

# Estrutura de Diretório

Igor Gustavo Hoelscher

Renan Arend

Rogério Corrêa Medeiros

# Introdução

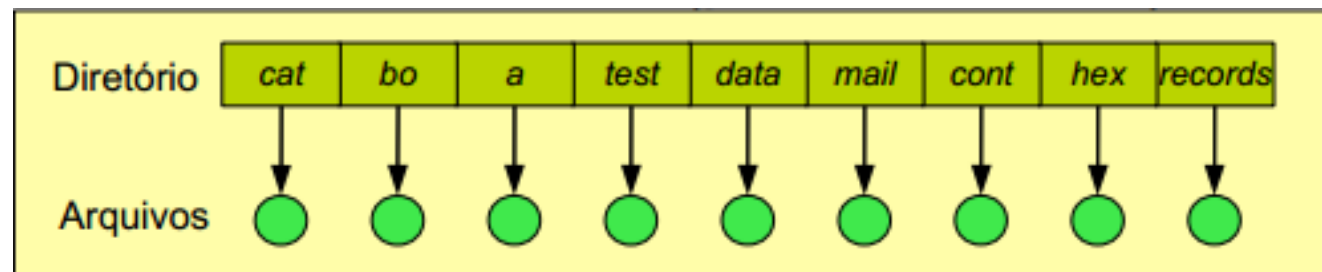
- ▶ Um disco possui pelo menos uma partição, considerado disco virtual, que contém informações sobre os arquivos dentro dela. Essas informações são mantidas em diretórios.
- ▶ Pode ser visto como uma tabela de símbolos que traduz os nomes dos arquivos em suas entradas do diretório. (Silberschatz, 2004)
- ▶ Operações sobre o diretório:
  - ▶ Busca de arquivos;
  - ▶ Criar um arquivo;
  - ▶ Apagar arquivo;
  - ▶ Listar diretório;
  - ▶ Renomear arquivo;
  - ▶ Percorrer sistema de arquivos;

# Estruturas Lógicas de um Diretório

- ▶ Diretório em um nível.
- ▶ Diretório em dois níveis.
- ▶ Diretórios estruturados em árvores.
- ▶ Diretórios com estruturas de grafos acíclicos.
- ▶ Diretório como estrutura de grafo geral.

