

Sistemas Operacionais

Interface do Sistema de Arquivos

Lesandro Ponciano

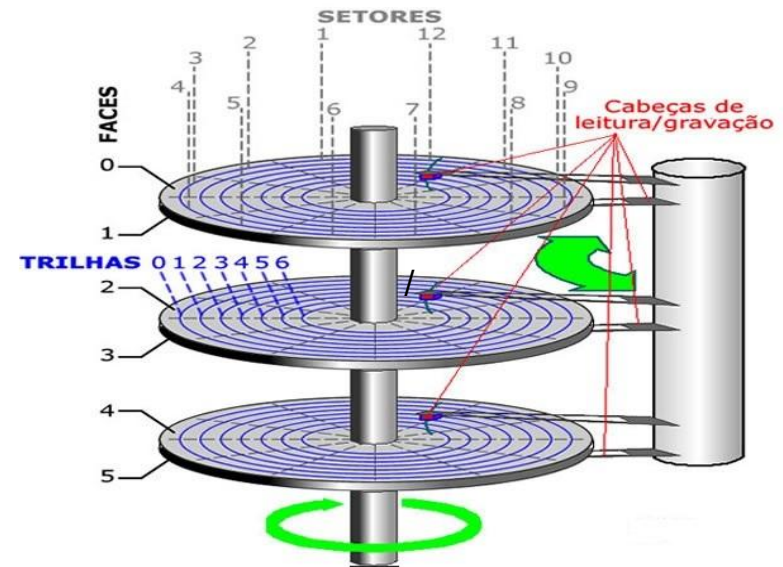
2024

Objetivos da Aula

- Explicar a função do sistema de arquivos
- Descrever as interfaces para sistemas de arquivos
- Discutir o projeto de sistemas de arquivos
- Examinar a proteção de sistemas de arquivos

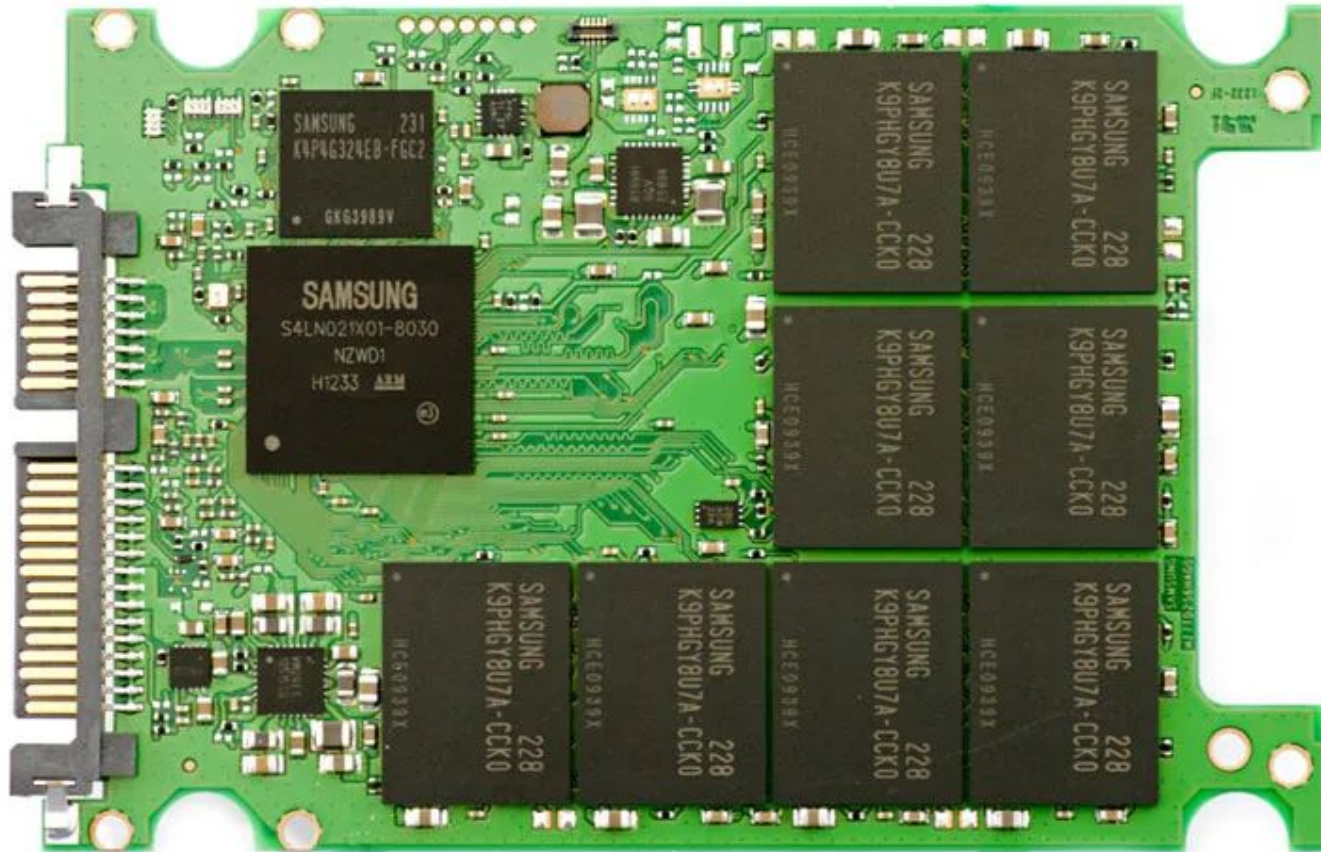
Armazenamento de Informações

- Computadores podem armazenar informações em várias mídias de armazenamento, ex:
 - Discos magnéticos
 - Fitas magnéticas
 - Discos ópticos
- O sistema operacional (SO) deve fornecer uma **visão lógica** uniforme do armazenamento de informações



