido à memória (ou iniciado)
  - B então sai



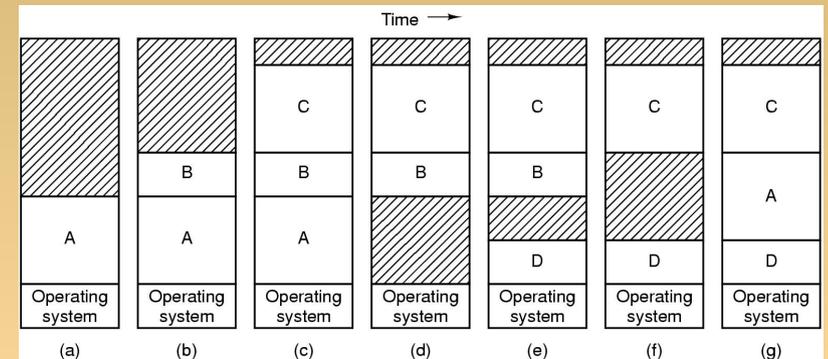
# Swapping

- Operação:
  - E A é trazido de volta
- Note que:
  - A agora está em outra porção da memória → devemos relocar os endereços
  - As partições são de tamanho variável



# Swapping

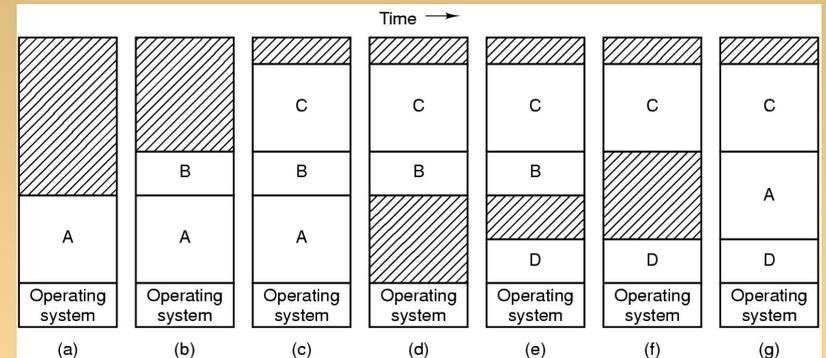
- Problema
  - A operação de swap pode criar muitos “buracos” na memória
    - Problema de fragmentação externa
- Solução: ???



# Swapping

- Problema

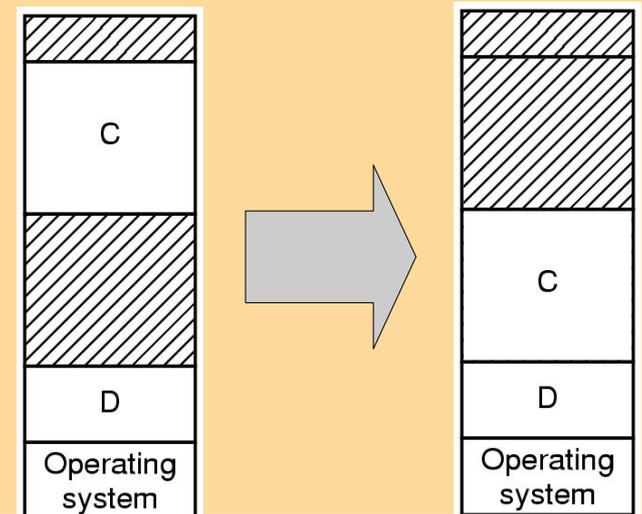
- A operação de swap pode criar muitos “buracos” na memória



- Problema de fragmentação externa

- Solução: Compactação de memória:

- Mova todos os processos o mais para baixo possível na memória – haverá um único espaço vazio acima
- Consome bastante CPU



# Swapping – Alocando Memória

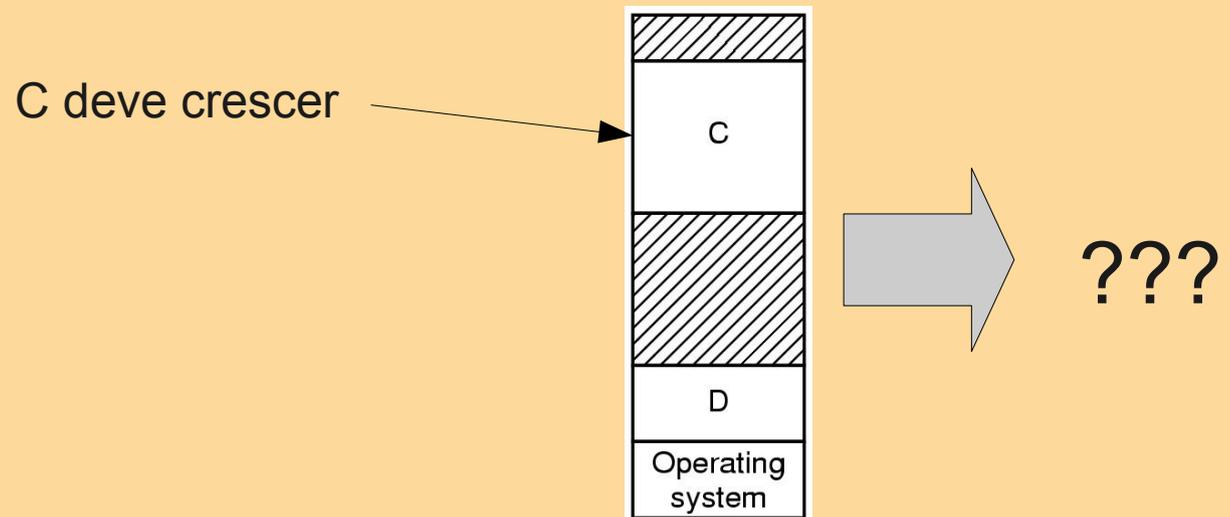
- A alocação de memória muda à medida em que
  - Os processos chegam à memória
  - Os processos deixam a memória
- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?

