

FACULDADE DE TECNOLOGIA SENAC RIO

SISTEMAS OPERACIONAIS

LAURO L. A. WHATELY

SISTEMAS DE ARQUIVO

SISTEMA DE ARQUIVOS

- Conceito de Arquivo
- Métodos de Acesso
- Estrutura de Diretórios
- Compartilhamento de Arquivos
- Proteção

OBJETIVOS

- Explicar a função de sistemas de arquivos.
- Descrever a interface para sistemas de arquivos.
- Discutir métodos de acesso, compartilhamento de arquivos e estruturas de diretórios.
- Explorar a proteção de sistemas de arquivos.

ARMAZENAMENTO A LONGO PRAZO

- Em muitas aplicações o arquivo é o elemento central.
- 3 requisitos essenciais:
 - Deve ser possível armazenar uma quantidade muito grande de informações;
 - A informação deve sobreviver ao término do processo que a usa;
 - Múltiplos processos tem que ser capazes de acessar as informações concorrentemente.

ARQUIVO

- Do ponto de vista do usuário o sistema de arquivos é uma das partes principais de um SO.
- Usuários desejam acessar arquivos, ler , escrever, salvar, etc.
- Para ajudar nestes objetivos, os sistemas operacionais oferecem os sistemas gerenciadores de arquivos.
- O Sistema Operacional gerencia dispositivos de armazenamento:
 - Esconde as dificuldades impostas pelo hardware;
 - Define uma abstração de armazenamento: Arquivo.

CONCEITO DE ARQUIVO

- Espaço de endereçamento lógico contíguo.
- Tipos:
 - Dados
 - Numérico,
 - Caractere,
 - Binário.
 - Programa.
 - O conteúdo é definido pelo criador do arquivo:
 - Ex. arquivo fonte, arquivo executável, arquivo áudio.

ESTRUTURA DE ARQUIVOS

- Nenhuma estrutura inerente - sequência de bytes
- Estrutura de registro simples
 - Linhas
 - Tamanho fixo
 - Tamanho variável
- Estruturas Complexas
 - Documentos formatados
 - Arquivo de carga relocável
- Pode simular os dois últimos com o primeiro método inserindo caracteres especiais de controle.
- Quem decide:
 - Sistema Operacional
 - Programa

EXEMPLO - MACOS



ATRIBUTOS DE ARQUIVOS

- **Nome** – única informação mantida em uma forma legível para o usuário.
- **Identificador** – identificador único (número) do arquivo pelo sistema de arquivos
- **Tipo** – necessária para sistemas que suportam diferentes tipos de arquivos.
- **Localização** – ponteiro para a posição do arquivo no dispositivo.
- **Tamanho** – tamanho atual do arquivo.
- **Proteção** – controla quem pode ler, escrever e executar.
- **Hora, data, e identificação do usuário** – dados para proteção, segurança e monitoração de uso.
- Informações sobre os arquivos são mantidas nas estruturas de diretórios, as quais são armazenadas no disco.

OPERAÇÕES EM ARQUIVOS

- Arquivo é um **tipo de dados abstrato**
- **Criar** (*create*)
- **Escrever** (*write*)
- **Ler** (*read*)
- **Reposicionamento de um arquivo** (*seek*)
- **Excluir** (*delete*)
- **Truncamento** (*truncate*)
- **Abrir** $[F_i]$ (*open* $[F_i]$) – procura na estrutura de diretório do disco pela entrada F_i , e move o conteúdo da entrada para a memória.
- **Fechar** $[F_i]$ (*close* $[F_i]$) – move o conteúdo da entrada F_i na memória para a estrutura de diretório no disco.

ARQUIVOS ABERTOS

- Alguns dados são necessários para gerenciar arquivos abertos:
 - **Ponteiro de Arquivo:** ponteiro para a última localização de leitura/escrita, por processo que tem um arquivo aberto
 - **Contador de arquivos abertos:** contador do número de vezes que um arquivo é aberto - para permitir a remoção dos dados da tabela de arquivos abertos quando o último processo fechar o arquivo
 - **Localização no disco do arquivo:** cache dos dados acessados
 - **Direitos de acesso:** informações de modo de acesso por processo

TRAVA (LOCK) DE ARQUIVOS ABERTOS

- Fornecido por alguns sistemas operacionais e sistemas de arquivos
- Acesso exclusivo a um arquivo
- Semelhante ao mecanismo para exclusão mútua:
 - trava compartilhada (shared lock) – para leituras concorrentes.
 - Trava exclusiva (exclusive lock) – acesso exclusivo para escrita.

