

# Sistemas Operacionais

**Professor:**  
**Cristiano Bonato Both**



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Sumário

- Escalonamento de disco
  - Objetivos
  - Requisitos
  - Estrutura
  - Métodos de acesso
  - Controle de acesso
  - Alocação
- Referências

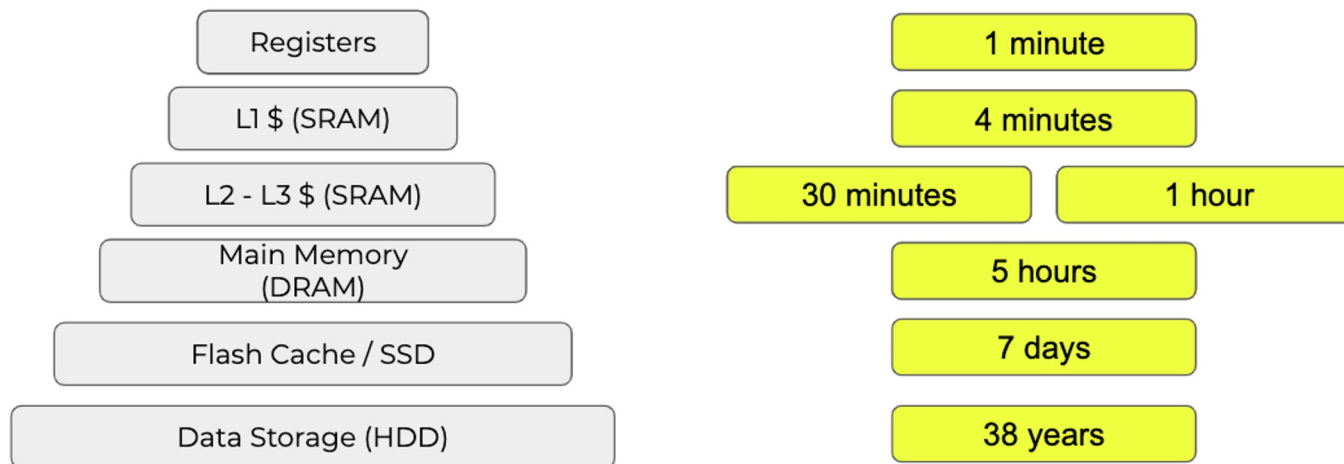


# Relembrando aulas passadas

- Hierarquia de memória
  - Situação hipotética: 1 ciclo = 1 minuto

## MEMORY HIERARCHY

1 cycle = 1 minute



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Objetivos do Sistema de Arquivos

- Criar um recurso lógico a partir de recursos físicos, através de uma interface coerente e fácil de usar
- Garantir a validade e coerência de dados
- Otimizar o acesso
- Fornecer suporte a outros sistemas de arquivos
- Suporte a vários usuários (multiprogramação)



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Conceito de Arquivo

- Informação que pode ser armazenada em diferentes tipos de mídia
  - O sistema operacional deve oferecer uma visão uniforme da informação independente do dispositivo físico de armazenamento
    - Visão lógica é o arquivo
- Arquivos são mapeados para dispositivos físicos
- Arquivos possuem:
  - nome, atributos, estrutura interna, tipo, método de acesso, operações



# Métodos de Acesso

- Acesso sequencial: feito através de chamadas de sistema do tipo *read* e *write*
  - Cada chamada de sistema *read* retorna ao processo os dados seguintes àqueles que foram lidos na chamada anterior
  - Método não adequado a todas aplicações
    - e.g., acesso e atualização a cadastros de funcionários
- Acesso relativo: provê uma chamada de sistema específica para indicar o ponto em que um arquivo deve ser lido/escrito
  - Implementado com abstração de “posição corrente no arquivo”



# Outros Tipos de Acesso

- Os métodos sequenciais e relativos não resolvem todos os tipos de acesso
  - e.g., localizar um registro a partir do conteúdo
- Necessidades de métodos de acesso mais sofisticados, tais como: sequencial indexado, indexado, direto, *hash*, etc.
  - Normalmente, implementados por programas específicos
  - Baseados nos métodos de acesso sequencial e relativo



# Operações Básicas sobre Arquivos

- Arquivo é um tipo abstrato de dados sobre o qual se pode efetuar uma série de operações
  - Criação (*create*)
  - Escrita (*write*) e leitura (*read*)
  - Reposicionamento em um ponto qualquer do arquivo (*file seek*)
  - Remoção (*delete*)
  - Abertura (*open*) e encerramento (*close*)
  - Adicionalmente: truncagem (*truncate*); renomeação (*rename*); *appending*, etc.
- Operações mais complexas podem ser criadas utilizando-se das operações básicas





