

# Sistemas Operacionais

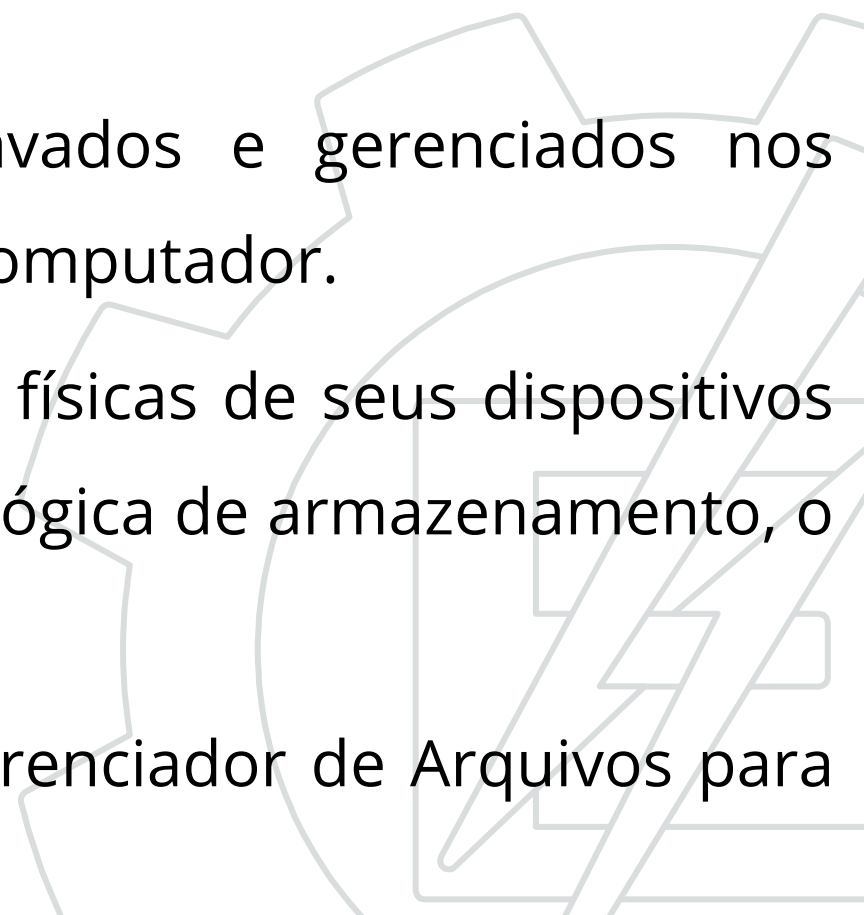
## Sistema de Arquivos

Prof. Otávio Gomes

[otavio.gomes@unifei.edu.br](mailto:otavio.gomes@unifei.edu.br)

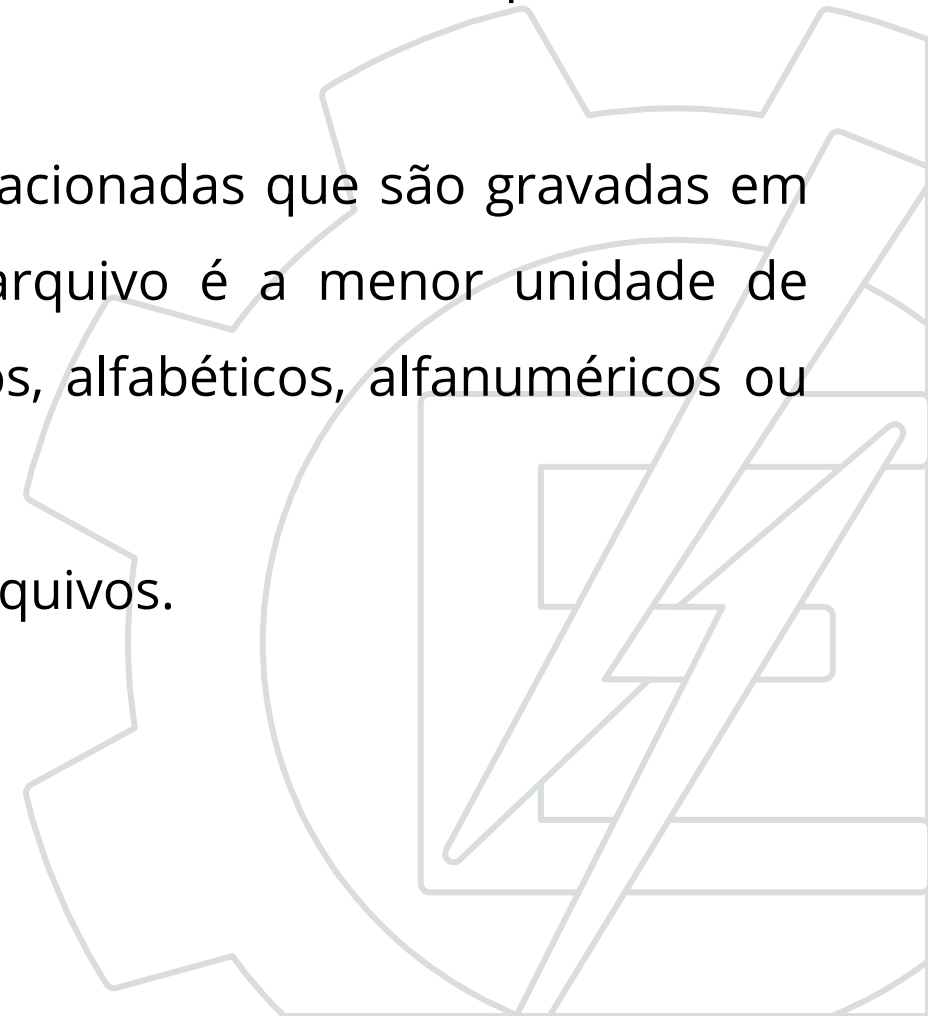


# Sistema de Arquivos

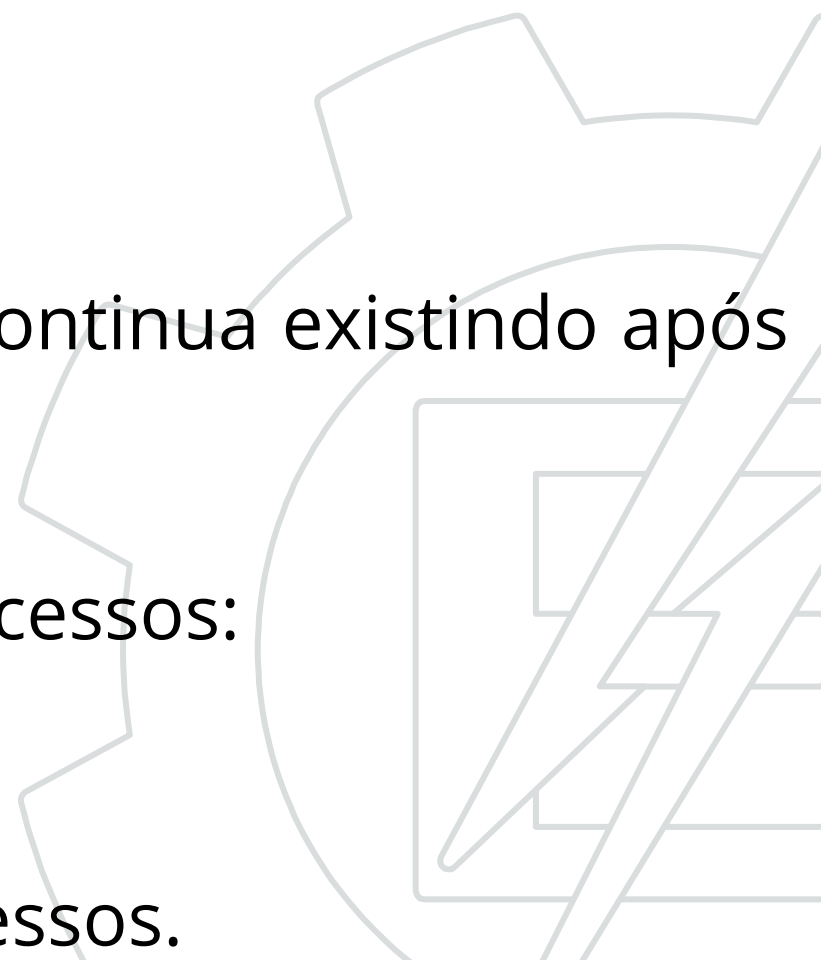
- É responsável pelo mapeamento do espaço de endereçamento para o armazenamento secundário (disco).
  - Define a maneira como os dados serão gravados e gerenciados nos dispositivos de armazenamento acessados pelo computador.
  - O sistema operacional abstrai, das propriedades físicas de seus dispositivos de armazenamento, a definição de uma unidade lógica de armazenamento, o arquivo.
  - A tabela de arquivos é utilizada pelo Sistema Gerenciador de Arquivos para guardar a organização física dos arquivos.
- 

# Sistema de Arquivos

- Os arquivos são mapeados pelo sistema operacional para dispositivos físicos. Esses dispositivos físicos são usualmente não voláteis; portanto, os conteúdos são persistentes entre reinicializações do sistema.
- Um arquivo é uma coleção nomeada de informações relacionadas que são gravadas em memória secundária. Da perspectiva do usuário, um arquivo é a menor unidade de armazenamento lógico secundário. Podem ser numéricos, alfabéticos, alfanuméricos ou binários.
- Diretórios são utilizados pelo usuário para organizar os arquivos.

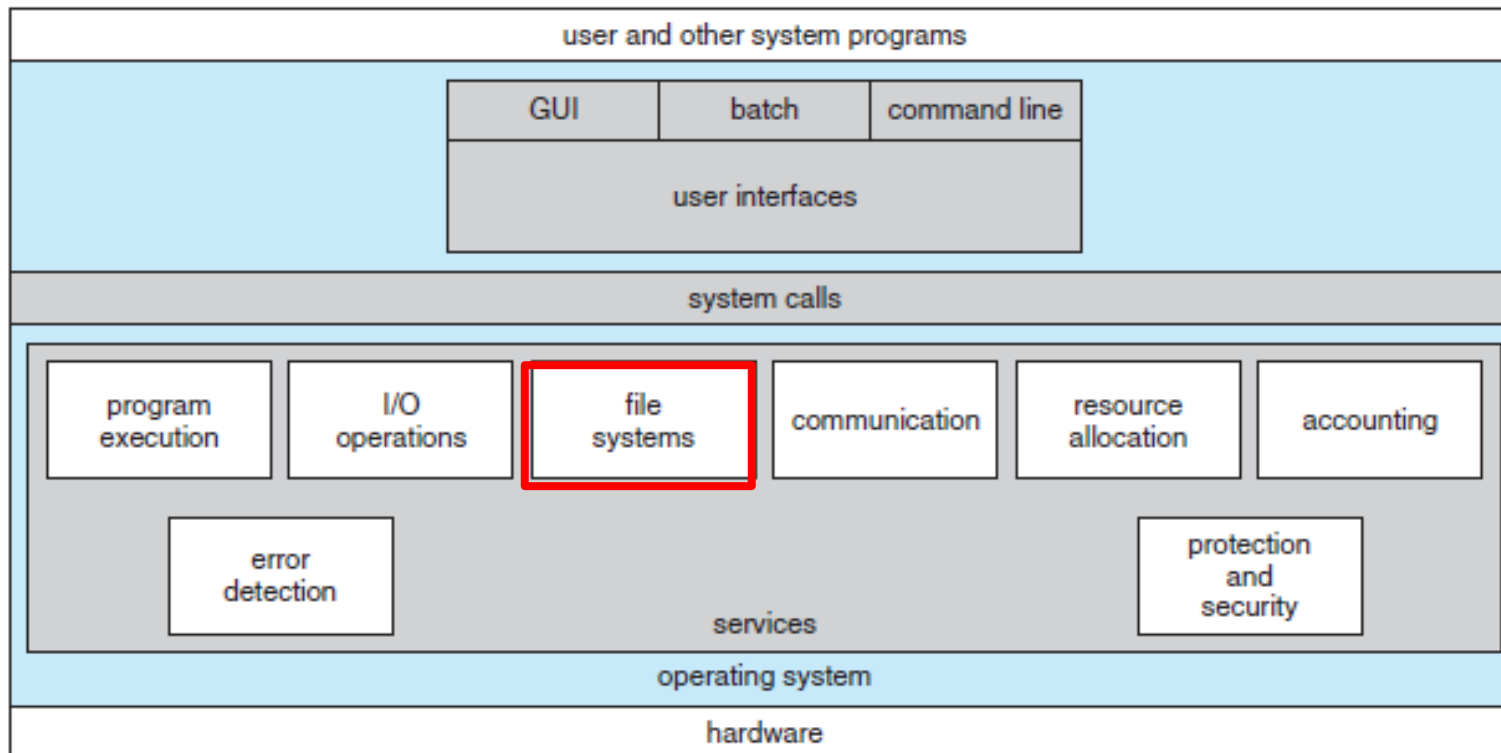


## Requisitos de armazenamento persistente

- Armazenar e recuperar - *big data*:
    - 3 Vs - Volume, Variedade e Velocidade.
  - A informação gerada por um processo continua existindo após a finalização do processo;
  - Dar acesso à informação a múltiplos processos:
    - Concorrência;
    - Informações independentes dos processos.
- 

## Sistema de Arquivos

- Parte do SO responsável pelo tratamento dos arquivos;
- Uma das partes mais visíveis ao usuário.



## **Ponto de vista do usuário** (alto nível)

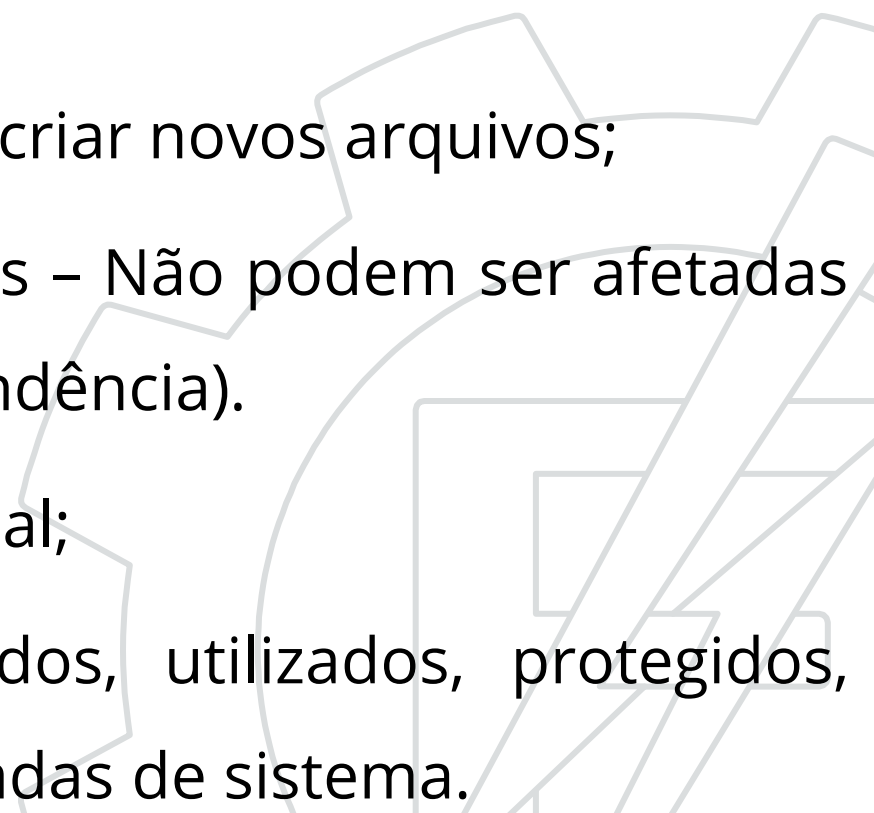
- Interface – como os arquivos aparecem / são apresentados;
- Como são nomeados e protegidos;
- Operações que podem ser realizadas.