# Swapping – Alocando Memória

- A Alocação de memória muda à medida em que
  - Os processos chegam à memória
  - Os processos deixam a memória
- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Se eles tiverem tamanho fixo, aloque o que ele precisa

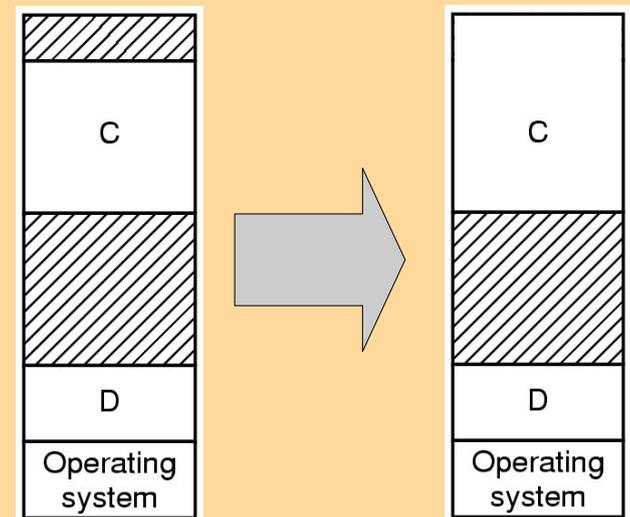
# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Mas e se o segmento de dados deles cresce com o tempo?



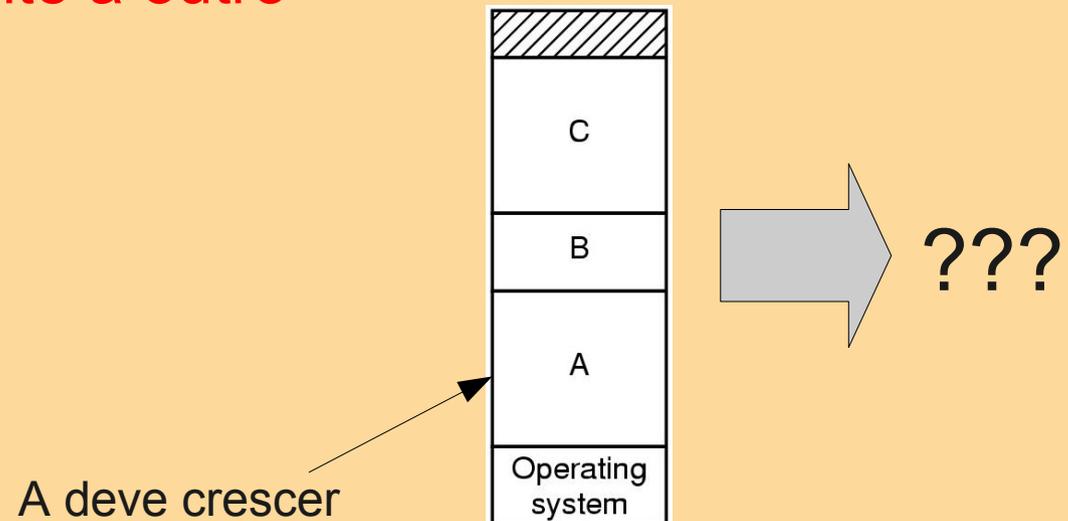
# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Mas e se o segmento de dados deles cresce com o tempo?
    - Se houver um “buraco” adjacente à memória atual do processo, ele pode ser alocado, e o processo cresce nesse buraco



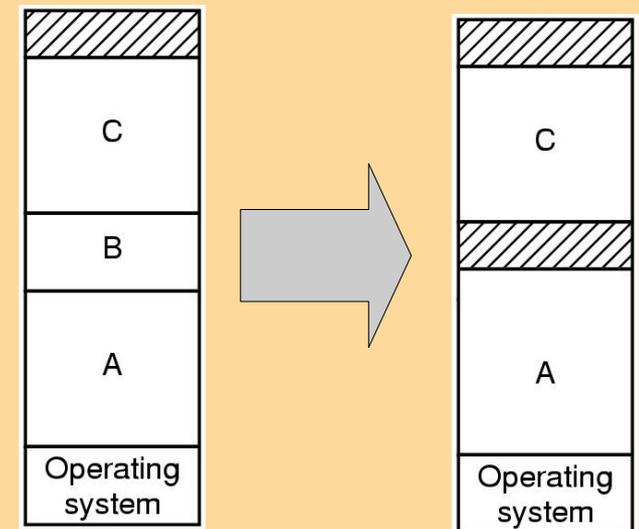
# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Mas e se o segmento de dados deles cresce com o tempo?
    - E se ele for adjacente a outro processo?



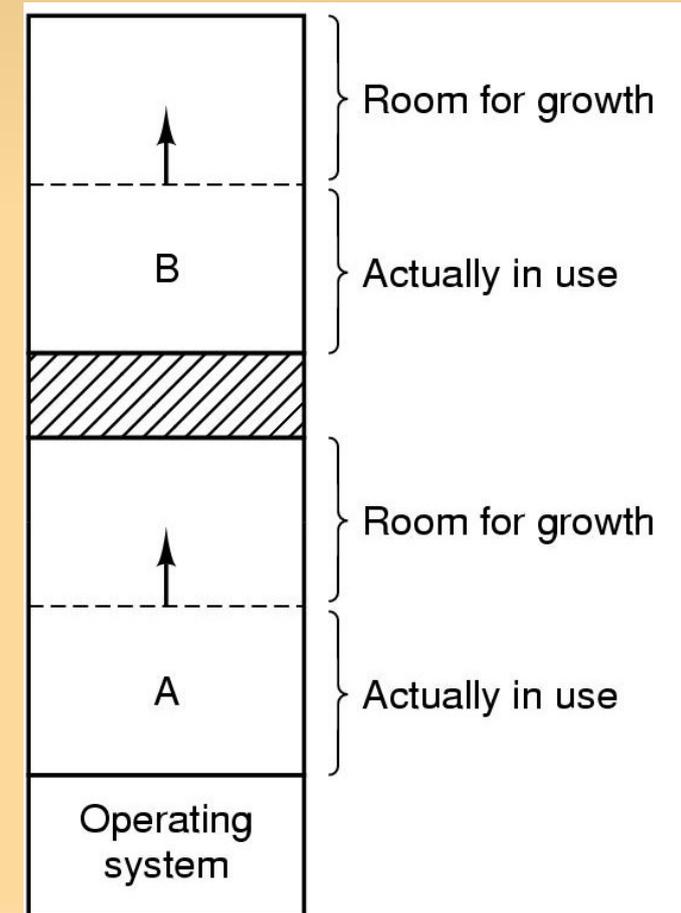
# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Mas e se o segmento de dados deles cresce com o tempo?
    - E se ele for adjacente a outro processo?
      - Ou o movemos a um buraco maior na memória
      - Ou um ou mais processos terão que ser removidos para criar esse buraco



