Curso de Ciência da Computação
Pontifícia Universidade Católica de Minas Gerais

Sistemas Operacionais

Capítulo XI - Sistemas de Arquivos

Conceito de Arquivo

- Espaço de endereçamento lógico contíguo
- Tipos:
  - Dados
    - numéricos
    - caractere
    - binário
  - Programas

Estrutura de Arquivos

- Nenhuma - seqüência de bytes
- Estrutura de registro simples
  - Linhas
  - Tamanho fixo
  - Tamanho variável
- Estrutura complexa
  - Documentos formatados
  - Arquivo de carga realocável
- Quem decide:
  - S.O.
  - Programa

Atributos de Arquivos

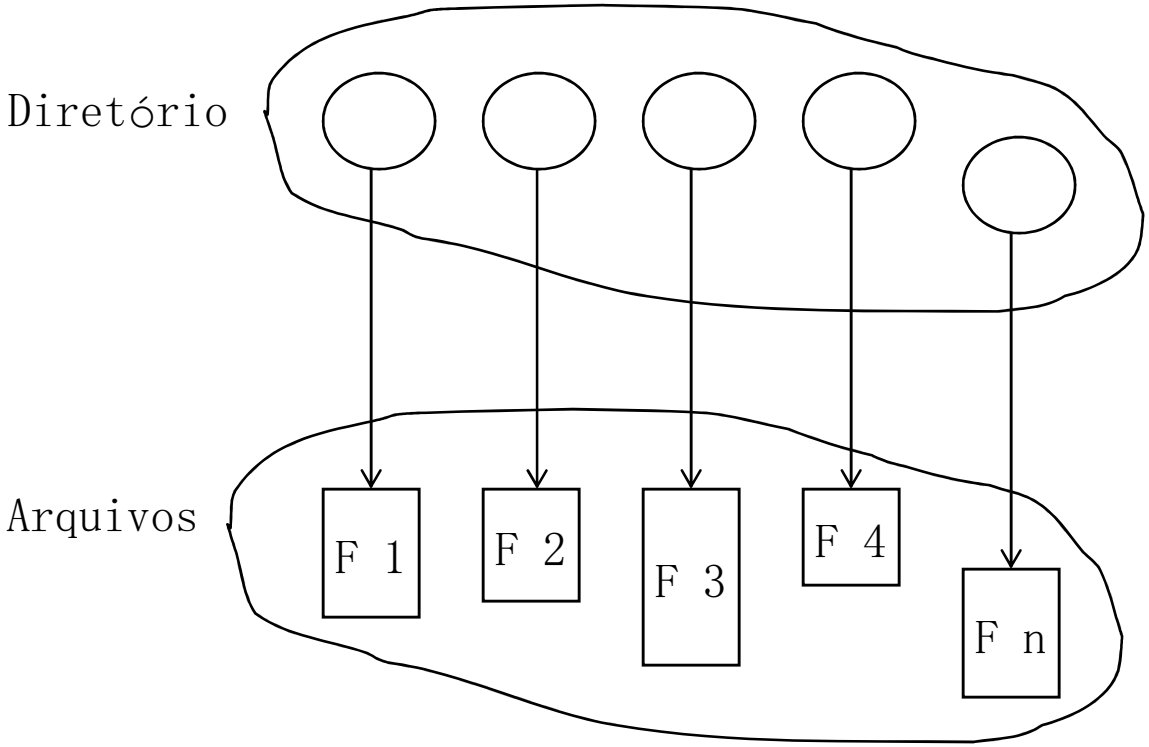
- **Nome** - única informação mantida na forma legível para o usuário.
- **Tipo** - necessário em sistemas que suportam vários tipos.
- **Localização** - apontador para a posição do arquivo no dispositivo.
- **Tamanho** - tamanho corrente do arquivo.
- **Proteção** - controla quem pode ler, escrever ou executar.
- **Data e identificação do usuário** - dados para proteção, segurança e monitoração de uso.
- Informações sobre arquivos são mantidas na estrutura de diretórios, residente em disco.

Operações com Arquivos

- criar
- gravar
- ler
- Reposicionar o ponteiro (seek)
- apagar
- truncar
- abrir(F_i) - pesquisar a estrutura do diretório no disco pela entrada de F_i, e mover o conteúdo da entrada para a memória.
- fechar (F_i) - mover o conteúdo da entrada F_i da memória para a estrutura de diretório em disco.

Estrutura de Diretório

- Uma coleção de nodos contendo informação sobre todos os arquivos.



A estrutura do diretório e os arquivos residem em disco.