## **Ponto de vista do Sistema Operacional** (baixo nível)

- Como os arquivos são armazenados fisicamente;
- Como os arquivos são referenciados (*links*).

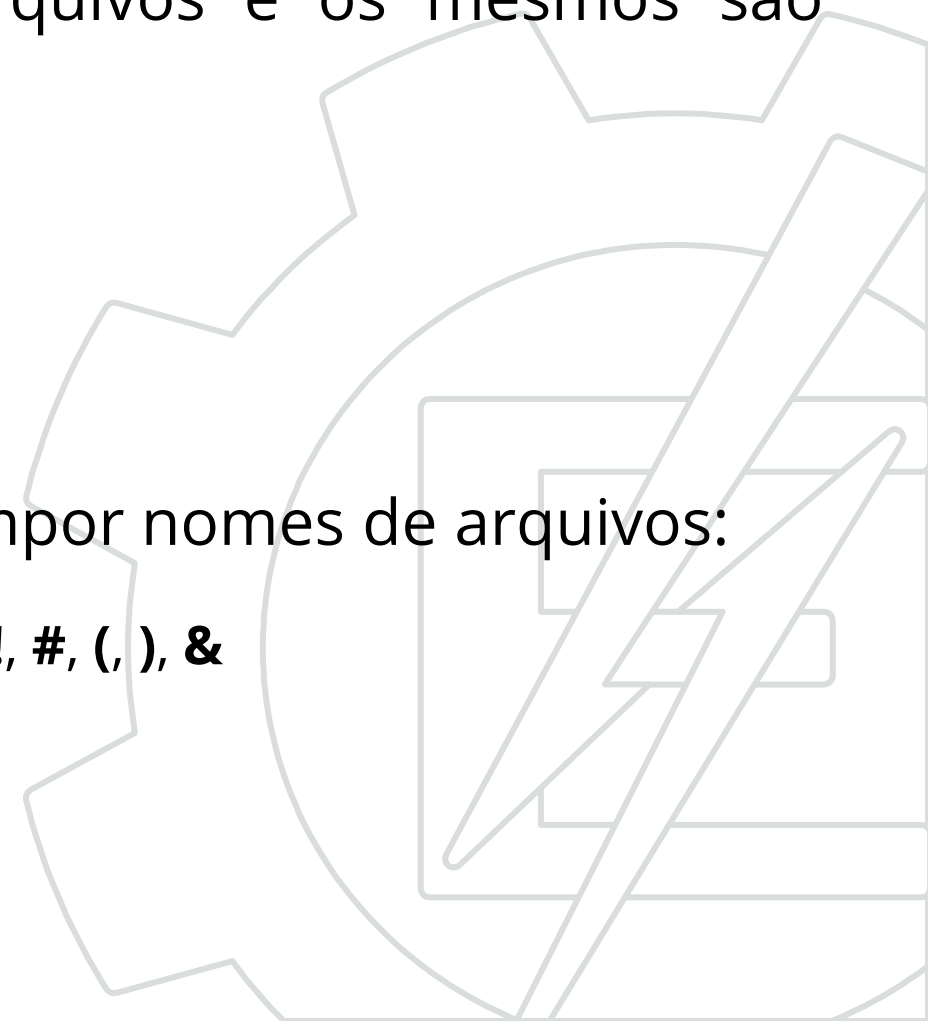


## **Arquivos - Armazenamento persistente**

- Criados pelos processos;
  - Processos podem ler ou escrever em arquivos ou criar novos arquivos;
  - Informações armazenadas devem ser persistentes – Não podem ser afetadas pela criação ou finalização de um processo (dependência).
  - Arquivos são gerenciados pelo Sistema Operacional;
  - Arquivos são estruturados, nomeados, acessados, utilizados, protegidos, implementados e gerenciados por meio de chamadas de sistema.
- 

## Nomes de arquivos

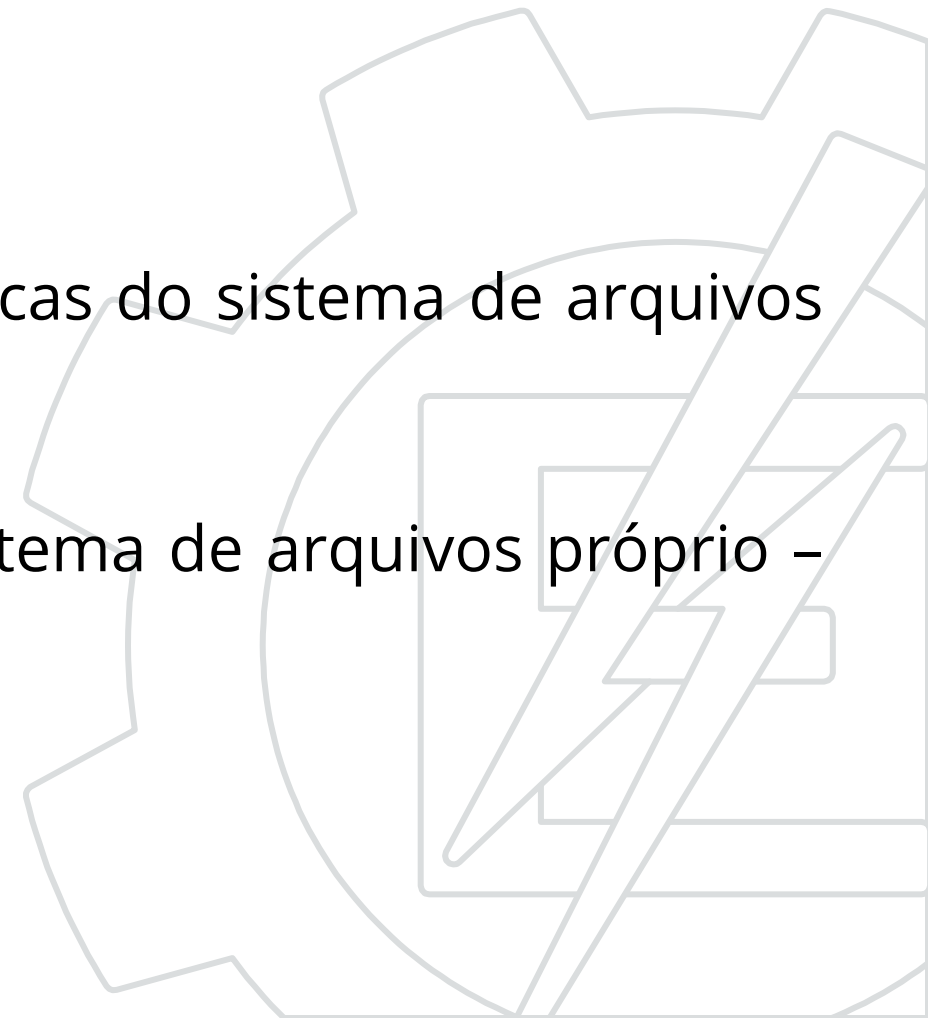
- Quando criados são fornecidos nomes aos arquivos e os mesmos são referenciados por meio de seus nomes;
- Tamanho: até 255 caracteres;
- Restrição: MS-DOS aceita de 1-8 caracteres;
- Letras, números, caracteres especiais podem compor nomes de arquivos:
  - Caracteres permitidos: **A-Z, a-z, 0-9, \$, %, ', @, {, }, ~, `, !, #, (, ), &**
  - Caracteres não-permitidos: **?, \*, /, \, ", |, <, >, :**





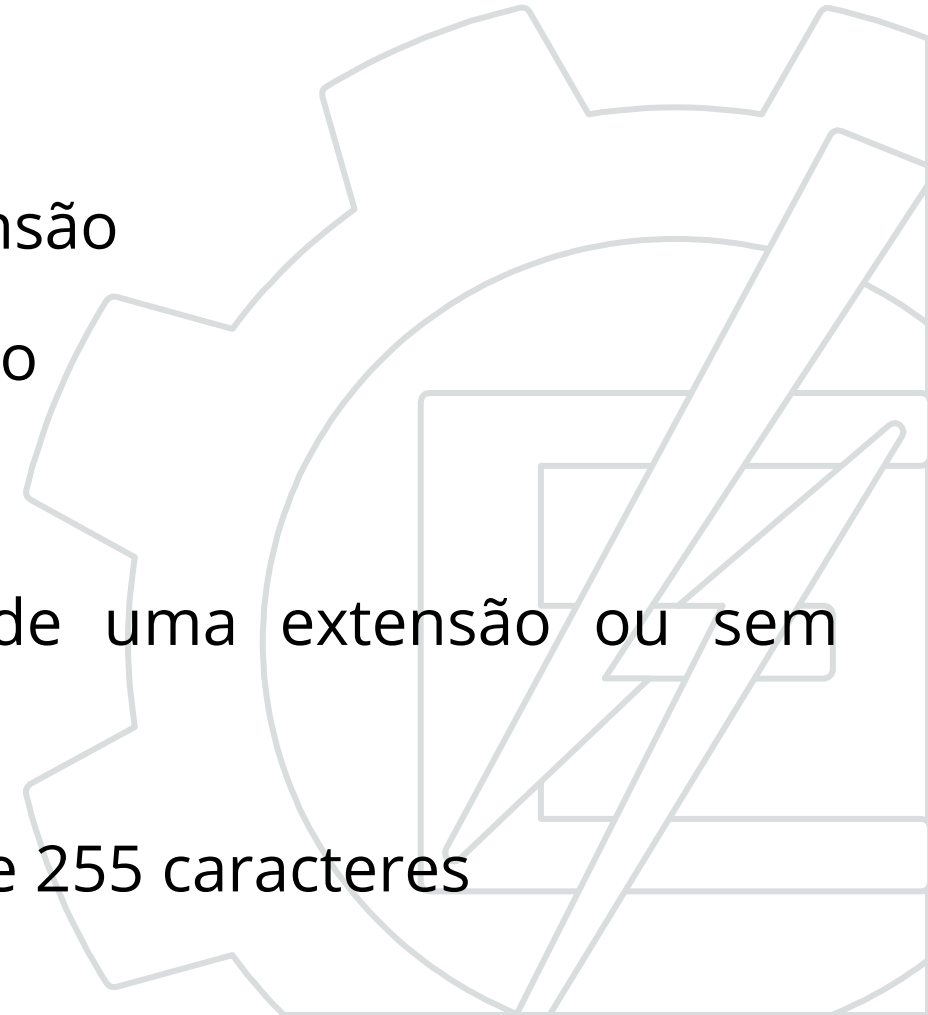
## Nomes de arquivos

- Alguns são sensíveis – *Case Sensitive*
  - UNIX é sensível, MS-DOS não é sensível:
- Win95 / WinNT / Win2000 herdaram características do sistema de arquivos do MS-DOS.
- Em particular, WinNT / Win2000 utilizam um sistema de arquivos próprio – **NTFS** (*New Technology File System*)



## Extensão dos arquivos

- Alguns SOs suportam uma extensão relacionada:
  - **MS-DOS:**
    - 1-3 caracteres – suporta apenas uma extensão
    - 8 caracteres para o nome + 3 para extensão
  - **UNIX:**
    - Mais de 3 caracteres – permite mais de uma extensão ou sem extensão
    - Nome + extensões não pode ser maior que 255 caracteres



## Extensão dos arquivos

- Geralmente associada a algum aplicativo: *doc, zip, jpg*;
- SO pode ou não associar as extensões aos aplicativos:
  - Unix não realiza a associação;
  - Windows realiza a associação;
  - Linux associa algumas:
    - Facilidade de uso;
    - Apenas no GUI.



# Sistema de Arquivos

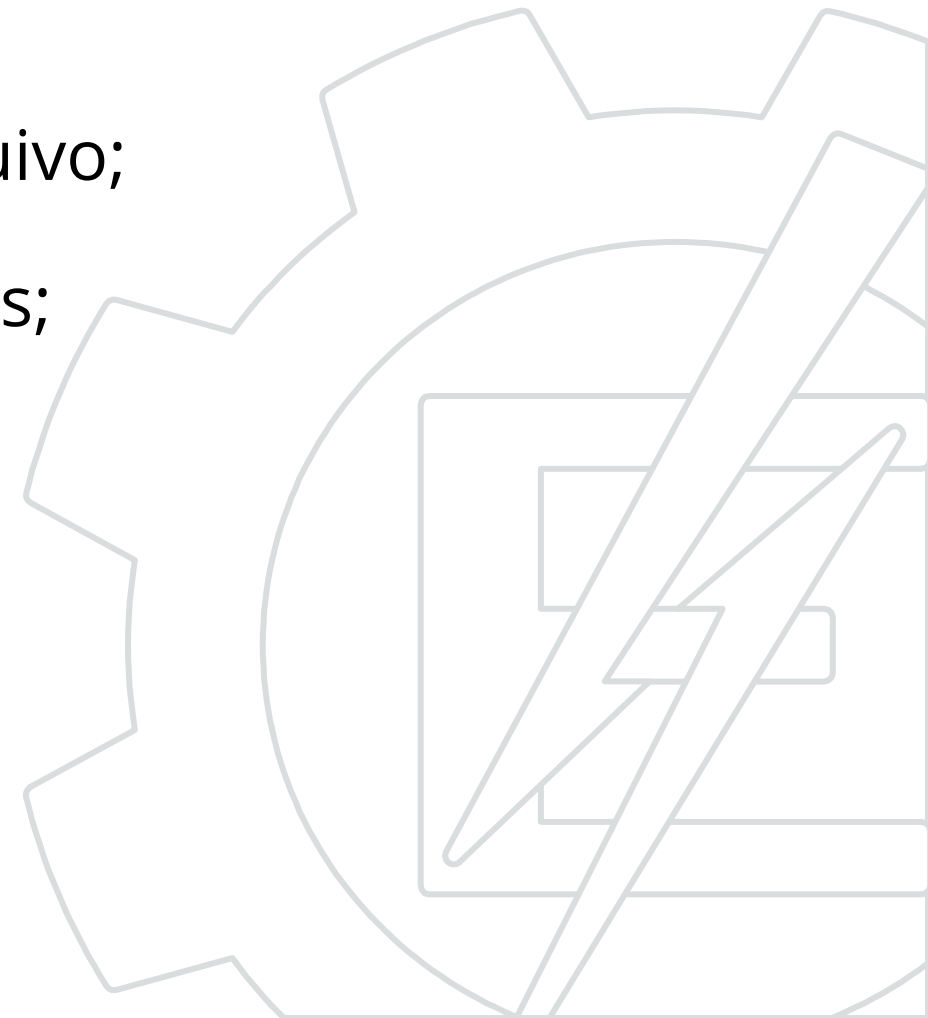
## Extensão dos arquivos



file type	usual extension	function
executable	exe, com, bin or none	read 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, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

### 1. Sequência não-estruturada de *bytes*:

- Para o SO são apenas conjuntos de bytes;
- SO não se importa com o conteúdo do arquivo;
- O significado deve ser dado pelas aplicações;
- Vantagem – Flexibilidade:
  - Os usuários colocam o que quiserem;
  - Nomeiam arquivos como quiserem.
- Utilizado em UNIX e Windows.