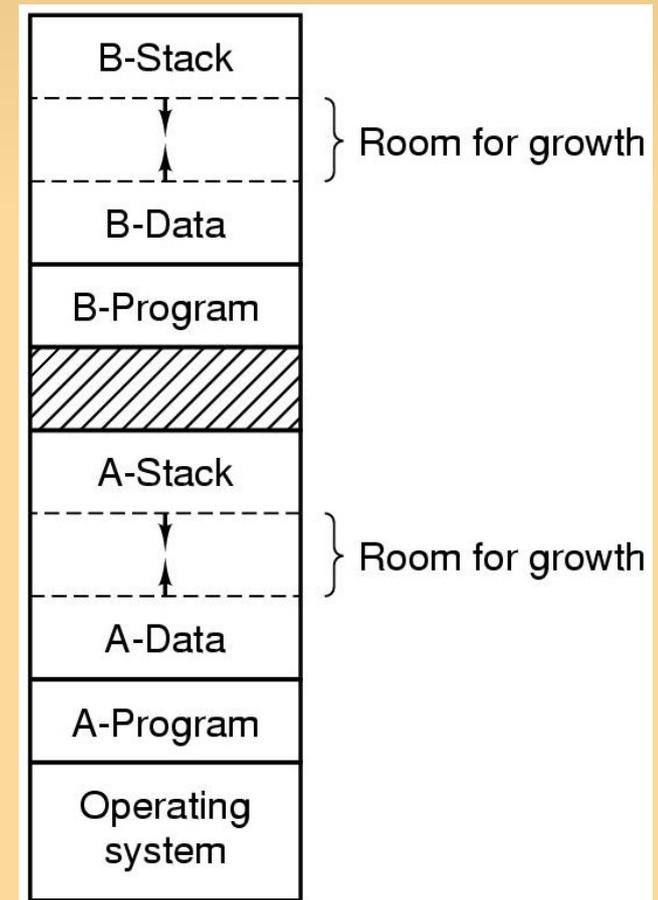
# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Alternativamente, podemos alocar memória a mais para cada processo carregado
    - Reduz o overhead de ter que fazer swap
    - Se ainda assim o processo for para o disco, somente a memória realmente em uso é gravada



# Swapping – Alocando Memória

- Quanto de memória devemos alocar a um processo quando ele é criado ou trazido a ela?
  - Se os processos tiverem dois segmentos que crescem (dados e pilha, por exemplo), podemos usar a mesma idéia
    - Nesse caso, a pilha cresce para baixo enquanto que o segmento de dados cresce para cima
  - Se ainda assim ficar sem espaço:
    - Swap, como antes



# Swapping

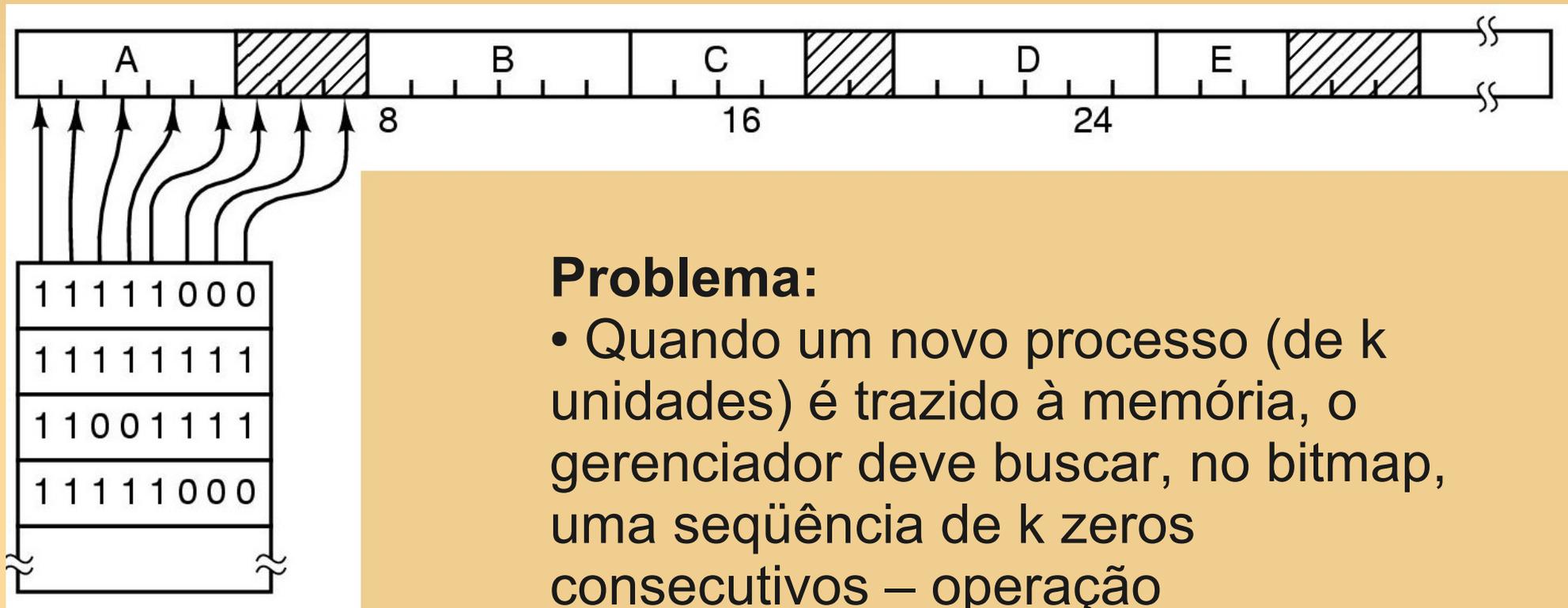
## Técnicas para alocação dinâmica de memória

