

# Sistemas Operacionais

## Métodos de Alocação

Acadêmicos: Caciano D. Mattiello  
Eduardo dos Santos  
Orientador: Prof. Msc. Bruno Ribas

# Métodos de Alocação

## **Motivo:**

Quando é necessário armazenar muitos arquivos no mesmo disco, surge o problema de alocação de espaço para esses arquivos, de modo que o disco seja efetivamente utilizado e os arquivos possam ser rapidamente acessados.

## **Métodos:**

- Contíguo;
- Encadeado;
- Indexado.

# Alocação Contígua

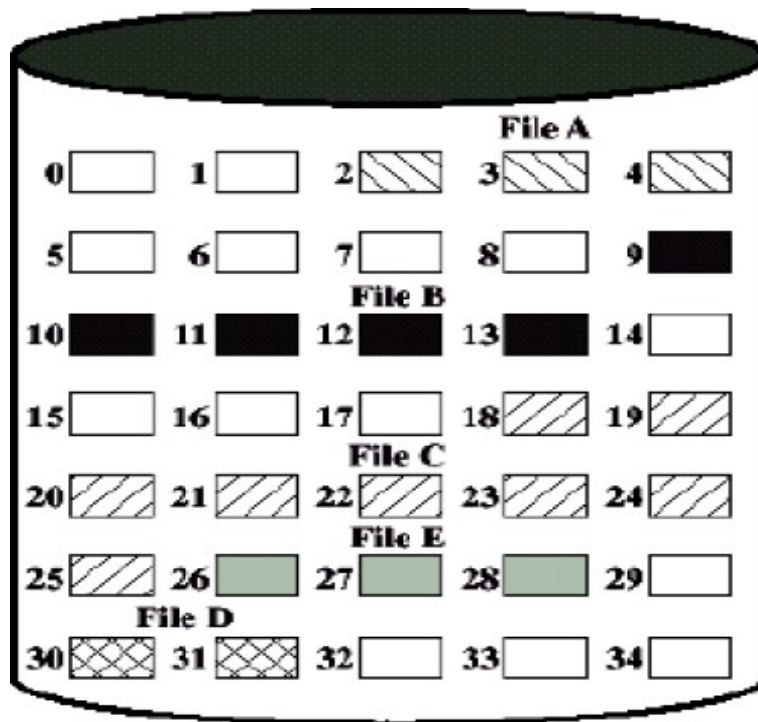
- Neste método cada arquivo ocupa um conjunto de blocos contíguos no disco. Assim o número de operações para acessar os arquivos que foram alocados contiguamente é mínimo.

É a melhor escolha quando:

- Arquivos não são deletados do disco;
- O tamanho dos arquivos é fixo.

## Alocação Contígua (2)

- Para a alocação de um arquivo, se ele tiver  $n$  blocos e iniciar na localização  $b$ , então ele irá ocupar os blocos  $b$ ,  $b+1$ ,  $b+2$  até  $b+n-1$ .
- A entrada de diretório para cada arquivo indicará o endereço do bloco inicial e o tamanho da área alocada para o arquivo.

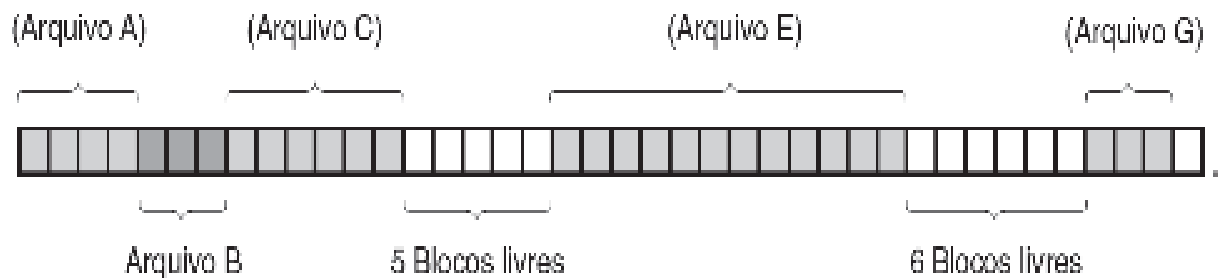
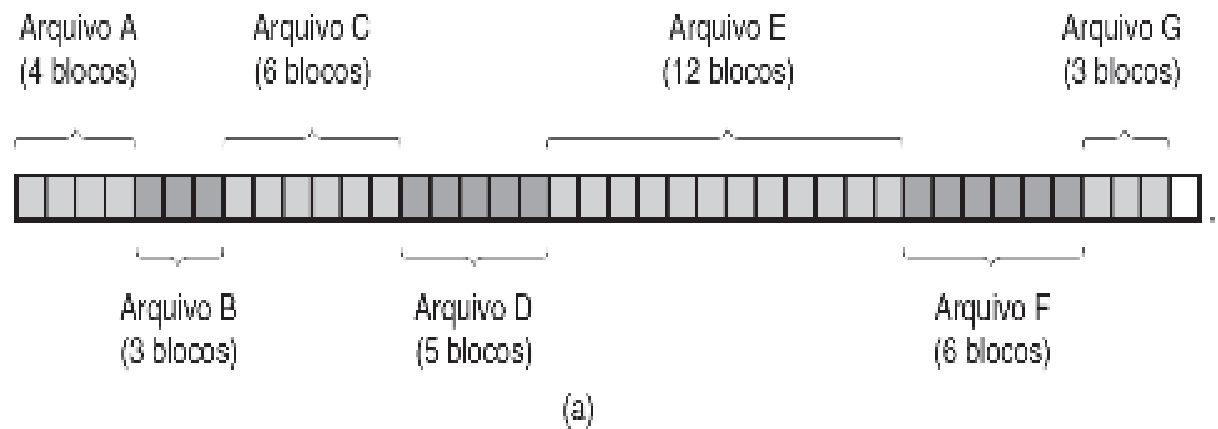