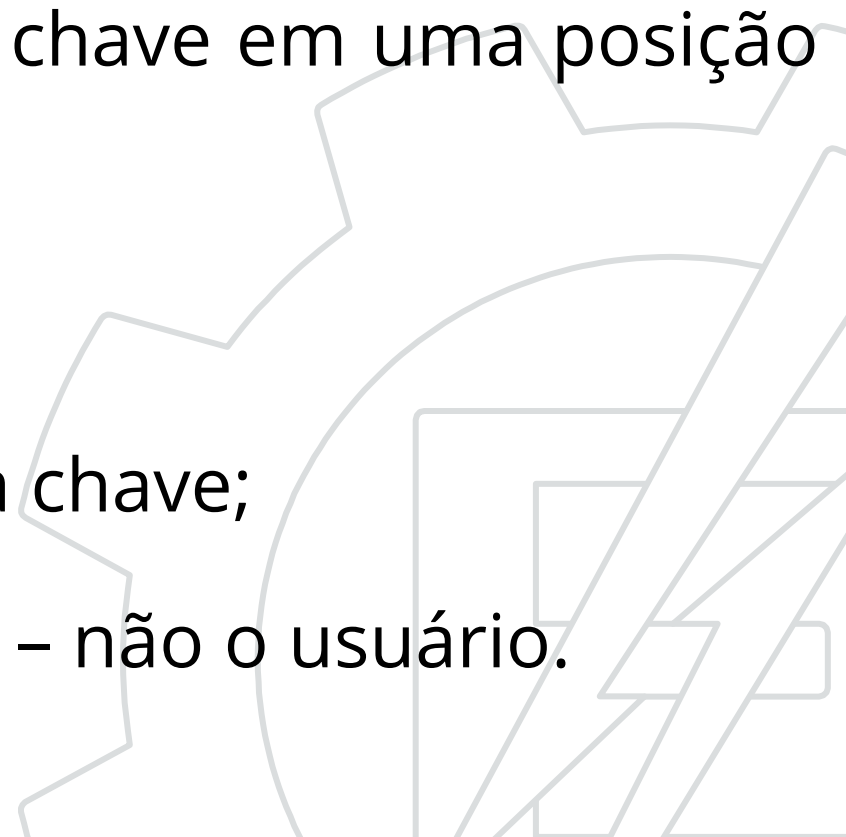


### 2. Sequência de registros de tamanho fixo:

- Cada qual com alguma estrutura interna (p.ex. 80 e 132 caracteres);
- Leitura/escrita são realizadas em registros:
  - Lê, sobrescreve ou adiciona um registro;
  - Utilizado em SOs mais antigos – *mainframes*;
  - 80 caracteres do cartão perfurado.
- Nenhum SO atual utiliza esse esquema.

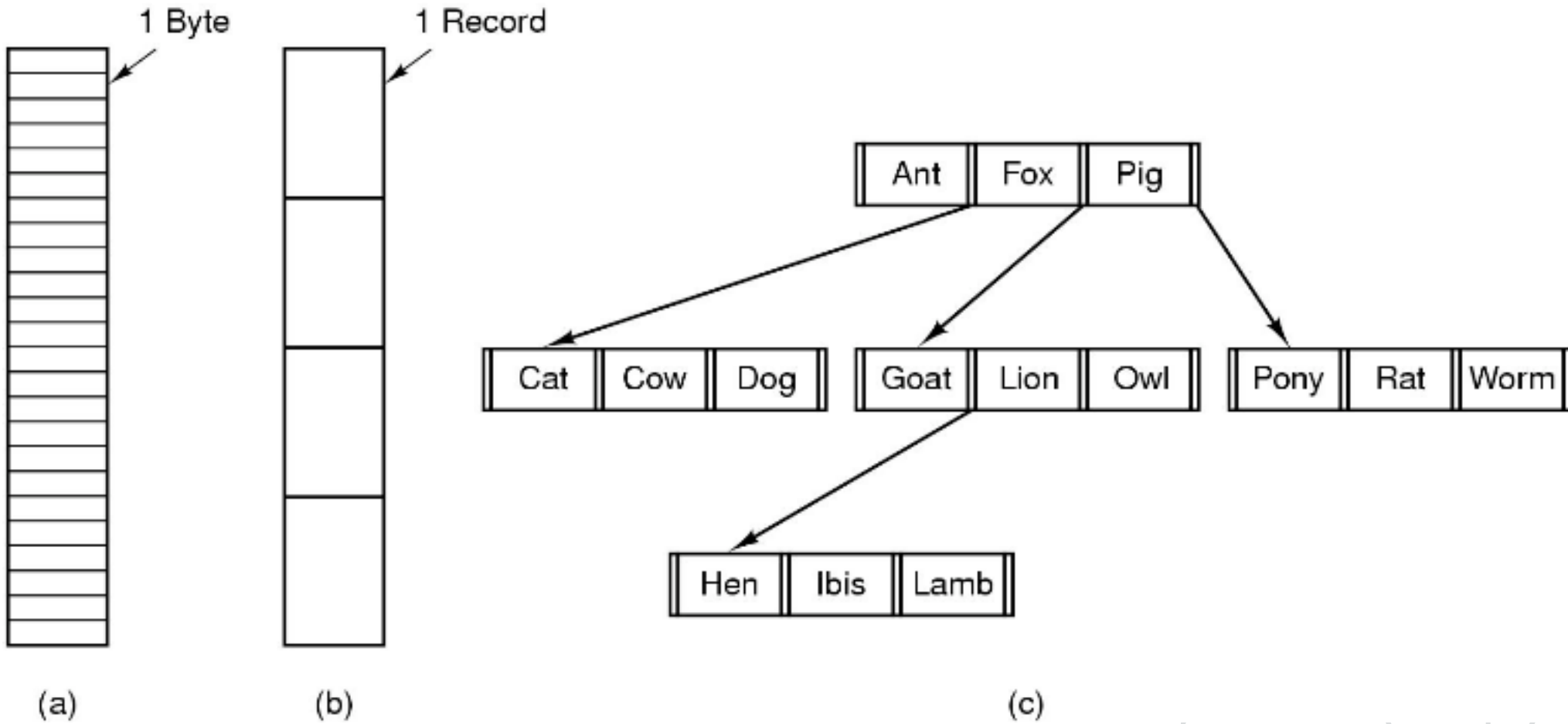


## 3. Árvores de registros (tamanho variado):

- Cada parte do arquivo possui um campo chave em uma posição fixa;
  - O arquivo consiste nesta árvore;
  - Operação: obter o registro com uma certa chave;
  - O SO decide onde colocar novos registros – não o usuário.
  - Utilizado em *mainframes* atuais.
- 

# Sistema de Arquivos

## Estrutura de arquivos



(a) Sequência de byte não estruturada. (b) Sequência de registros de comprimento fixo. (c) Árvore de registros.

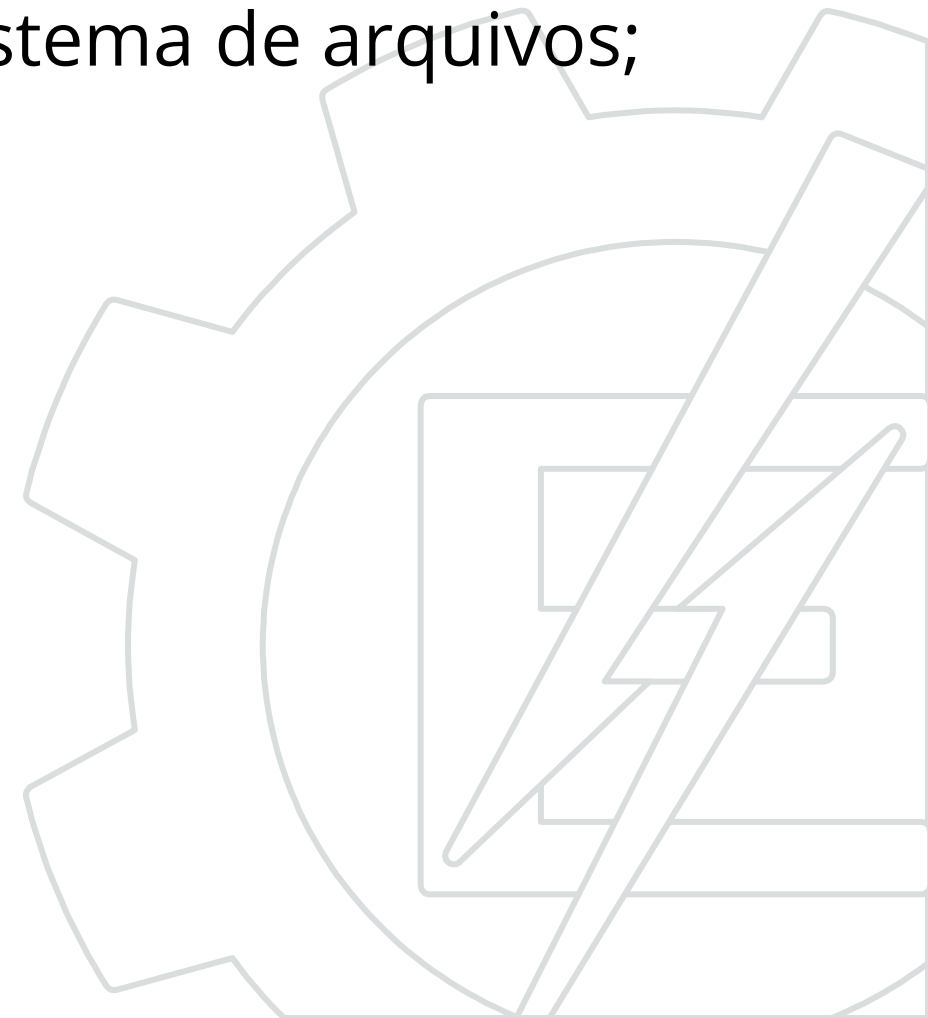


# Sistema de Arquivos

## Tipos de arquivos

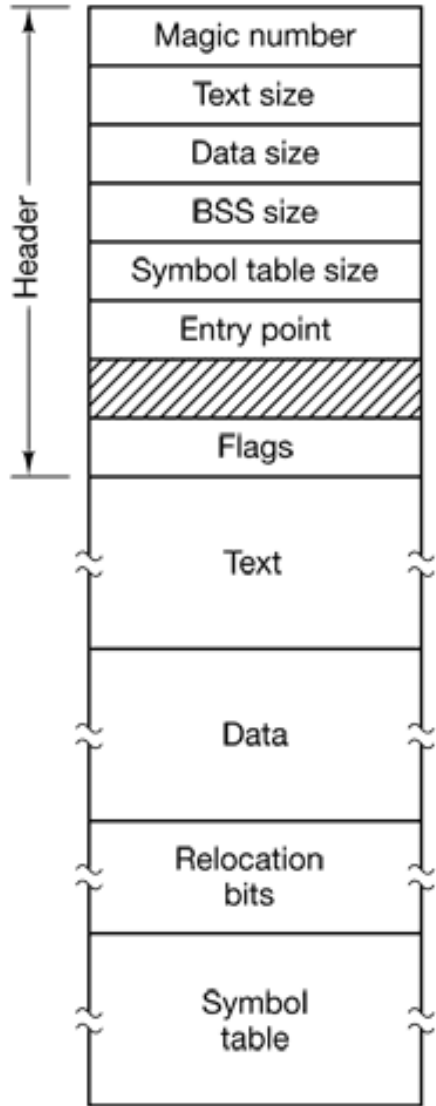
- Arquivos **regulares** – informações dos usuários;
- **Diretórios** – arquivos para estruturar o sistema de arquivos;
- Arquivos do **sistema**: */dev* e */proc*
  - *Arquivos de caracteres especiais (input/output).*
  - *Arquivos especiais de bloco (disks).*

Símbolo identificador	Tipo de Arquivo
-	Arquivo regular
d	Diretórios
c	Arquivos de dispositivos de caracteres
b	Arquivos de dispositivos de blocos
s	Sockets Local
p	Pipes
l	Links Simbólicos



