



Sistemas Operacionais

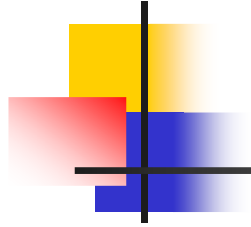
Unidade Seis

Prof. Flávio Márcio de Moraes e Silva



Conceitos Básicos

- Sistema de arquivos:
 - É o módulo do SO mais visível para os usuários, portanto deve ter interface amigável que facilite sua manipulação
 - Recurso implementado em software e não em hardware
 - Hardware = espaço em disco magnético na forma de setores que podem ser acessados individualmente
 - Software = abstração lógica de um disco magnético
 - Um software é subdividido em blocos pelo SO (similar a paginação de memória), cada bloco é formado por um ou mais setores. (blocos lógicos mapeados em blocos físicos)
 - Acesso a disco 10000 vezes mais lento que à memória
 - Sistema de arquivos = estruturas de dados e algoritmos que otimizam o acesso a disco



Objetivos do sistema de arquivos

- Fornecer mecanismos para o usuário manipular arquivos e diretórios
- Garantir a validade e coerência de dados
 - Minimizar ou eliminar o risco de perda/alteração de dados
- Otimizar o acesso
- Fornecer suporte a outros sistemas de arquivos
- Suporte a vários usuários (multiprogramação)
 - Uso compartilhado (proteção e acesso concorrente)



Requisitos mínimos: ponto de vista do usuário

- Cada usuário deve ser capaz de:
 - Criar, apagar, ler e alterar arquivos
 - Controlar as permissões de acesso a seus arquivos
 - Nomear arquivos de forma simbólica
 - Estruturar os arquivos de forma a adequá-los a suas necessidades específicas
 - Criação de diretórios e subdiretórios
 - Realizar *back-ups* e recuperar arquivos em caso de problemas



Requisitos mínimos: ponto de vista do sistema

- O sistema operacional deve ser capaz de:
 - Descrever a localização de todos os arquivos e de seus atributos
 - Via diretório
 - Gerenciar espaço físico do disco
 - Alocar blocos livres a arquivos em criação/expansão
 - Liberar blocos de arquivos removidos
 - Mecanismos para localizar eficientemente blocos (setores) que compõem arquivos



Conceito de Arquivo

- Arquivos são recipientes que contém dados. Dados estes que o usuário resolveu colocar juntos: um programa, uma planilha, um texto, uma foto, etc
- Diretórios são conjuntos de referências para arquivos. Usados para organizar os arquivos em uma hierarquia de grupos.
- Partições são abstrações que permitem a partir de discos físicos criar discos lógicos. São auto-contidas. Uma partição pode conter mais de um disco, ou um disco pode ser dividido e 2 ou mais partições.



Atributos e Operações em Arquivos

- Atributos:
 - Nome
 - Tipo de conteúdo
 - Data e hora: criação, último acesso, última alteração, etc
 - ID do usuário que criou o arquivo
 - Lista de usuários com permissão no arquivo
- Operações básicas (chamadas de sistema):
 - Criação, destruição, trocar de nome, abrir, fechar, etc
 - Leitura, alteração, escrita, execução, posicionar(seek), etc
 - Alterar permissões
 - Operações mais complexas podem ser implementadas como rotinas usando operações simples. Ex: impressão do arquivo.



Controle de acesso

- Identificar o usuário (login)
- A partir do momento que o usuário acessa um terminal, todos os comandos executados no mesmo tem os direitos de acesso do usuário.
- Para cada arquivo existe um lista de direitos de acesso
 - Usuário e direitos (leitura, escrita, execução, etc)
 - Pode-se criar grupos de usuários e dar permissões ao grupo
- Esta lista é criada e mantida pelo dono do arquivo
- Exemplo: planilha de notas
 - Dono: professor/leitura e escrita
 - Alunos/leitura



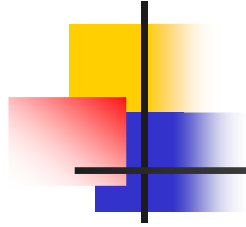
Estrutura interna de um arquivo

- Forma como os dados estão dispostos dentro do arquivo
 - Textos: parágrafos e linhas
 - Programas: segmentos de código, de dados e pilha de execução.
 - Cadastro de pessoas: registros
 - Imagens: pixels
- Muitos tipos de arquivos são criados a cada dia.
- SO ignora a estrutura interna dos arquivos, para ele cada arquivo é uma seqüência de bytes. Exceto executáveis, cuja estrutura interna é definida pelo SO, que precisa carregá-los para a memória no momento da execução.
- Extensões de arquivos são convenções usadas para orientar o usuário.