Operações de Diretório

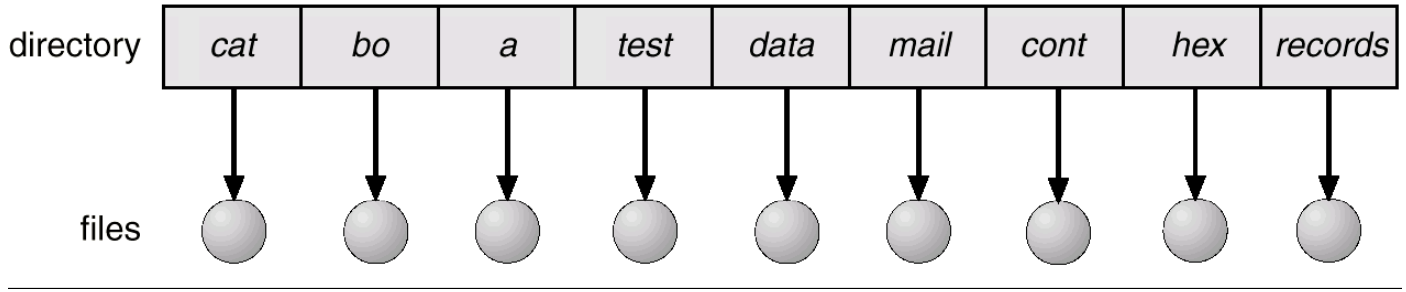
- Pesquisar por um arquivo
- Criar um arquivo
- Remover arquivos
- Listar diretório
- Renomear arquivos

Organização de Diretórios

- Eficiência - localizar arquivos rapidamente.
- Identificação - conveniente para os usuários.
  - Dois usuários podem dar mesmo nome a arquivos distintos.
  - Um mesmo arquivo pode ter vários nomes

Diretório de Um Nível

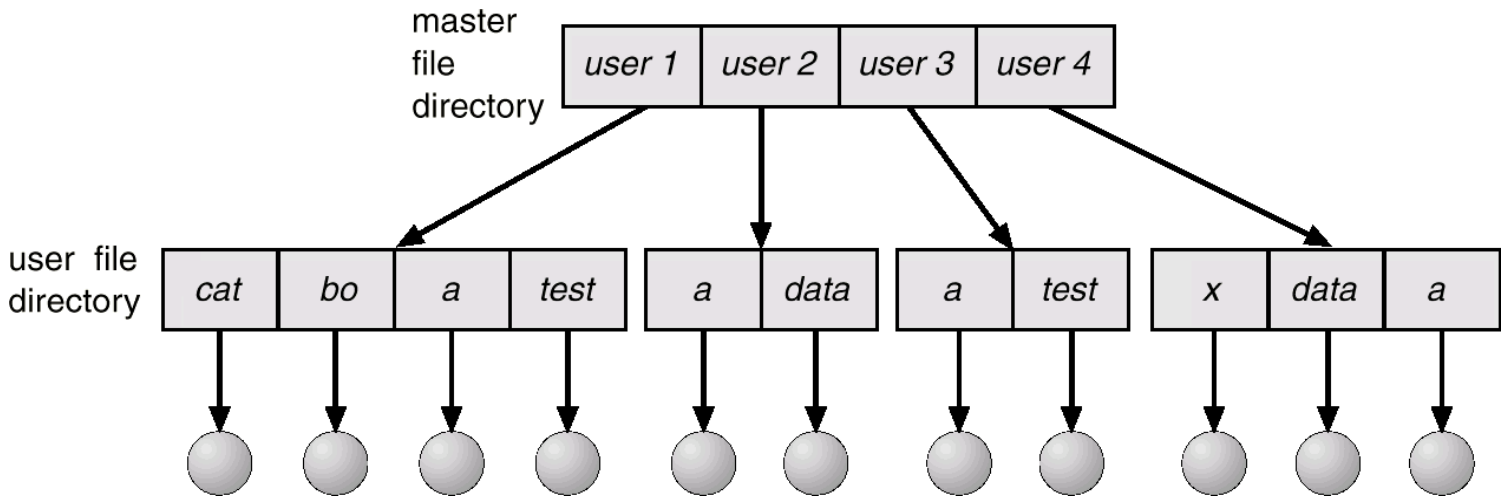
- Um único diretório para todos os usuários.