- Mapas de Bits (Bitmaps):
  - Memória é dividida em unidades de alocação
    - Pode conter vários KB
  - Cada unidade corresponde a um bit no bitmap:
    - 0 → livre
    - 1 → ocupado
  - Tamanho do bitmap depende do tamanho da unidade e do tamanho da memória;
    - unidades de alocação pequenas → bitmap grande;
    - unidades de alocação grandes → perda de espaço;

# Swapping

## Técnicas para alocação dinâmica de memória

- Mapas de Bits (Bitmaps):



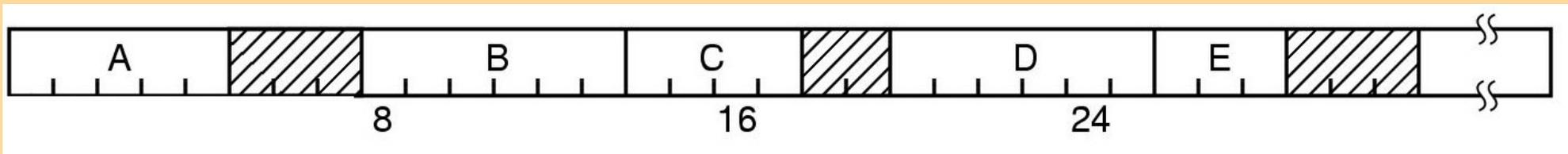
### Problema:

- Quando um novo processo (de k unidades) é trazido à memória, o gerenciador deve buscar, no bitmap, uma seqüência de k zeros consecutivos – operação potencialmente lenta

# Swapping

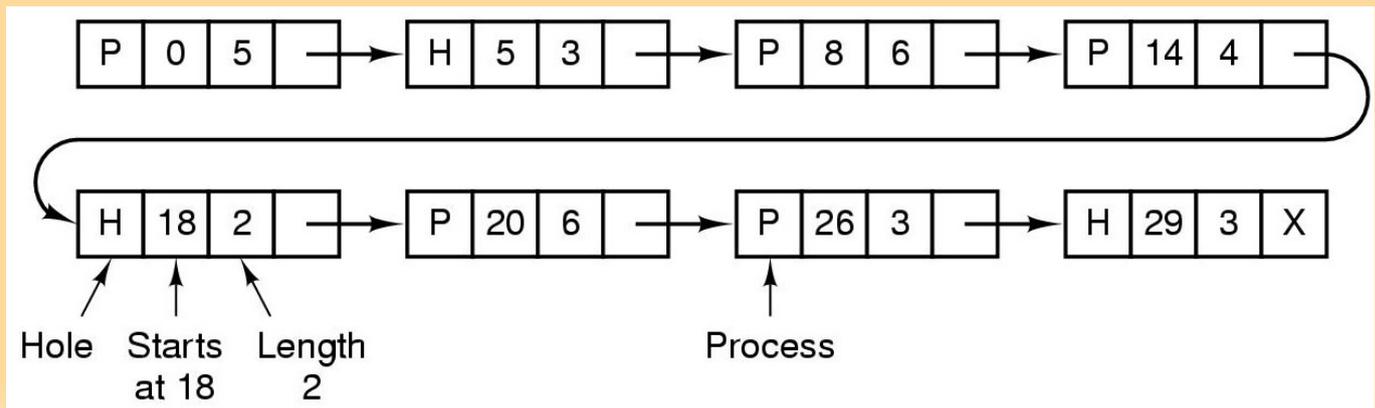
## Técnicas para alocação dinâmica de memória

- Listas Ligadas:
  - Manter uma lista ligada de segmentos de memória livres e alocados
    - Cada segmento ou contém um processo ou é um buraco vazio entre dois processos



Memória

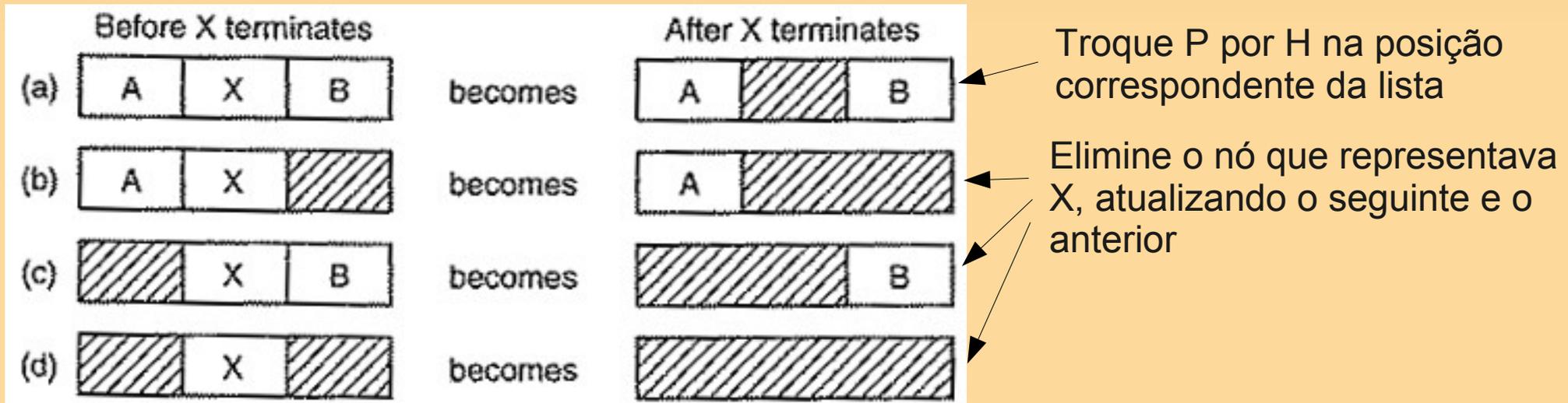
Representação:  
lista ordenada por  
endereço