# Controle de Acesso

- Importante controlar o acesso aos arquivos, devido a questões de segurança e de confidencialidade
- Objetivo é evitar acessos indevidos a arquivos
- Baseado na identificação dos usuários
  - Sistema de autenticação padrão (*login + password*)
  - Usuários possuem direitos de acesso
- Solução típica:
  - Lista de acesso e grupos



# Implementação de Arquivos

- Arquivos são implementados através da criação de uma estrutura de dados
- *Descritor de arquivo* é um registro que mantém informações sobre o arquivo
- Informações típicas (atributos):
  - Nome do arquivo
  - Tamanho em *bytes*
  - Data e hora da criação, do último acesso da última modificação
  - Identificação do usuário que criou o arquivo
  - Listas de controle de acesso
  - Localização no disco físico do arquivo



# Tabelas de Descritores de Arquivos

- Descritores de arquivos são armazenados no próprio disco
- Problema de desempenho
  - Acesso ao disco para ler o descritor de arquivos é lento
  - Solução é manter descritor em memória, enquanto o arquivo estiver em uso
- Sistema de arquivos mantém os descritores de arquivos em memória em uma estrutura de dados do SO
  - *Tabela de Descritores do Arquivo Abertos (TDAA)*

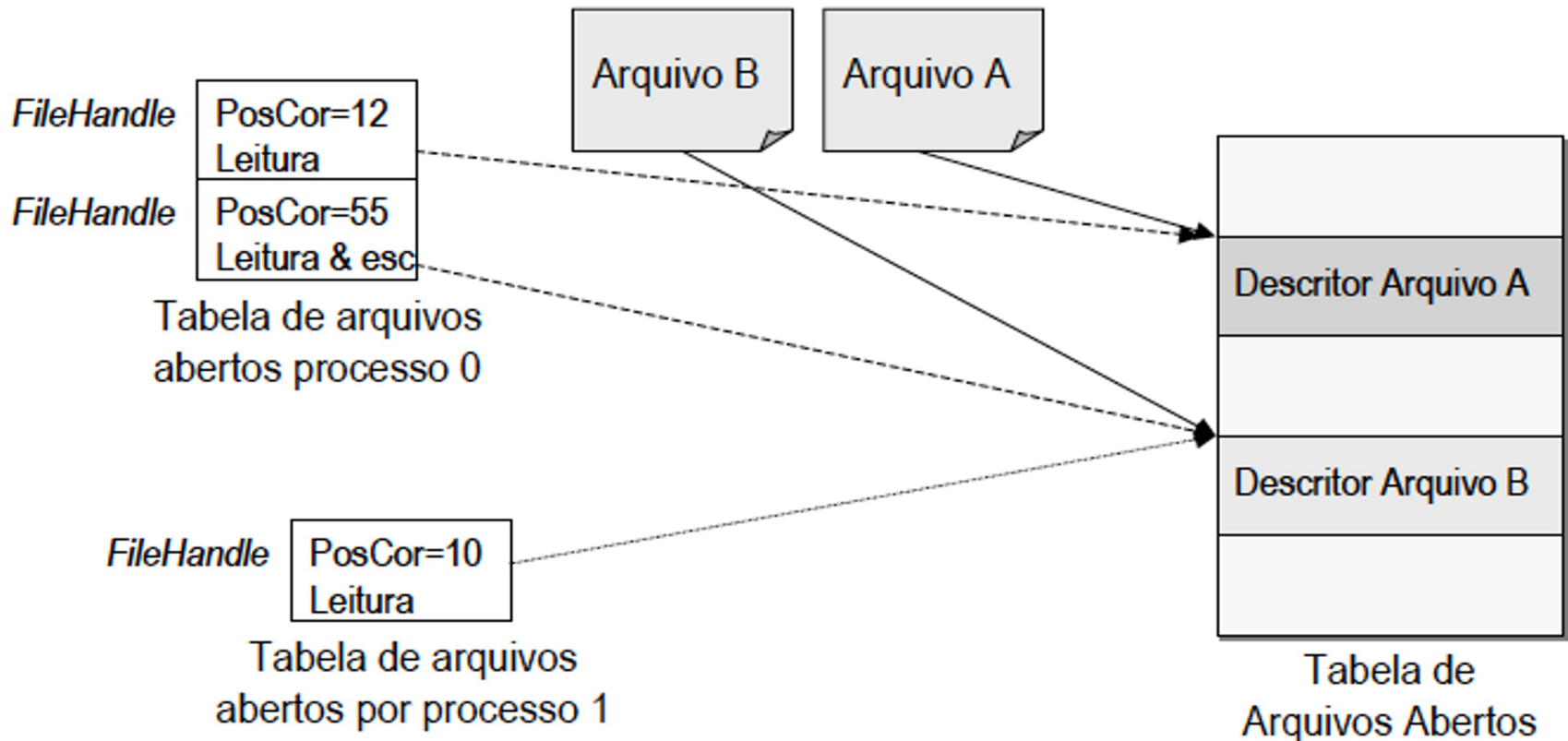


# Tabelas de Arquivos Abertos por Processo

- Informações relacionadas com arquivos são de dois tipos:
  - Não variam conforme o processo que está acessando o arquivo
    - e.g., tamanho do arquivo corrente
  - Dependem do processo que está acessando o arquivo
    - e.g., posição corrente
- Informações dependentes do processo são armazenadas em uma tabela a parte, mantida pelo processo (TAAP)
  - e.g., posição corrente no arquivo, tipo de acesso e apontador para a entrada correspondente na TDAA
- Entrada na TAAP serve para referenciar o arquivo
  - *File handle*



# Emprego Conjunto das Tabelas TAAP e TDAA



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



JESUÍTAS BRASIL



Somos infinitas possibilidades

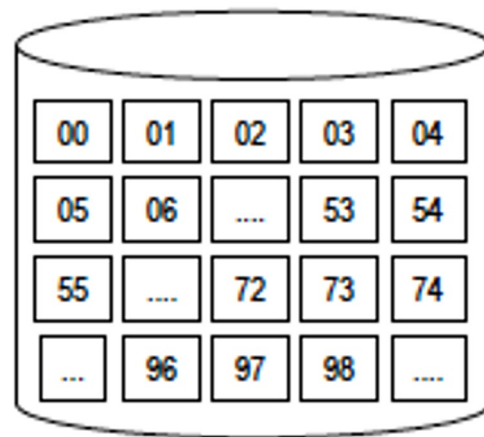
# Gerenciamento do Dispositivo de Armazenamento

- Pontos a serem tratados:
  - Relação entre número de setores do disco que compõem um bloco
  - Alocação de blocos no disco
  - Recuperação de blocos liberados
  - Localização de dados no disco
- Existe uma relação entre a política de alocação com a política de gerência de espaço livre



# Alocação do Espaço em Disco

- Como alocar espaço em disco de forma que os arquivos sejam armazenados de forma eficiente e que permita acesso rápido
  - Alocar blocos livres suficientes para armazenar o arquivo
  - Blocos lógicos do disco são numerados sequencialmente
- Duas formas básicas
  - Contígua (alocação contígua)
  - Não-contígua (alocação encadeada e alocação indexada)