Problemas com identificação

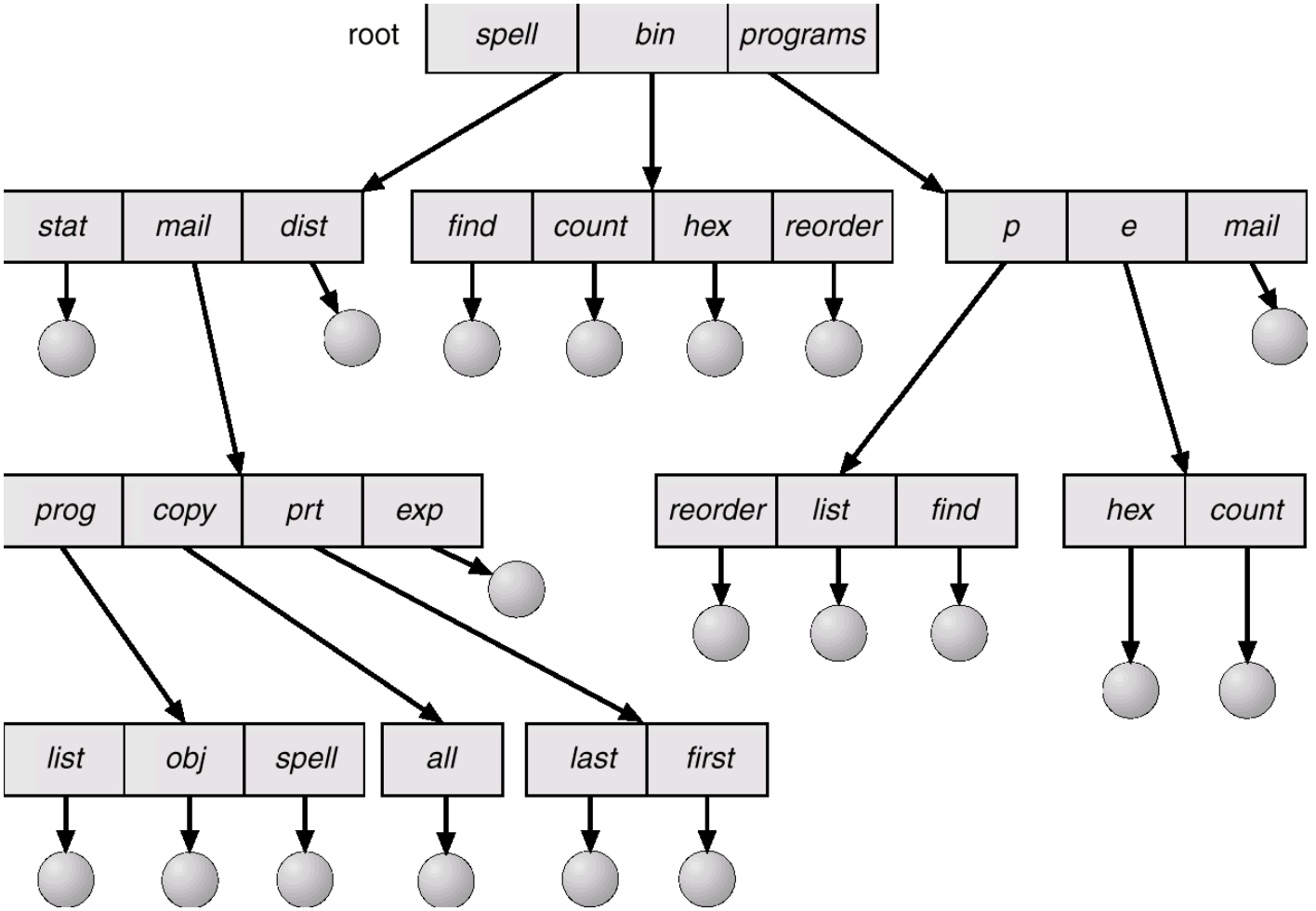
Diretório de Dois Níveis

- Diretório separado para cada usuário



- Arqs. podem ter mesmo nome para usuários distintos
- Busca eficiente
- Nome do caminho (path)

Diretórios com Estrutura em Árvore



Diretórios com Estrutura em Árvore

- Caminho absolutos ou relativos
- Criação de novos arquivos é feita no diretório corrente.
- Remover arquivo

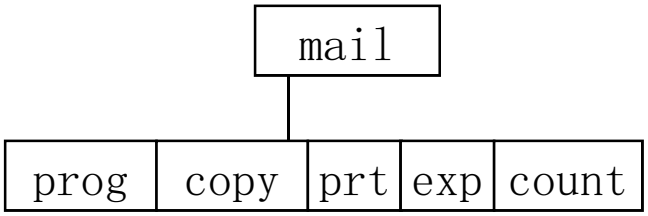
  rm <nome>

- Criação de novos subdiretórios é feita no diretório corrente.