# Swapping

## Técnicas para alocação dinâmica de memória

- Listas Ligadas:
  - A lista ordenada por endereço fica fácil de ser atualizada:



# Swapping

## Técnicas para alocação dinâmica de memória

- Existem três métodos que podem ser usados para selecionar uma região para um processo.
  - Se a lista estiver ordenada por endereço
- Os algoritmos de alocação são:
  - Primeira escolha (First fit):
    - Percorre a lista de segmentos até que encontre um buraco grande o bastante
    - Quebre o buraco em dois pedaços – um para o processo e um para a memória não usada
    - Variação: inicie a busca a partir de onde parou, na vez anterior (next fit)

# Swapping

## Técnicas para alocação dinâmica de memória

- Os algoritmos de alocação são:
  - Melhor escolha (Best fit):
    - Busca a lista inteira, até o fim, e toma o menor buraco que seja adequado
    - Não quebra buracos grandes que poderiam ser úteis mais tarde
    - Problema: permite o surgimento de buracos minúsculos
  - Pior escolha (worst fit):
    - Sempre tome o maior buraco disponível
    - Reduz a chance de buracos minúsculos e inúteis

# Swapping

## Técnicas para alocação dinâmica de memória

Dois processos, **P** e **Q**, são alocados nessa ordem:

14kbytes (P)

20kbytes (Q)

Sentido em que a memória é percorrida



Áreas livres iniciais

Melhor Escolha

Pior Escolha

Primeira Escolha

# Swapping

## Técnicas para alocação dinâmica de memória

Dois processos, **P** e **Q**, são alocados nessa ordem:

14kbytes (P)

20kbytes (Q)

Sentido em que a memória é percorrida



Áreas livres iniciais



Melhor Escolha

Pior Escolha

Primeira Escolha

# Swapping

## Técnicas para alocação dinâmica de memória

Dois processos, **P** e **Q**, são alocados nessa ordem:

14kbytes (P)

20kbytes (Q)