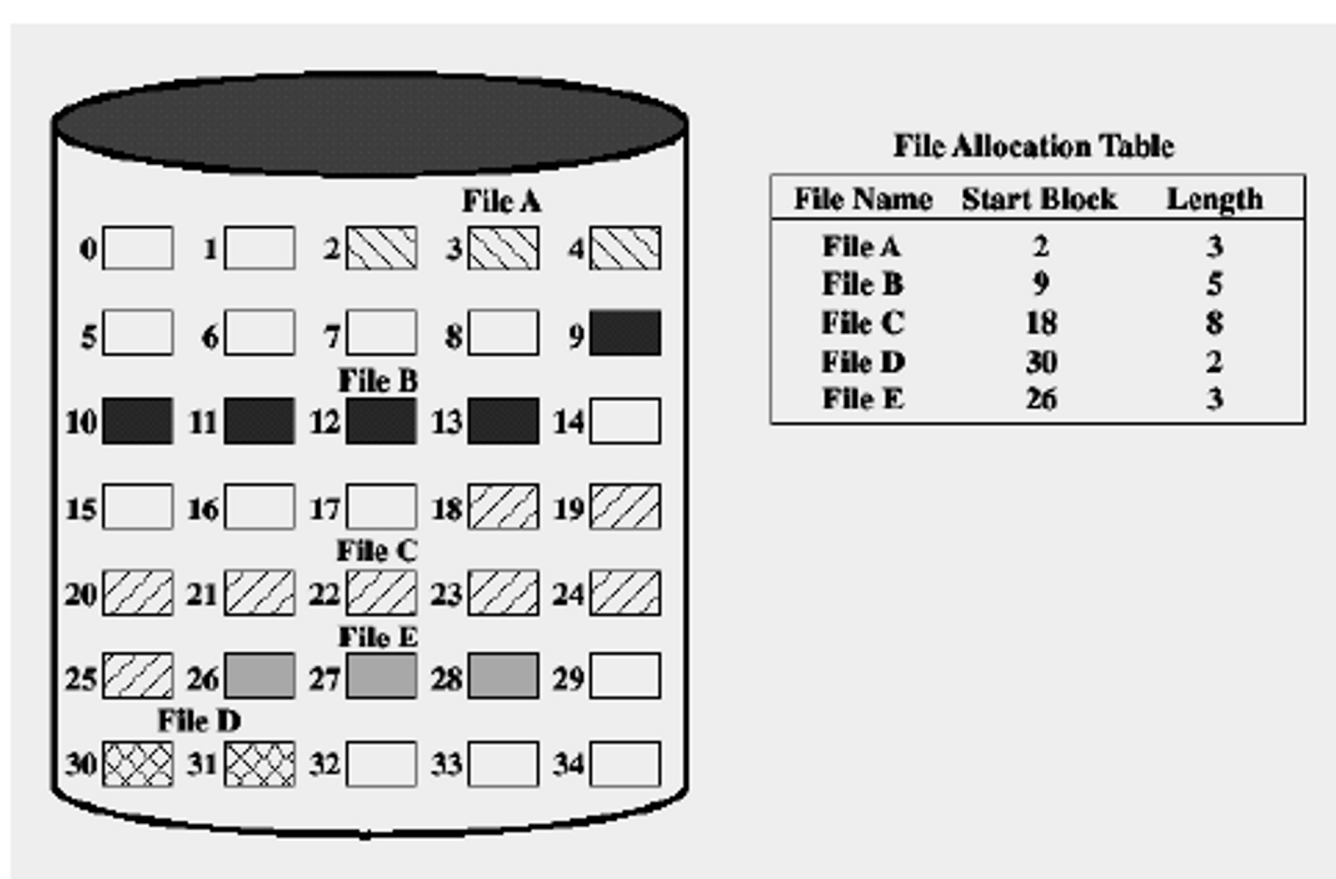
# Alocação Contígua

- Arquivo é uma sequência de blocos lógicos contíguos alocados no momento da criação
- Endereço no disco são lineares
  - Bloco lógico  $i$  e  $i+1$  são armazenados fisicamente em sequência
  - Reduz a necessidade de *seek*, já que blocos estão na mesma trilha
    - No pior caso necessita apenas a troca de cilindro
- Arquivo é descrito através de uma entrada na forma:
  - bloco físico inicial
  - tamanho do arquivo em blocos





# Esquema de Alocação Contígua



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Problemas com Alocação Contígua

- Problema 1: encontrar espaço para um novo arquivo
  - Técnicas de gerência de memória
  - Gera fragmentação externa
- Problema 2: determinar o espaço necessário a um arquivo
  - Arquivos tendem a crescer e não há espaço contíguo disponível?
  - Pré-alocar um espaço máximo para o arquivo
    - Fragmentação interna