  mkdir <nome>

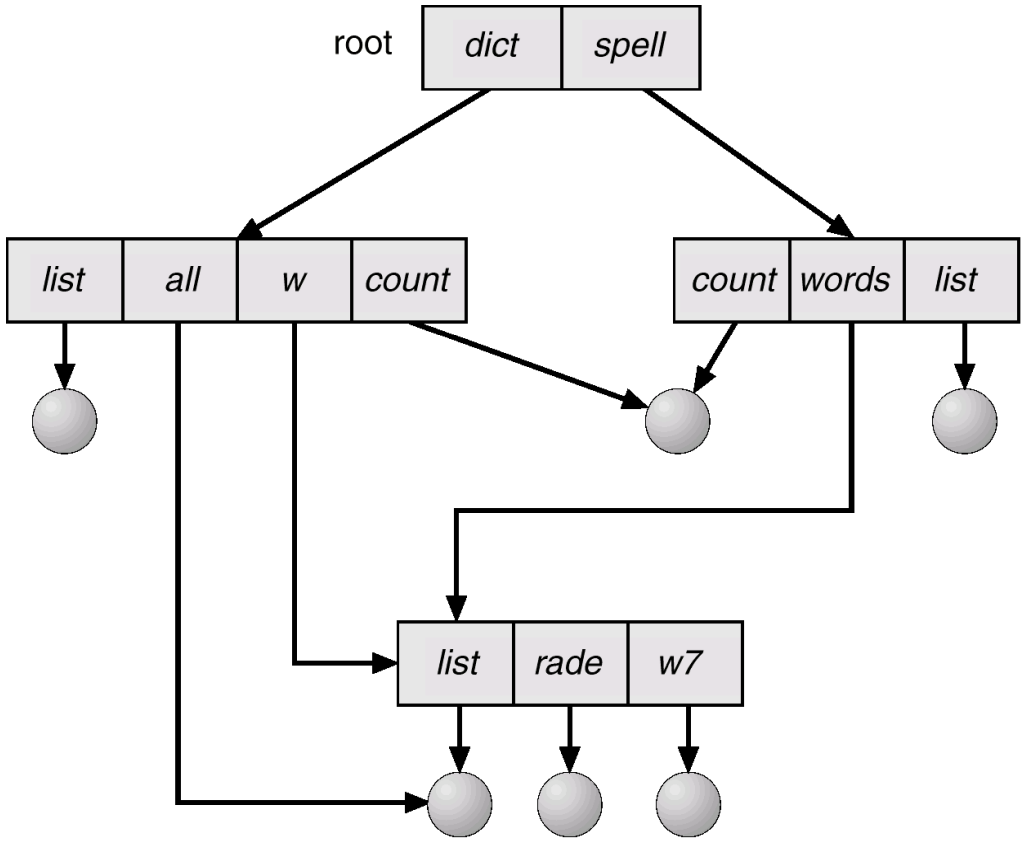
  Exemplo: se dir. corrente /spell/mail

  mkdir count



Diretórios com Estrutura de Grafos

- Possui diretórios e arquivos compartilhados

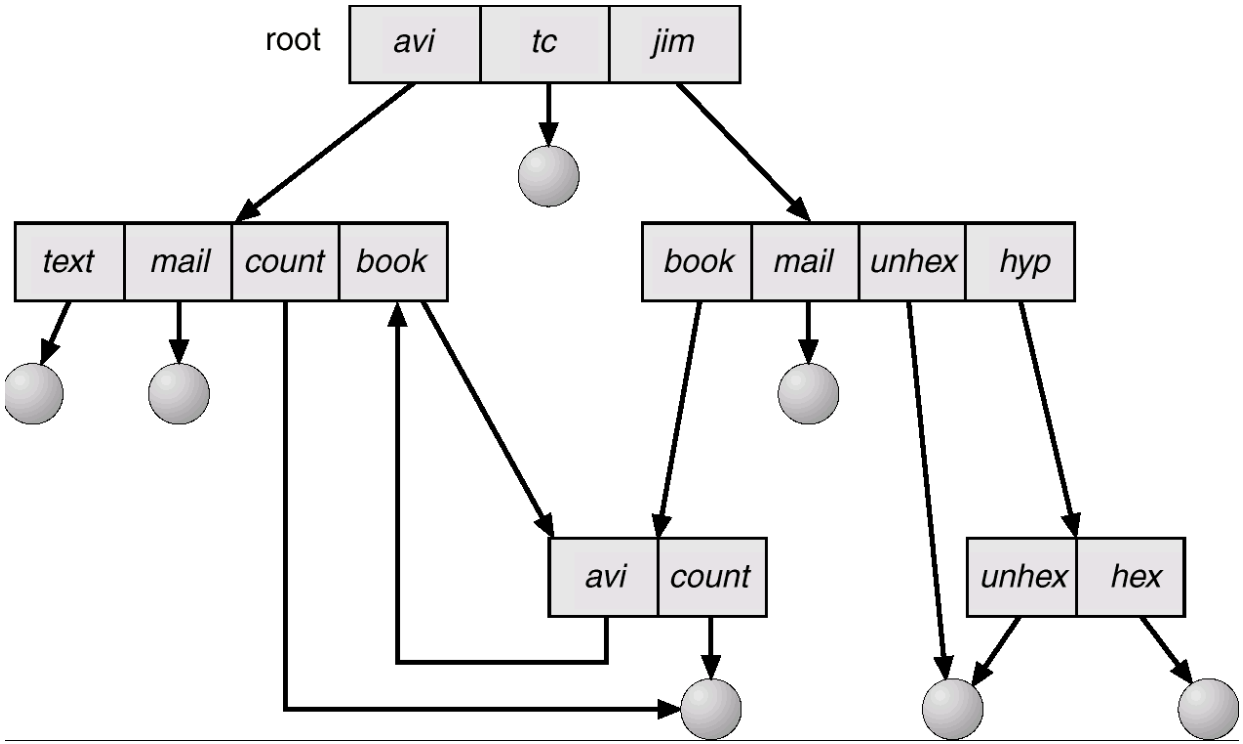


Diretórios com Estrutura de Grafos

- Dois nomes diferentes (aliasing)
- Se *dict* remove *list* $\Rightarrow$ ponteiro pendente.

  Solução:
  - Contadores de referências.

Diretório em Grafo Genérico



Diretório em Grafo Genérico

- Como garantir que não haverá ciclos?
  - Permitir links apenas para arquivos e não diretórios.
  - Coleta de lixo.
  - Cara vez que um link for adicionado, utilizar um algoritmo de detecção de ciclos.

Proteção

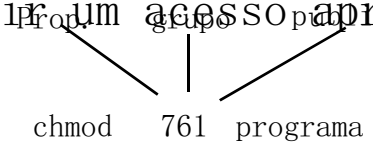
- O proprietário do arquivo deve ser capaz de controlar:
  - o que pode ser usado
  - por quem
- Tipos de acesso
  - Leitura
  - Escrita
  - Execução
  - Adição no fim
  - Remoção
  - Listagem

Listas e Grupos de Acesso

- Modos de acesso: leitura.escrita e execução
- Três classes de usuários

			RWX
a) proprietário	7	⇒	1 1 1
b) grupos	6	⇒	1 1 0
c) acesso público	1	⇒	0 0 1

- O administrador pode criar grupos e adicionar usuários a eles.
- Para um arquivo particular (ex: programa) ou subdiretório, definir um acesso apropriado.



Estrutura de Sistemas de Arquivos

- Estrutura de arquivo
  - Coleção de informação relacionada
  - File system reside em memória secundária (disco).

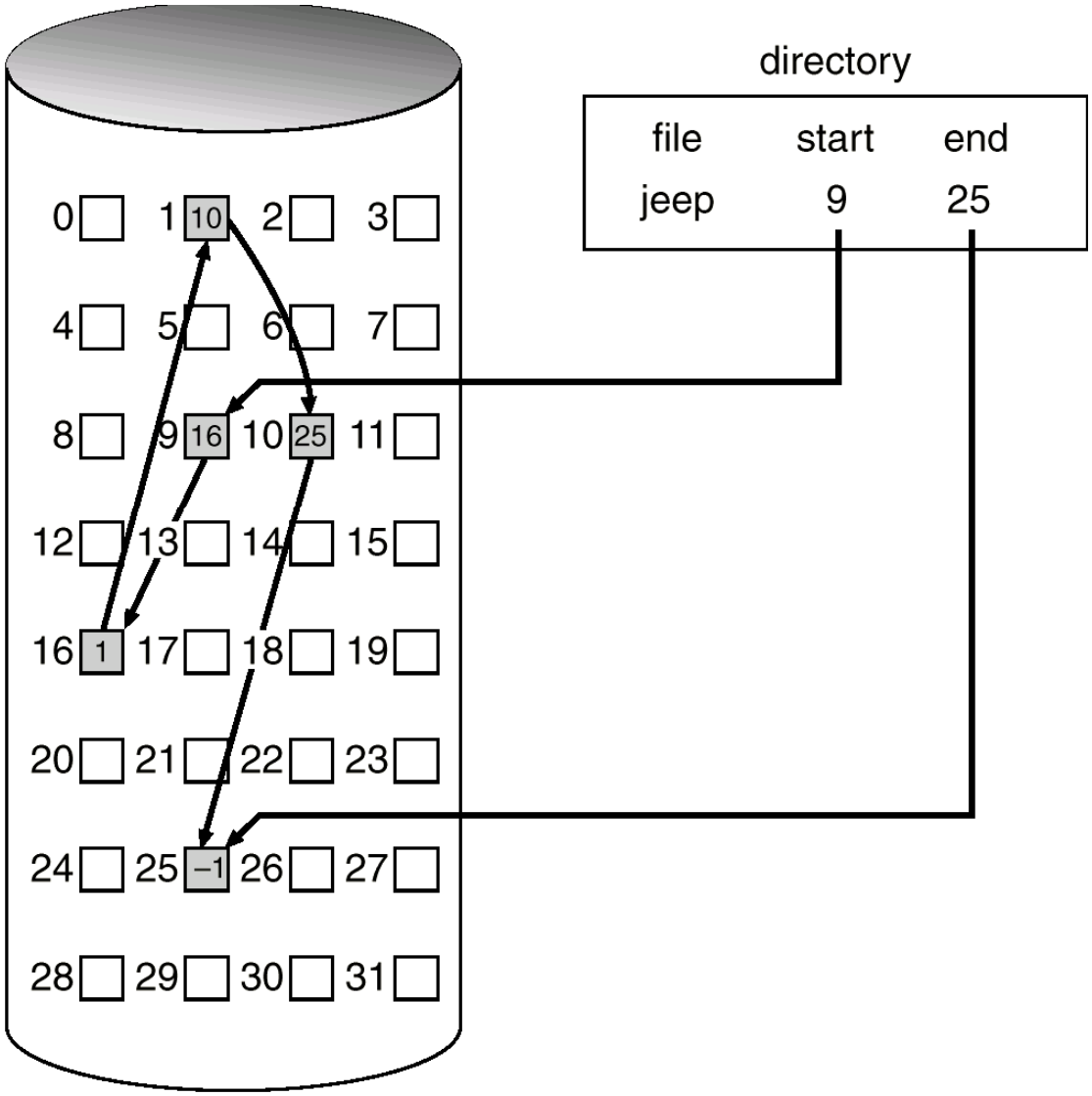
- *File control block* – estrutura de armazenamento que consiste da informação sobre o arquivo