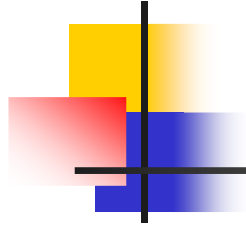
Métodos de acesso

- Seqüencial: cada chamada de sistema retorna dados seguintes aos que foram lidos na chamada anterior. Exemplos: compiladores, impressão, copiar o conteúdo de um local para outro, etc
- Acesso relativo: incluir na chamada de sistema a posição do arquivo a ser lida. Ex: em uma lista de registros buscar o de número 100. Seqüencialmente o desempenho seria muito ruim.
- Posição corrente: comandos separados para posicionar, ler e escrever. Leitura e escrita podem ser seqüenciais, após o posicionamento.
- Acessos mais complexos: SGBD + BD



Implementação de arquivos

- Arquivos são implementados através da criação, para cada arquivo no sistema, de uma estrutura de dados
- O 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
 - Local do disco físico onde o conteúdo do arquivo foi colocado
 - Etc.



Tabelas de descritores de arquivos

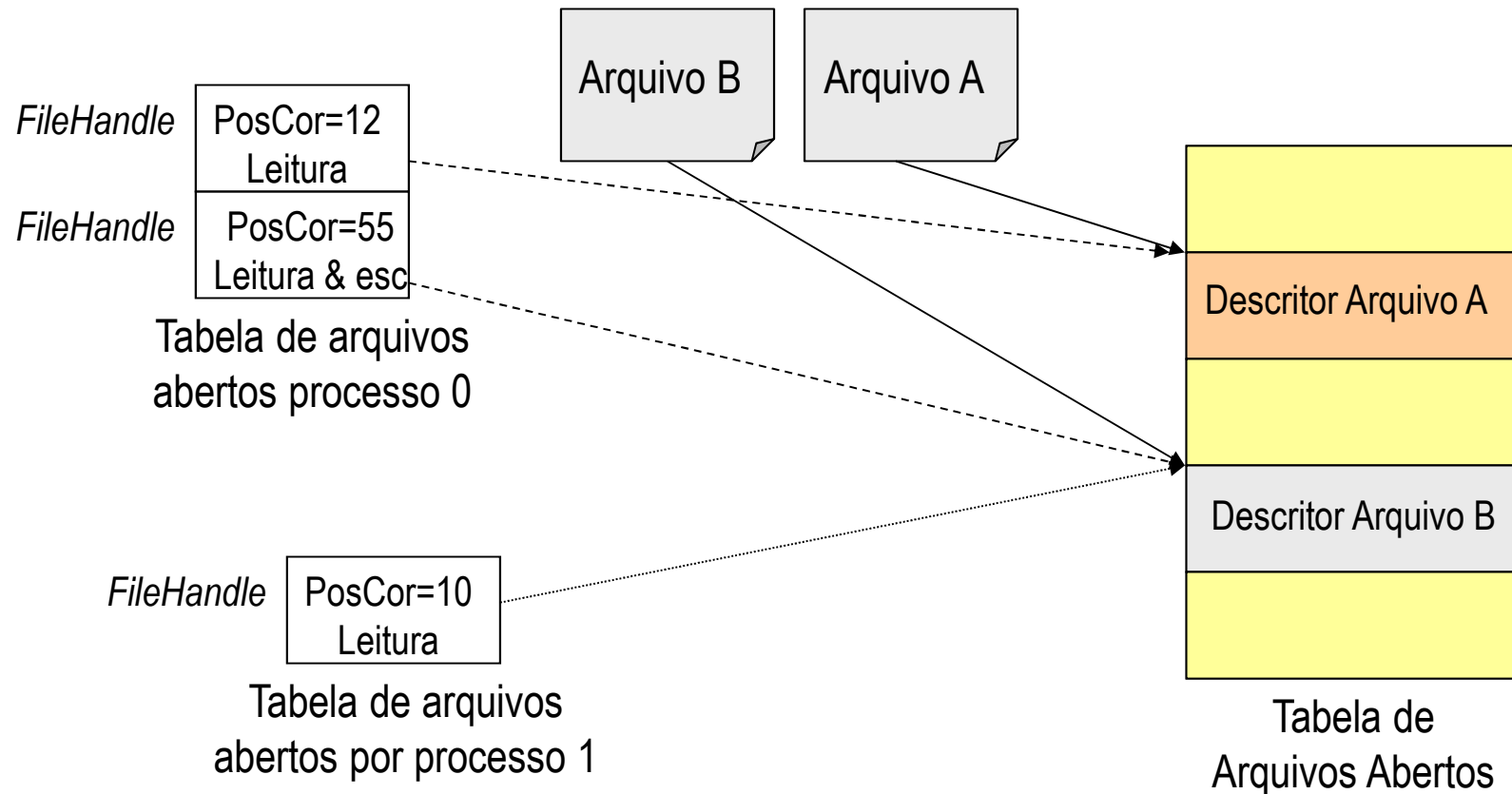
- Descritores de arquivos são armazenados no próprio disco
 - Na realidade no mesmo disco lógico (partição)
- Problema de desempenho
 - Acesso ao disco para ler o descritor de arquivos é lento
 - Solução é manter descritor em memória enquanto o arquivo estiver em uso
 - A indicação se arquivo está em uso normalmente é feita pelo próprio usuário (aplicação) através de chamadas do tipo *open* e *close*
- O sistema de arquivos mantém os descritores de arquivos em memória em uma estrutura de dados do sistema operacional
 - Tabela de descritores de arquivos abertos (TDAA)