**File Allocation Table**

File Name	Start Block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

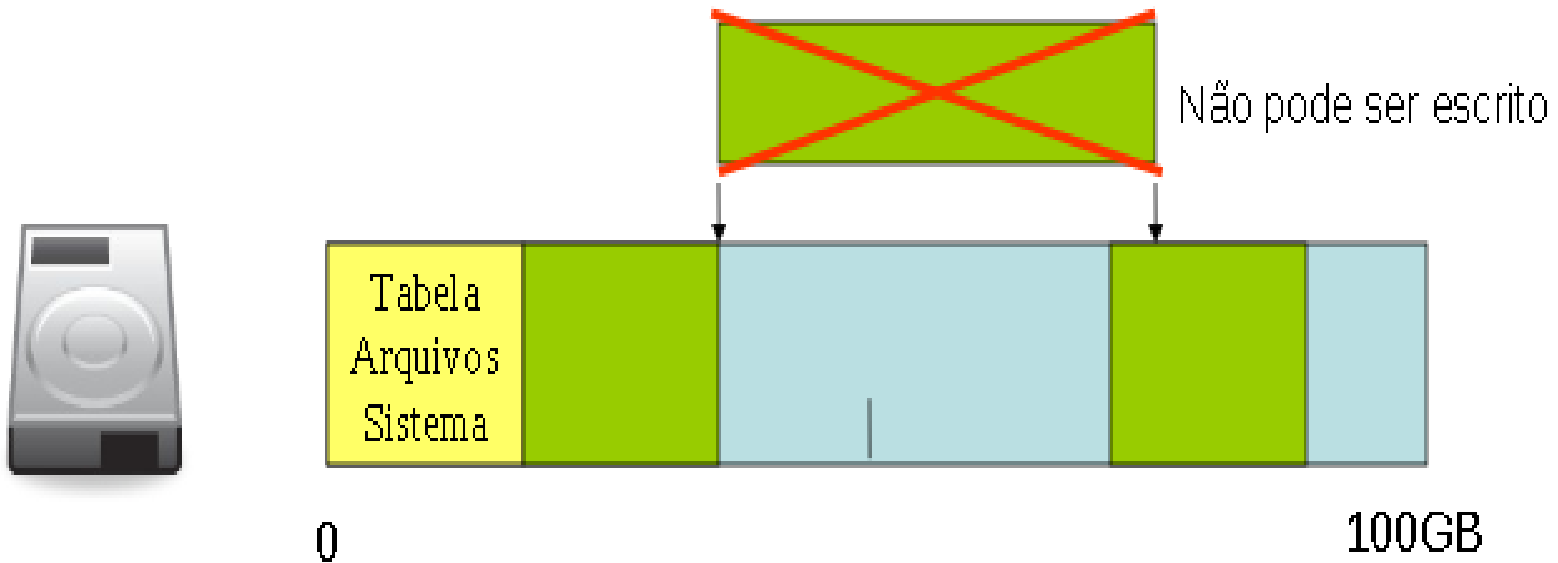
# Alocação Contígua

Para resolver o problema de fragmentação externa utiliza-se a compactação, porem a operação pode levar horas.