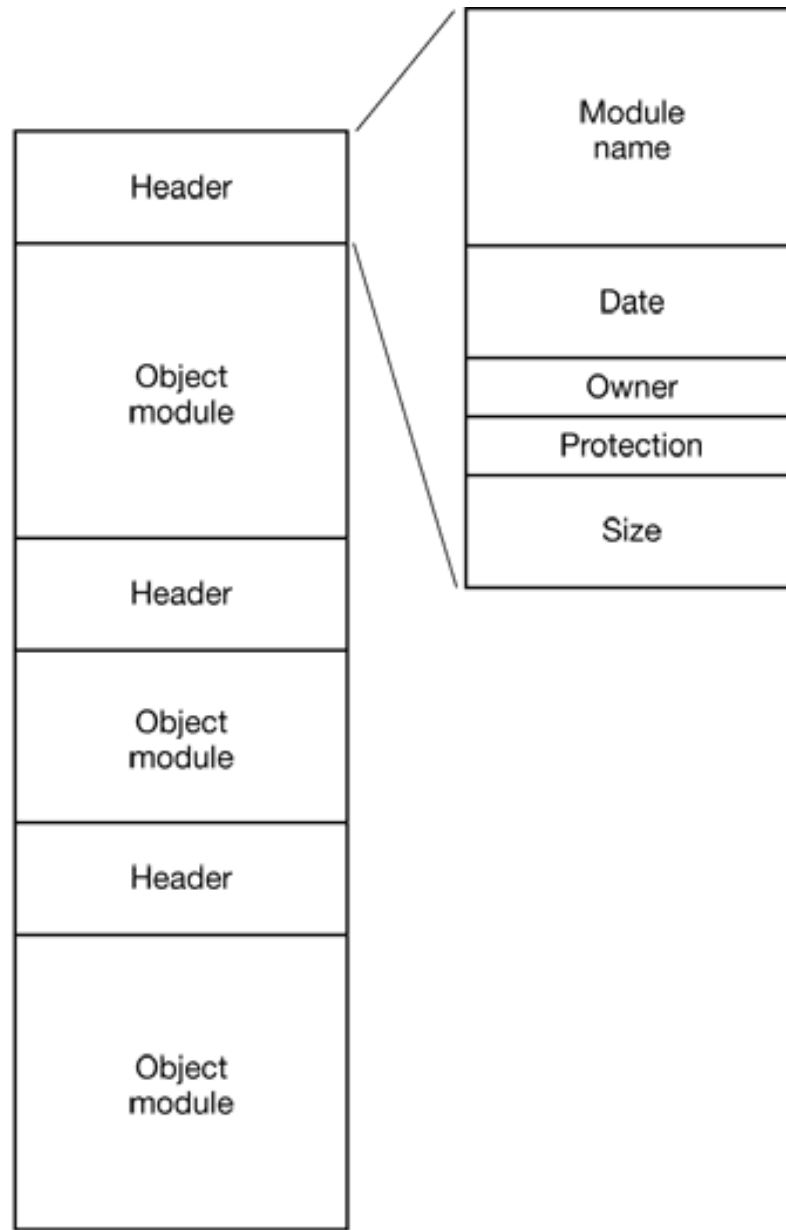
# Sistema de Arquivos

## Bloco de controle de arquivo (FCB)



(a)

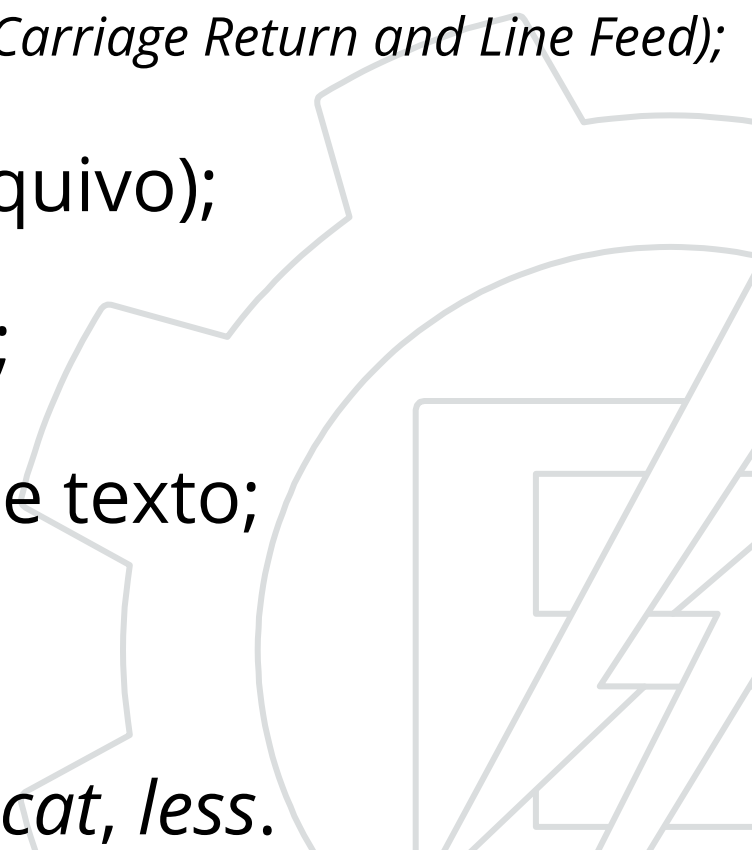
(a) Arquivo executável.



(b)

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

### 1. ASCII:

- Consistem de linhas de texto com CR+LF (*Carriage Return and Line Feed*);
  - Facilitam integração de programas (via arquivo);
  - Podem ser exibidos e impresso como são;
  - Podem ser editados em qualquer editor de texto;
  - Mais portátil e inter-operável.
  - Manuseio / Impressão de arquivos: *more, cat, less*.
- 

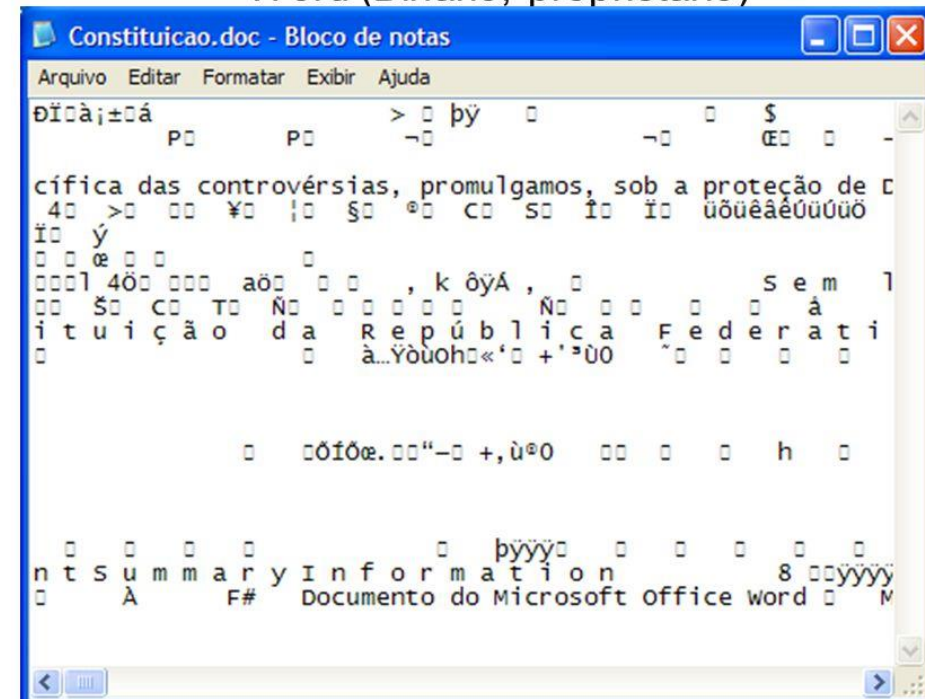
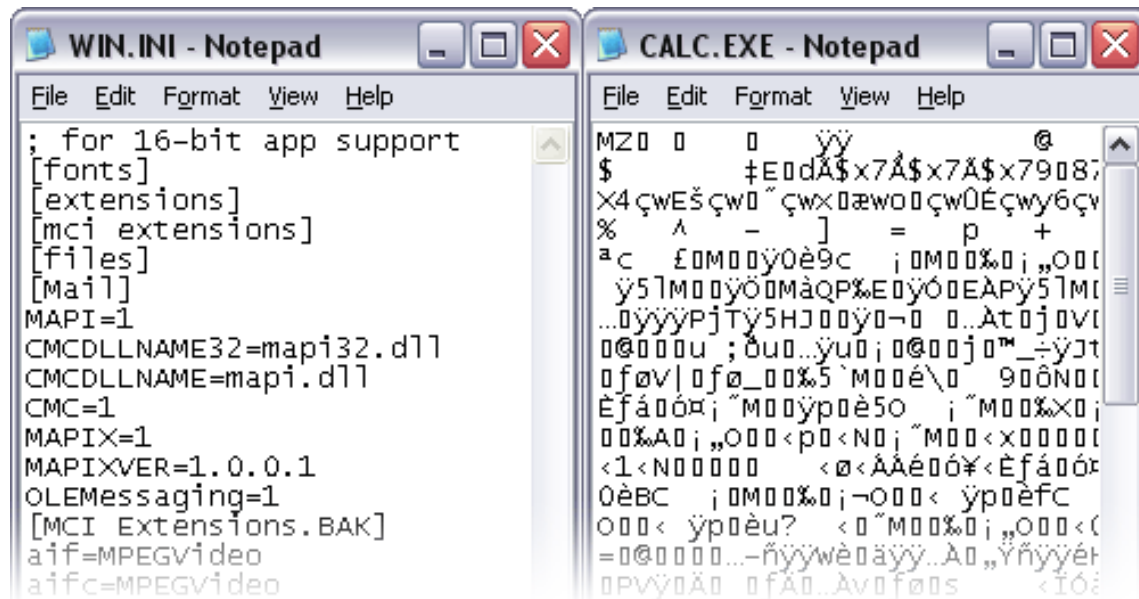
# Sistema de Arquivos

## Tipos de arquivos regulares

### 2. Binário:

- Todo arquivo não-ASCII;
- Estrutura interna conhecida apenas pelos aplicativos que os utilizam – exemplo: programa executável.

Word (Binário, proprietário)



# Sistema de Arquivos

## Atributos de Arquivos

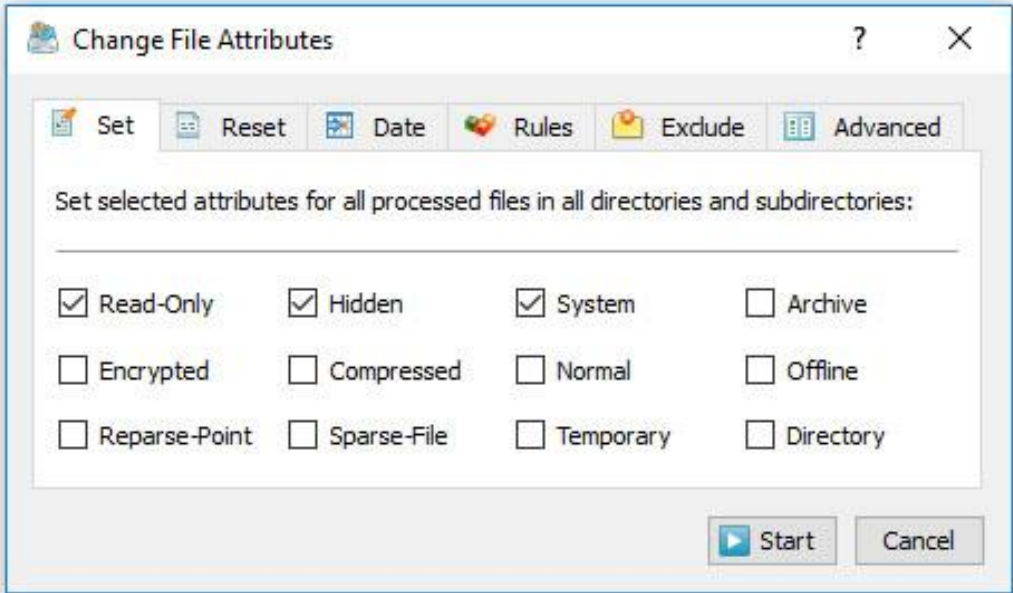
Os atributos de um arquivo variam de um sistema operacional para outro, mas normalmente são os seguintes:

- Nome.
- Identificador.
- Tipo.
- Localização.
- Tamanho.
- Proteção.
- Hora, data e identificação do usuário



# Sistema de Arquivos

## Atributos de arquivos:



### Show File Attributes in Linux

a append only

c compressed

d no dump

e extent format

i immutable

j data journaling

s secure deletion

t no tail-merging

u undeletable

A no atime updates

D synchronous directory updates

S synchronous updates

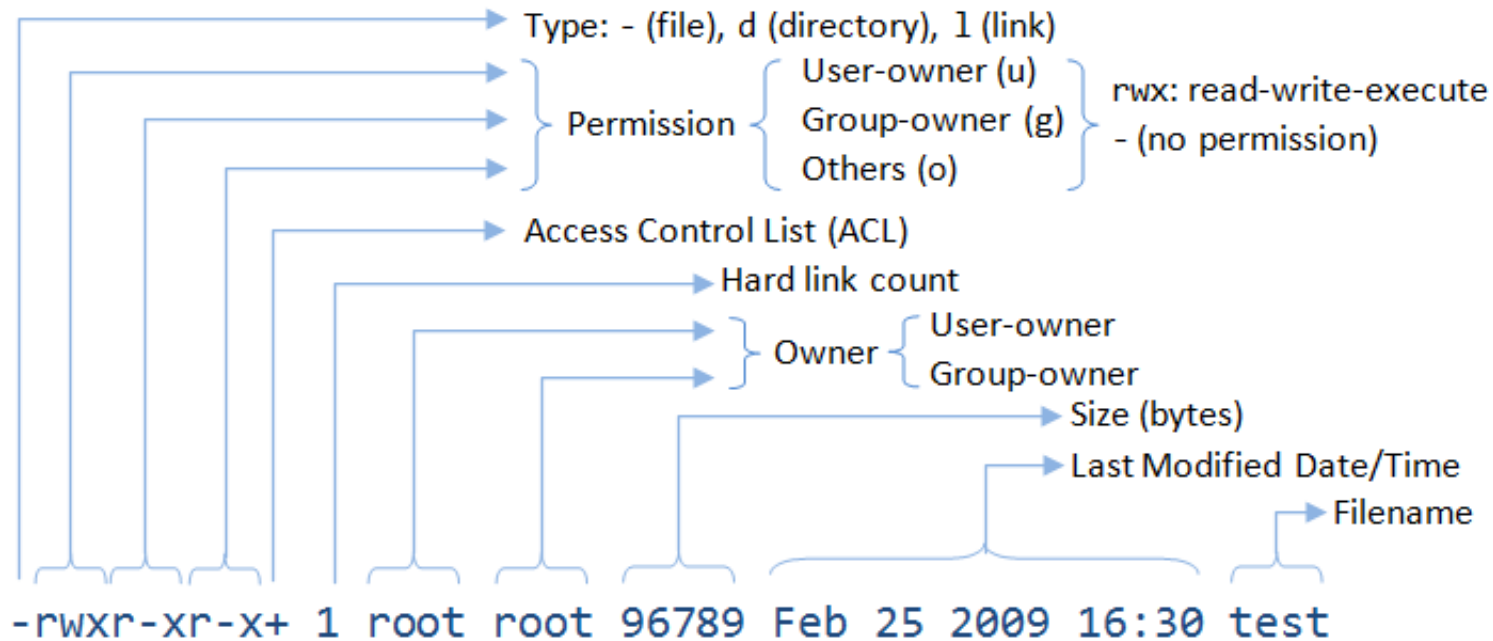
T top of directory hierarchy

Comando	Permissão
# chmod 777 file	rwXrwxrwx
# chmod 755 file	rwXr-xr-x
# chmod 644 file	rw-r--r-
# chmod 000 file	-----
# chmod a-w files	r-xr-xr-x
# chmod o-x files	rwsrwsrw
# chmod go-rwx files	rwX-----
# chmod u+rw files	rw-----
# chmod a+x files	--X--X--X
# chmod ug+rx files	r-xr-x---

String	Binário	Decimal	Permissões
---	000	0	Nenhuma permissão
--x	001	1	Executar
-w-	010	2	Escrever
-wx	011	3	Escrever e Executar
r--	100	4	Ler
r-x	101	5	Ler e Executar
rw-	110	6	Ler e Escrever
rwX	111	7	Ler, Escrever e Executar