Tabelas de arquivos abertos por processo (TAAP)

- Informações relacionadas com arquivos são de dois tipos:
 - Não variam conforme o processo que está acessando o arquivo
 - ex: tamanho do arquivo
 - Dependem do processo que está acessando o arquivo
 - ex: posição corrente
- Informações dependentes do processo são armazenadas em uma tabela à parte mantida pelo processo (TAAP)
 - ex: 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



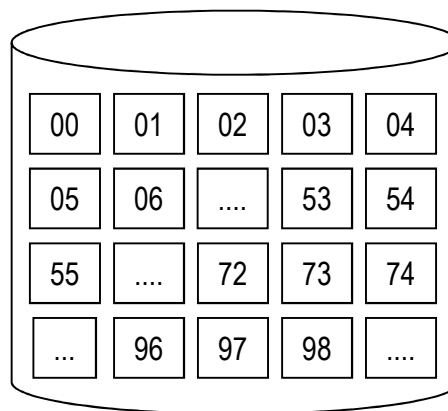


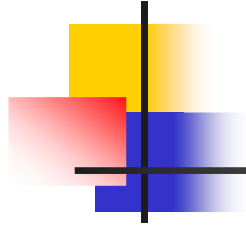
Gerenciamento do dispositivo de armazenamento

- Problema: arquivos devem ser armazenados no disco!!
 - Unidade de manipulação dos dados no dispositivo físico (bloco)
- Pontos a serem tratados:
 - Relação número de setores do disco que compõem um bloco
 - Não necessita ser 1:1
 - 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)

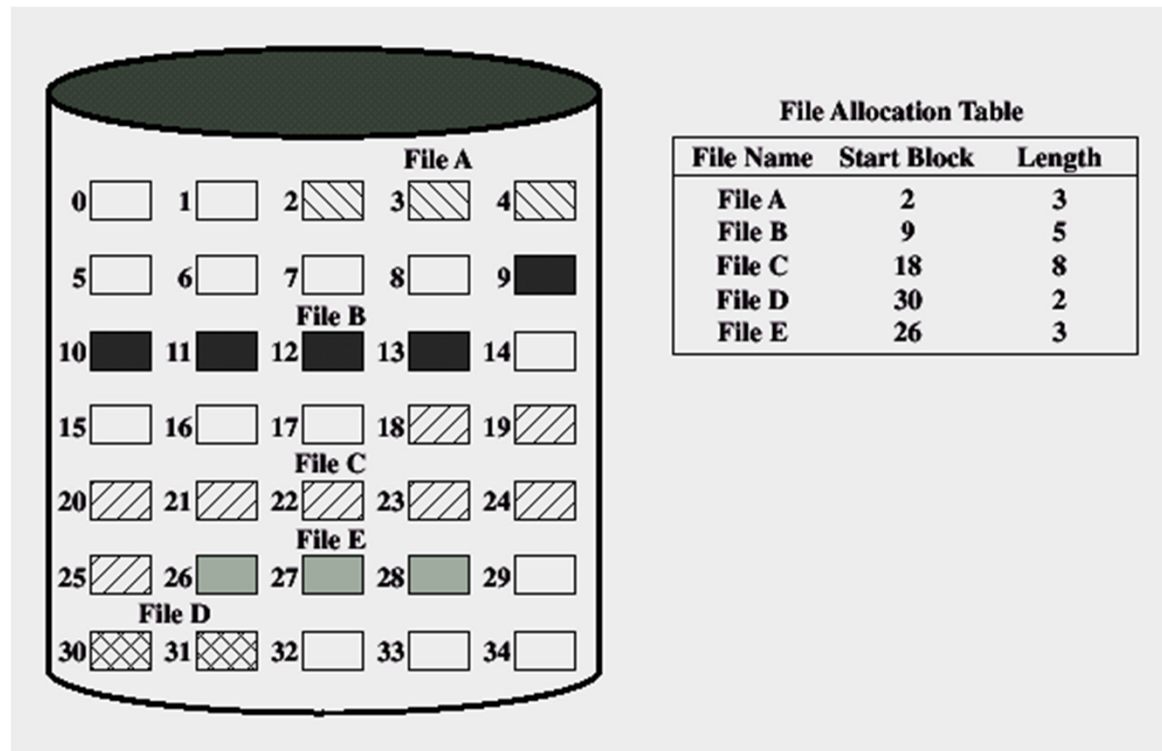


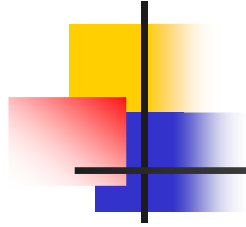


Alocação contígua

- Arquivo é uma sequência de blocos lógicos contíguos alocados no momento da criação
- Endereços 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 alocação contígua