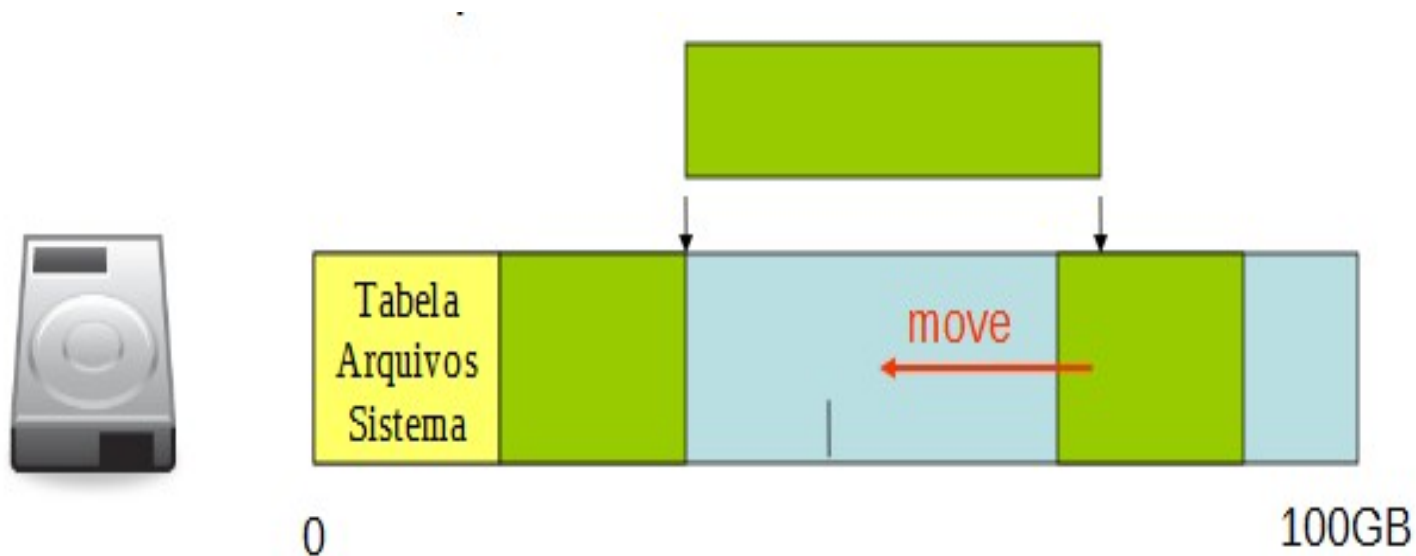
# Alocação Contígua

Como resolver a fragmentação externa?



# Alocação Contígua

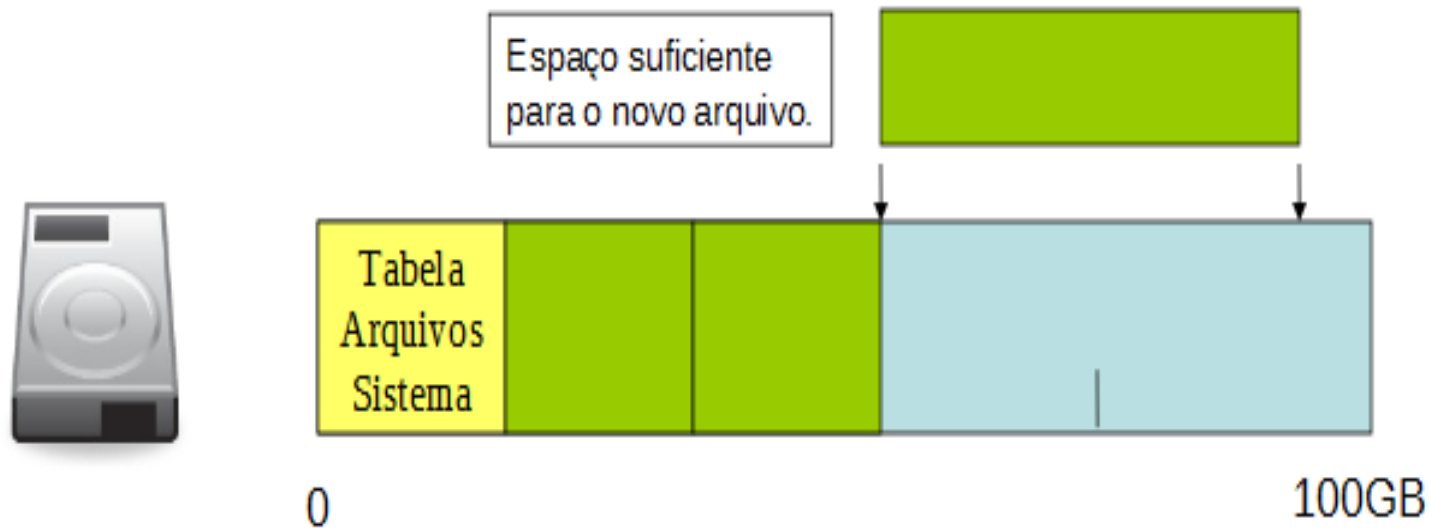
Para remover a fragmentação externa é necessário realizar uma desfragmentação.