Alocação Contígua

- Cada arquivo ocupa um conjunto de blocos contíguos no disco.
- Simples - apenas a posição inicial (# bloco) e tamanho (número de blocos) são necessários.
- Acesso Randômico.
- Gasto inútil de espaço - fragmentação externa.
- Arquivos não podem crescer.

Alocação Encadeada

- Cada arquivo é uma lista encadeada de blocos de disco. Blocos podem ficar espalhados em qualquer posição no disco.
- Alocado conforme necessidade

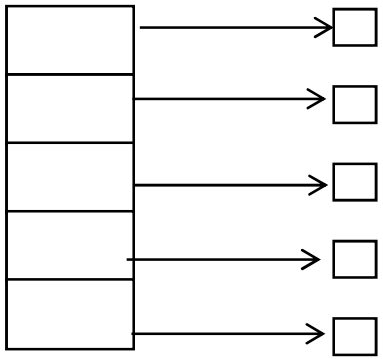

directory
file
start
end
jeep
9
25
0
1
10
2
3
4
5
6
7
8
9
16
10
25
11
12
13
14
15
16
1
17
18
19
20
21
22
23
24
25
−1
26
27
28
29
30
31

Alocação Encadeada

- Simples - necessita apenas do endereço inicial
- Sistema de gerenciamento do espaço livre
  - não há desperdício de espaço
  - exceto o ponteiro do bloco para próximo bloco
- Sem acesso randômico
- *File-allocation table (FAT)* - alocação de espaço em disco usado pelo MS-DOS e OS/2.