Problemas com alocação contígua

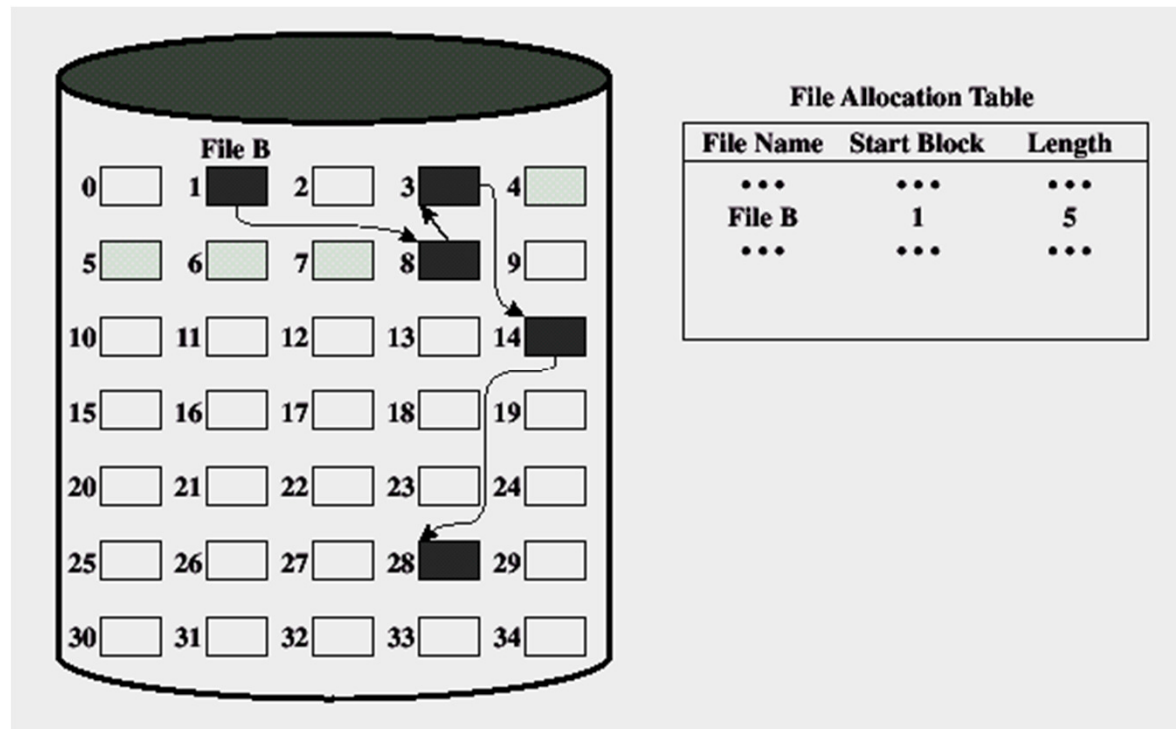
- Problema 1: encontrar espaço para um novo arquivo
 - Técnicas de gerência de memória
 - ex; *first-fit*, *best-fit*, *worst-fit*
 - Gera fragmentação externa
 - Necessidade de compactação
- Problema 2: determinar o espaço necessário a um arquivo
 - Arquivos tendem a crescer, e se não há espaço contíguo disponível?
 - Aborta execução do programa com erro
 - Recopia o programa para uma zona maior
 - Pré-alocar um espaço máximo para o arquivo
 - Fragmentação interna



Alocação encadeada

- Soluciona os problemas da alocação contígua
 - Relação a dimensionamento do tamanho e crescimento de arquivos
- Alocação é baseada em uma unidade de tamanho fixo (bloco lógico)
 - Análogo à paginação
- Arquivo é uma lista encadeada de blocos
 - Cada bloco contém um ponteiro para o próximo bloco
- 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





Prós e contras da alocação encadeada

- Elimina a fragmentação externa
- Arquivos podem crescer indefinidamente
 - Não há necessidade de compactar o disco
- O acesso a um bloco 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)