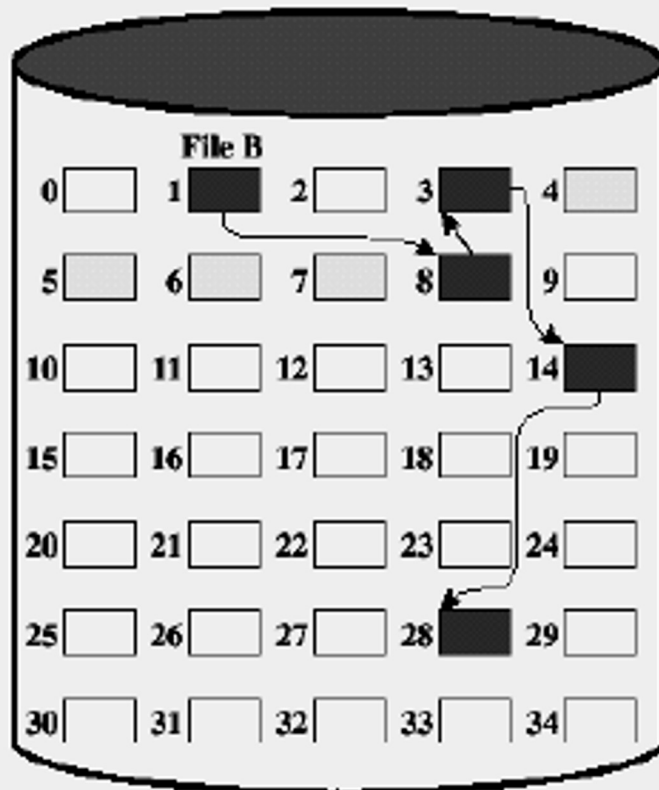


# Alocação Encadeada

- Soluciona os problemas da alocação contígua
- Alocação é baseada em uma unidade de tamanho fixo (bloco lógico)
- Arquivo é uma lista encadeada de blocos
- Arquivo é descrito em uma entrada na forma:
  - bloco inicial do arquivo
  - bloco final do arquivo ou tamanho do arquivo em blocos



# Esquema de Alocação Encadeada



File Allocation Table

File Name	Start Block	Length
...	...	...
File B	1	5
...	...	...

Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Prós e Contras da Alocação Encadeada

- Elimina a fragmentação externa
- Arquivos podem crescer indefinidamente
- O acesso a um bloco  $i$  implica em percorrer a lista encadeada
  - Afeta o desempenho
  - Adequado para acesso sequencial a arquivos
- Confiabilidade
  - Erro provoca a leitura/escrita em bloco pertencente a outro arquivo



# Exemplo: *File Allocation Table* (FAT)

- Variação de alocação encadeada
- FAT é uma tabela de encadeamento de blocos lógicos
  - Uma entrada na FAT para cada bloco lógico do disco (sistema de arquivos)
  - Composta por um ponteiro (endereço do bloco lógico)
  - Arquivo é descrito por uma sequência de entradas na FAT, cada entrada apontando para a próxima entrada