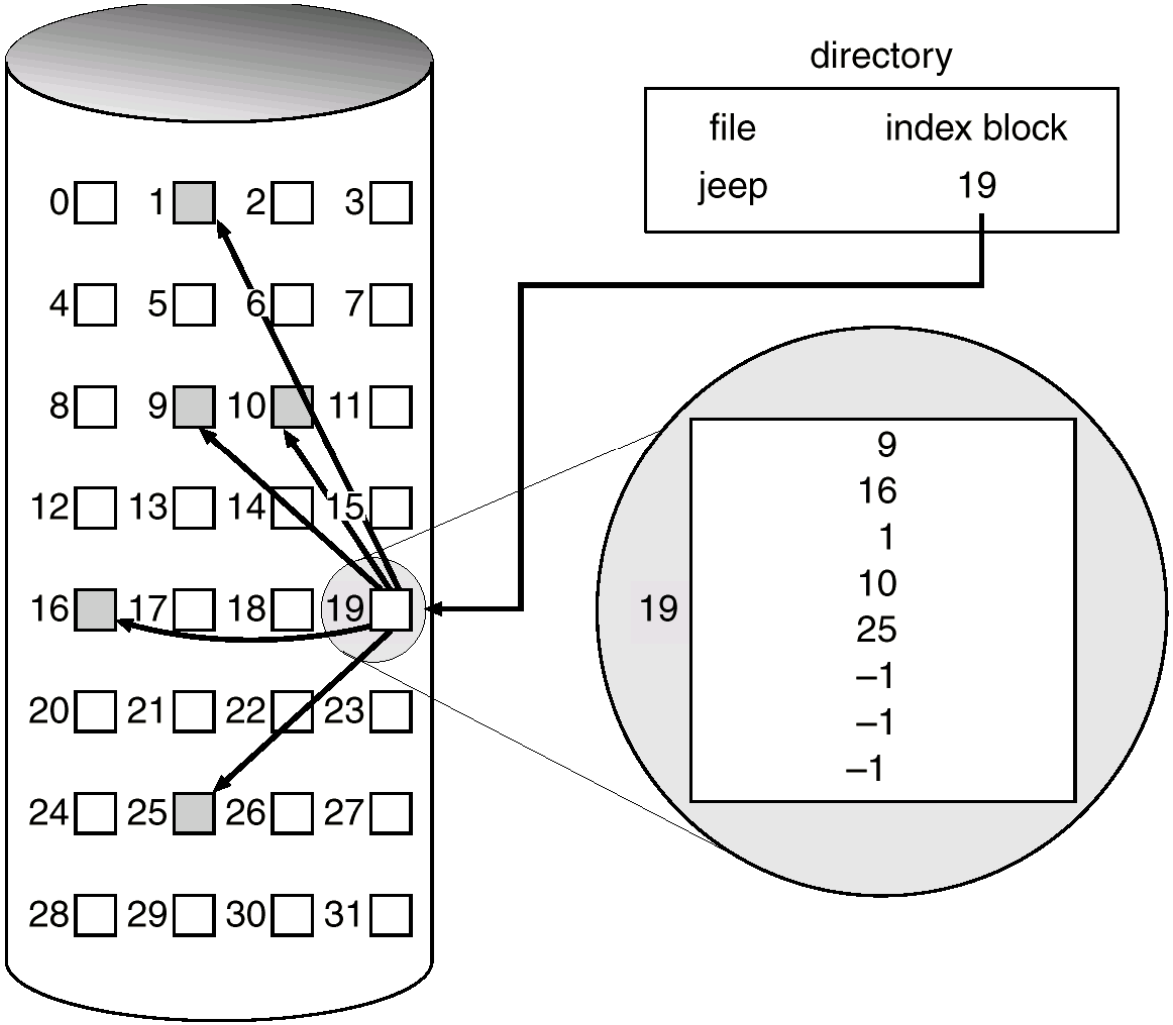
Alocação Indexada

- Reune todos os ponteiros em um bloco de índice



index table

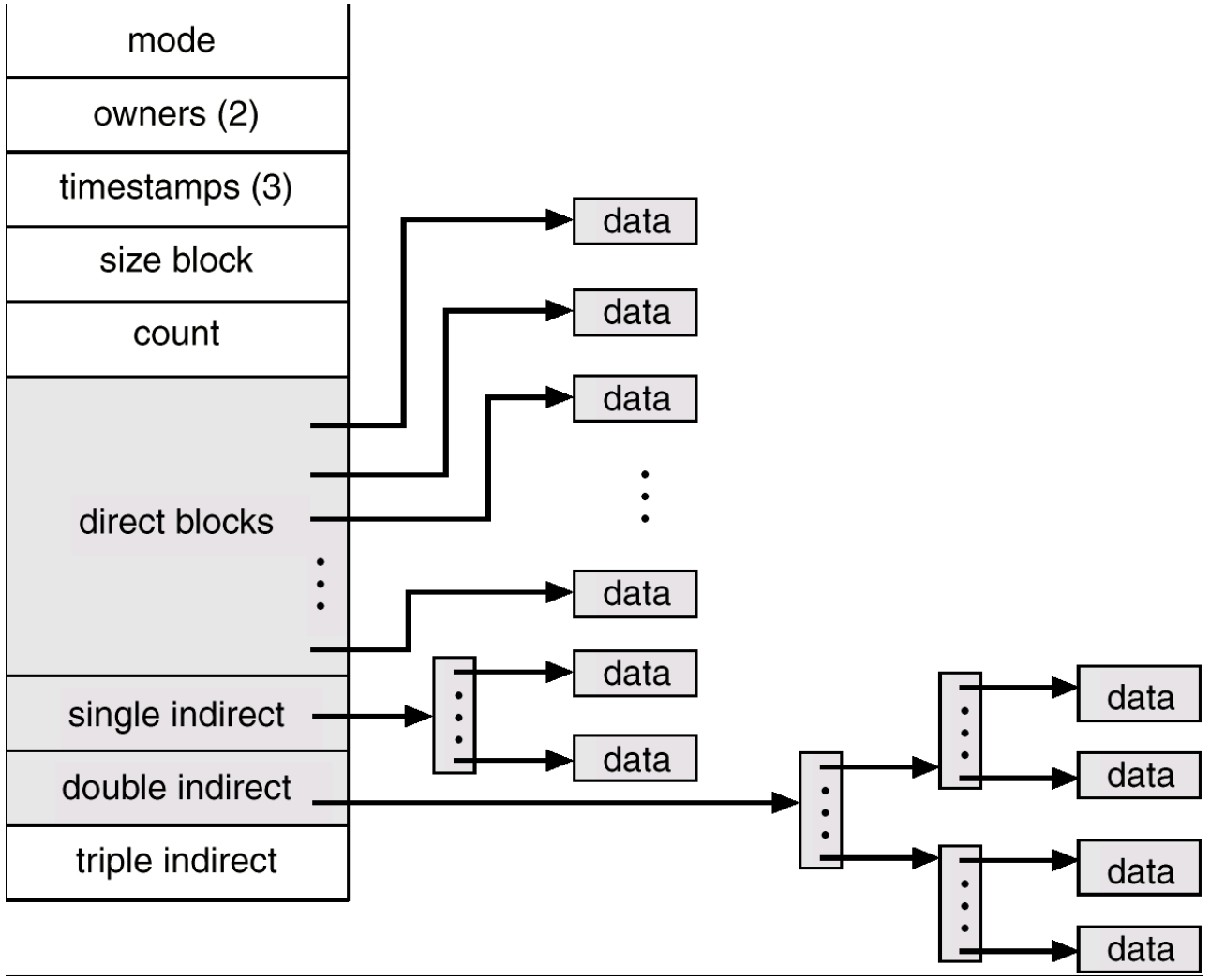
Alocação Indexada



Alocação Indexada

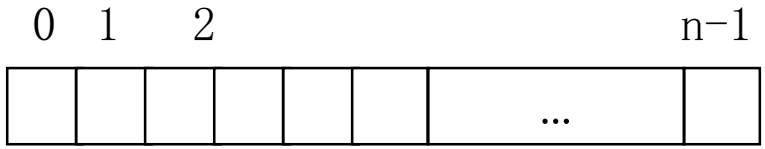
- Necessita de uma tabela de índices
- Acesso randômico
- Acesso dinâmico sem fragmentação externa, mas com overhead para indexação.
- Desperdiça bloco para ponteiro
  - Um arquivo que use 2 blocos precisa de um bloco a mais para os ponteiros - na realidade utiliza 3 blocos

Esquema combinado: UNIX (4K bytes por bloco)



Gerência de Espaço Livre

- Vetor de bits (n blocos)



$$bit[i] = \begin{cases} 0 \Rightarrow \text{bloco}[i] \text{ livre} \\ 1 \Rightarrow \text{bloco}[i] \text{ ocupado} \end{cases}$$

- Mapa de bits requer espaço extra
- Exemplo:

  tamanho bloco = 2^{12} bytes

  tamanho disco = 2^{30} bytes (1 gigabyte)

  $n = 2^{30}/2^{12} = 2^{18}$ bits (ou 32K bytes)

- Fácil conseguir arquivos contíguos: 1000011

Gerência de Espaço Livre

- Lista encadeada
  - Não é fácil conseguir arquivos contíguos
  - Não há desperdício de espaço

Implementação de Diretórios

- Lista linear de nomes de arquivos com ponteiros para os blocos de dados.
  - simples de programar
  - consome tempo para executar
- Tabelas Hashing - lista linear com estrutura de dados hashing.
  - reduz tempo de busca em diretórios