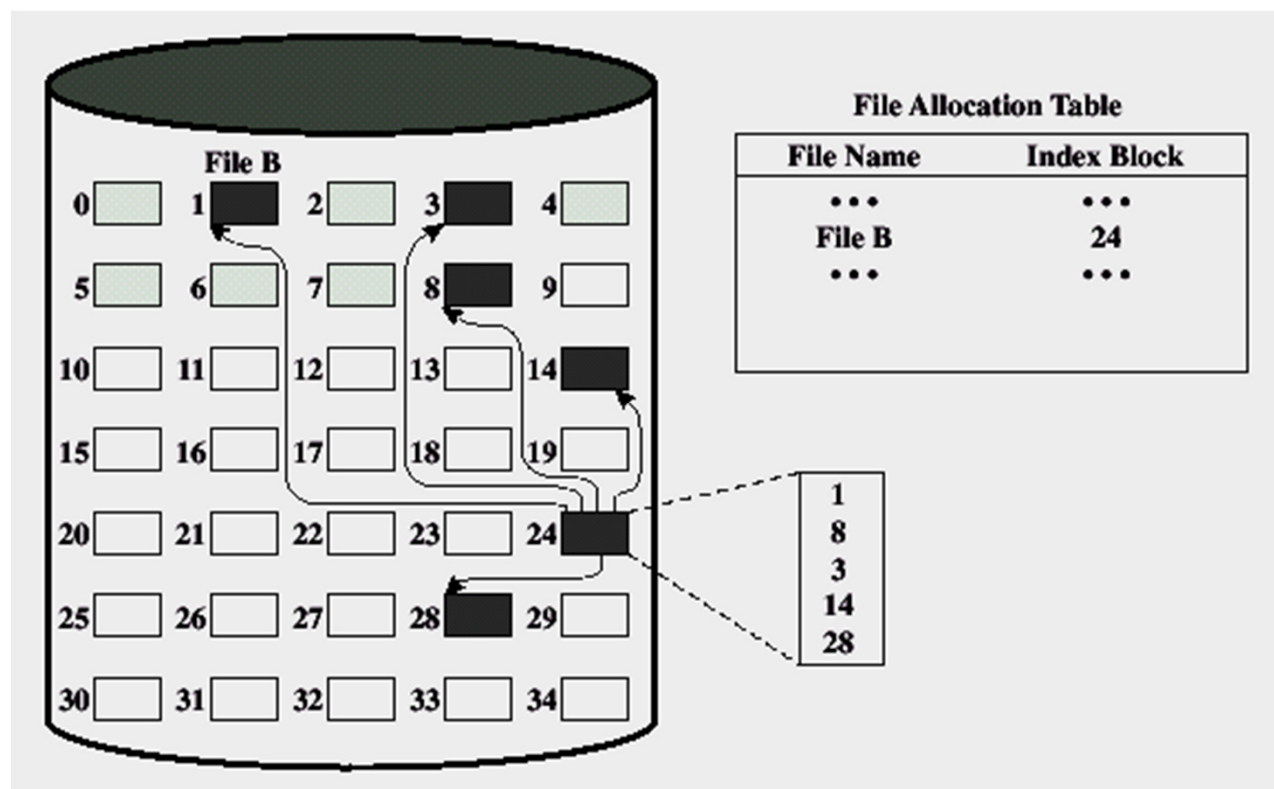
- Variação de alocação encadeada
- FAT é uma tabela de encadeamento de blocos lógicos
 - Uma entrada na FAT para cada *bloco lógico* do disco (sistema de arquivos)
 - Composta por um ponteiro (endereço do bloco lógico)
 - Arquivo é descrito por uma sequência de entradas na FAT, cada entrada apontando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

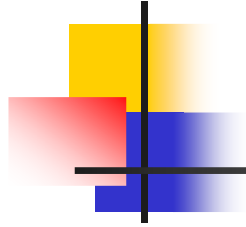


Alocação indexada

- 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





Prós e contras da alocação indexada

- Permite o acesso randômico 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 é utilizar dois tamanhos de blocos, um para índice e outro para dados
 - ex: i-nodes e bloco de dados em sistemas UNIX