Ex: Disco magnético

Solid State Drive (SSD)



Todo em transistores, sem operação mecânica.

Principais componentes: Interface de interconexão, controlador, memórias flash

Conceito de **Arquivo**

- O SO abstrai das propriedades físicas de seus dispositivos de armazenamento a definição de uma unidade lógica de armazenamento, o **arquivo**
- O SO mapeia arquivos para dispositivos físicos
 - Geralmente dispositivos não voláteis
 - O conteúdo persiste após falhas e reinicializações
- Um arquivo é um conjunto nomeado de informações relacionadas que são gravadas em memória secundária

Atributos de Arquivos

- Os atributos de um arquivo variam de um SO para outro, mas normalmente são:
 - **Nome**: informação legível ao ser humano
 - **Identificador**: número que identifica o arquivo no sistema de arquivos
 - **Tipo**: característica dos dados armazenados, tais como: arquivo texto, arquivo executável
 - **Tamanho**: quantidade de dados que ele armazenada (bytes)
 - **Proteção**: informações de controle de acesso que determinam quem pode fazer e que tipo de operações pode fazer sobre o arquivo
 - **Hora, data e identificação do usuário**

Operações sobre Arquivos

- Um arquivo é um tipo abstrato de dados (TAD), portanto, há uma definição das operações que podem ser realizadas sobre eles
 - Criação de um arquivo
 - Gravação de um arquivo
 - Leitura de um arquivo
 - Reposicionamento dentro de um arquivo
 - Exclusão de um arquivo
 - Truncamento de um arquivo
- O SO fornece chamadas de sistema para que tais operações possam ser realizadas

Tabela de Arquivos Abertos

- O SO mantém uma tabela de arquivos abertos, com as informações associadas a um arquivo aberto
 - Ponteiro do arquivo
 - Contagem de arquivos abertos
 - Localização do arquivo em disco
 - Direitos de acesso

Tipos de Arquivos

- Uma forma comum para a implementação de tipos de arquivos é incluir o tipo como parte do nome do arquivo, que tem duas partes
 - nome
 - *extensão*
- O SO usa a extensão para indicar o tipo do arquivo e o tipo de operações que podem ser feitas sobre ele

file type	usual extension
executable	exe, com, bin or none
object	obj, o
source code	c, cc, java, perl, asm
batch	bat, sh
markup	xml, html, tex
word processor	xml, rtf, docx
library	lib, a, so, dll
print or view	gif, pdf, jpg
archive	rar, zip, tar
multimedia	mpeg, mov, mp3, mp4, avi

Estrutura de Arquivos

- Arquivos têm estruturas internas que correspondem à expectativa dos programas que os lêem
 - .doc, entendido por editores de texto
 - .pdf, entendido por leitores de pdf
 - .exe, entendido como executável pelo Windows
- Alguns arquivos devem estar de acordo com uma estrutura obrigatória que é entendida pelo SO
 - ex.: arquivo executável precisa ter uma estrutura específica que permite determinar onde fica a primeira instrução
 - Alguns SOs podem dar suporte a múltiplas estruturas de arquivos

Estrutura Interna dos Arquivos

- Internamente pode ser difícil para o SO localizar um deslocamento dentro de um arquivo
- Normalmente um sistema de disco têm um tamanho de **bloco** bem definido que é determinado pelo tamanho de um setor do disco
 - Todo I/O de disco é realizado em unidades de um bloco e todos os blocos têm o mesmo tamanho
- Um registro lógico pode ser menor que um bloco, assim vários registros lógicos podem ser **empacotados** em um mesmo bloco

Fragmentação do Disco

- A quantidade de registros que cabe em um bloco físico é determinada por
 - Tamanho do registro lógico
 - Tamanho do bloco físico
 - Técnica de empacotamento usada
- Um arquivo é uma sequência de blocos
 - Parte do último bloco de cada arquivo é, em geral, perdida
 - Tal desperdício dentro do bloco gera fragmentação interna
 - Quanto maior o tamanho do bloco, maior a fragmentação interna

Métodos de Acesso

- As informações dos arquivos podem ser acessadas de várias maneiras
 - Acesso sequencial
 - Acesso direto
 - Acesso baseado em índice

Acesso Sequencial

- As informações existentes no arquivo são processadas em ordem, um registro (linha, palavra, byte) após outro
 - Os registros são lidos sequencialmente a partir do início
 - As escritas são realizadas no fim do arquivo
- A partir de uma posição corrente, pode-se
 - Ler o próximo registro
 - Ir para o fim do arquivo
 - Ir para o início do arquivo