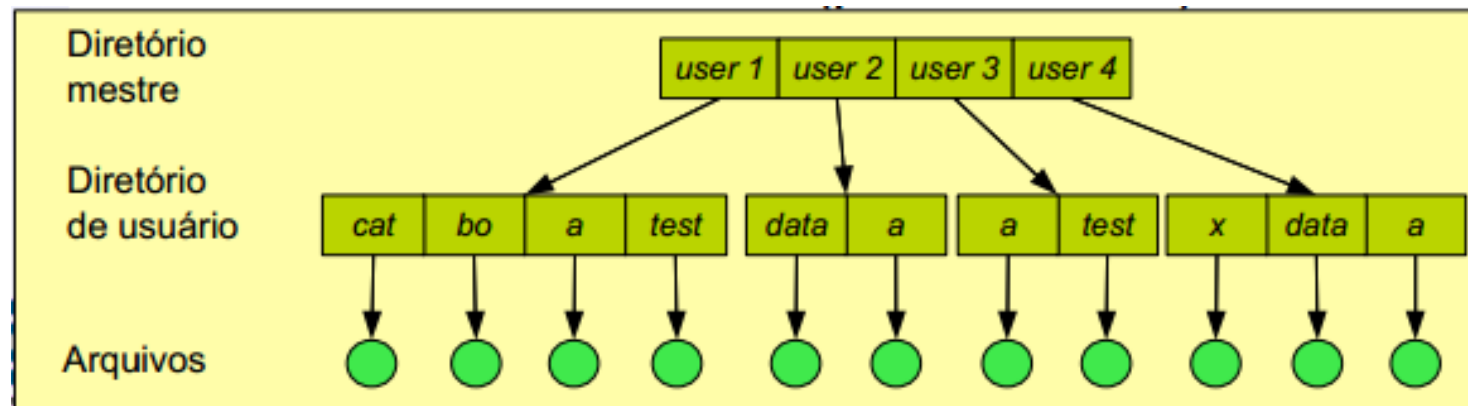
# Diretório em Um Nível

- ▶ Todos os arquivos estão contidos no mesmo diretório.
- ▶ Não recomendado para sistemas com mais de um usuário.
- ▶ Problema da “colisão de nomes”.
- ▶ Pouco eficiente na busca de arquivos.



# Diretório em Dois Níveis

- ▶ Cada usuário tem um diretório pessoal (UFD - User File Directory).
- ▶ O segundo nível, MFD(Master File Directory), é indexado pelo nome do usuário ou pelo número de contabilidade.
  - ▶ Cada ponto de entrada do MFD aponta para o UFD de um usuário



# Diretório em Dois níveis

- ▶ Permite usuários com nomes de arquivo iguais.
- ▶ Isolamento entre os usuários.
  - ▶ Vantagem quando usuários são independentes.
  - ▶ Desvantagem quando usuários cooperam em alguma tarefa e precisam de arquivos compartilhados.
- ▶ Alguns sistemas permitem acesso por um usuário ao diretório de outro.

# Diretórios Estruturados em Árvore

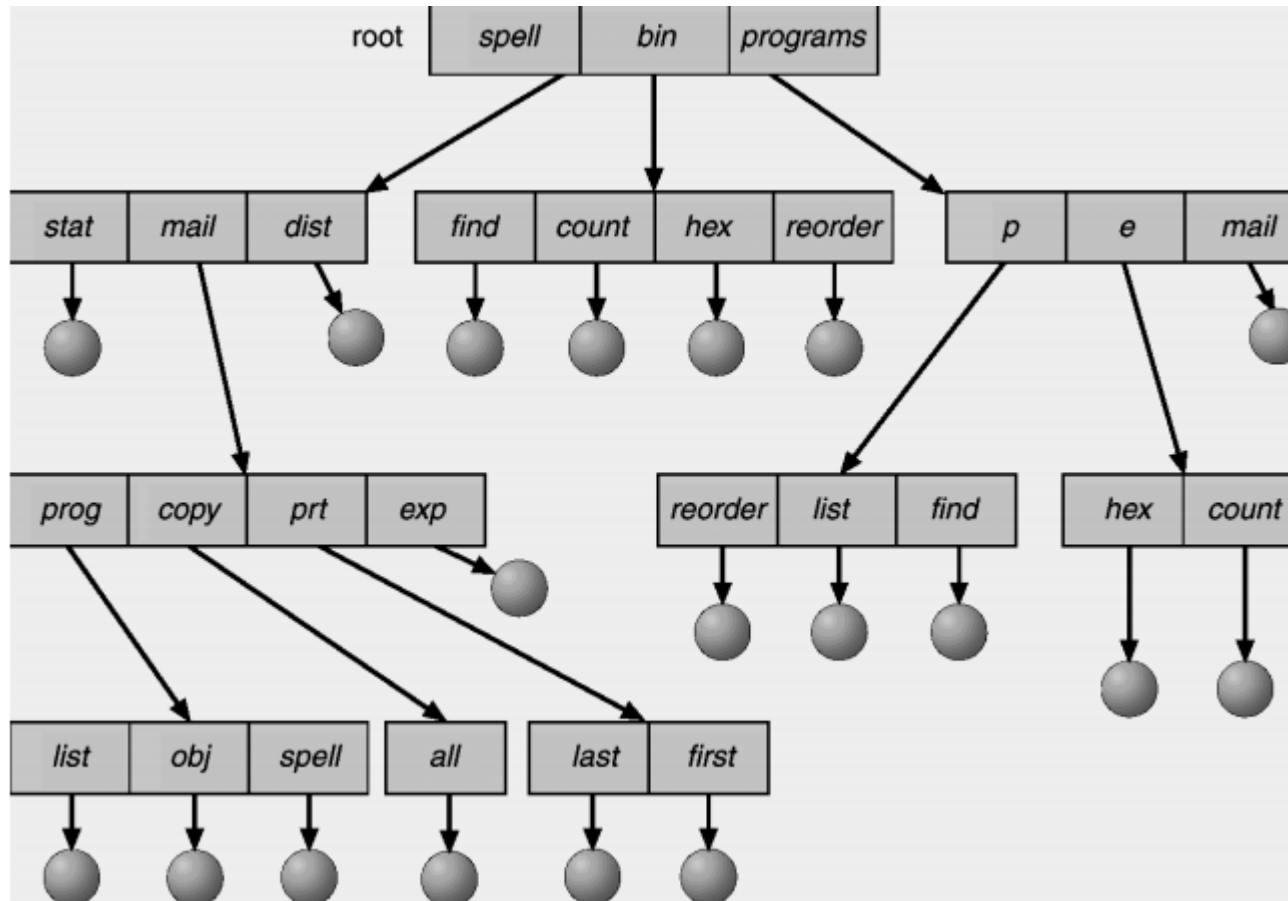
- ▶ Generalização do **diretório a dois níveis**
- ▶ Os usuários podem criar seus próprios sub-diretórios para organizar seus arquivos;
- ▶ A árvore possui um **diretório raiz**;
- ▶ Cada arquivo possui um **pathname** (nome do arquivo composto pelos nomes de diretórios que formam o caminho da raiz até ele).