TRAVA (LOCK)- EXEMPLO JAVA (1)

```
import java.io.*;
import java.nio.channels.*;
public class LockingExample {
    public static final boolean EXCLUSIVE = false;
    public static final boolean SHARED = true;

    public static void main(String arsg[]) throws IOException {
        FileLock sharedLock = null;
        FileLock exclusiveLock = null;
        try {
            RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");
            // get the channel for the file
            FileChannel ch = raf.getChannel();
            // this locks the first half of the file - exclusive
            exclusiveLock = ch.lock(0, raf.length()/2, EXCLUSIVE);
```

TRAVA (LOCK)- EXEMPLO JAVA (2)

```
/* Now modify the data ... */  
// release the lock  
ExclusiveLock.release();  
// this locks the second half of the file - shared  
sharedLock = ch.lock(raf.length()/2+1, raf.length(), SHARED);  
/** Now read the data . . . */  
// release the lock  
exclusiveLock.release();  
} catch (java.io.IOException ioe) {  
    System.err.println(ioe);  
}finally {  
    if (exclusiveLock != null)  
        exclusiveLock.release();  
    if (sharedLock != null)  
        sharedLock.release();  
}  
}  
}
```

TIPOS DE ARQUIVOS - NOME E EXTENSÃO

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

MÉTODOS DE ACESSO

- **Acesso Sequencial**

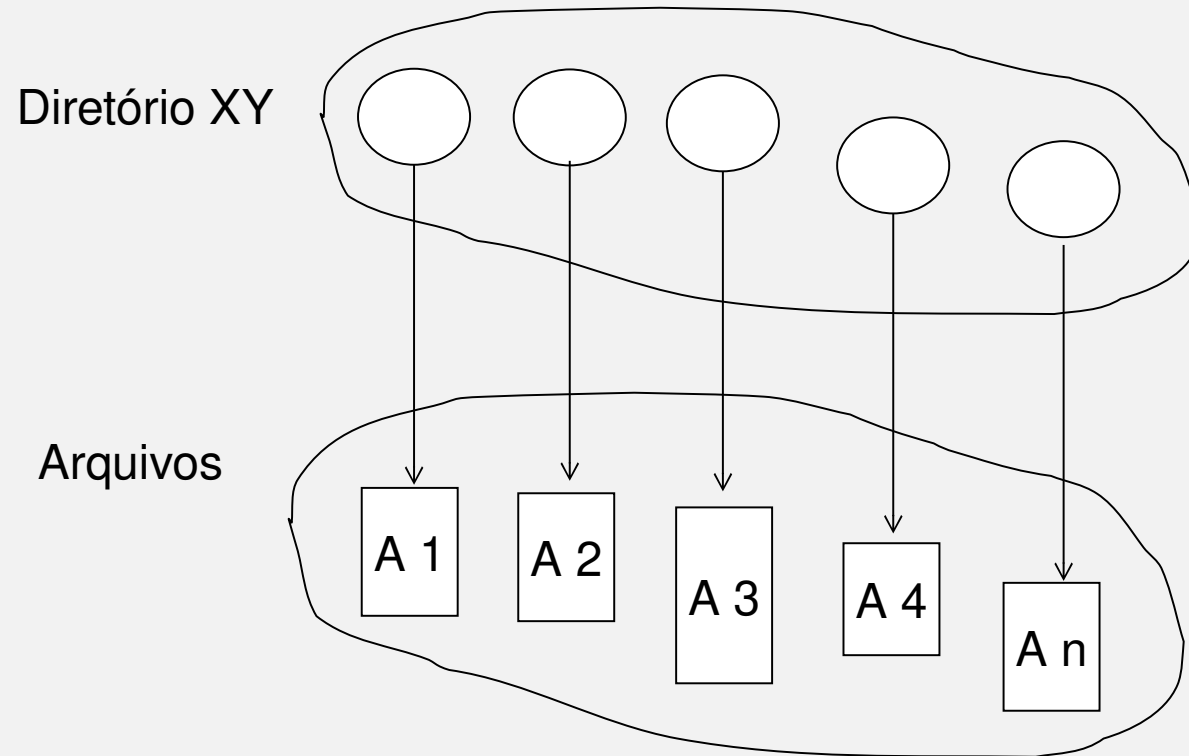
- Os primeiros tipos de sistema ofereciam somente este tipo de acesso → fitas
- Lê uma posição do arquivo e avança para a próxima → não é possível ler fora de ordem