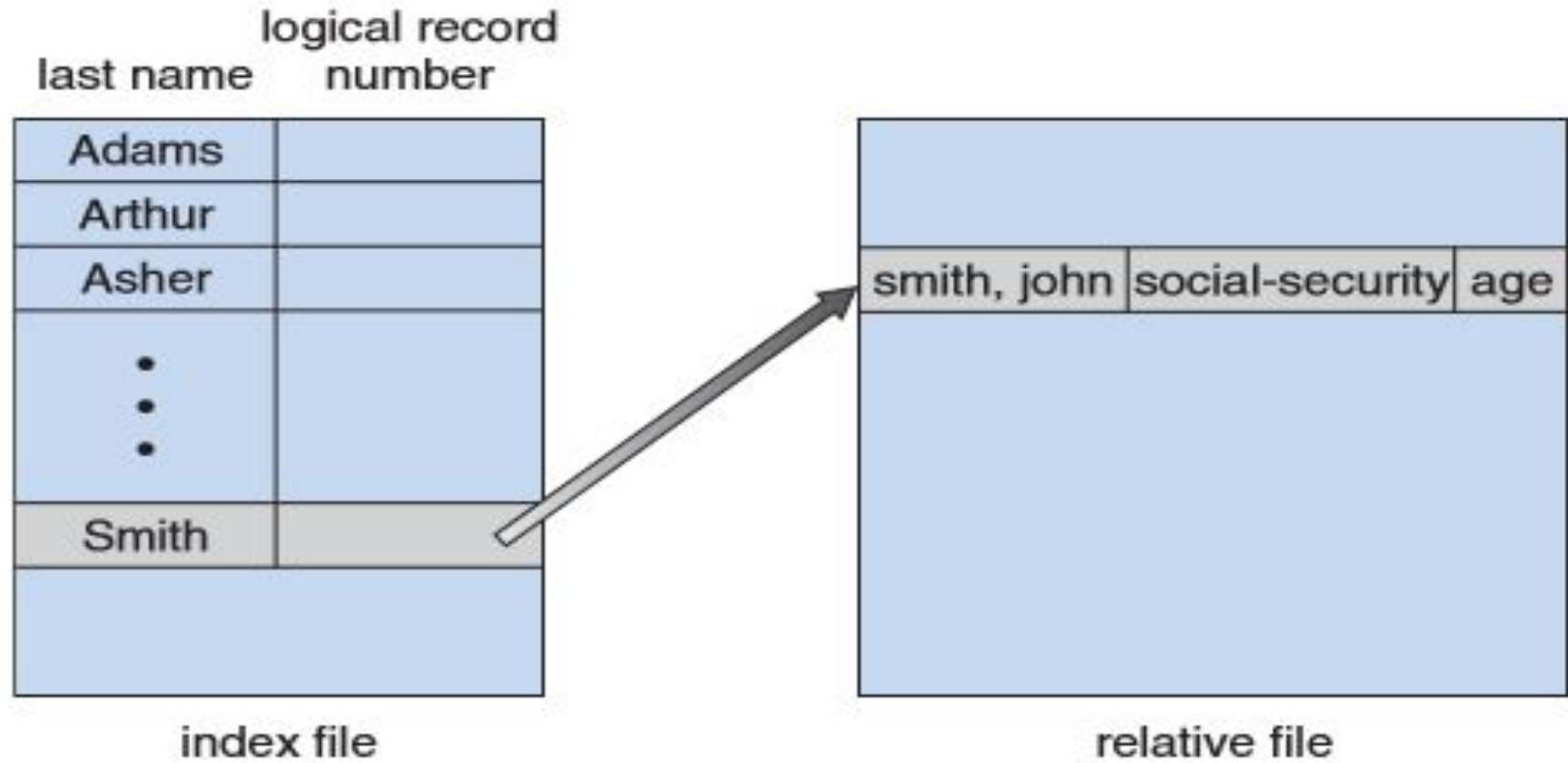
Acesso Direto

- Não há restrições quanto à ordem de leitura ou gravação em um arquivo de acesso direto
- O arquivo é considerado uma sequência de blocos ou registros lógicos numerados
 - Cada bloco tem tamanho fixo
 - Pode-se ler o bloco 10 e, em seguida, o bloco 14
- Os bancos de dados, frequentemente, utilizam esse tipo de acesso

Acesso Baseado em Índice

- Existem dois arquivos
 - Um arquivo de índices
 - Um arquivo relacionado
- O arquivo de índice contém um termo de indexação e um apontador para a posição do registro lógico no arquivo relacionado
 - O índice pode ser ordenado, por exemplo, em ordem alfabética
- O arquivo relacionado contém os registros lógicos