# Alocação Contígua

Porém, é muito cara

- A ação pode ser lenta, pode envolver todos os arquivos no disco.
- O disco torna-se não disponível durante este procedimento





# Alocação Contígua (3)

- Problemas:
  - Tem como dificuldade encontrar espaço para um novo arquivo;
  - Apresenta o problema de fragmentação externa, que é resolvido com o uso de compactação, porém uma operação altamente custosa.
- Vantagens:
  - Cada arquivo ocupa um conjunto contíguo de blocos no disco, assim o número de operações para acessar os arquivos que foram alocados contiguamente é mínimo.
- Quando usar?
  - Quando tamanho de arquivos forem fixos;
  - Os arquivos não são deletados do disco;  
Por exemplo, CD-ROM ( ISO9660).

## Alternativa ao Sistema Contíguo

- Fixar os tamanhos de alocação e vazio em blocos pequenos e iguais;
- Quebrar um novo arquivo em blocos pequenos ;
- Colocar os novos arquivos no espaço vazio bloco a bloco.

# Alocação Encadeada

- Este método é utilizar lista encadeadas para alocação;
- Pode-se utilizar a lista também para gerenciar o espaço livre.

## Alocação encadeada (2)

- Vantagens:
  - Evita fragmentação externa, um novo arquivo pode ser colocado no espaço vazio disponível efetivamente;
  - Fácil de implementar;
  - Tabela de arquivos do sistema menor em relação ao contíguo;
  - Arquivos poder alterar seu tamanho sem muito custo.
- Contras:
  - Acesso aleatório é custoso com grande número de acessos a disco;
  - Gera fragmentação interna.

## Melhorando a alocação encadeada

- Agrupa os dados da lista em uma tabela;
- As entradas na tabela são armazenadas contiguamente;
- Cada entrada corresponde a um bloco de alocação do arquivo no dispositivo de armazenamento;
- O tamanho da tabela depende da capacidade do dispositivo e do tamanho do bloco;
- Método conhecido como FAT (File Allocation Table).

# FAT

- Vantagens:
  - Tabela pode ser carregada em memória para maior velocidade de busca de blocos;
  - Evita o problema do custo para acesso aleatório.
- Contras:
  - Tabela do FAT é grande;
  - Fragmentação interna.

# Alternativa

- Novamente dividir para conquistar!
- Ao invés de carregar toda a tabela em memória para localizar os blocos de dados, particionamos ela em pequenos pedaços;
- Assim somente poucos pedaços da tabela precisam ser lido.

## Alocação Indexada (2)

- Vantagens
  - Duas operações de leitura localizam todos os blocos de dados de um arquivo;
  - Não necessita tanta memória alocada como na FAT.
- Contras
  - Necessidade do tamanho do inode ser variável, dificultando uma leitura eficiente na leitura desta tabela.