# Diretórios Estruturados em Árvore

- ▶ Qualquer arquivo (ou subdiretório) pode ser identificado de forma não ambígua através de seu pathname
  - ▶ Conceito de **diretório corrente**, **caminho absoluto** e **caminho relativo**
- ▶ **Diretório corrente** (diretório de trabalho)
  - ▶ Qualquer nó da árvore
- ▶ **Caminho Absoluto**
  - ▶ Quando se referencia um arquivo a partir da raiz da árvore
- ▶ **Caminho Relativo**
  - ▶ Quando se referencia um arquivo a partir do diretório corrente



# Diretórios Estruturados em Árvore

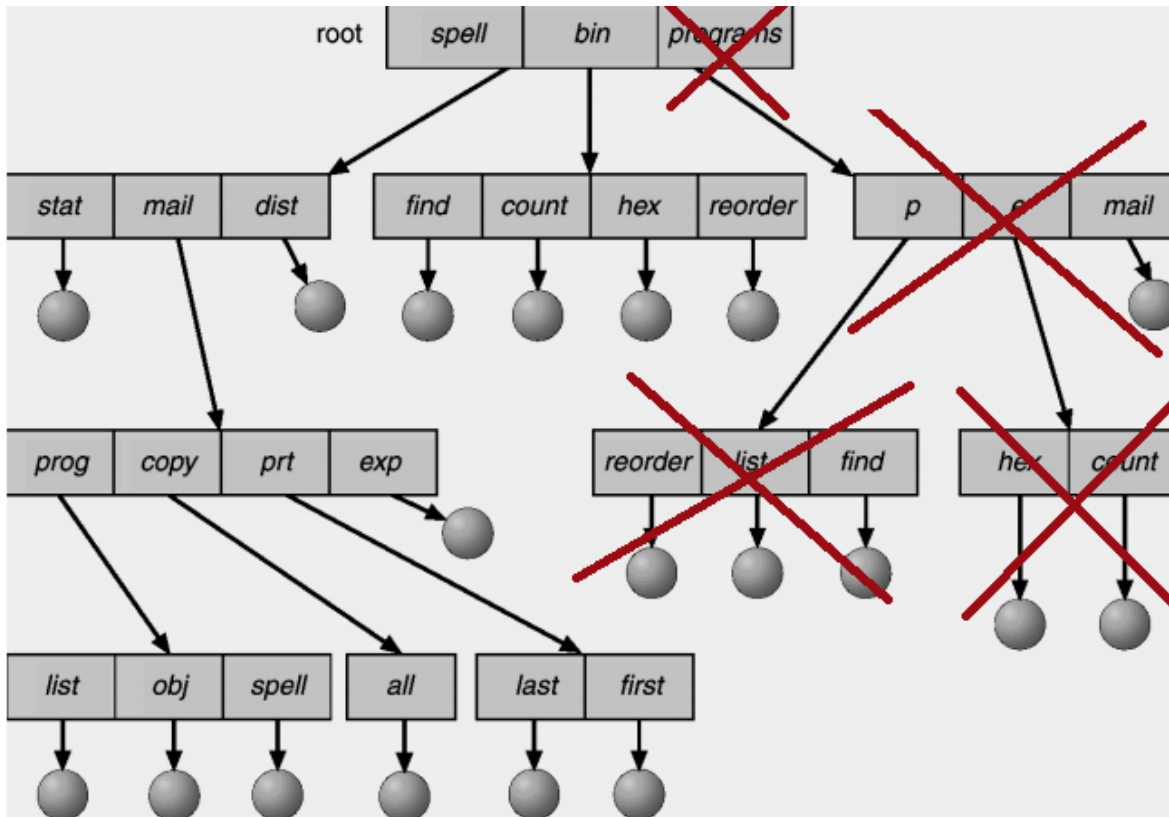


# Diretórios Estruturados em Árvore

- ▶ Nome de caminho absoluto ou relativo.
- ▶ A criação de um novo arquivo é feita no diretório atual.
- ▶ Excluir um arquivo
  - ▶ `rm <nome_arquivo>`
- ▶ A criação de um novo subdiretório é feita no diretório atual
  - ▶ `mkdir <nome_diretório>`
- ▶ Exemplo: se o diretório atual é `/programs`
  - ▶ `mkdir mail`

# Diretórios Estruturados em Árvore

E se excluir um diretório raiz?  
É excluída toda a sub-árvore iniciada neste diretório raiz.



Exclusão de “programs”  
⇒ exclui toda a sub-árvore iniciada em “programs”.