- **Acesso Direto**

- Baseado no modelo de arquivo em disco
- Permite que blocos sejam lidos e gravados em qualquer ordem
- Essencial para vários tipos de aplicações
- É o método mais usado

ESTRUTURA DE DIRETÓRIO

Uma coleção de nodos contendo informações sobre todos arquivos.



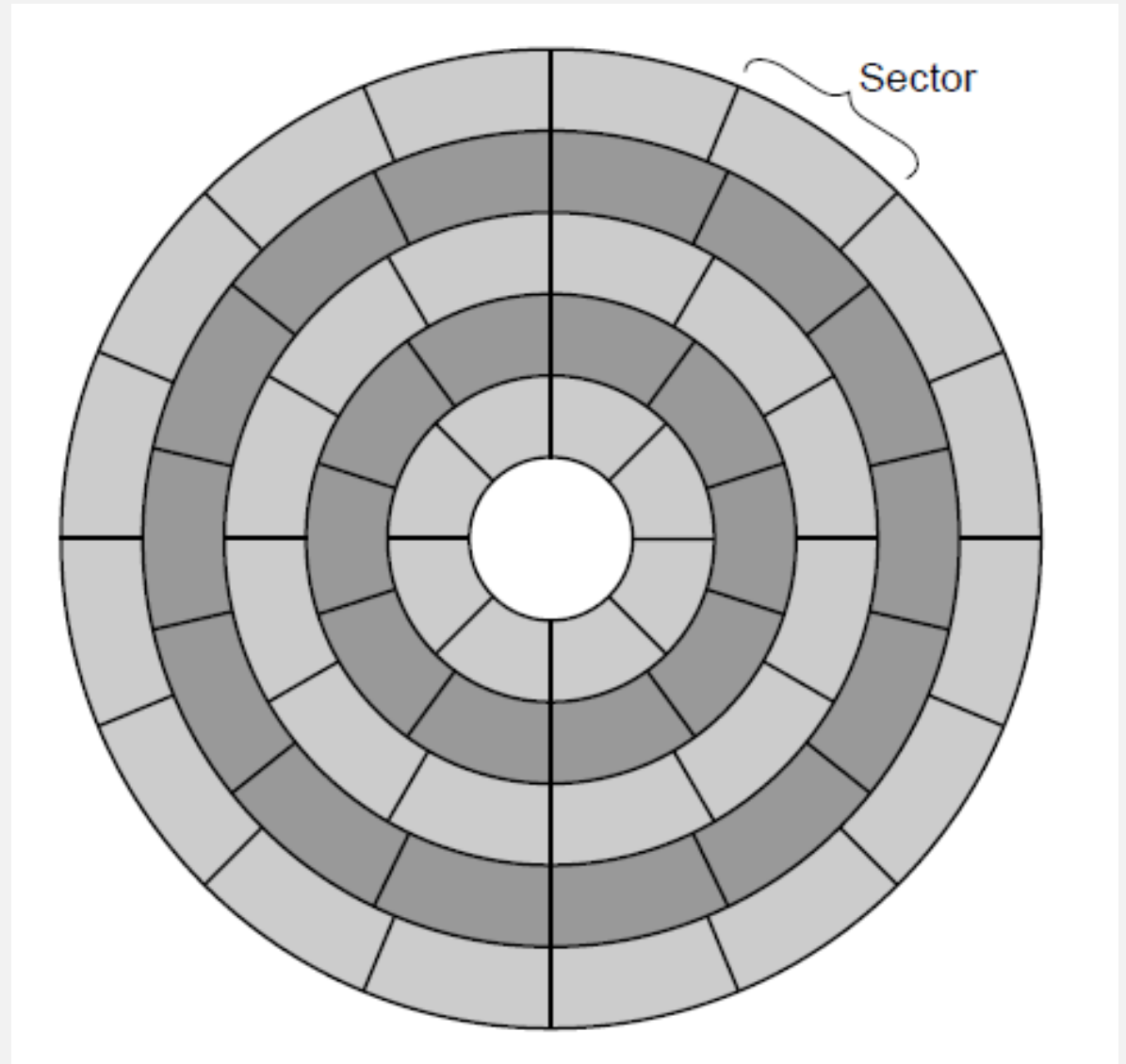
Tanto a estrutura de diretórios quanto de arquivos residem no disco