Acesso Baseado em Índice



Operações em Arquivos Usando C++

```
1 // basic file operations
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5
6 int main () {
7     ofstream myfile;
8     myfile.open ("example.txt");
9     myfile << "Writing this to a file.\n";
10    myfile.close();
11    return 0;
12 }
```

```
1 // writing on a text file
2 #include <iostream>
3 #include <fstream>
4 using namespace std;
5
6 int main () {
7     ofstream myfile ("example.txt");
8     if (myfile.is_open())
9     {
10        myfile << "This is a line.\n";
11        myfile << "This is another line.\n";
12        myfile.close();
13    }
14    else cout << "Unable to open file";
15    return 0;
16 }
```

```
1 // reading a text file
2 #include <iostream>
3 #include <fstream>
4 #include <string>
5 using namespace std;
6
7 int main () {
8     string line;
9     ifstream myfile ("example.txt");
10    if (myfile.is_open())
11    {
12        while ( getline (myfile,line) )
13        {
14            cout << line << '\n';
15        }
16        myfile.close();
17    }
18
19    else cout << "Unable to open file";
20
21    return 0;
22 }
```

Operações em Arquivos Usando C++

`open (filename, mode);`

<code>ios::in</code>	Open for input operations.
<code>ios::out</code>	Open for output operations.
<code>ios::binary</code>	Open in binary mode.
<code>ios::ate</code>	Set the initial position at the end of the file. If this flag is not set, the initial position is the beginning of the file.
<code>ios::app</code>	All output operations are performed at the end of the file, appending the content to the current content of the file.
<code>ios::trunc</code>	If the file is opened for output operations and it already existed, its previous content is deleted and replaced by the new one.

```
1 ofstream myfile;  
2 myfile.open ("example.bin", ios::out | ios::app | ios::binary);
```

class	default mode parameter
<code>ofstream</code>	<code>ios::out</code>
<code>ifstream</code>	<code>ios::in</code>
<code>fstream</code>	<code>ios::in ios::out</code>

Atividade de Fixação

- 1) Dê exemplo de quatro operações sobre arquivos.
- 2) O "tipo" do arquivo é geralmente representado por sua extensão. Nesse contexto, qual a importância do tipo do arquivo para o SO?
- 3) Por que ocorre a fragmentação do disco? Exemplifique uma situação em que tal fragmentação ocorre.
- 4) Diferencie os três tipos de acesso a registros de um arquivo: acesso sequencial, acesso direto e acesso baseado em índice.