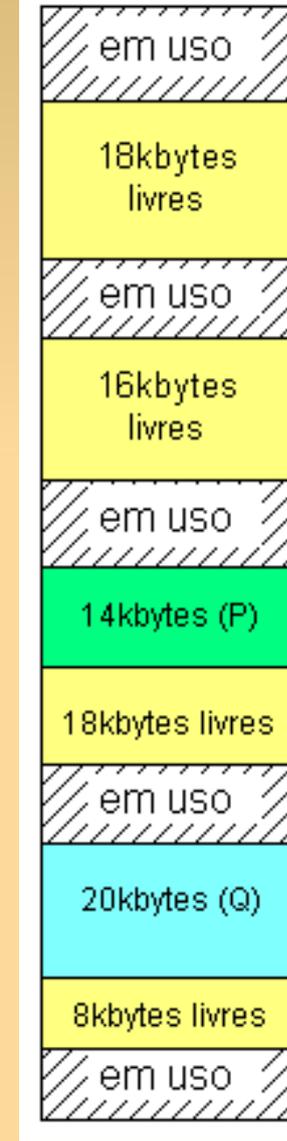
Sentido em que a memória é percorrida



Áreas livres iniciais



Melhor Escolha



Pior Escolha

Primeira Escolha

# Swapping

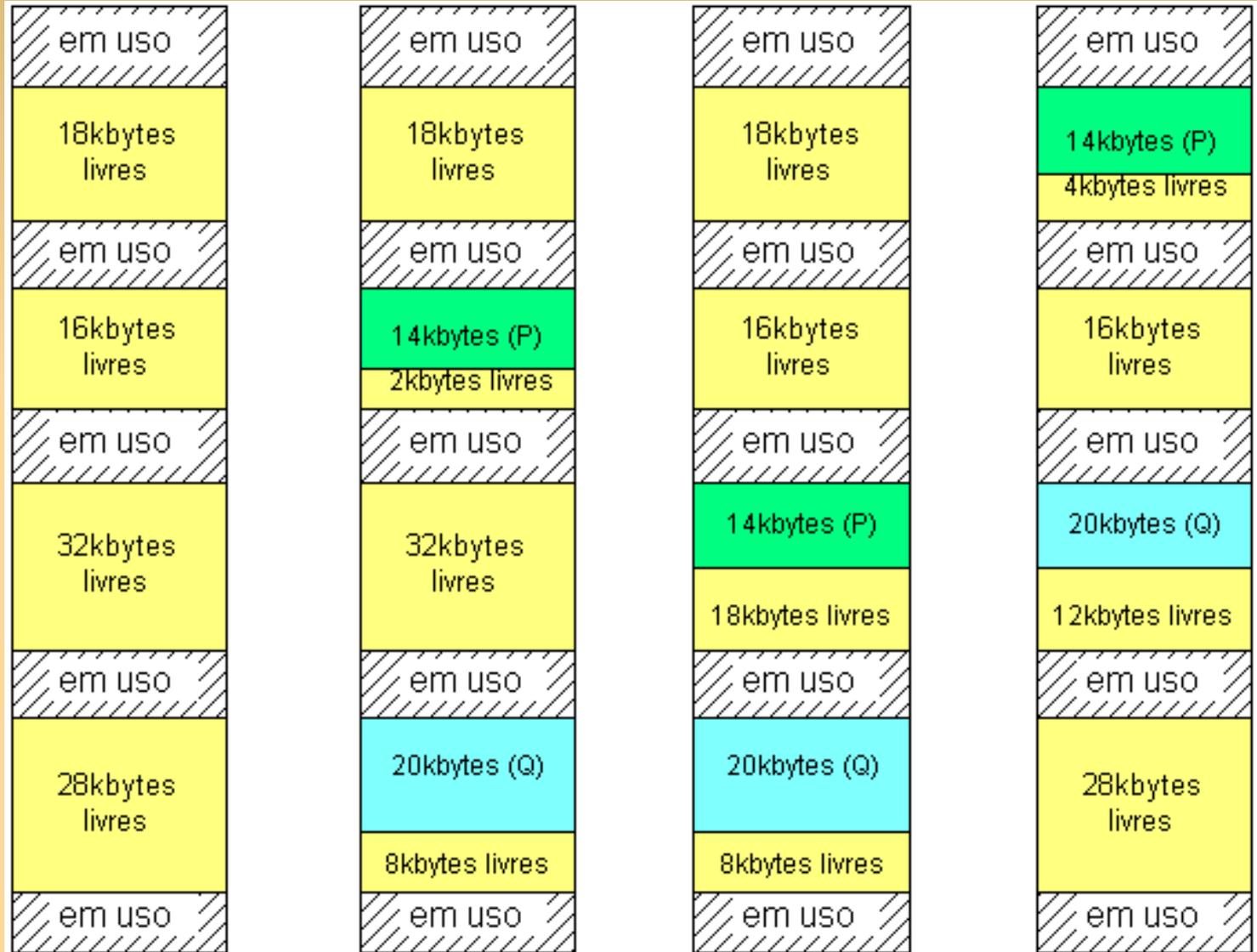
## Técnicas para alocação dinâmica de memória

Dois processos, **P** e **Q**, são alocados nessa ordem:

14kbytes (P)

20kbytes (Q)