## Atributos de arquivos:

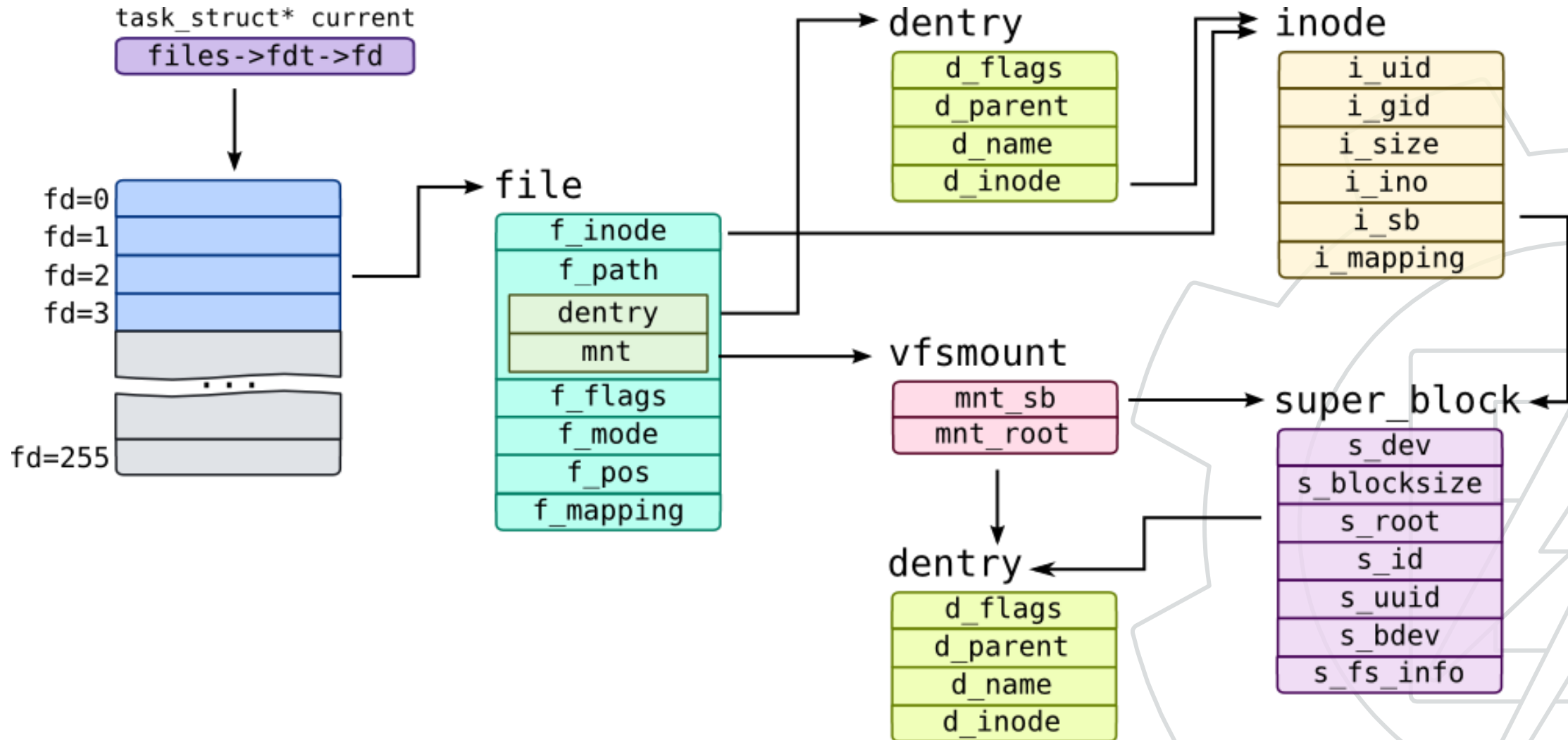
- Além do nome e dados, todo arquivo possui outras informações a respeito dele – atributos (ou metadados).
- A lista de atributos varia de SO para SO.





# Sistema de Arquivos

## Atributos de arquivos:



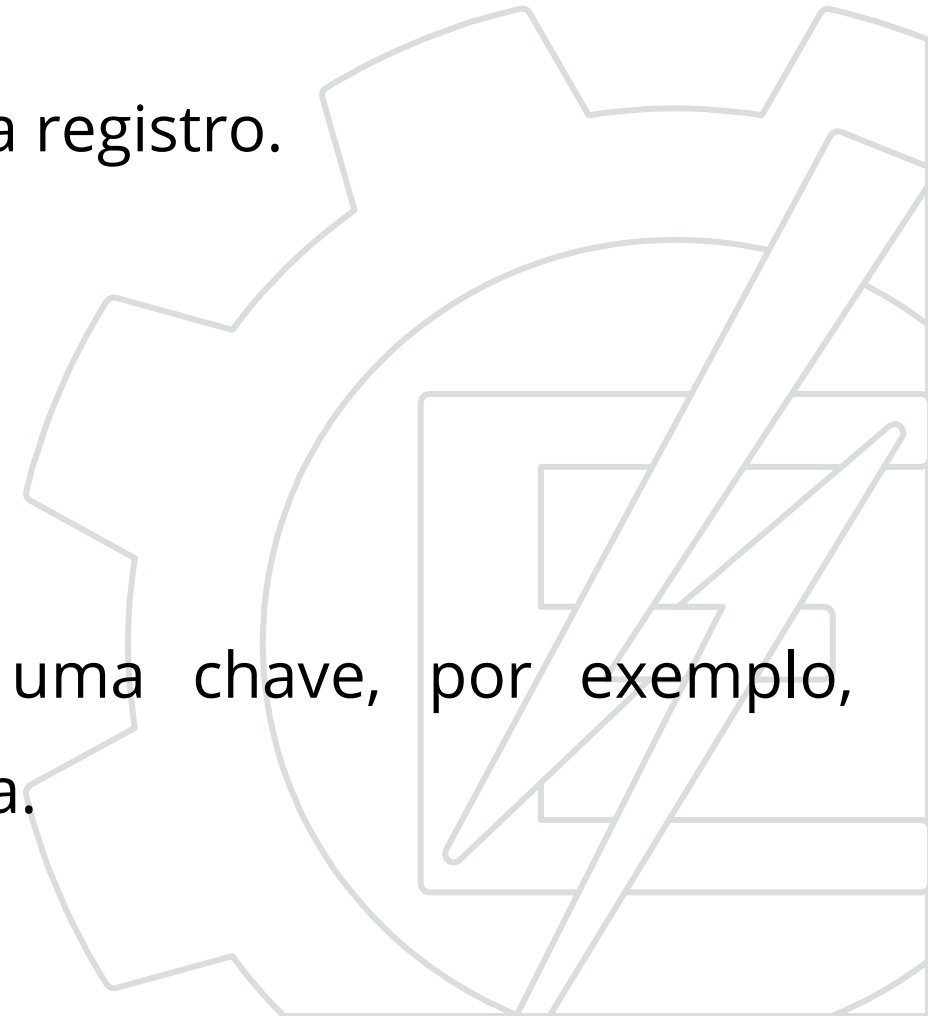


## **SOs mais antigos:**

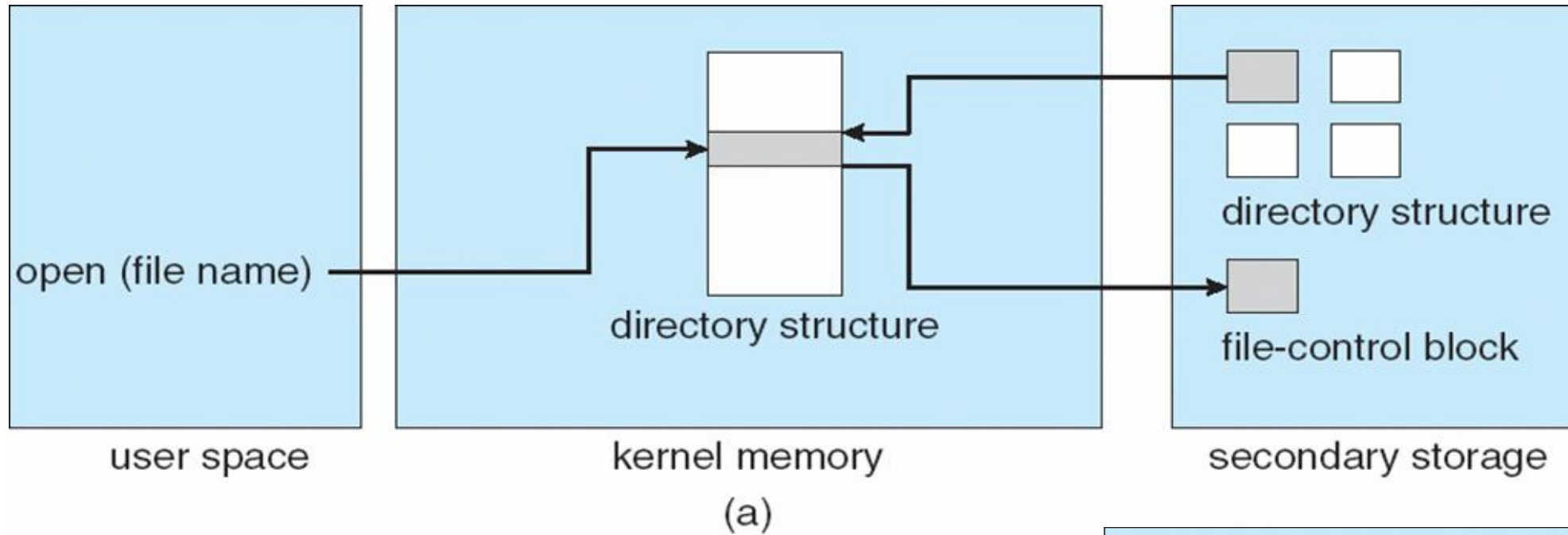
- Apenas acesso sequencial no disco;
- Leitura seguindo a ordem *byte a byte*, ou registro a registro.

## **SOs mais modernos:**

- Realizam acesso aleatório no disco;
- Acesso realizado fora de ordem através de uma chave, por exemplo, utilizando a base de dados de uma empresa aérea.



# Acesso aos Arquivos

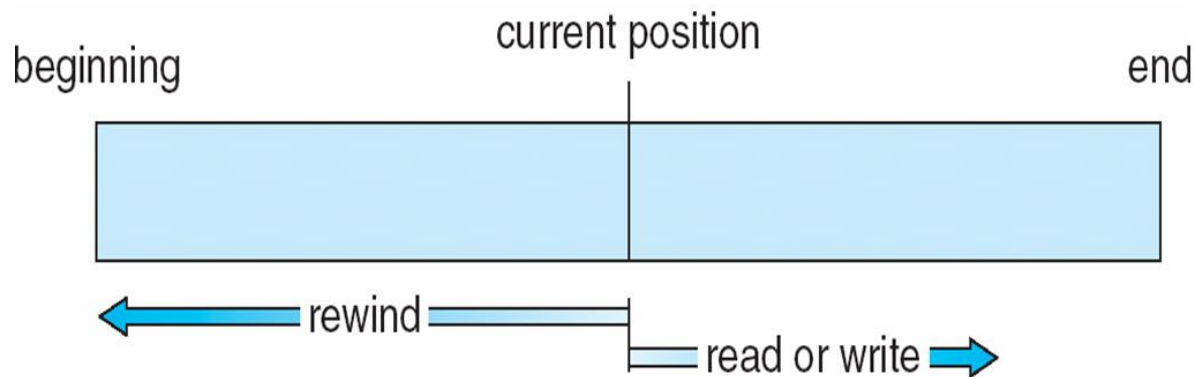


file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks or pointers to file data blocks

*File Control Block (FCB)*

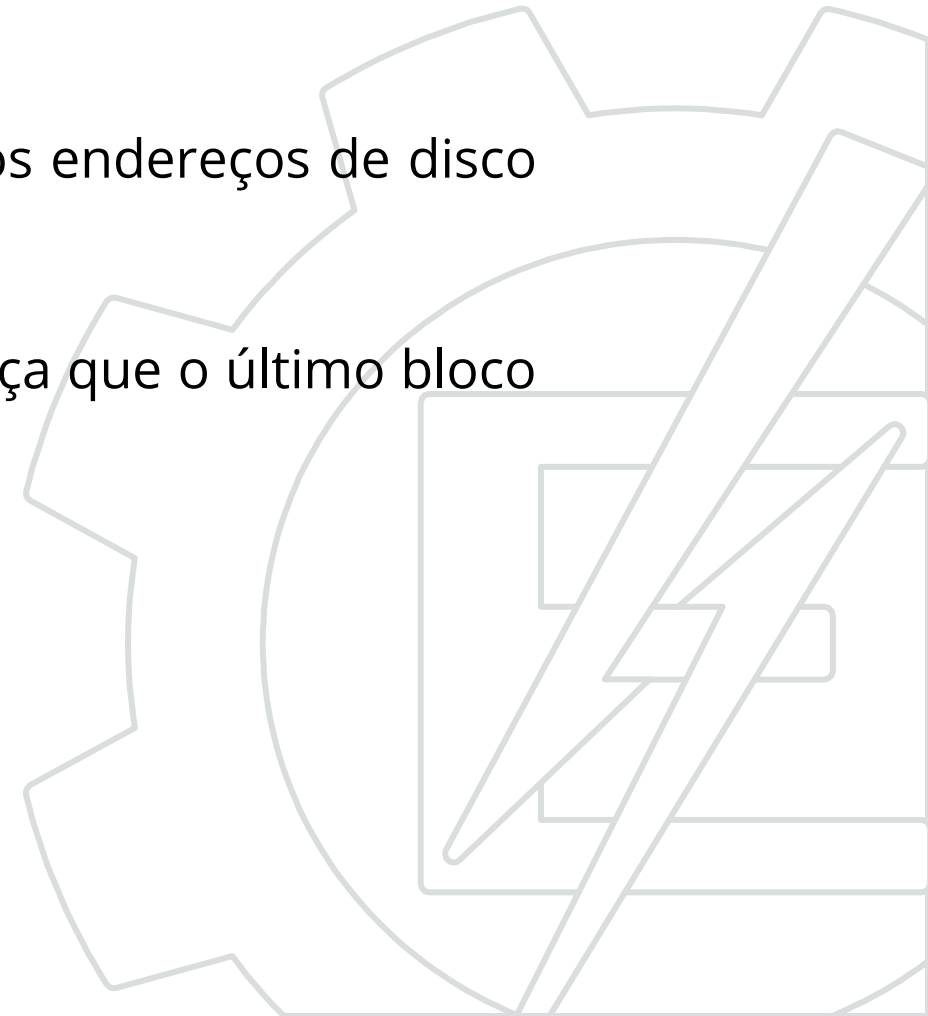
realiza a leitura sequencial.