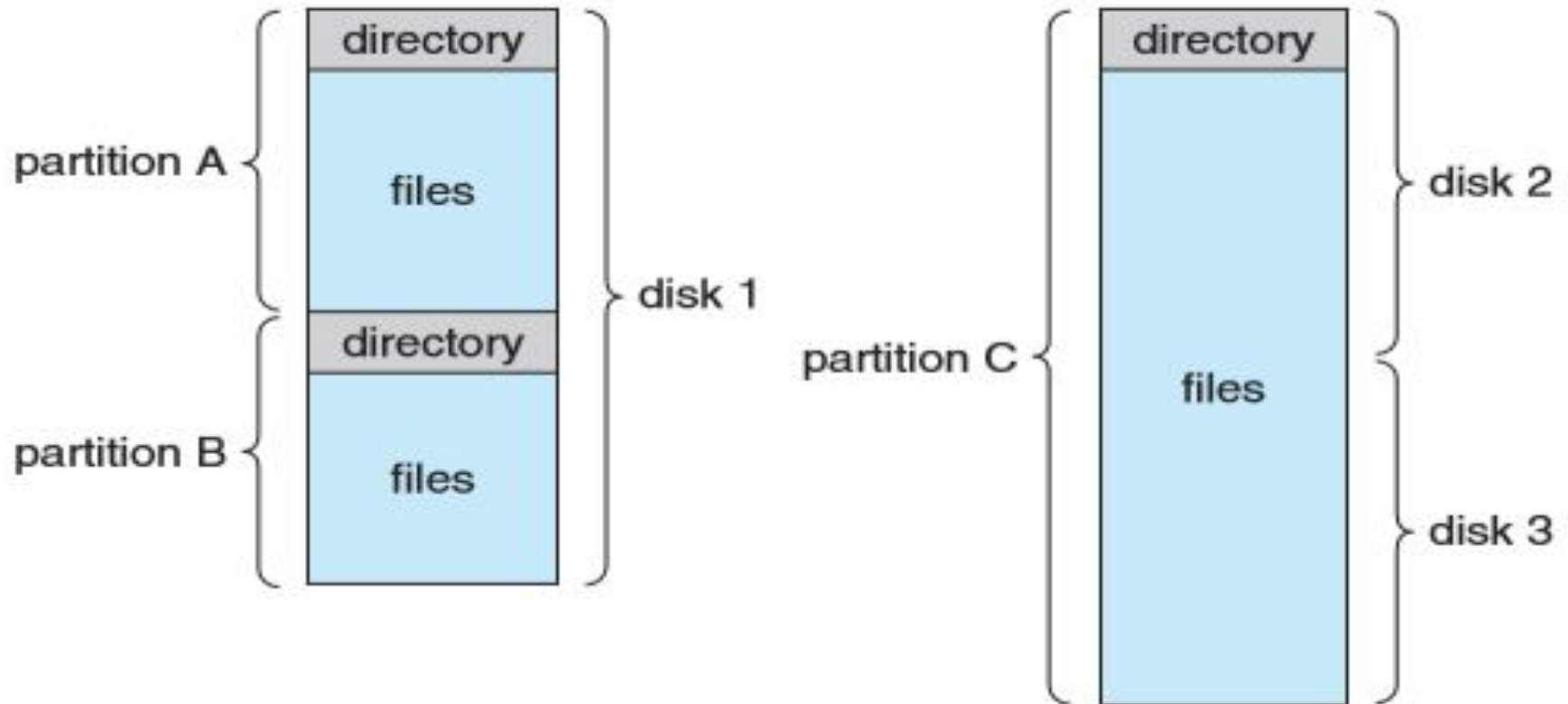
Organização dos Dispositivos

- Um dispositivo de armazenamento pode ser usado em sua totalidade para um sistema de arquivos ou pode ser **particionado**
- O particionamento é útil para
 - Limitar o tamanho do sistema de arquivos
 - Alocar vários tipos de sistemas de arquivos ao mesmo dispositivo
 - Deixar parte do dispositivo disponível para outras finalidades, como espaço para *swapping*

Organização dos Dispositivos

- Dispositivos de armazenamento podem ser reunidos em conjuntos
 - *Redundant Array of Independent Disks* (RAID) ou Conjunto Redundante de Discos Independentes
- Qualquer entidade que possui um sistema de arquivos é chamada um **volume**
 - Dispositivo inteiro; partição; RAID; etc,
- Cada volume contém um diretório de dispositivo (ou diretório) que contém informações sobre o sistema

Organização Típica



Estrutura de Diretórios

- Mesmo dentro de um sistema de arquivos, é útil segregar arquivos em grupos para gerenciar e manipular esses grupos
- Essa organização envolve o uso de diretórios
- Diretórios podem
 - ser criados de acordo com uma diversidade de critérios
 - atender a uma diversidade de propósitos

Operações sobre Diretórios

- Operações típicas que podem ser realizadas em diretórios
 - Busca de um arquivo
 - Criação de um arquivo
 - Exclusão de um arquivo
 - Listagem de arquivos em um diretório
 - Renomeação de um arquivo
 - Varredura do sistema de arquivos

Estrutura de Diretórios no Linux

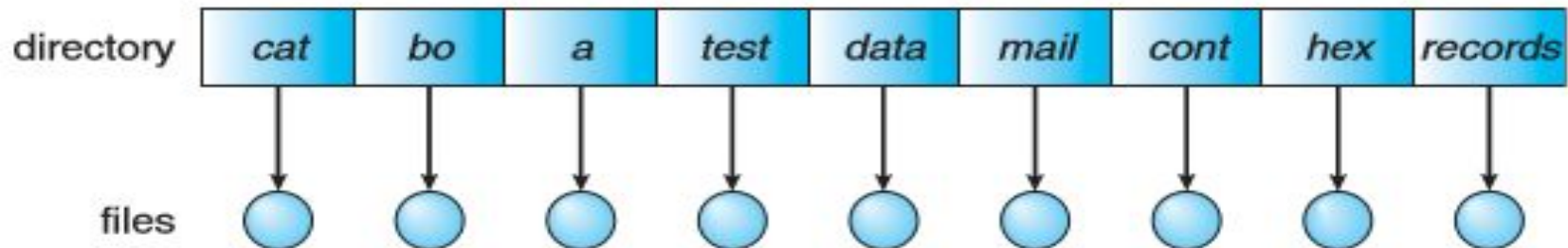
- /:** diretório raiz, contém todos os diretórios indicados abaixo
- /bin:** contém arquivos binários de comandos essenciais do sistema.
- /boot:** contém arquivos de boot (inicialização; boot-loader; Grub); kernel
- /dev:** contém dispositivos de entrada/saída: floppy, hardisk, cdrom, modem
- /etc:** contém arquivos de configuração (scripts) e inicialização.
- /home:** diretório local (home) de usuários.
- /lib:** contém bibliotecas e módulos(drives): compartilhadas com frequência.
- /mnt:** diretório de montagem de dispositivos, sistemas de arquivos e partições.
- /opt:** diretório onde são instalados programas não oficiais da distribuição
- /proc:** diretório virtual (RAM) onde rodam os processos ativos
- /root:** diretório local do superusuário (root)
- /sbin:** arquivos de sistema essenciais (binários do superusuário)
- /tmp:** arquivos temporários gerados por alguns utilitários.
- /usr:** arquivos de usuários nativos da distribuição
- /var:** arquivos de log e outros arquivos variáveis.