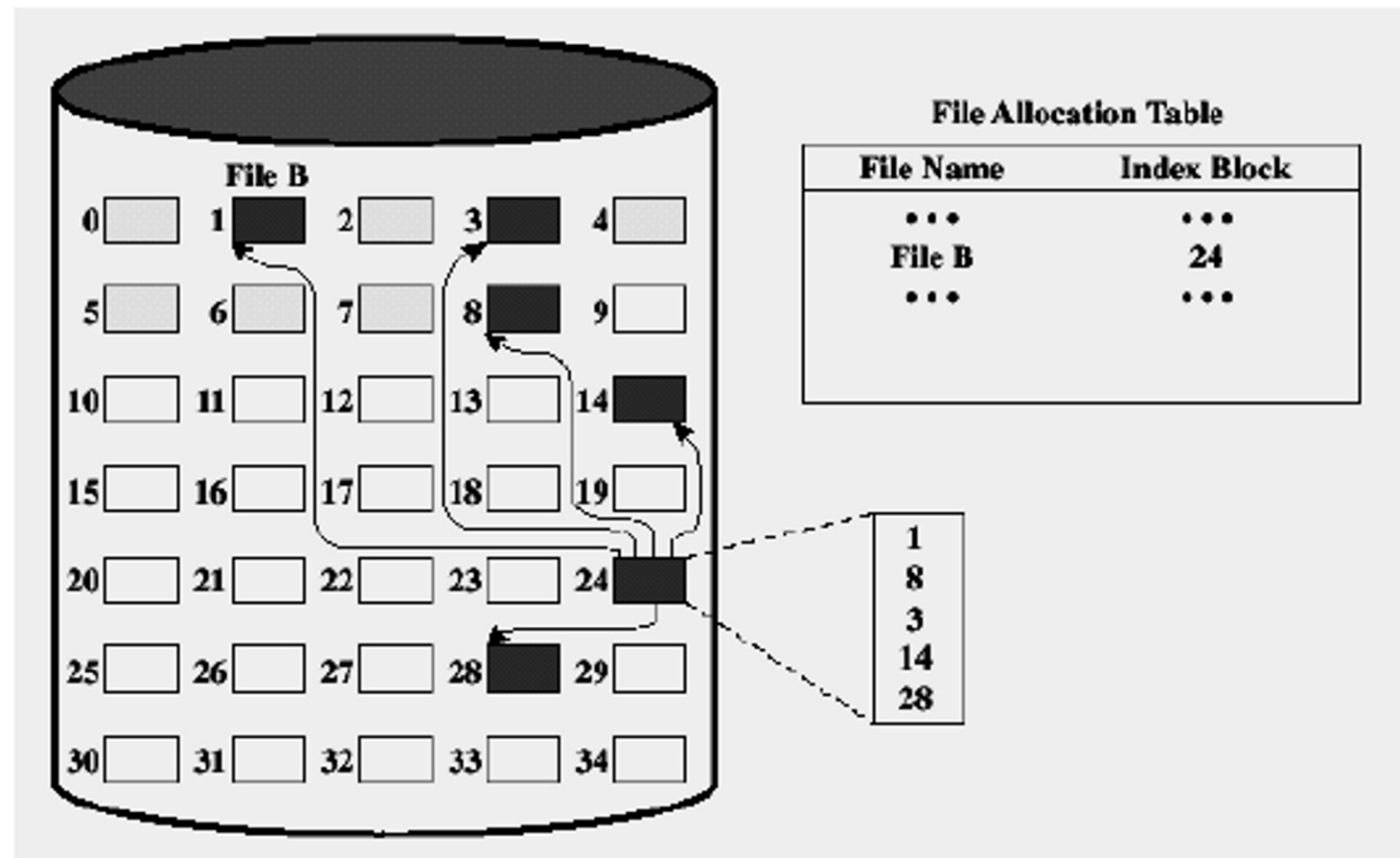


# Alocação Indexada

- Busca resolver o problema de “ponteiros esparramados” pelo disco que a alocação encadeada provoca
- Mantém, por arquivo, um índice de blocos que o compõe
- O índice é mantido em um bloco
- Diretório possui um ponteiro para o bloco, onde está o índice associado a um determinado arquivo



# Esquema de Alocação Indexada



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



JESUÍTAS BRASIL



Somos infinitas possibilidades



# Prós e Contras da Alocação Indexada

- Permite o acesso aleatório a blocos independentes de sua posição relativa no arquivo
- Tamanho máximo do arquivo é limitado pela quantidade de entradas suportadas pelo bloco
  - Muito pequeno (limita tamanho do arquivo)
  - Muito grande (desperdiça espaço em disco)
- Solução: utilizar dois tamanhos de blocos, um para índice e outro para dados
  - *e.g.*, *i-nodes* e blocos de dados em Sistemas UNIX



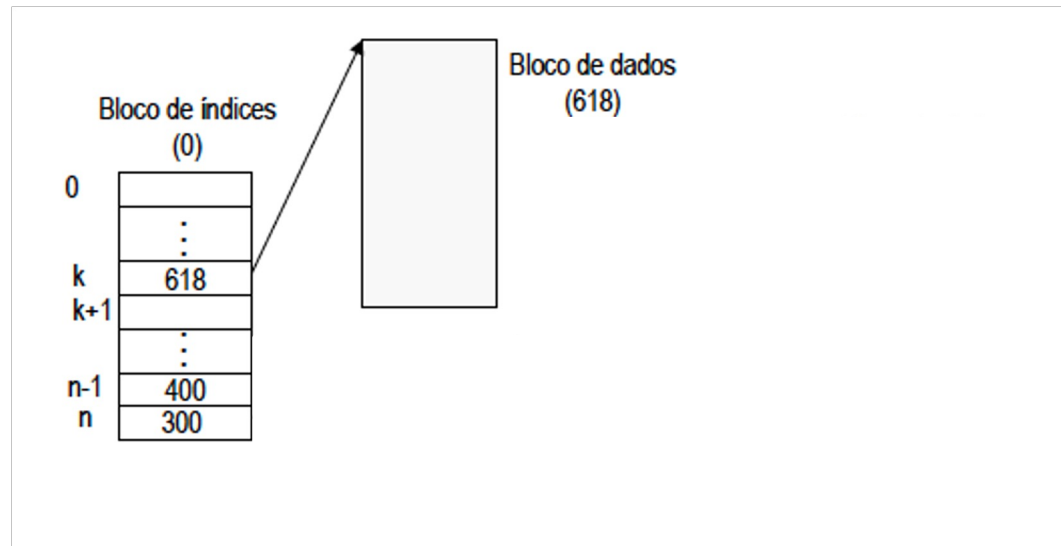
# Variação em Alocação Indexada

- Buscam resolver o problema do tamanho do bloco de índices
- Três métodos básicos:
  - Encadeado
  - Multinível
  - Combinado



# Método Encadeado

- O índice mantém ponteiros para os blocos que compõem o arquivo com exceção da última entrada
  - Mantém um ponteiro para outro bloco, onde índice continua



Fonte: OLIVEIRA, Rômulo; et al. Sistemas Operacionais. 4a edição.