- 
- The diagram illustrates a tape drive's state. A horizontal light blue rectangle represents the tape. Above the rectangle, the word "beginning" is at the left end, "current position" is above a vertical line, and "end" is at the right end. Below the rectangle, a thick blue arrow points left, labeled "rewind", and a thin blue arrow points right, labeled "read or write".



# Acesso aos Arquivos

- Operações mais comuns (*system calls*):
  - **Create**: o arquivo é criado sem dados.
  - **Delete**: o arquivo é removido do disco.
  - **Open**: permite que o SO busque os atributos e lista dos endereços de disco na memória.
  - **Close**: libera o espaço ocupado pelo *Open*, também força que o último bloco de dados seja escrito no disco.
  - **Read**: Lê do arquivo para um *buffer*.
  - **Write**: escreve dados no arquivo.
  - **Append**: escreve dados ao final do arquivo.
  - **Seek**: para acesso aleatório – procura dados.



# Acesso aos Arquivos

- Operações mais comuns (*system calls*):
  - ***Get attributes***: obtém os atributos de um arquivo.
  - ***Set attributes***: altera os atributos de um arquivo.

Read Only Operations	Modification Operations		
	Regular Files	Directories	Inodes
execve, chdir, access, chroot, readlink, uselib, statfs, stat, lstat, stat64, lstat64, oldstat, getdents, getdents64, readdir	open, truncate, truncate64	creat, link, unlink, mknod, rename, mkdir, rmdir, acct, symlink, open	chmod, lchown, utime, oldlstat, chown, lchown32, chown32

# Acesso aos Arquivos

## Process management

Call	Description
pid = fork()	Create a child process identical to the parent
pid = waitpid(pid, &statloc, options)	Wait for a child to terminate
s = execve(name, argv, environp)	Replace a process' core image
exit(status)	Terminate process execution and return status

## File management

Call	Description
fd = open(file, how, ...)	Open a file for reading, writing, or both
s = close(fd)	Close an open file
n = read(fd, buffer, nbytes)	Read data from a file into a buffer
n = write(fd, buffer, nbytes)	Write data from a buffer into a file
position = lseek(fd, offset, whence)	Move the file pointer
s = stat(name, &buf)	Get a file's status information

## Directory- and file-system management

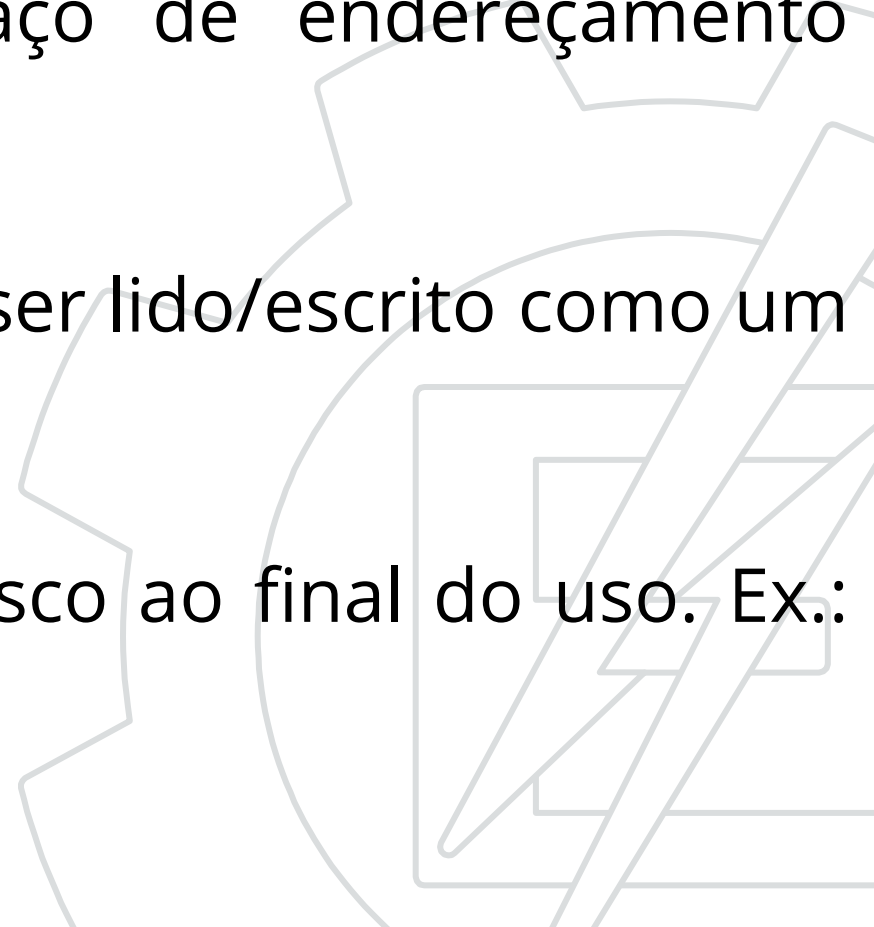
Call	Description
s = mkdir(name, mode)	Create a new directory
s = rmdir(name)	Remove an empty directory
s = link(name1, name2)	Create a new entry, name2, pointing to name1
s = unlink(name)	Remove a directory entry
s = mount(special, name, flag)	Mount a file system
s = umount(special)	Unmount a file system

## Miscellaneous

Call	Description
s = chdir(dirname)	Change the working directory
s = chmod(name, mode)	Change a file's protection bits
s = kill(pid, signal)	Send a signal to a process
seconds = time(&seconds)	Get the elapsed time since Jan. 1, 1970



## Arquivos mapeados na memória:

- Alguns SO mapeiam arquivos no espaço de endereçamento (virtual) de um processo.
  - Acesso mais rápido ao arquivo, que pode ser lido/escrito como um arranjo de bytes na memória.
  - Arquivos modificados são gravados no disco ao final do uso. Ex.: bibliotecas compartilhadas (*includes*).
  - *System calls: **map** e **unmap**.*
- 

# Sistema de Arquivos

## Arquivos mapeados na memória:

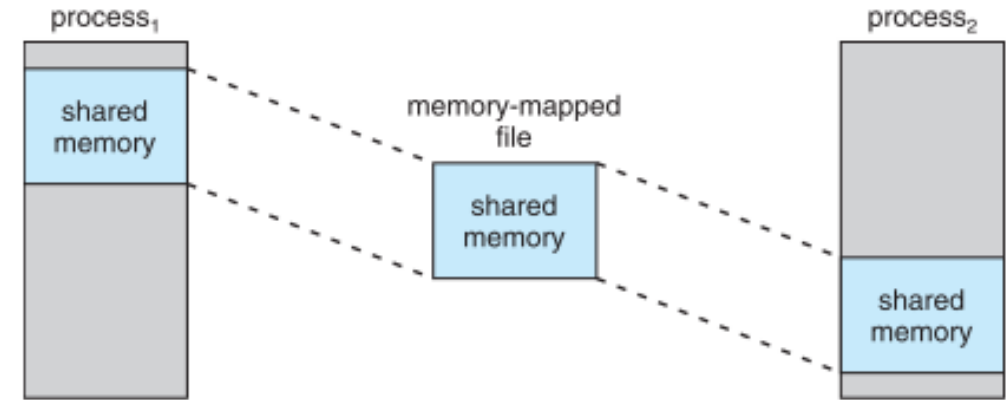
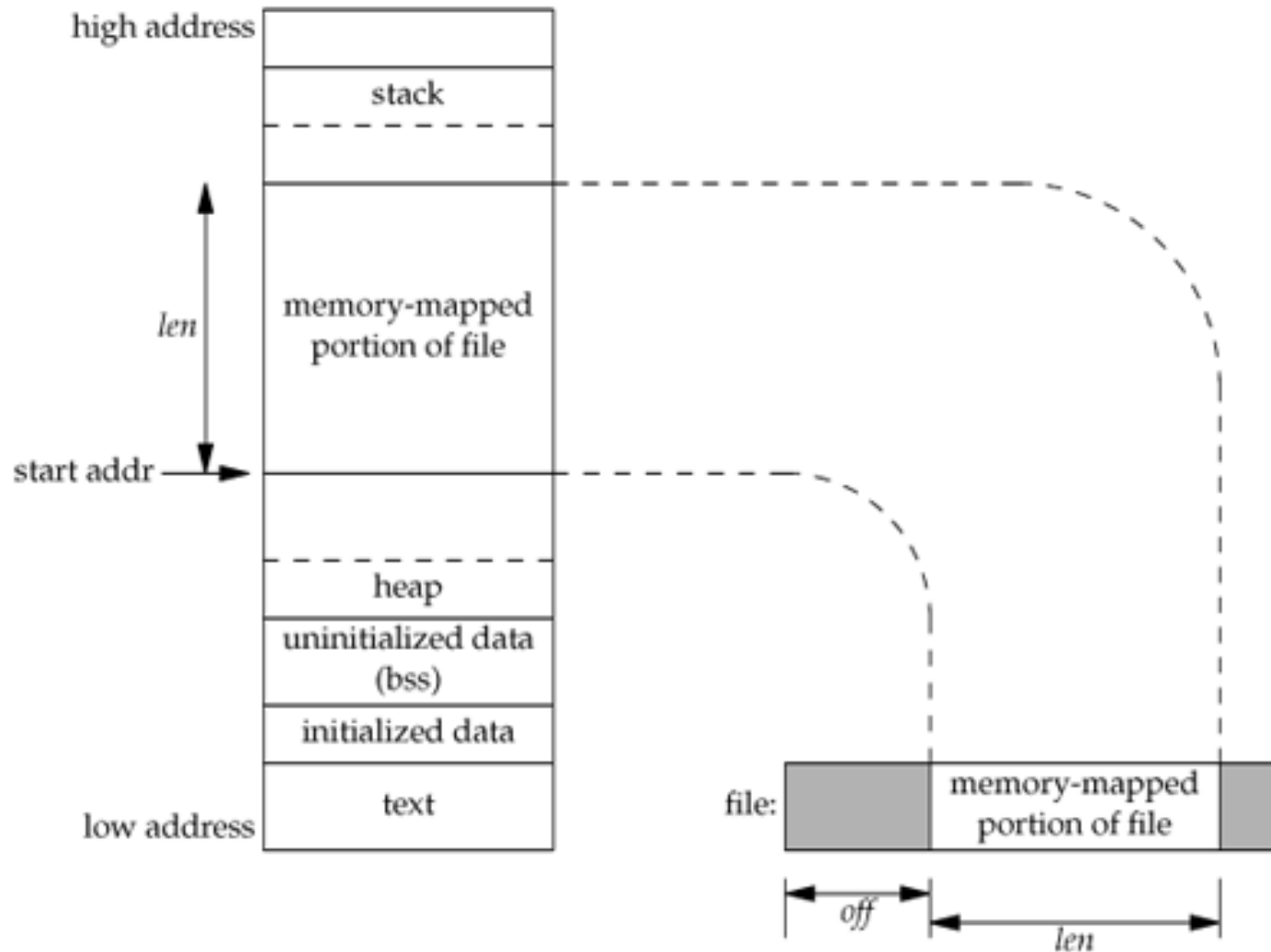


Figure 9.23 Shared memory using memory-mapped I/O.



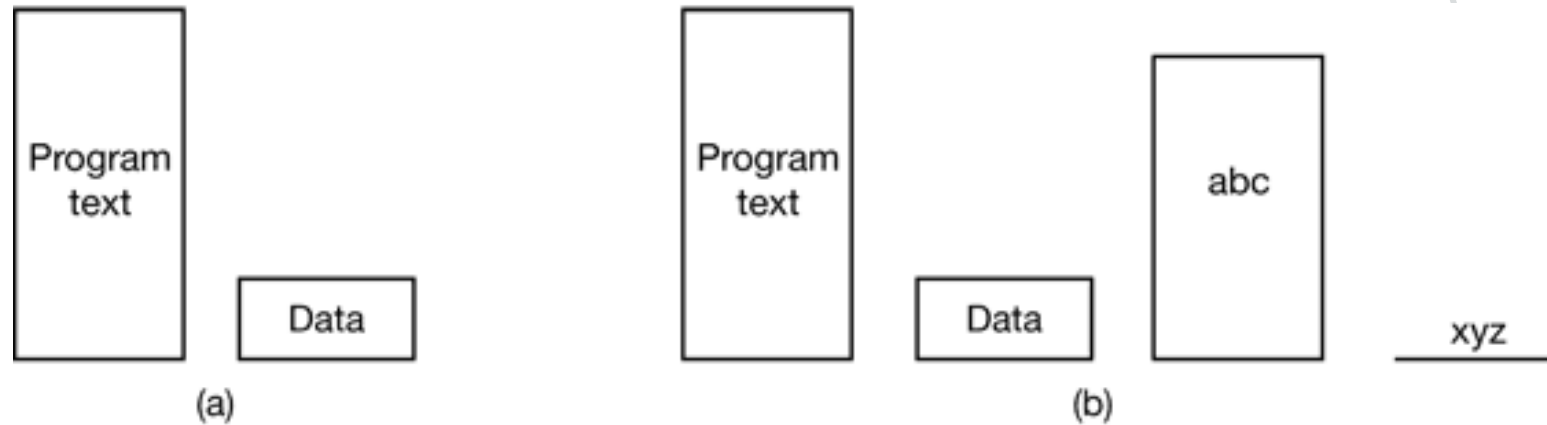
## **Arquivos mapeados na memória:**

- Funciona melhor com segmentação.
- Facilita o mapeamento através da utilização de segmentos.