ESTRUTURA DE DISCO

- Disco pode ser dividido em **partições**
- Discos ou partições podem ser protegidas por **RAID** contra falhas
- Disco ou partição pode ser usada **raw** – sem um sistemas de arquivo, ou **formatada** com um sistema de arquivo
- Partições são conhecidas também como minidiscos ou *slices*
- Entidade que contém um sistema de arquivos é conhecido como **volume**
- Cada volume contendo um sistema de arquivos também mantém as informações deste em **diretório do dispositivo** ou **índice do volume**
- Assim como **sistemas de arquivos de propósito geral** existem muitos **sistemas de arquivos de propósito específico**, frequentemente todos dentro do mesmo sistema operacional ou computador

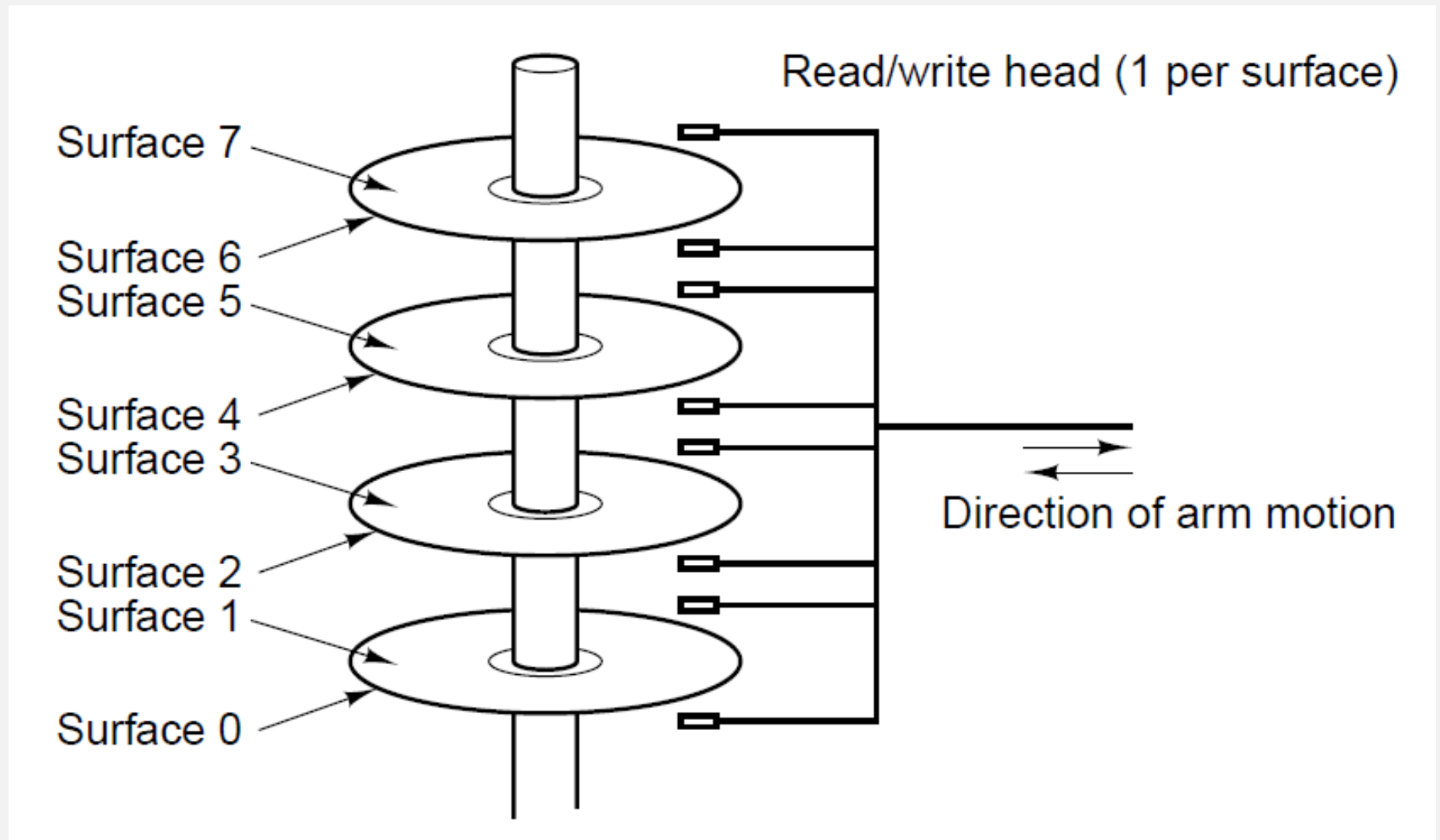
DISCO MAGNÉTICO

- Organizado em setores e trilhas.
- Trilhas são círculos concêntricos na superfície do disco. Cada trilha é dividida em setores (normalmente 512bytes).
- Cada trilha (desde mais externa - trilha 0, para a mais interna) possui o mesmo número de bits.
- A formatação na fábrica cria os setores e trilhas.
- A formatação lógica é a preparação do disco para os sistemas de arquivos.