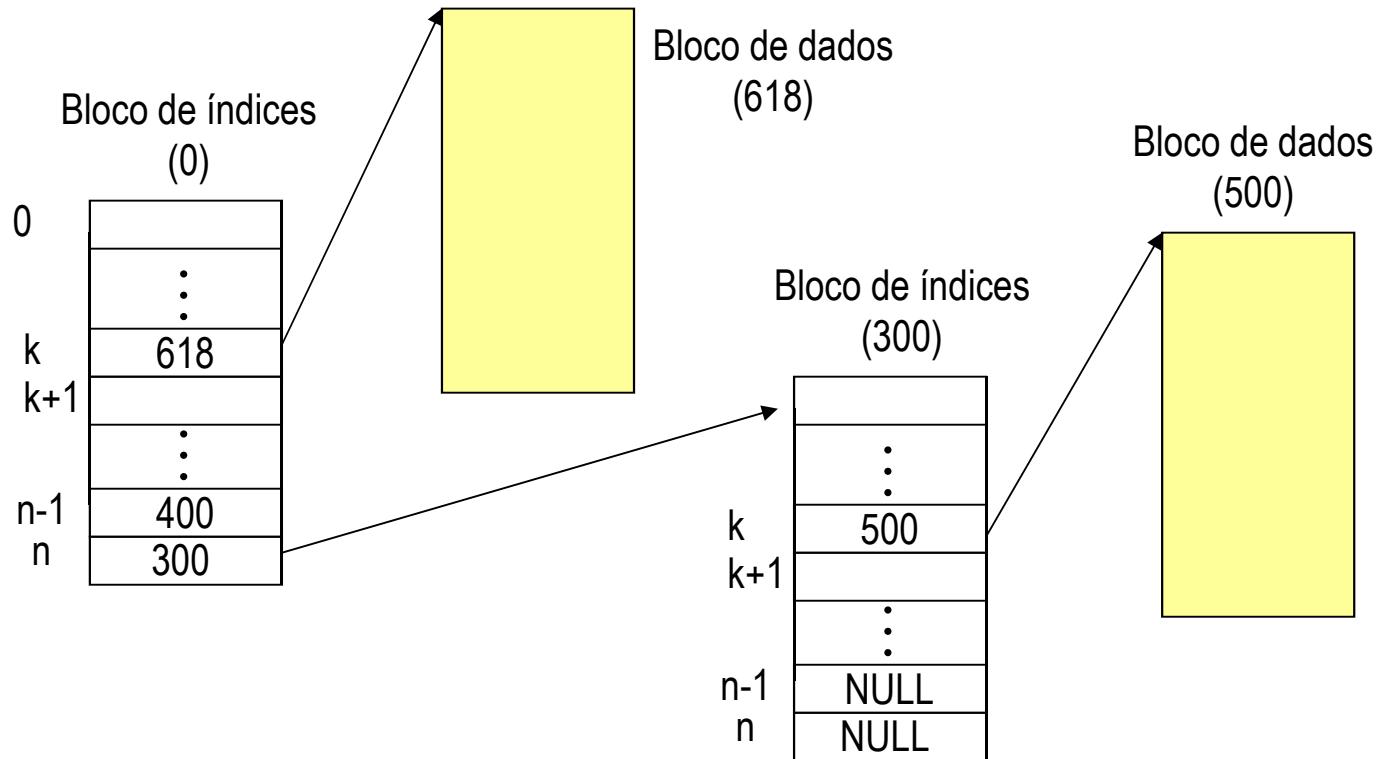


Variações em alocação indexada

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

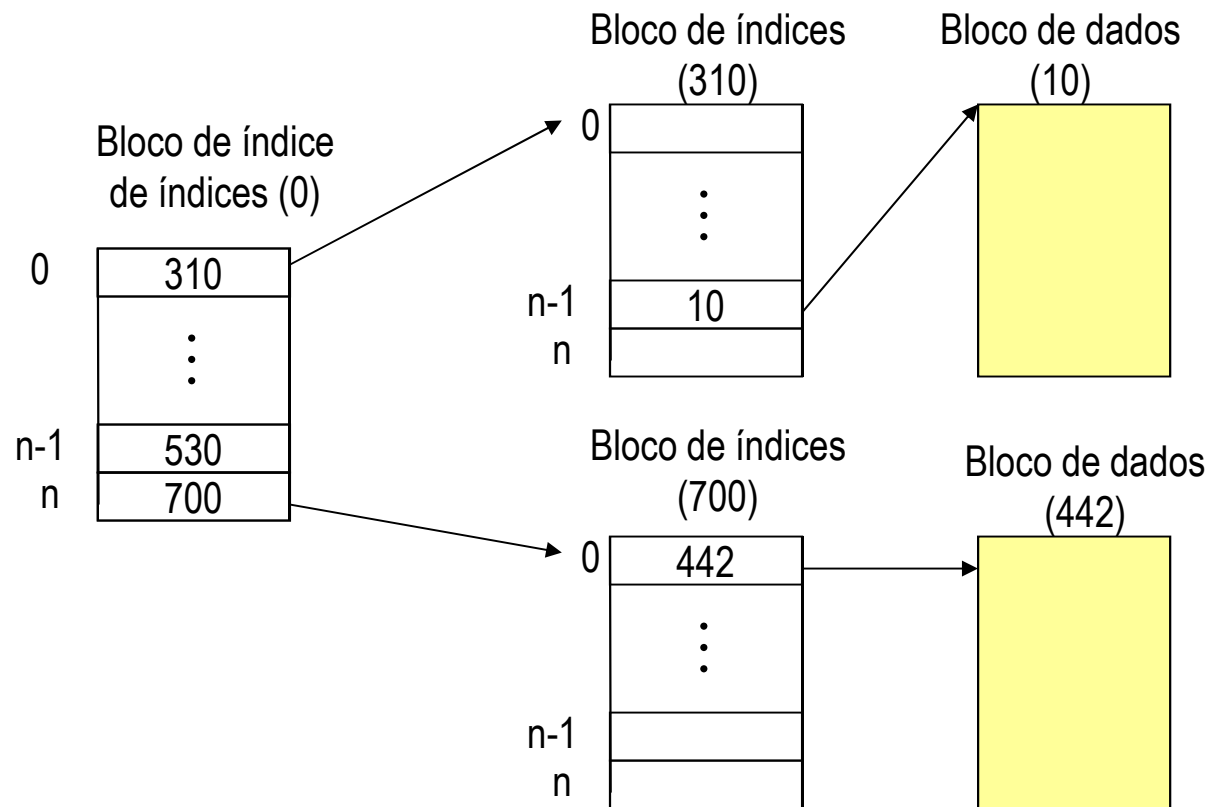
Método encadeado

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



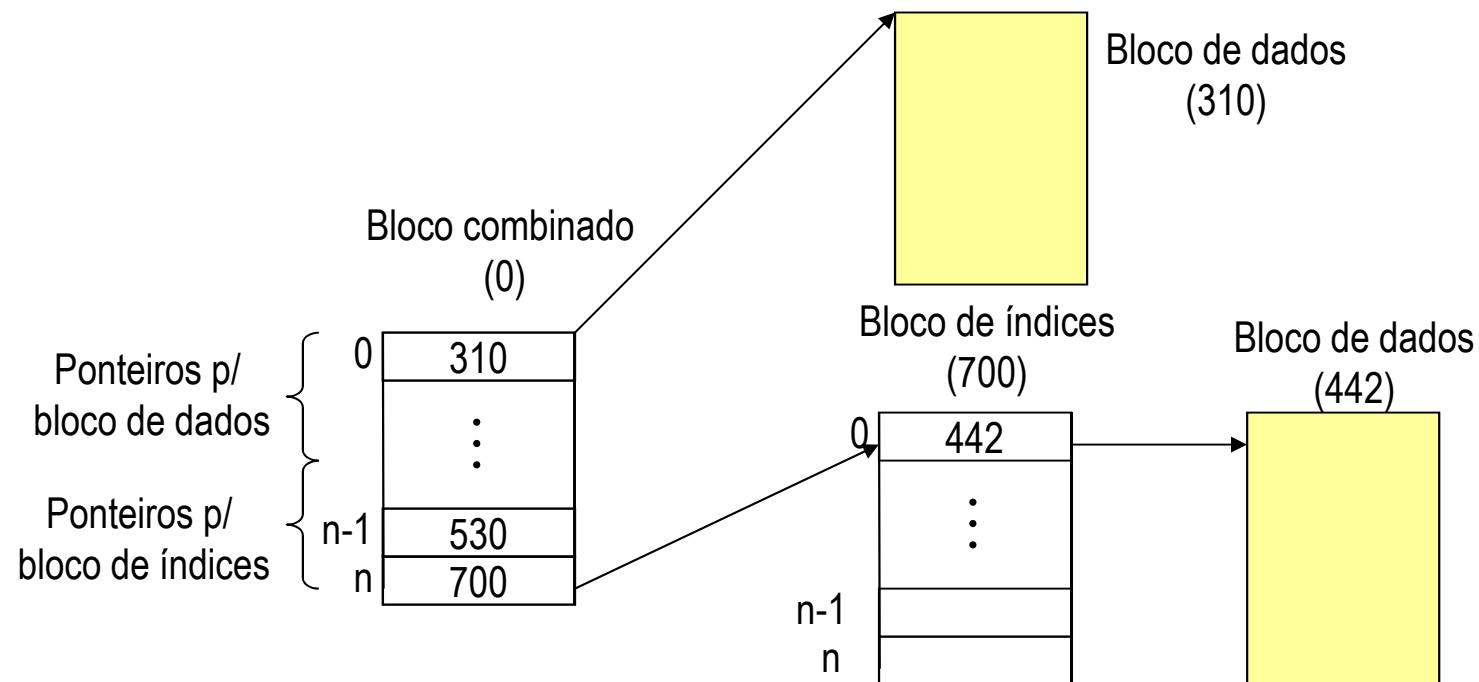
Método multinível

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

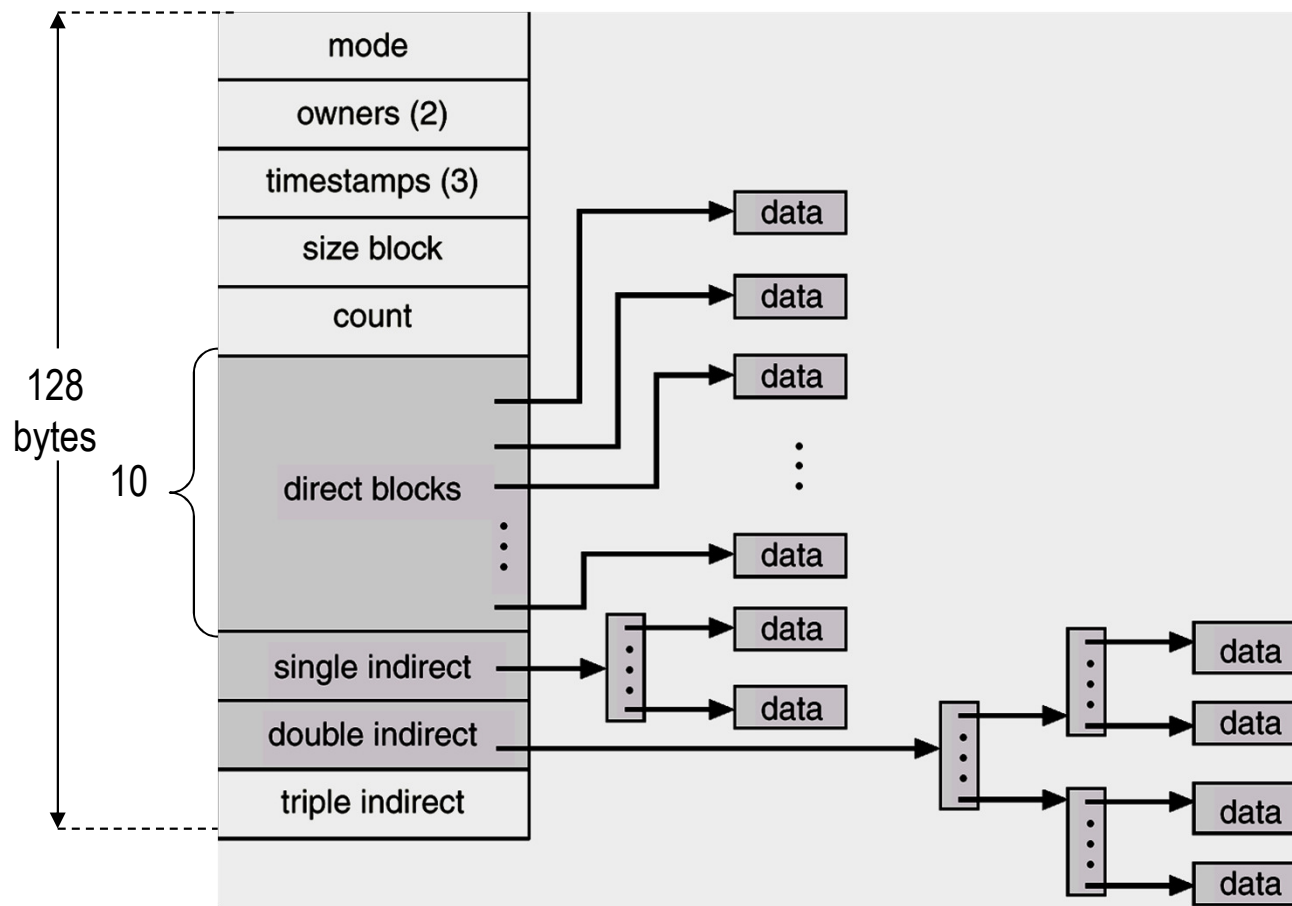


Método combinado

- Métodos 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



Exemplo: estrutura de *i-nodes* (UNIX)





Resumo dos tipos de alocação

- Alocação contígua
 - Só armazena endereço do primeiro bloco
 - Acesso randômico é possível (bloco inicial + deslocamento)
 - Gera fragmentação externa no disco
- Alocação encadeada
 - Armazena endereço do primeiro bloco
 - Problema de desempenho (*seek*)
 - Não recomendado para acesso randômico
- Alocação indexada
 - Visa solucionar problemas dos tipos anteriores
 - Análise de desempenho (tamanho + tempo de acesso) é complexa
 - Depende da estrutura de índice e do tamanho de arquivo



Conclusão: qual o melhor método de alocação?

- Depende do tipo de acesso que o sistema faz a seus arquivos
 - Sequencial *versus* randômico
- Fator adicional:
 - Evolução tecnológica (novos *hardwares*) e de desempenho forçam a coexistência de diferentes sistemas de arquivos