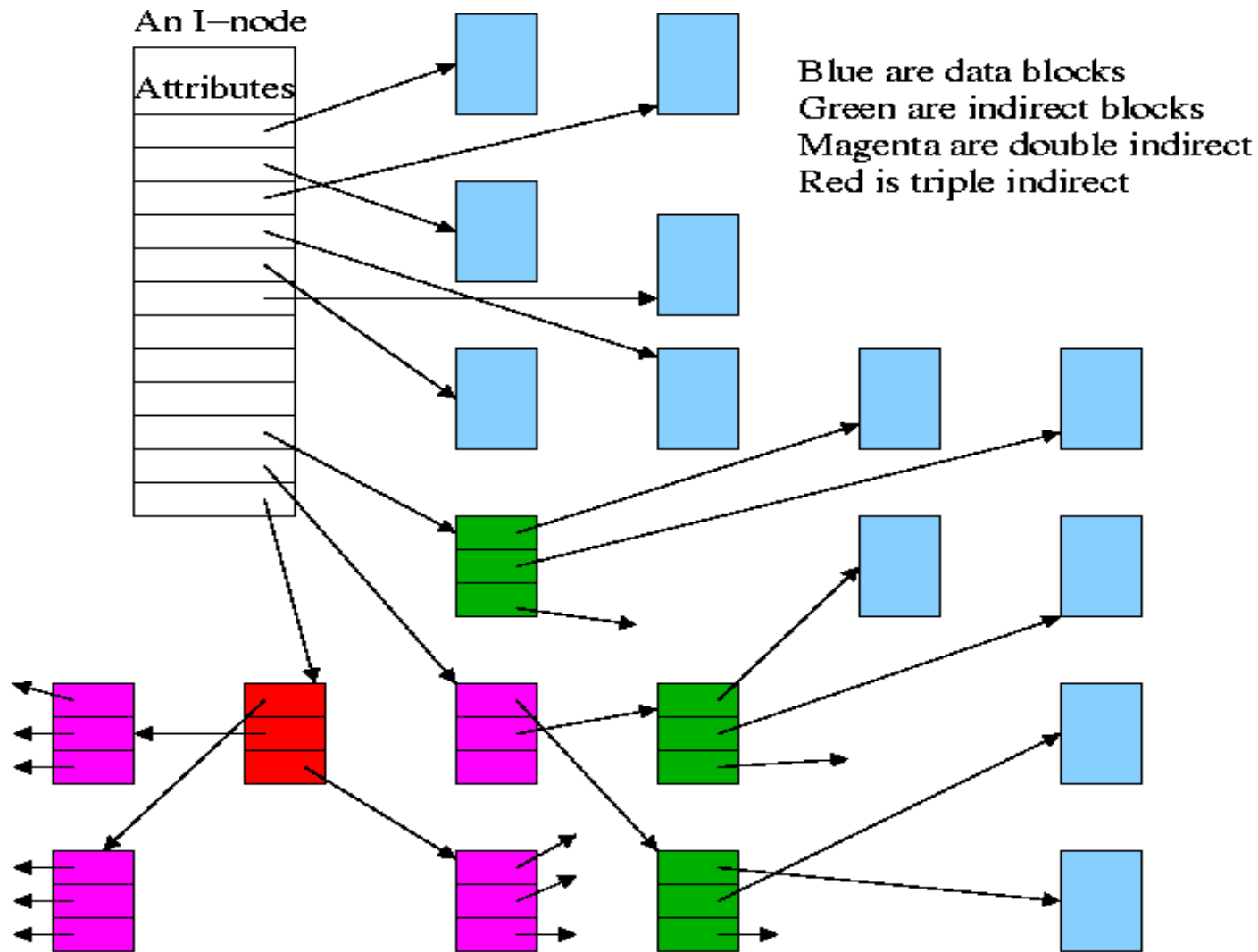
# Outra alternativa com indexação

- Dividir mais ainda!
- Funcionamento:
  - Cada inode terá tamanho **FIXO**;
  - Dividimos o sistema em Blocos Diretos, Bloco indireto, Bloco duplo indireto e Bloco triplo indireto;
  - Primeiramente é usado os blocos diretos (inodes) para guardar os endereços dos blocos de dados;
  - Depois de usados os endereços diretos, começa a usar os blocos indiretos.

# Alocação indexada direta/indireta

- Bloco de inode de endereços indireto:
  - Bloco que aponta para outro bloco com endereços de blocos de dados;
  - Assim se o bloco tiver 4 bytes e um bloco de dados tem 1 Kbytes, o total sera de 256 endereços para blocos de dados.
- Aumentado para duplamente indireto:
  - Assim como no bloco indireto, o segundo bloco apontará para outro de endereços para blocos de dados, aumentando a quantidade de blocos alocáveis para um arquivo possível ainda mais.

# Alocação Indexada com blocos indiretos



# Alocação Indexada com blocos indiretos

- Vantagens:

- Inode tem tamanho fixo;
- Aumenta significativamente o tamanho máximo de um arquivo!

Ex: Considerando tamanho de um endereço de blocos 4 bytes ( $2^2$ ) e tamanho de bloco 4 Kbytes ( $2^{12}$ ) e:

- Blocos Diretos: 12
- Bloco Indireto: 1
- Bloco Duplamente Indireto: 1
- Bloco Triplamente Indireto: 1

Teríamos  $12 \cdot 2^2 + 2^{(2 \cdot 12 - 2)} + 2^{(3 \cdot 12 - 4)} + 2^{(4 \cdot 12 - 6)} = 4$  TBytes tamanho máximo para um arquivo.

# Sistemas de alocação usados atualmente

- Maioria dos sistemas hoje utilizam indexação.

## Sistemas e métodos de alocação:

- ISO9660(CDROM) → contíguo;
- EXT4, XFS (\*NIX) → Indexado;
- FAT32 (Windows anteriores a XP e o XP) → Lista encadeada;
- NTFS (Windows XP e versões mais novas) → Indexado;