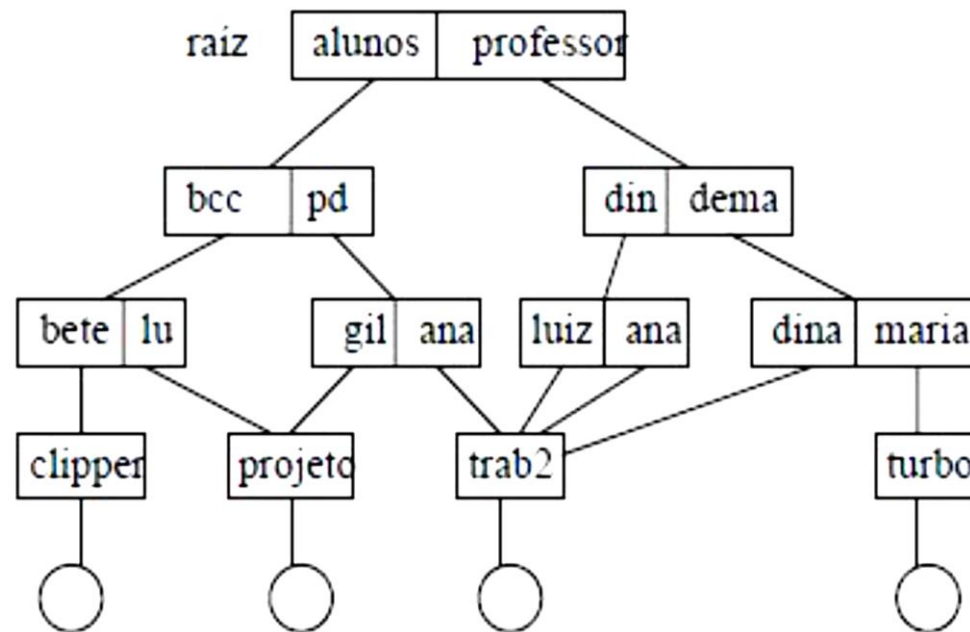
# Prós e Contras da Estutura em Árvore

- ▶ **Vantagem:**
  - ▶ Procura eficiente por arquivos
  - ▶ Possibilidade de agrupamento de arquivos
- ▶ **Desvantagem:**
  - ▶ Compartilhamento de arquivos

# Diretórios em Grafos Acíclicos

- ▶ Imaginem que dois projetistas compartilham do mesmo arquivo. Eles desejam que esse arquivo esteja originalmente em um subdiretório localizado em cada um de seus próprios diretórios.
- ▶ Uma estrutura de árvore não permite esse tipo de compartilhamento de diretórios e arquivos.
- ▶ Já na estrutura de grafo acíclico, o mesmo arquivo ou subdiretório poderá estar em dois diretórios diferentes.

# Diretórios em Grafos Acíclicos



# Diretórios em Grafos Acíclicos

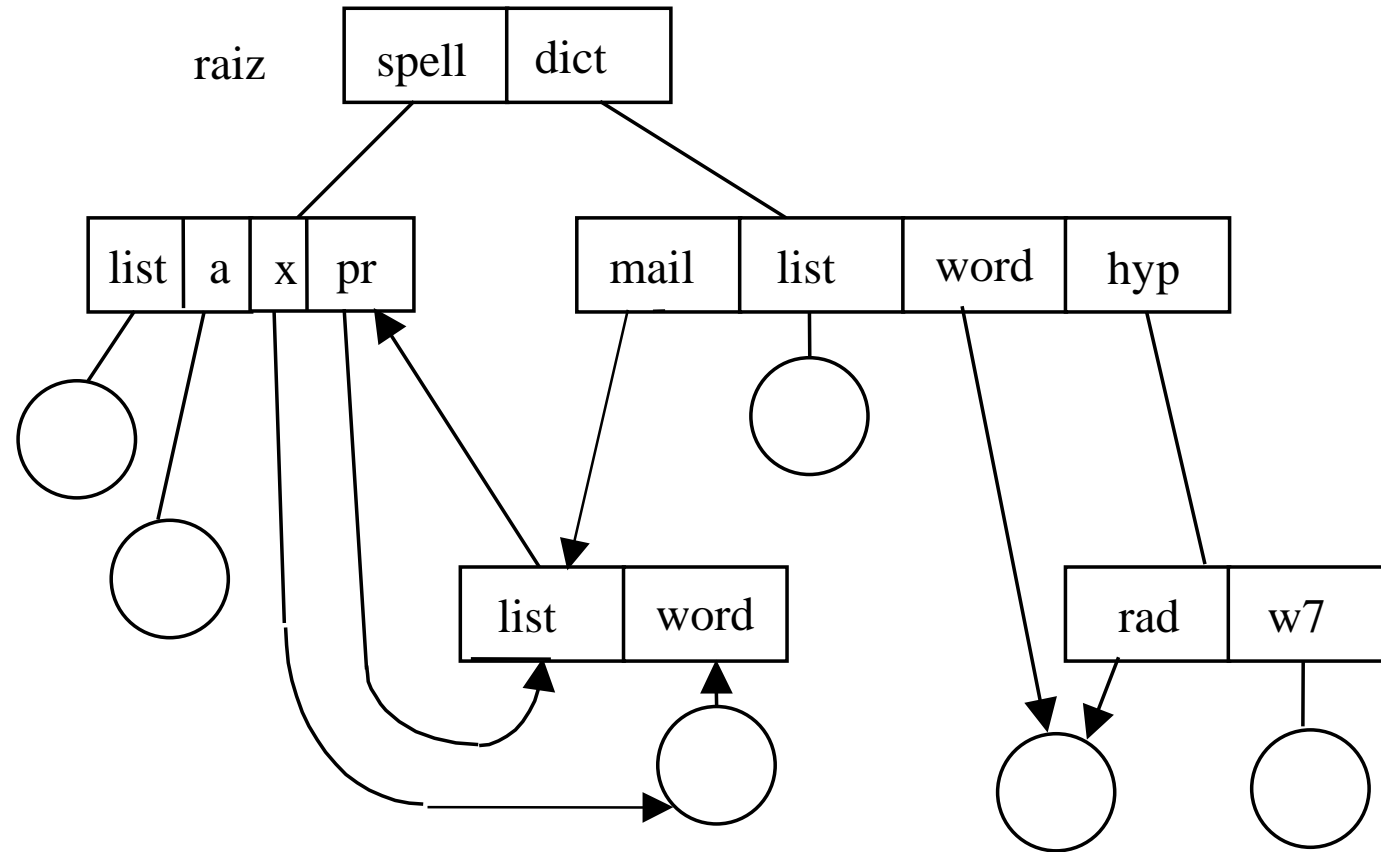
- ▶ Duas abordagens são usadas para implementar esse tipo de estrutura. A mais fácil de ser tratada é a criação de uma nova entrada de diretório conhecida como link. O link nada mais é do que um ponteiro para o arquivo ou diretório compartilhado.
- ▶ Quando acessado, o link resolve o caminho para localizar o arquivo original.
- ▶ Outra abordagem seria duplicar todas as informações compartilhadas para cada diretório. No entanto, qualquer modificação ou exclusão torna complexa manutenção da consistência entre os arquivos.
- ▶ Ao usar links, quando o arquivo original, os links não irão mais resolver o caminho, porque ele não existe mais, e poderão ser excluídos no momento em que há uma tentativa de acesso. Porém, se o arquivo original manter uma lista de links que apontam para si, esses links podem ser excluídos imediatamente.