Organização dos Diretórios

- Diretório de Um Nível
- Diretório de Dois Níveis
- Diretórios Estruturados em Árvore
- Diretórios em Grafo Acíclico
- Diretórios em Grafo Geral

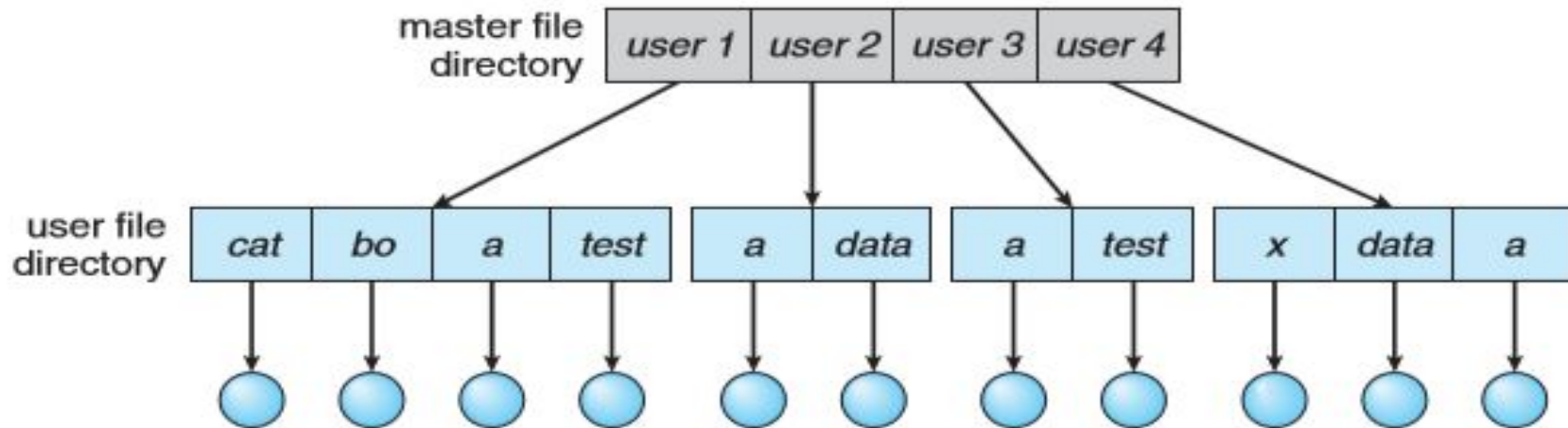
Diretório de Um Nível

- Todos os arquivos ficam no mesmo diretório
 - É fácil de suportar e entender
 - Torna-se complexo quando a quantidade de arquivos aumenta ou quando o sistema possui muitos usuários



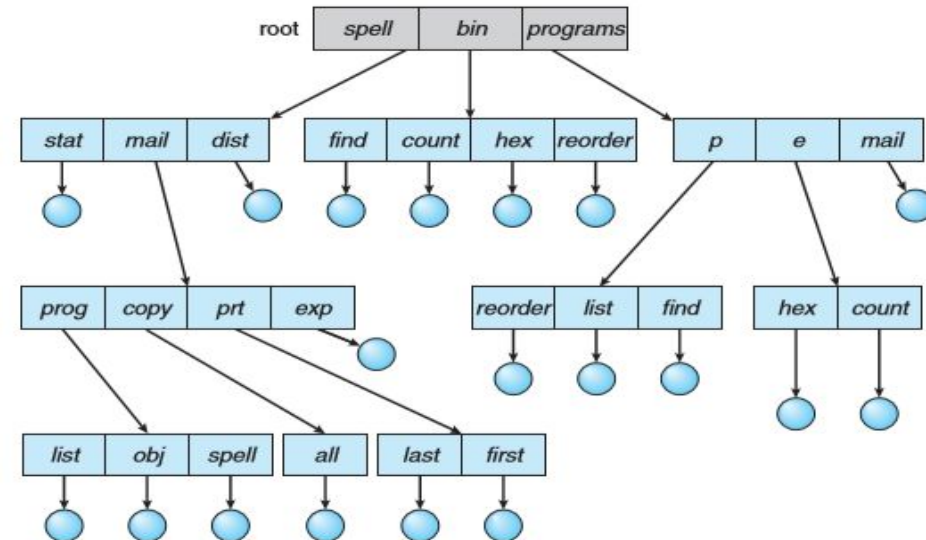
Diretório de Dois Níveis

- Diretório de arquivos do usuário (*user file directory*, UFD)
 - Há uma estrutura de diretórios para cada usuário
 - Trata a complexidade do crescimento do número de usuários, pois cria-se um novo diretório para os diretórios de cada usuário
 - Torna mais difícil a cooperação entre usuários