Sentido em que a memória é percorrida



Áreas livres iniciais

Melhor Escolha

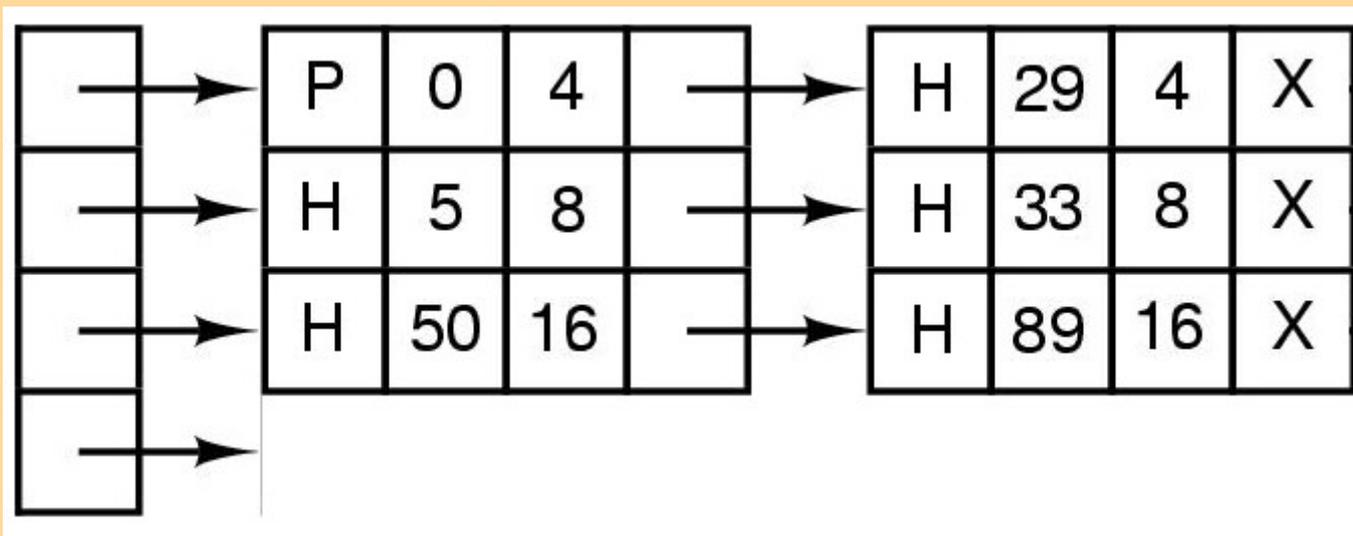
Pior Escolha

Primeira Escolha

# Swapping

## Técnicas para alocação dinâmica de memória

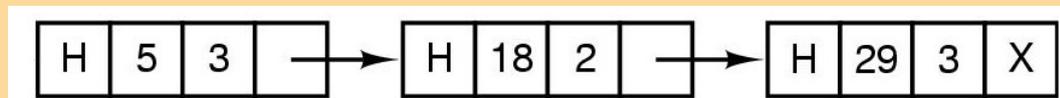
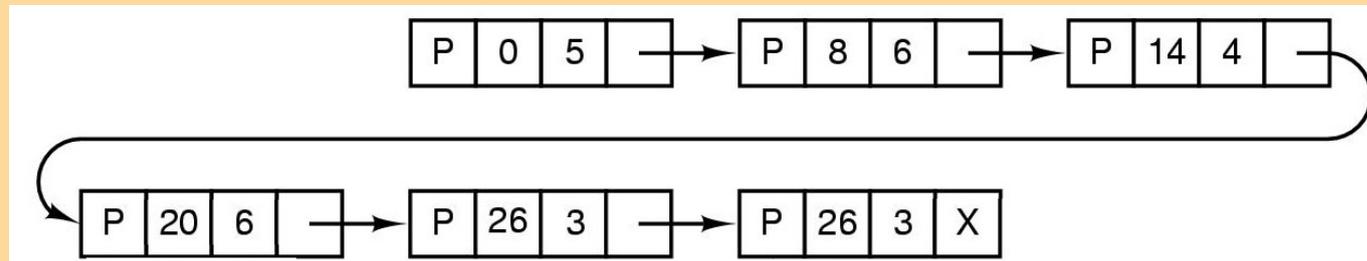
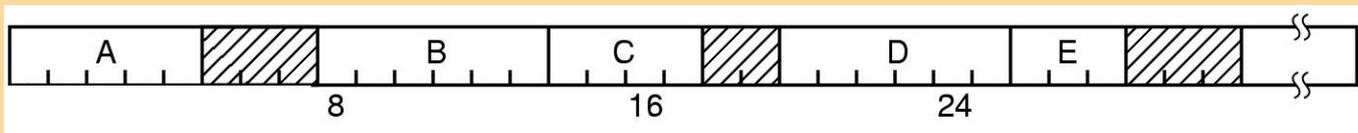
- Os algoritmos de alocação são:
  - Quick fit
    - Mantém listas separadas para alguns dos tamanhos mais comumente requisitados
      - Uma para cada tamanho
    - Problema quando processo é desalocado:
      - Encontrar os vizinhos ao buraco que ele deixou, para união



# Swapping

## Técnicas para alocação dinâmica de memória

- Algoritmos de alocação:
  - Todos podem ser melhorados se mantivermos buracos e processos em listas separadas
    - Há ganho de desempenho
    - Ainda assim, quando um processo é retirado da memória, mesclar buracos é de alta complexidade.



# Swapping

## Técnicas para alocação dinâmica de memória

- Principais Conseqüências dos algoritmos:
  - Melhor escolha: deixa o menor resto, porém após um longo processamento poderá deixar “buracos” muito pequenos para serem úteis.
  - Pior escolha: deixa o maior espaço após cada alocação, mas tende a espalhar as porções não utilizadas sobre áreas não contínuas de memória e, portanto, pode tornar difícil alocar grandes jobs.
  - Primeira escolha: tende a ser um meio termo entre a melhor e a pior escolha, com a característica adicional de fazer com que os espaços vazios migrem para o final da memória.

# Memória Virtual

- O tamanho da memória cresce rapidamente
  - O tamanho do software cresce muito mais rápido
  - Há necessidade de rodar programas grandes demais para a memória
- Programas fazem referência à memória
  - Ex:
    - `MOV REG, 1000`
  - Não necessariamente esse é o endereço real.

# Memória Virtual

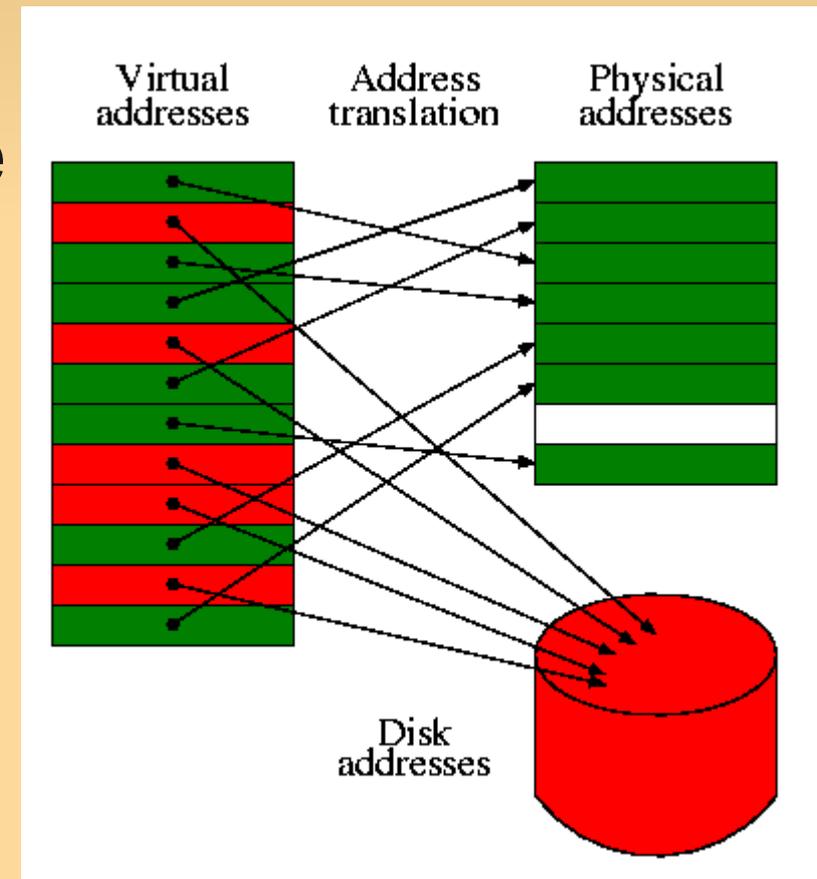
- Os endereços físicos (reais) podem ser gerados de muitas maneiras:
  - Mapeamento direto
  - Usando registradores base
  - etc
- Endereços referenciados pelos programas são chamados de virtuais
  - Na ausência de memória virtual, o endereço virtual bate com o físico
  - Com memória virtual, é tratado pela MMU

# Memória Virtual

- Espaço de Endereçamento Físico de um processo:
  - Formado por todos os endereços físicos/reais aceitos pela memória principal (RAM);
- Espaço de Endereçamento Virtual de um processo:
  - Formado por todos os endereços virtuais que esse processo pode gerar;