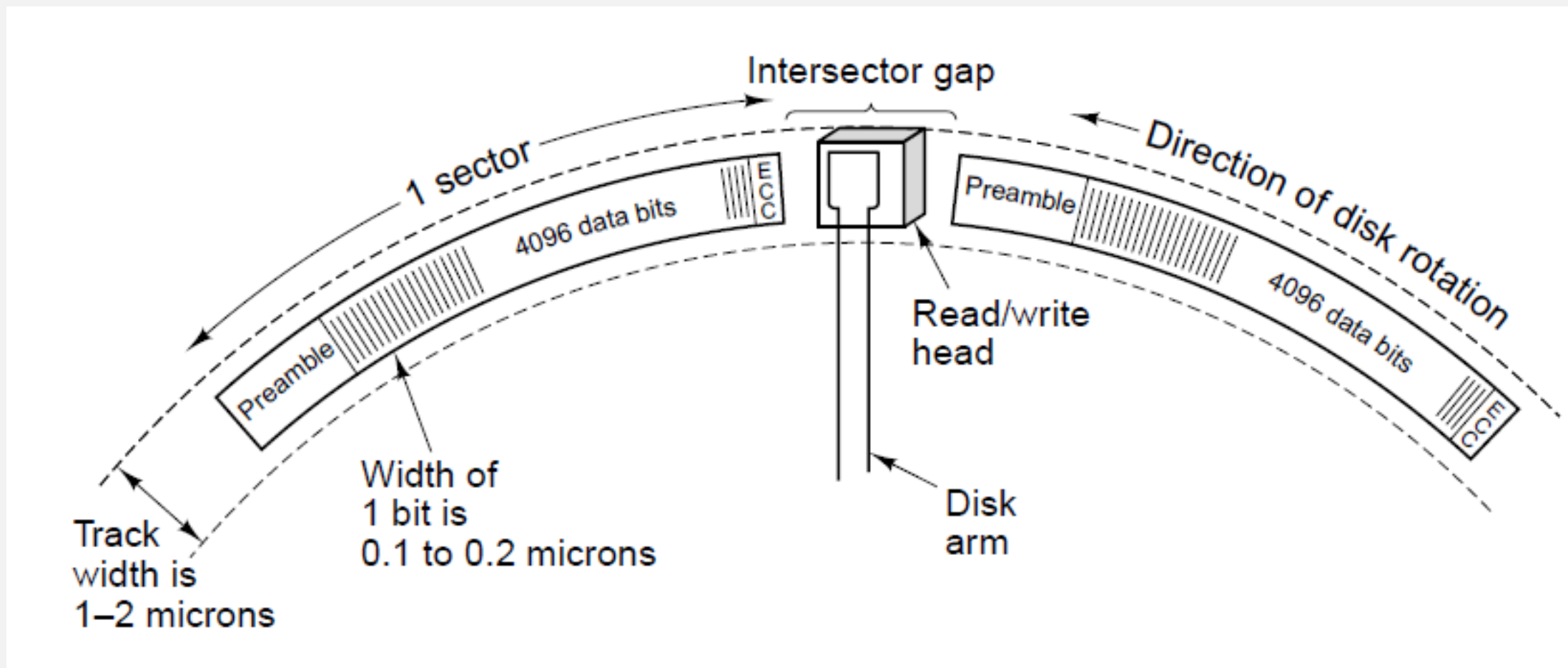
DISCO MAGNÉTICO

Um disco com 4 pratos

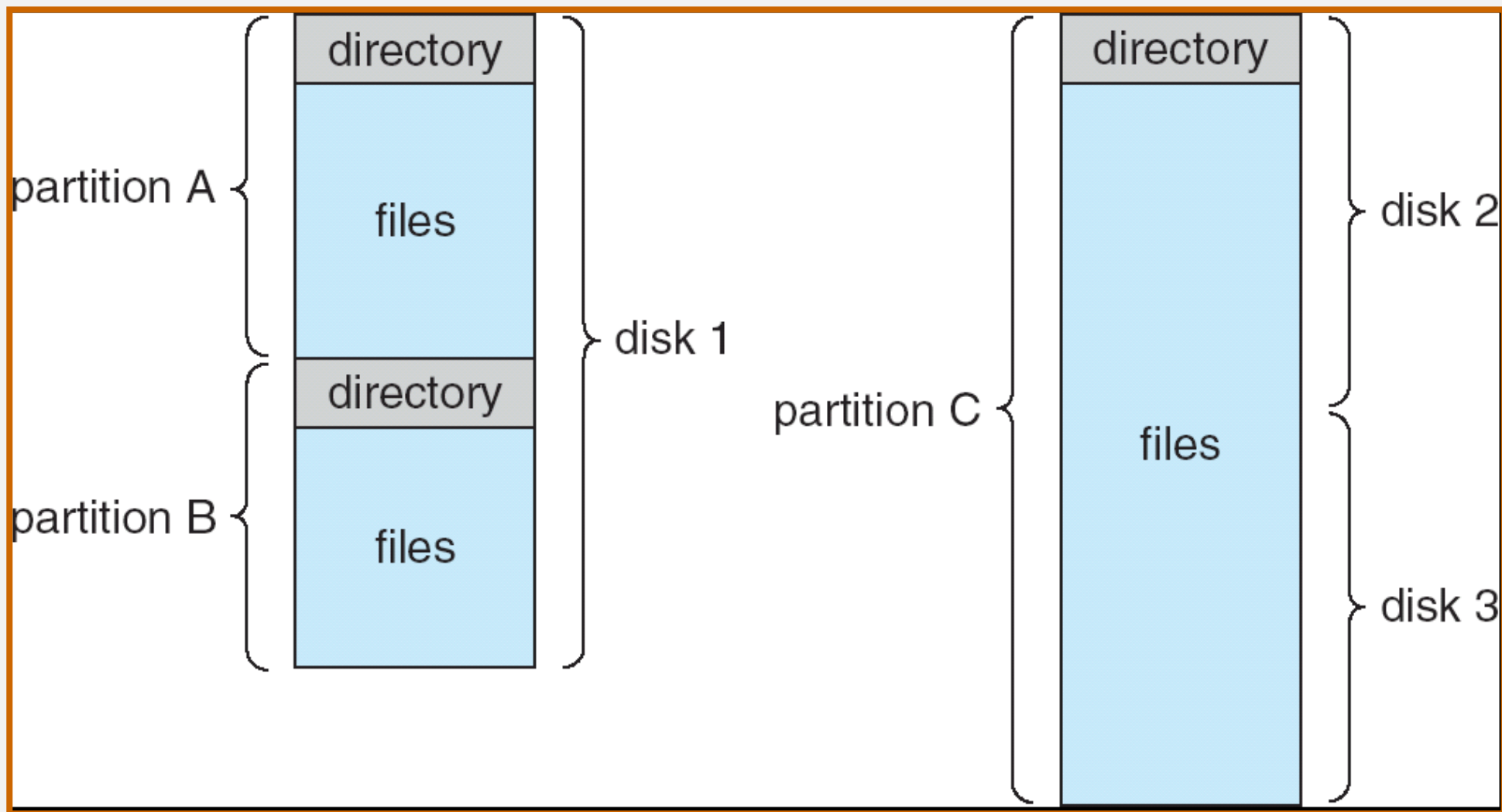


DISCO MAGNÉTICO