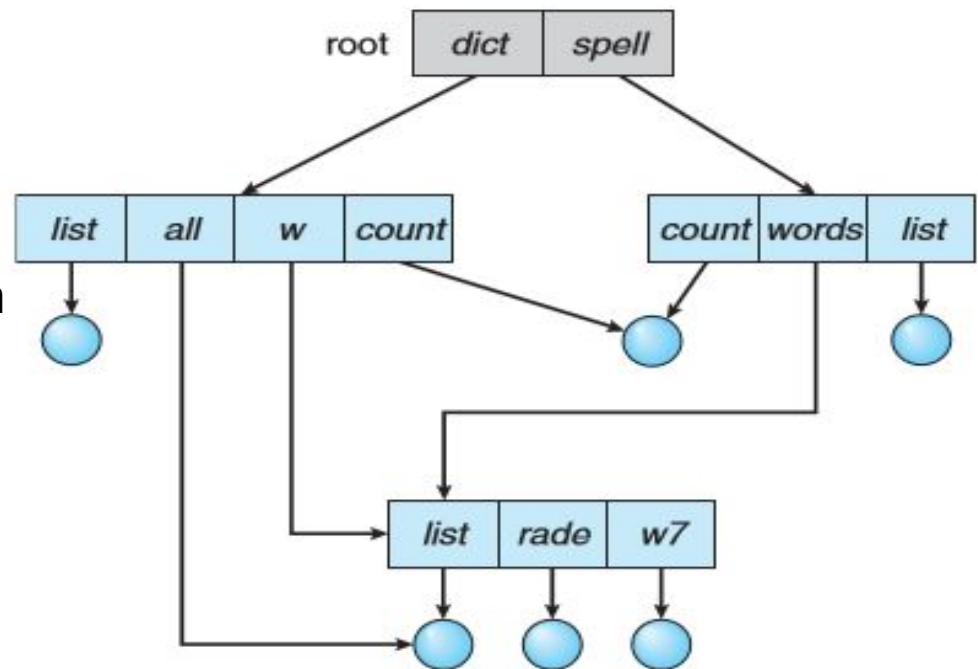
Diretórios em Forma de **Árvore**

- Permite que usuários criem seus próprios subdiretórios, o que gera uma árvore de diretórios com diversos níveis
- Diretório corrente e nome de caminho (PATH)
- **Caminho absoluto**
 - Caminho completo desde o /
 - Ex: root/spell/mail/prt/first
- **Caminho relativo**
 - Caminho a partir do diretório corrente
 - Ex: prt/first se o diretório corrente for root/spell/mail/



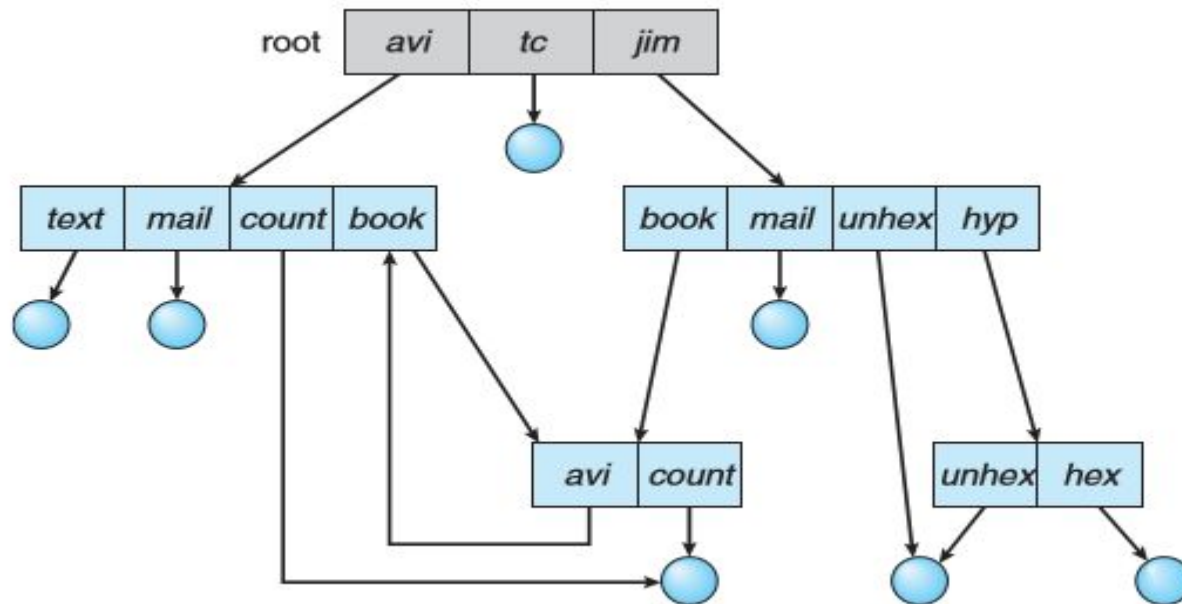
Diretórios em Grafo Acíclico

- Permite o compartilhamento de diretórios/arquivos por dois ou mais diretórios (ou sistemas)
- O diretório/arquivo compartilhado aparecerá nos dois sistemas de arquivos ao mesmo tempo
- Não são permitidos ciclos (é um grafo acíclico)
- É uma generalização da estrutura de diretórios em árvore



Diretórios em Grafo Geral

- Possibilita o compartilhamento de diretórios/arquivos por dois ou mais diretórios e permite que existam ciclos



Montagem do Sistema de Arquivos

- Um sistema de arquivos precisa ser montado para ficar disponível para processos do sistema
- Para montar um sistema de arquivos, o SO recebe o nome do dispositivo e o **ponto de montagem**
- Ponto de montagem é a localização dentro da estrutura de arquivos onde o sistema de arquivo deve ser anexado
 - Normalmente o ponto de montagem é um diretório vazio

Compartilhamento de Arquivos

- Compartilhamento é desejável para usuários que querem colaborar e reduzir o esforço para alcançar um objetivo de computação
- Múltiplos Usuários
 - **Proprietário** (*owner*), usuários que têm acesso total a todas as operações sobre o arquivo/diretório
 - **Grupo** de arquivos ou diretórios, subconjunto de usuários que podem compartilhar o acesso ao arquivo/diretório
- Sistema de Arquivos Remotos
 - *Distributed file system (DFS); Network file system (NFS)*