- Necessidade de “fazer conviver” diferentes sistemas de arquivos em um mesmo computador
 - Suporte a múltiplos sistemas de arquivos



Controle de Acesso por Classe de Usuário

- Proprietário
 - Usuário que criou o arquivo
- Usuário especificado
 - O proprietário especifica quem pode acessar o arquivo
- Grupo
 - Membros do mesmo grupo tem permissão de acessar quaisquer arquivos pertencente ao grupo
- Público
 - O arquivo é publico, todos podem acessá-lo



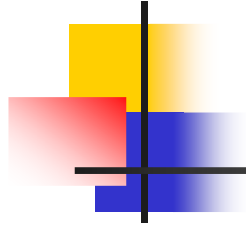
Cópia de Segurança e Recuperação

- Os SO's quanto ao armazenamento devem ser tolerantes a falhas
- Devem fornecer técnicas para a recuperação de dados afetados
- Armazenam cópias redundantes (backup)
 - Realizar cópias periódicas de segurança dos arquivos em disco (SO e dados)
 - Não só copia arquivos, como também a estrutura de diretórios da partição e com isso é possível restaurar os arquivos no mesmo local de origem



Proteção e Integridade

- O SO possui rotinas para proteger os dados dos arquivos com relação ao acesso e a sua integridade
- O acesso pode ser protegido com
 - Autenticação e permissões
 - Criptografia de arquivos
- A integridade pode ser implementada com:
 - Discos em RAID
 - Cópias de backup
 - Sistemas de arquivos log-estruturados (operações atômicas)