# Diretórios em Grafos Acíclicos

- ▶ No UNIX, quando os links são simbólicos e o arquivo original é excluído, o sistema deixa que o usuário perceba que estes links não resolvem mais o diretório para o qual apontavam e aí então os exclua.
- ▶ No entanto, quando são criados links não-simbólicos (hard links), o sistema mantém um contador de referências para o arquivo ou diretório original. No momento da exclusão, ela somente será efetivada quando o contador chegar a zero, ou seja, todas as referências para o local forem também excluídas.



# Diretórios em Grafos Gerais



# Diretórios em Grafos Gerais

- ▶ Essa estrutura elimina um dos maiores problemas da estrutura de grafos acíclicos, que é exatamente não permitir ciclos.
- ▶ Com a existência de ciclos, os algoritmos de tratamento, principalmente de busca, são muito mais complexos. Algoritmos mal projetados podem manter um sistema buscando um arquivo infinitamente, por estar percorrendo em ciclos.
- ▶ Ainda, permitir a existência de ciclos provoca uma anomalia conhecida como auto-referência. Ou seja, o contador de referências, pode ser diferente de zero mesmo que não existam mais hard links. Assim, tem-se um problema no momento da exclusão de um arquivo compartilhado.
- ▶ Uma técnica usada é a Coleta de Lixo (*Garbage Collection*).

# Diretórios em Grafos Gerais

- ▶ Algoritmos são muito complexos e demorados, ainda mais quando esses grafos são arquivos reais, gravados em disco. Desse modo, trabalhar com uma estrutura em grafo acíclico é muito mais fácil.
- ▶ É necessário apenas que não se permita a criação de ciclos quando são estabelecidos novos links. No entanto, algoritmos para a verificação de ciclos também são caros computacionalmente. Atualmente, o mais fácil a se fazer numa busca é evitar links. Ou seja, o arquivo original será alcançado em algum momento, em seu diretório original, e qualquer referência será evitada durante a varredura de disco.