Proteção

- Tipos de acesso, ou tipos de operações que um dado usuário (ou grupo de usuários) tem acesso
 - Leitura
 - Gravação
 - Execução
 - Acréscimo
 - Exclusão
 - Listagem
- Conceder um acesso é conceder uma permissão


Controle de Acesso

- O esquema mais geral é associar a cada arquivo/diretório uma lista de controle de acesso (ACL – *Access-control list*) que especifica os nomes e os tipos de acesso permitidos a cada usuário
 - O controle de acesso é dependente da identidade
- Três classificações de usuários associados a cada arquivo/diretório
 - Proprietário
 - Grupo
 - Universo

Permissões no Linux

```
$ ls -l dante
```

```
-rw-r--r-- 1 user1 staff 1319 Mar 15 11:23 dante
```



rw-r--r--

┌───┬───┬───┐
└───┴───┴───┘

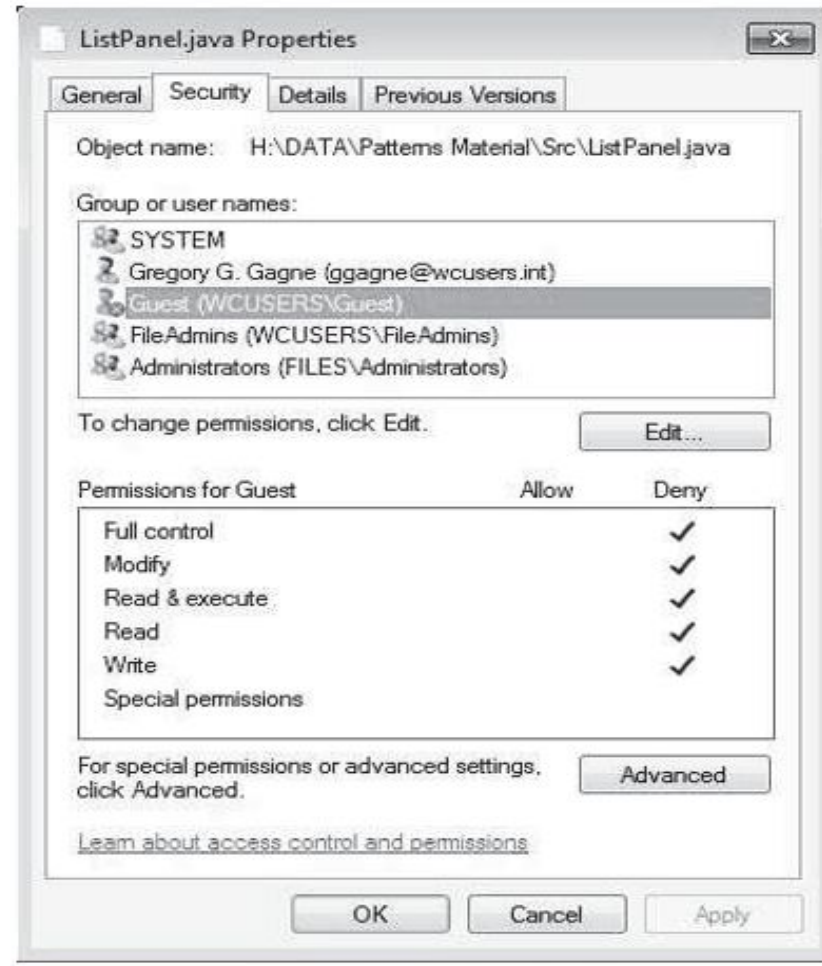
↓ ↓ ↓

Owner Group Other

r = Readable
w = Writable
x = Executable
- = Denied

<https://www.vivaolinux.com.br/dica/Os-usuarios-e-permissoes-no-sistema-GNULinux>

Permissões no Windows



Atividade de Fixação

- 1) Discuta o conceito e a importância dos seguintes itens
 - Particionamento do dispositivo
 - Montagem do sistema de arquivos
 - Proteção por meio de permissões

- 2) Compare as estruturas abaixo
 - Diretórios estruturados em Árvore
 - Diretórios em Grafo Acíclico
 - Diretórios em Grafo Geral

Referências

TANENBAUM, Andrew S. Sistemas operacionais modernos. 3. ed. São Paulo: Pearson Prentice Hall, 2009. xvi, 653 p. ISBN 9788576052371

SILBERSCHATZ, Abraham; GALVIN, Peter B.; GAGNE, Greg. Fundamentos de sistemas operacionais: princípios básicos. Rio de Janeiro, RJ: LTC, 2013. xvi, 432 p. ISBN 9788521622055

PEREIRA, Thiago Emmanuel; BRASILEIRO, Francisco; SAMPAIO, Livia. A study on the errors and uncertainties of file system trace capture methods. In: Proceedings of the 9th ACM International on Systems and Storage Conference. ACM, 2016. p. 14.

Sistemas Operacionais

Prof. Dr. Lesandro Ponciano

<https://orcid.org/0000-0002-5724-0094>