## Arquivos mapeados na memória:

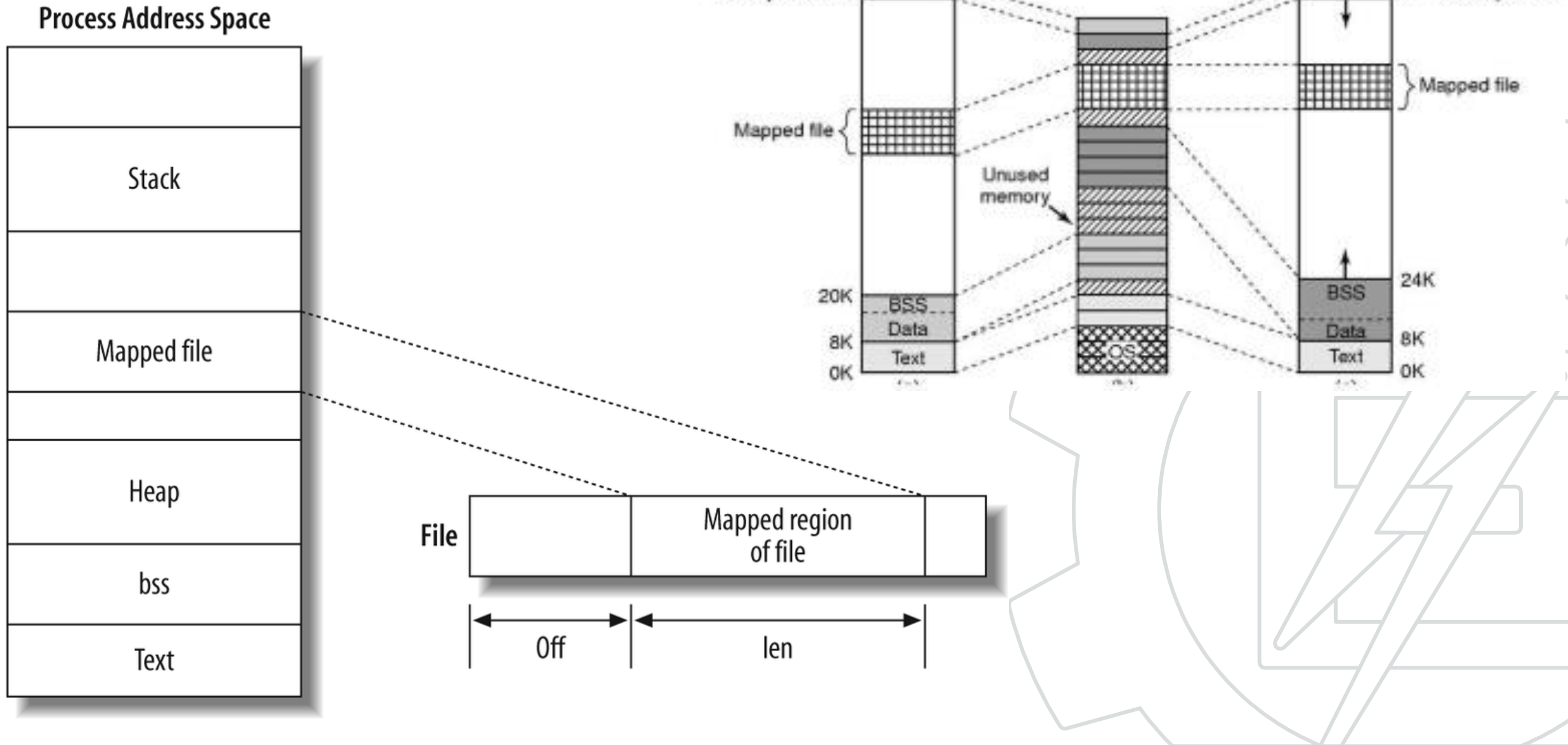
- Funciona melhor com segmentação.
- Facilita o mapeamento através da utilização de segmentos.



- Suponha dois segmentos:
  - Queremos ler de um arquivo e criar outro;
  - O programa mapeia o arquivo de origem (*abc*);
  - Cria um segmento vazio para o arquivo de destino (*xyz*).

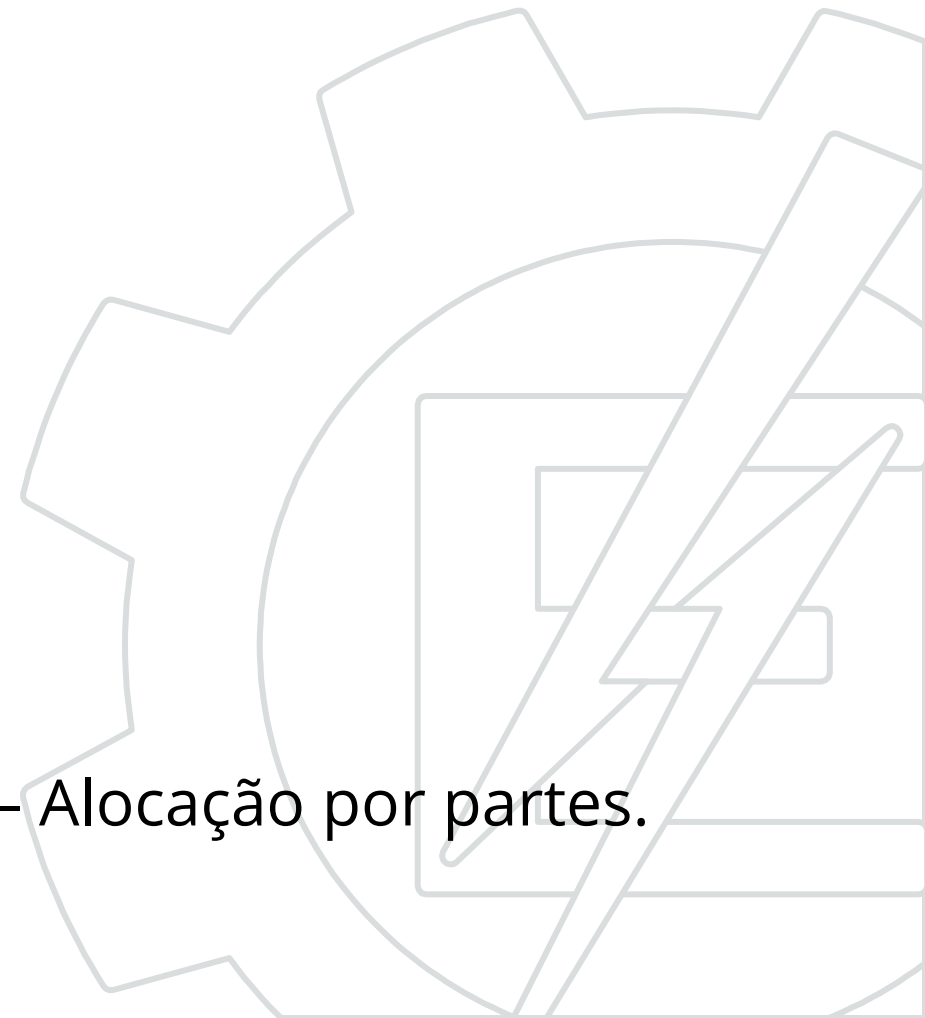
# Sistema de Arquivos

## Arquivos mapeados na memória:



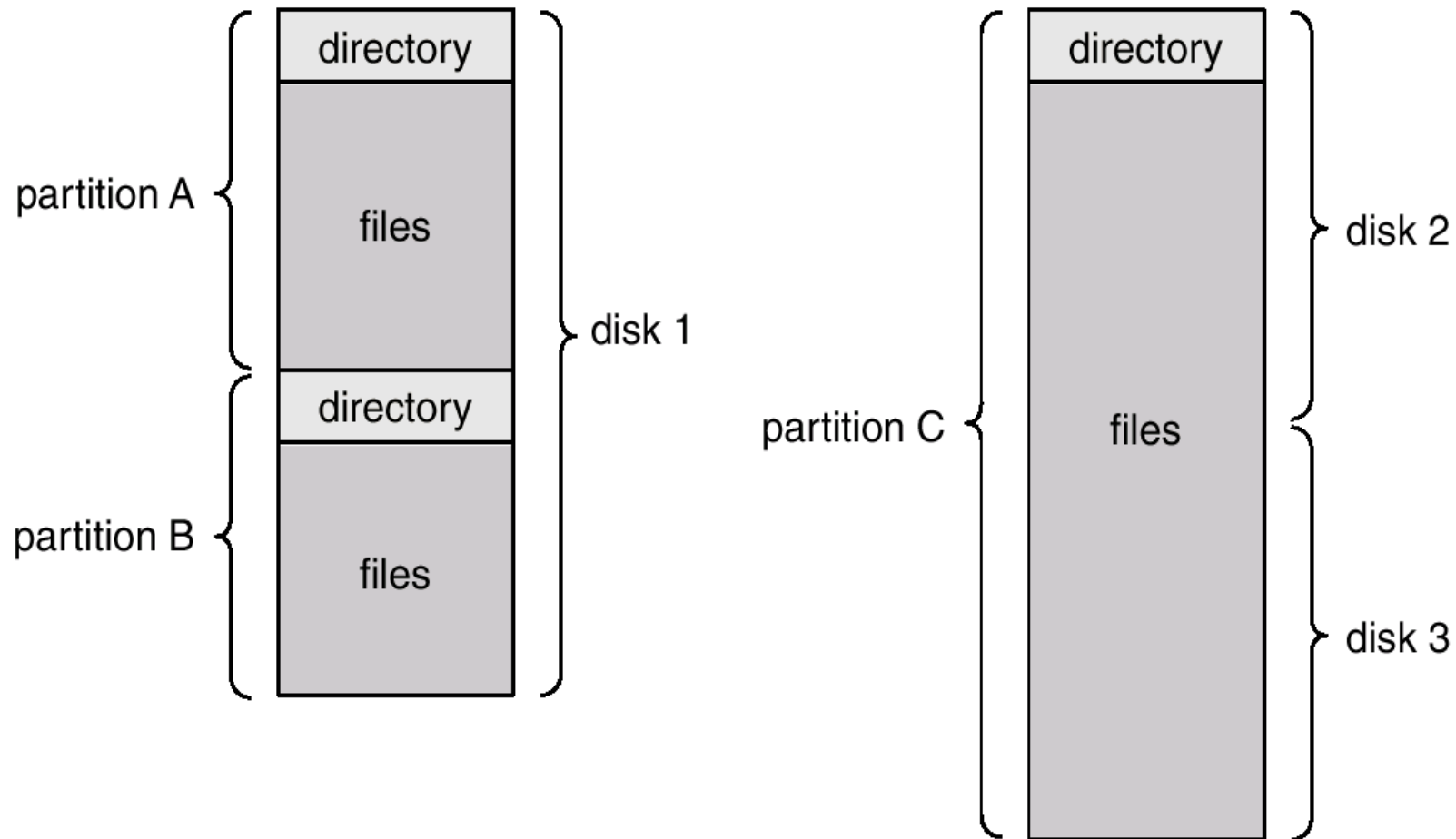
## **Arquivos mapeados na memória:**

- Vantagens:
  - Compartilhamento e trabalho colaborativo.
- Problemas:
  - Uso do espaço virtual para o arquivo;
  - Sincronização;
  - Arquivo pode ser maior que o espaço virtual – Alocação por partes.



## Diretórios:

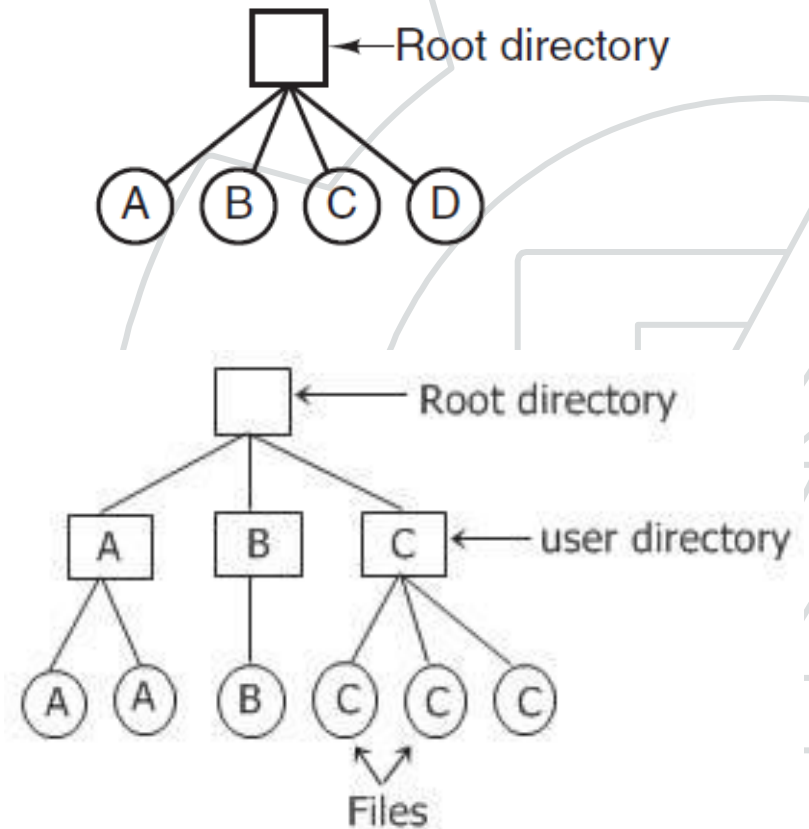
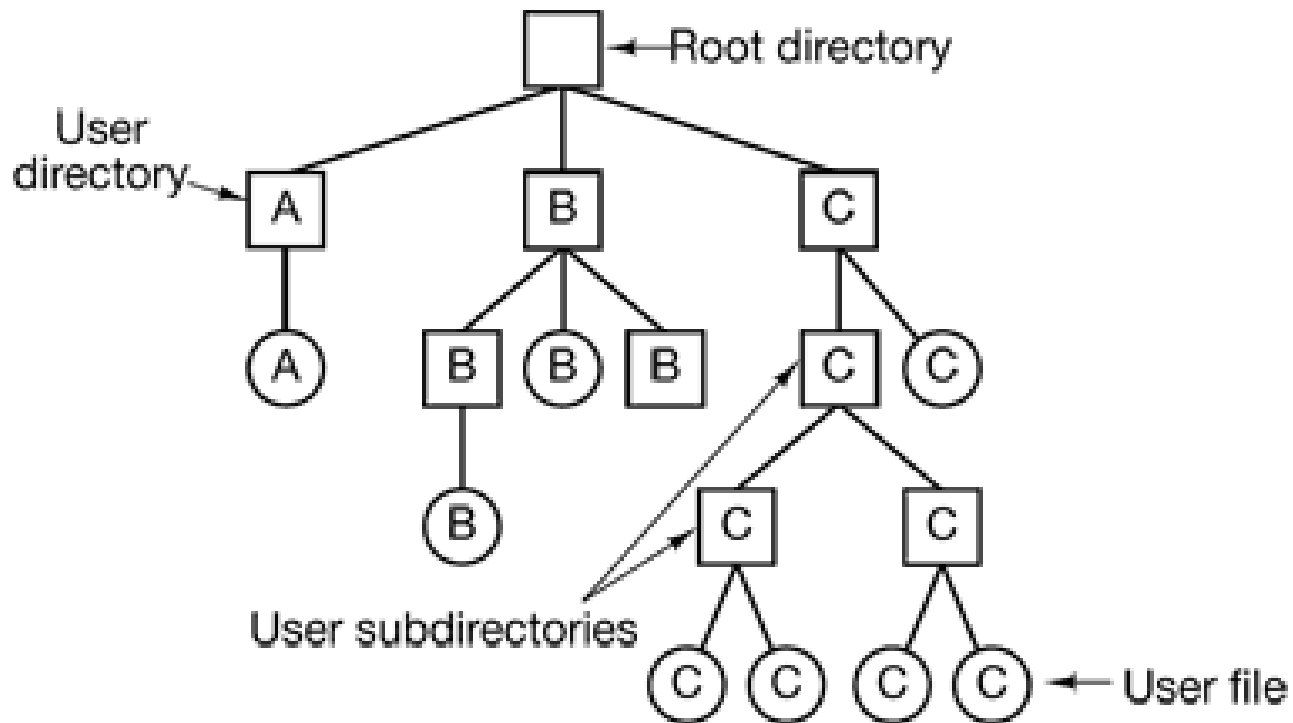
- São arquivos que mantêm a estrutura do sistema de arquivos.