Uma porção de uma trilha com dois setores



ESTRUTURA DE DISCO



OPERAÇÕES EM DIRETÓRIOS

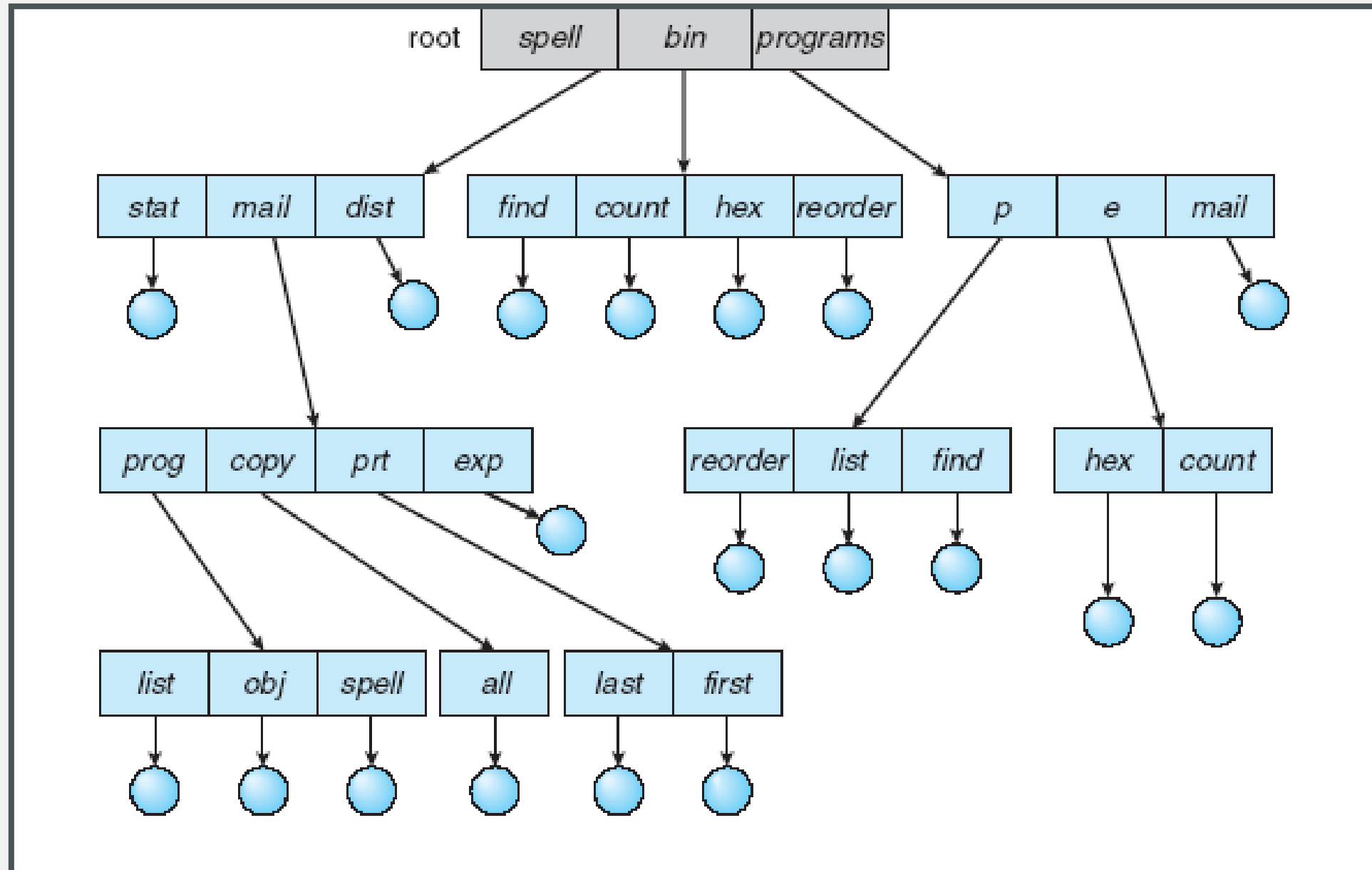
- Procurar por um arquivo
- Criar um Arquivo
- Excluir um Arquivo
- Listar um diretório
- Alterar o nome de um arquivo
- Percorrer o Sistema de Arquivos