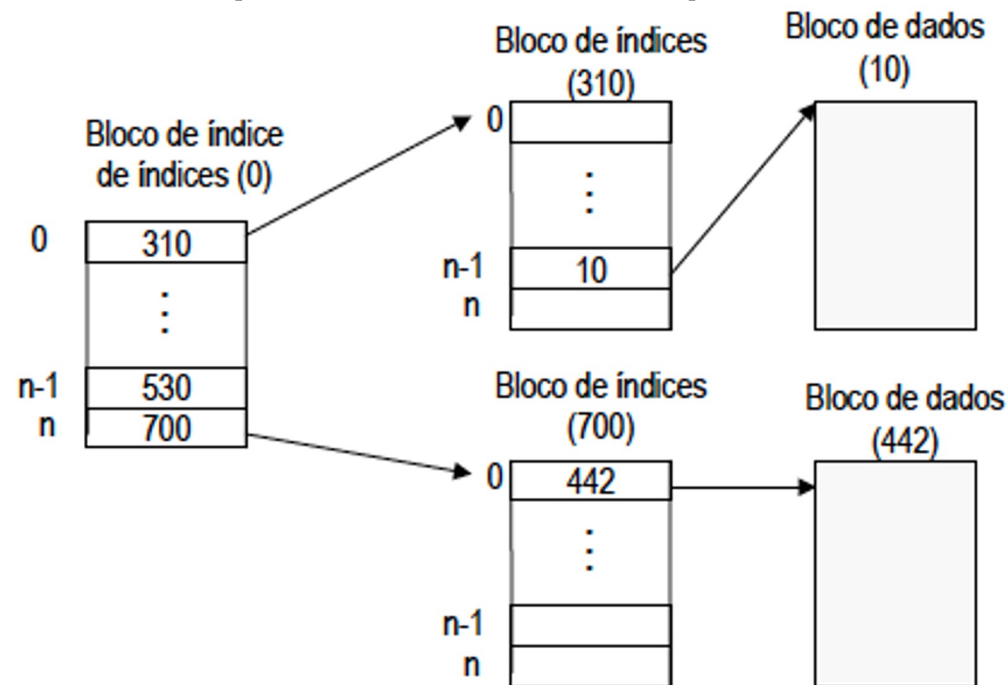
JESUÍTAS BRASIL



Somos infinitas possibilidades

# Método Multinível

- Mantém um índice de índice
  - Não resolve completamente o problema de limite



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



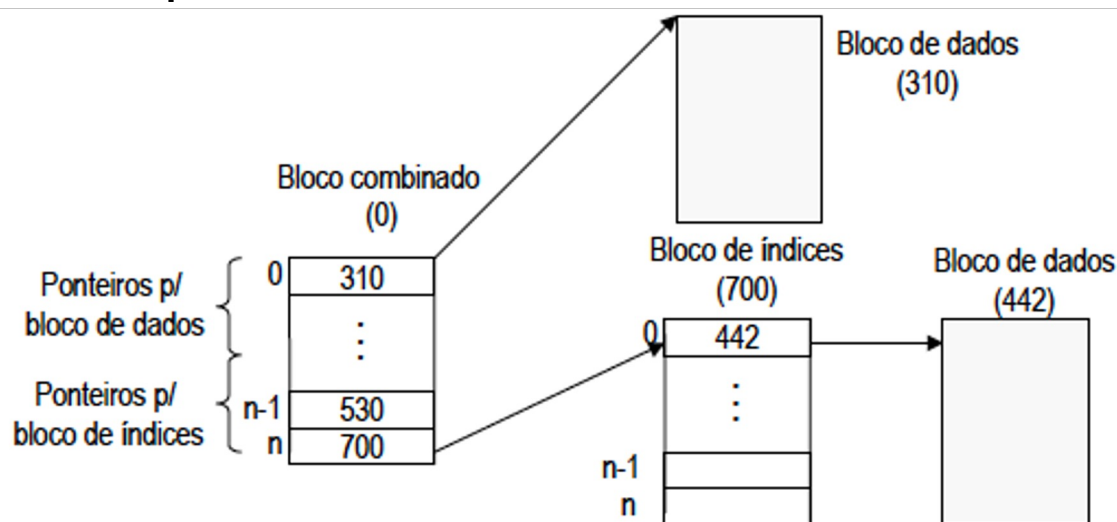
JESUÍTAS BRASIL



Somos infinitas possibilidades

# Método Combinado

- Método encadeado e multinível em uma única estrutura de dados
- O que justifica essa combinação?
  - Acesso otimizado a blocos de dados: método indexado
  - Limite de arquivos: multinível



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.