  - *colisões* - 2 nomes de arquivos na mesma posição

Sistemas de Arquivos Conhecidos

- **Ext:** Significando "Extended file system" ou "Sistema de arquivos extendido", primeiro sistema de arquivos criado unicamente para o linux em 1992.
- **Ext2:** Suportava discos com até 2 TB e não suportava journaling. Porém, já era um avanço considerável do Ext. Por não usar journaling pode ser usado em pendrives e derivados.
  - Journaling é um serviço de log de atividade do sistema arquivos
  - registra as mudanças que serão feitas no sistema de arquivos e depois grava as mudanças no disco.
  - utiliza arquivos que guardam informações sobre outros arquivos (metadados) e arquivos com as mudanças que serão escritas no disco

Sistemas de Arquivos Conhecidos

- **Ext3**: Igual ao Ext2, se diferenciando apenas por ter journaling.
- **Ext4**: Possui várias funções como redução na fragmentação do sistema, criptografia, etc.

Sistemas de Arquivos Conhecidos

- **ReiserFS**: A sua criação foi um avanço para sistemas linux. Foi substituído pelo Reiser4.
  - O Reiser4 ficou estagnado, sabe o por que? O seu principal desenvolvedor, Hans Reiser, foi preso em 2008.
- **FAT**: FAT é um sistema de arquivos da Microsoft e tem algumas versões: FAT16, FAT32, exFAT.
  - melhor escolha para pendrives e derivados, já que não possuem journaling
  - velocidade à mais em momentos de escrita e leitura.