ESTRUTURA DE DIRETÓRIO

Diretório em árvores:

- Cada usuário pode ter tantos diretórios quanto necessário, podendo agrupar seus arquivos da melhor maneira.
- Cada usuário tem um diretório corrente:
 - ✓ Caso um arquivo não esteja no diretório corrente, deve se especificar um caminho.
 - ✓ Nomes de caminhos podem ser absolutos (começa no diretório raiz) ou relativos (a partir do diretório corrente).
 - ✓ Um usuário pode acessar arquivos de outros usuários

DIRETÓRIOS EM ÁRVORE



DIRETÓRIOS EM ÁRVORE

- Procura eficiente
- Capacidade de Agrupamento
- Diretório Corrente (diretório de trabalho)
 - `cd /spell/mail/prog`
 - `type list`

DIRETÓRIOS EM ÁRVORE

- Procura eficiente
- Diretório Corrente (diretório de trabalho)
 - `cd /spell/mail/prog`
 - `type list`
- Caminho **absoluto** ou **relativo**
- Criação de arquivos novos é feita no diretório corrente.
- Apagar um arquivo
 - `rm <file-name>`
- Criação de novos subdiretórios é feita no diretório corrente.
 - `mkdir <dir-name>`

Exemplo: se o diretório corrente é `/spell/mail`

`mkdir count`

PROTEÇÃO

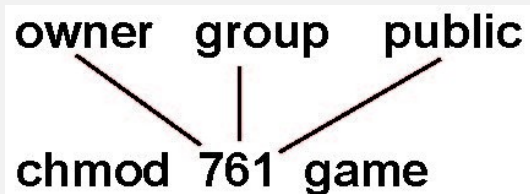
- Dono/Criador do arquivo deve estar apto a controlar:
 - O que pode ser feito
 - Por quem
- Tipos de Acesso
 - **Leitura**
 - **Escrita**
 - **Execução**
 - **Adição** (*Append*)
 - **Exclusão**
 - **Listagem**

LISTAS DE ACESSO E GRUPOS

- Modos de acesso: leitura (**read**), escrita(**w**rite), execução (**ex**ecute)
- Três classes de usuários

				RWX
a) acesso de dono (<i>owner</i>)	7	→	1 1 1	RWX
b) acesso de grupo (<i>group</i>)	6	→	1 1 0	RWX
c) acesso público (<i>public</i>)	1	→	0 0 1	

- Peça para o administrador criar um grupo (nome único), digamos “dev”, e adicionar alguns usuários ao grupo.
- Para um arquivo ou subdiretório particular (digamos *game*), defina um acesso apropriado.



- Associe um grupo a um arquivo:

chgrp dev ./game

EXEMPLO - UNIX

-rw-rw-r--	1 pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5 pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2 pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2 pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1 pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1 pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4 pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3 pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3 pbg	staff	512	Jul 8 09:35	test/

DESEMPENHO DO SISTEMA DE ARQUIVOS

- O acesso ao disco é mais lento que o acesso a memória principal.
- A técnica mais usada para reduzir o acesso ao disco é usar uma cache de blocos:
 - A cache de blocos é uma coleção de blocos que pertencem ao disco mas que são mantidos na memória principal para aumentar o desempenho.
 - Alguns sistemas otimizam sua cache de disco utilizando diferentes algoritmos de substituição de blocos, dependendo do tipo de acesso ao arquivo
 - Leitura antecipada de blocos

ARQUIVOS EM JAVA

- Em programas Java, a entrada e saída (E/S) é controlada por objetos stream:
 - *stream de entrada*: dados entram em um programa.
 - *stream de saída*: dados saem do um programa.

- Os tipos mais comuns de streams são **arquivos de dados externos**.

- 4 Grupos principais de Classes de E/S:

	Entrada	Saída
Binário	InputStream	OutputStream
Texto	Reader	Writer

- 4 Classes de E/S mais usadas:

	Entrada	Saída
Binária	ObjectInputStream	ObjectOutputStream
Texto	BufferedReader	PrintWriter

- Arquivo binário é aquele que é lido e escrito em caracteres de 8 bits.
Ex.: imagens, vídeo, áudio, programas executáveis.
- Arquivo texto é lido e escrito em caracteres Unicode de 16 bits.
Ex.: arquivos criados por editors de texto, arquivos html.