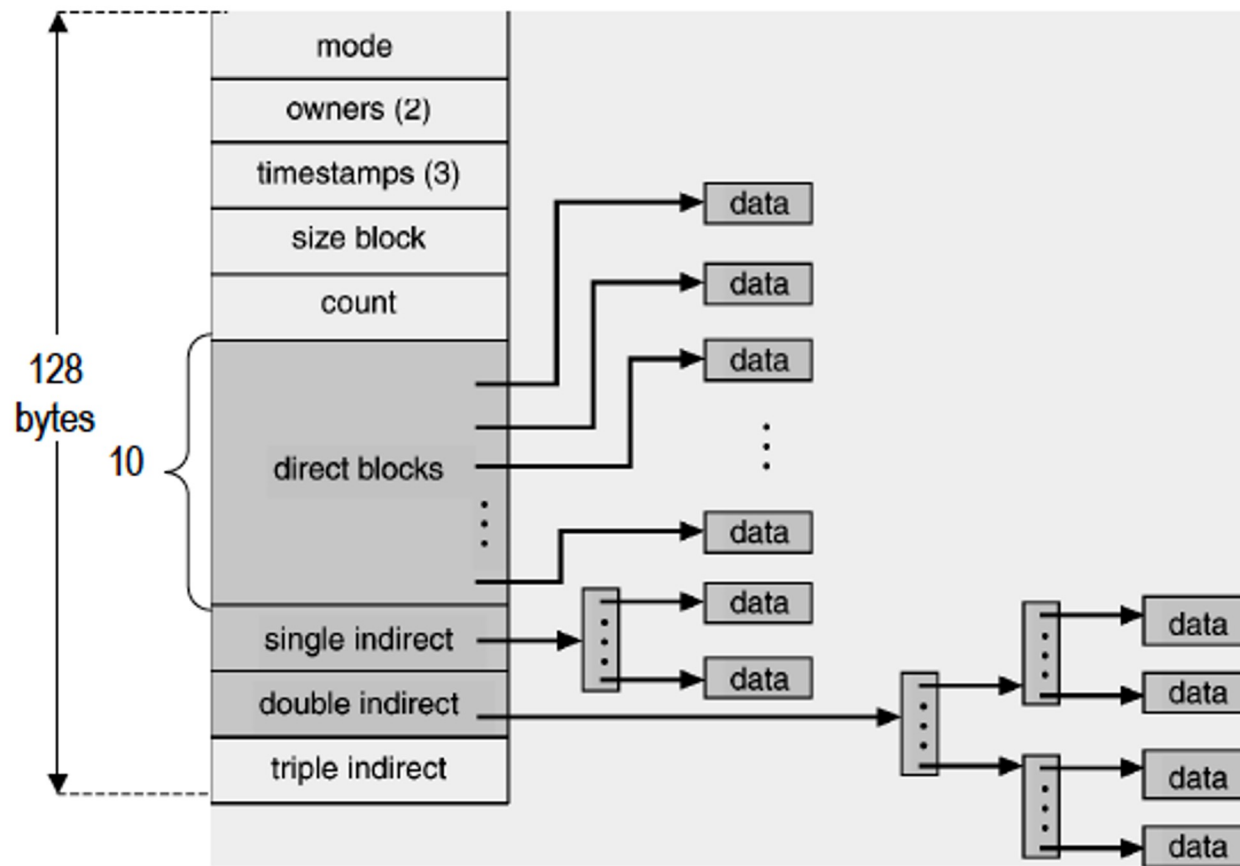


JESUÍTAS BRASIL



Somos infinitas possibilidades

# Exemplo: Estrutura de *i-nodes* (UNIX)



Fonte: OLIVEIRA, Rômulo; *et al.* Sistemas Operacionais. 4a edição.



JESUÍTAS BRASIL



Somos infinitas possibilidades

# Referências Bibliográficas

- Baseado nos originais de: OLIVEIRA, Rômulo; CARÍSSIMI, Alexandre; TOSCANI, Simão.
  - Livro: Sistemas Operacionais. Porto Alegre: Bookman, 4a. ed. 2010.
- SILBERSCHATZ, A.; GALVIN, Peter; GAGNE Greg, Operating System Concepts Essentials. John Wiley & Sons, Inc. 2th edition, 2013.
- TANENBAUM, Andrew S. Sistemas operacionais modernos. 3a. ed. São Paulo: Pearson, 2009-2013. p. 653.



JESUÍTAS BRASIL



Somos infinitas possibilidades