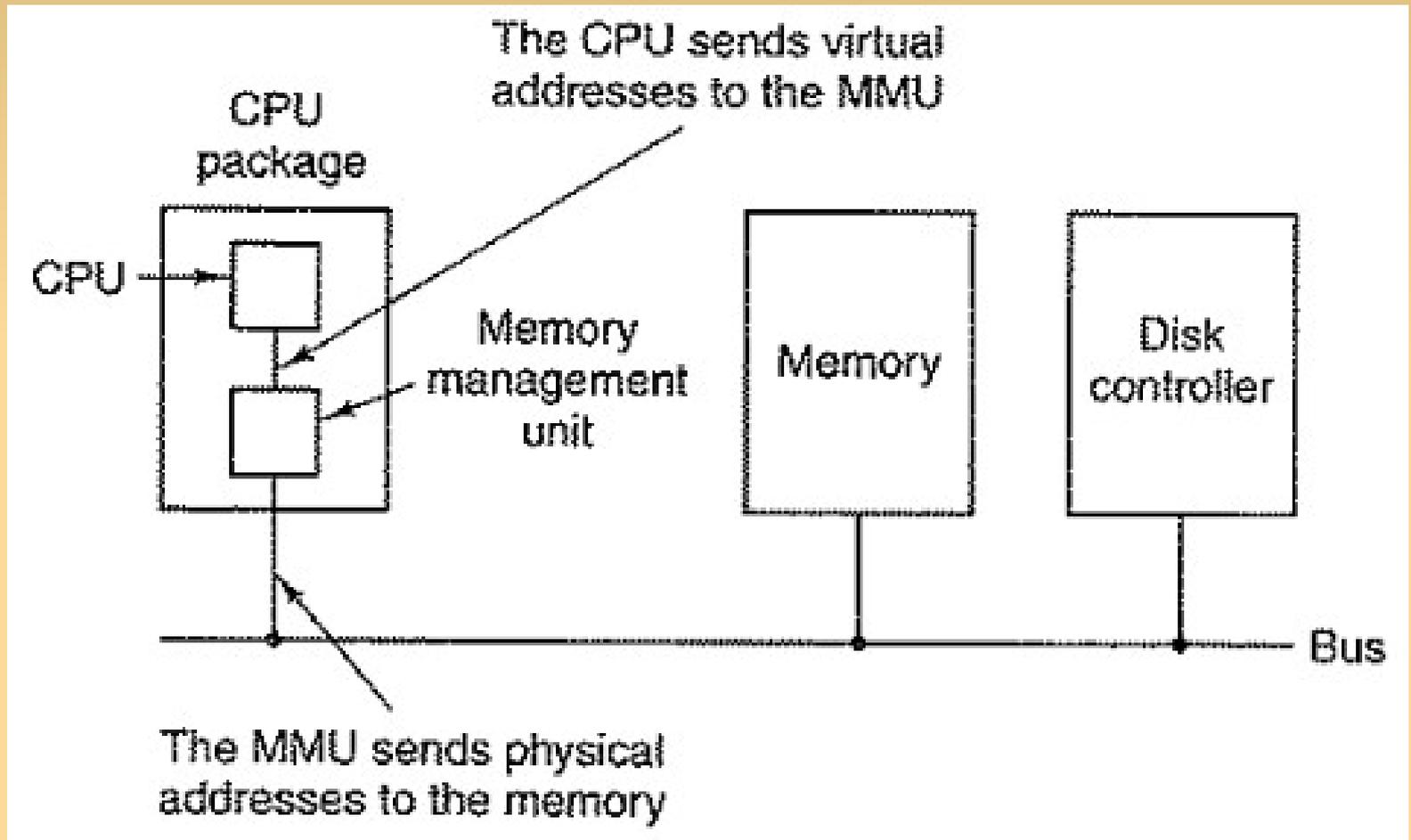
# Memória Virtual

- Um processo em Memória Virtual faz referência a endereços virtuais e não a endereço reais de memória RAM;
- No momento da execução de uma instrução, o endereço virtual é traduzido para um endereço real, pois a CPU manipula apenas endereços reais da memória RAM → MAPEAMENTO;



# Memória Virtual

A MMU mapeia o endereço virtual à memória física



# Memória Virtual

- Memória virtual é então uma técnica para troca de processos na memória
  - Programas maiores que a memória eram divididos em pedaços menores chamados overlays
    - Solução adotada nos anos 60
  - Quando um programa iniciava, tudo que é carregado na memória é o gerenciador de overlay
    - Imediatamente carrega e roda o overlay 0
  - Vantagem: expansão da memória principal;
  - Desvantagem: custo muito alto;

# Memória Virtual

- Memória virtual é uma técnica para troca de processos na memória
  - Sistema operacional era responsável por:
    - Realizar o chaveamento desses pedaços entre a memória principal e o disco;
    - Cabia ao programador dividir o programa em overlays;
- Memória Virtual automatiza a divisão em overlays
  - Cada programa tem seu próprio espaço de endereços, que é quebrado em porções → as páginas

# Memória Virtual

- Páginas:
  - Quando o programa referencia uma parte de seu endereço que está na memória, o hardware faz o mapeamento na hora
  - Se, contudo, essa parte não estiver na memória:
    - O S.O. traz do disco o pedaço que falta
    - Executa novamente a instrução que falhou
    - Enquanto isso, o escalonador pode colocar outro processo para rodar
- Memória virtual é semelhante ao swap
  - Swap aloca espaço para todo o processo, MV não

# Memória Virtual

- Técnicas de MV:
  - Paginação:
    - Blocos de tamanho fixo (endereços contínuos) chamados de páginas;
    - SO mantém uma fila de todas as páginas;
    - Endereços Virtuais formam o espaço de endereçamento virtual;
    - O espaço de endereçamento virtual é dividido em páginas virtuais;
    - Mapeamento entre endereços reais e virtuais (MMU);

# Memória Virtual

- Técnicas de MV:
  - Segmentação:
    - Blocos de tamanho arbitrário chamados segmentos;
- Arquitetura (hardware) tem que possibilitar a implementação tanto da paginação quanto da segmentação;