 - Sombreamento de páginas (operações atômicas)



Proteção e Integridade

- Sistemas de arquivos log-estruturados
 - Registra as transações em arquivos de log e com isso elas podem ser revertidas se for preciso (rollback) ou terminadas mais tarde quando for conveniente (roll forward)
- Sombreamento de páginas
 - Grava as transações em um novo bloco e uma vez validada a informação, libera o bloco antigo como espaço livre, podendo ser revertida se houver falha



Sistema de Banco de Dados

- Um banco de dados (BD) é uma coleção de dados independente, integrada, estruturada e gerenciada de modo central
- Do ponto de vista do SO, um banco de dados é composto por um arquivo e um programa de gerenciamento (SGBD)
- Ao contrário dos arquivos que muitas vezes armazenam informações repetidas, um BD armazena somente uma vez a informação e permite que usuários e aplicativos as acessem e as formatem ao seu modo



Sistema de Banco de Dados

- Os BDs incorporam um mecanismo de consulta aos dados e suas permissões são definidas de forma independente das permissões do SO. Os BDs compartilham os dados de um modo mais eficiente (o bloqueio ocorre ao nível de tabela ou de registro), e o acesso simultâneo é controlado
- Sistemas de BDs usam técnicas padronizadas de organização dos dados (relacional, hierárquica, etc)