# Sistema de Arquivos

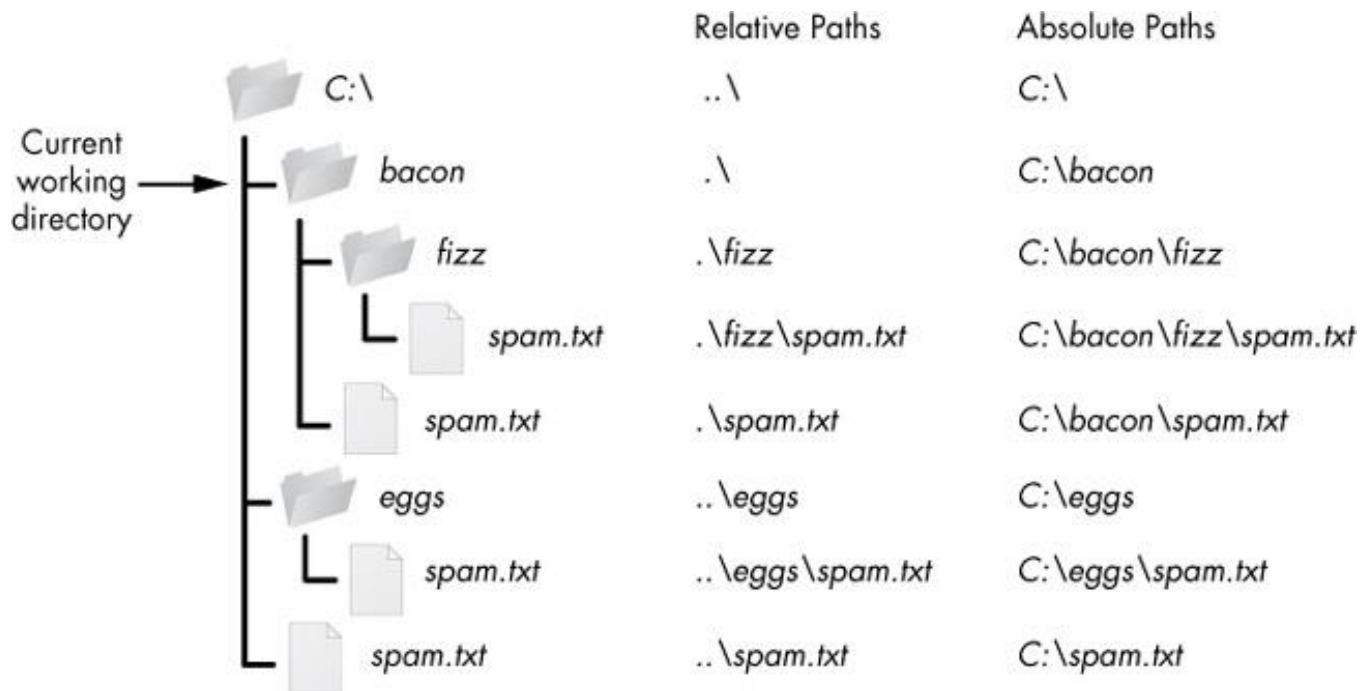
## Diretórios:

- São arquivos que mantêm a estrutura do sistema de arquivos.
- Podem ser organizados como: nível único, dois níveis e de modo hierárquico.



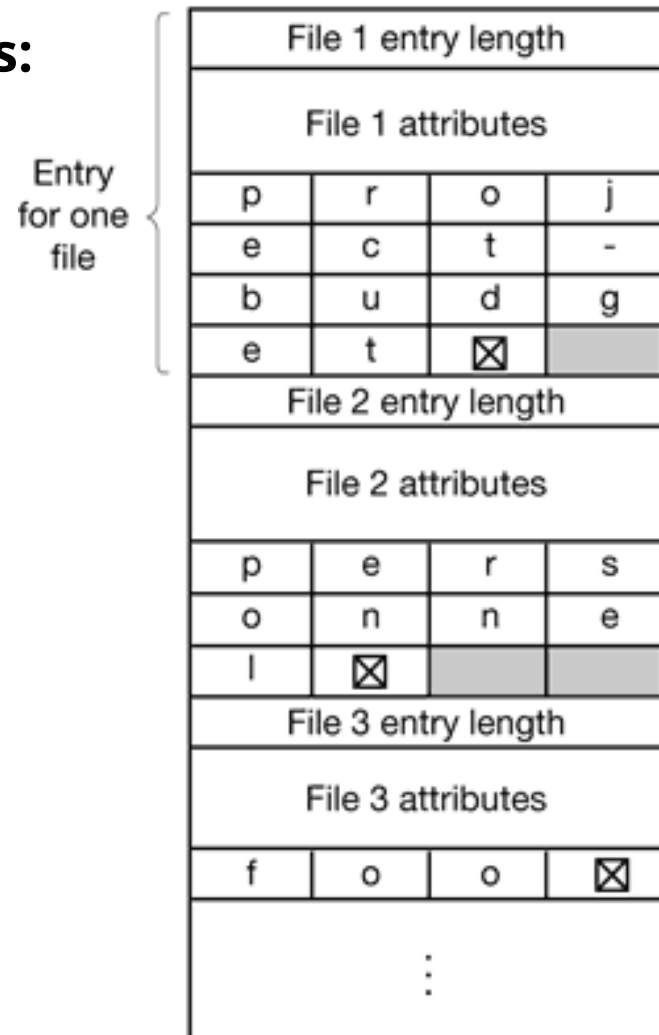
## Diretórios - caminhos:

- Existem dois métodos para acessar os arquivos:
  - Caminho absoluto - caminho desde a raiz
  - Caminho relativo - considera o diretório atual

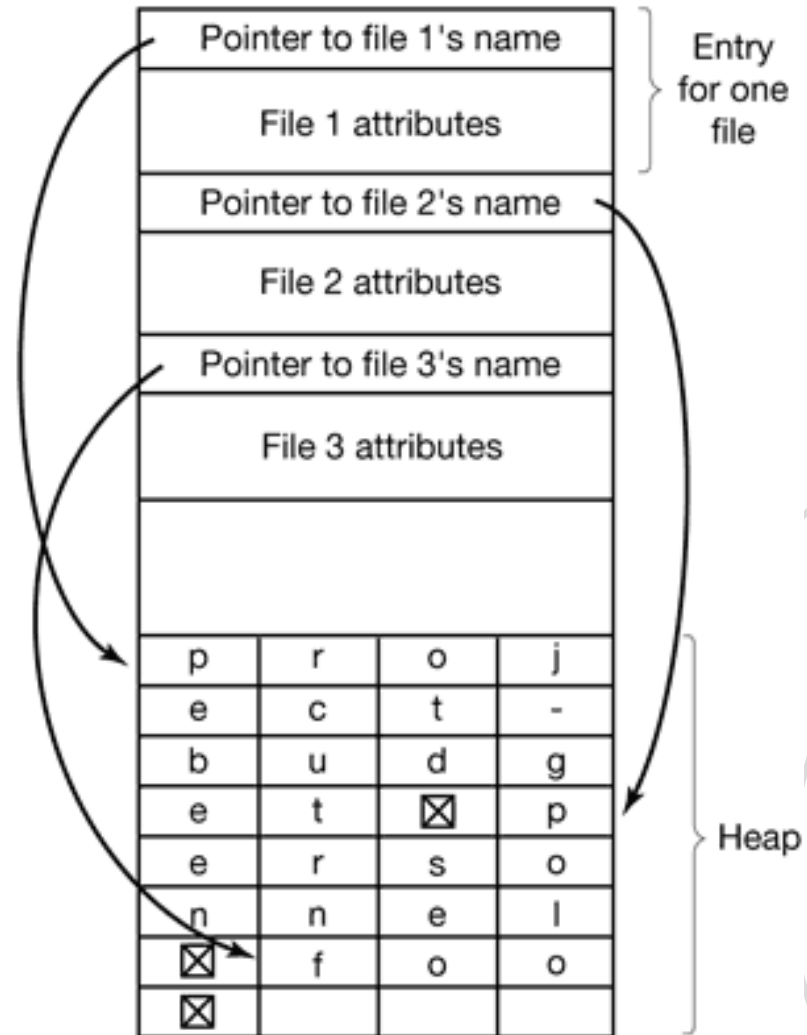


# Sistema de Arquivos

## Diretórios:



(a)



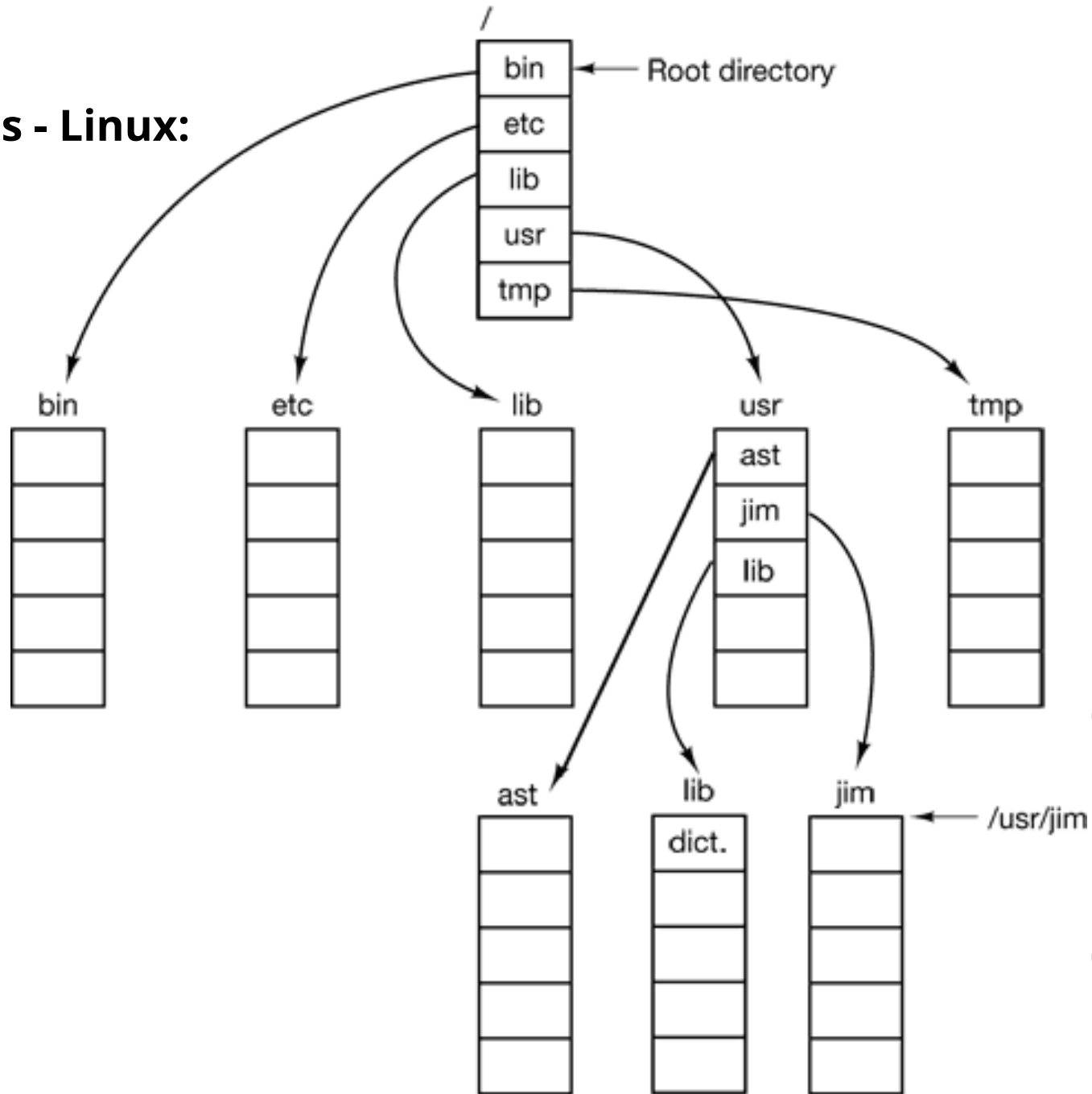
(b)

Modos de implementar nomes longos em diretórios. (a) *In-line*. (b) *In a heap*.



# Sistema de Arquivos

## Diretórios - Linux:

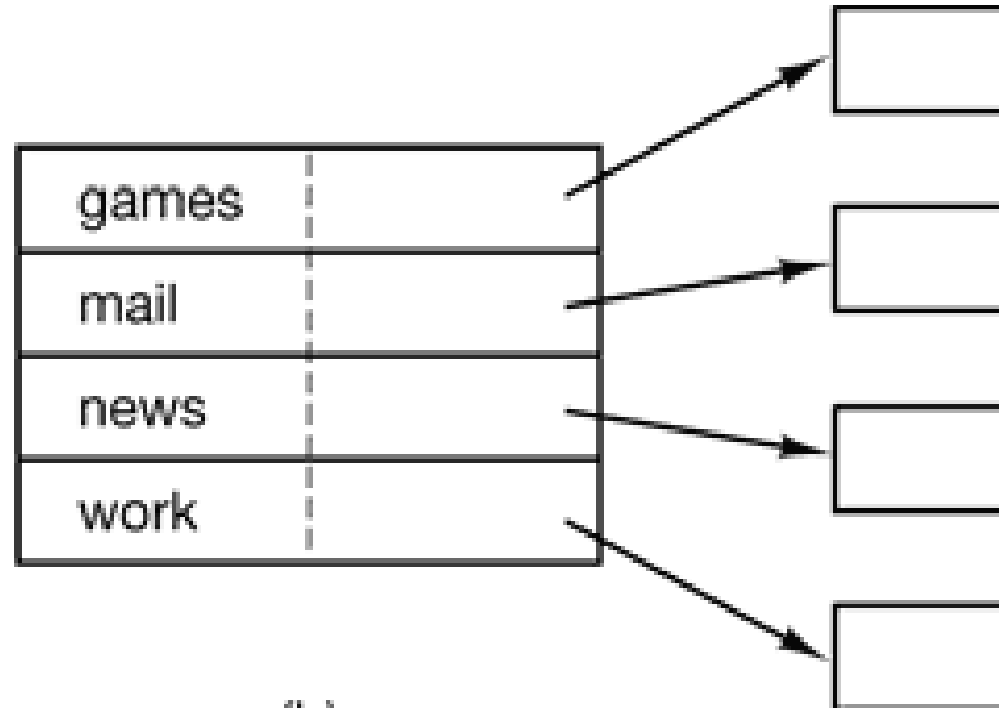


# Sistema de Arquivos

## Diretórios - Linux:

games	attributes
mail	attributes
news	attributes
work	attributes

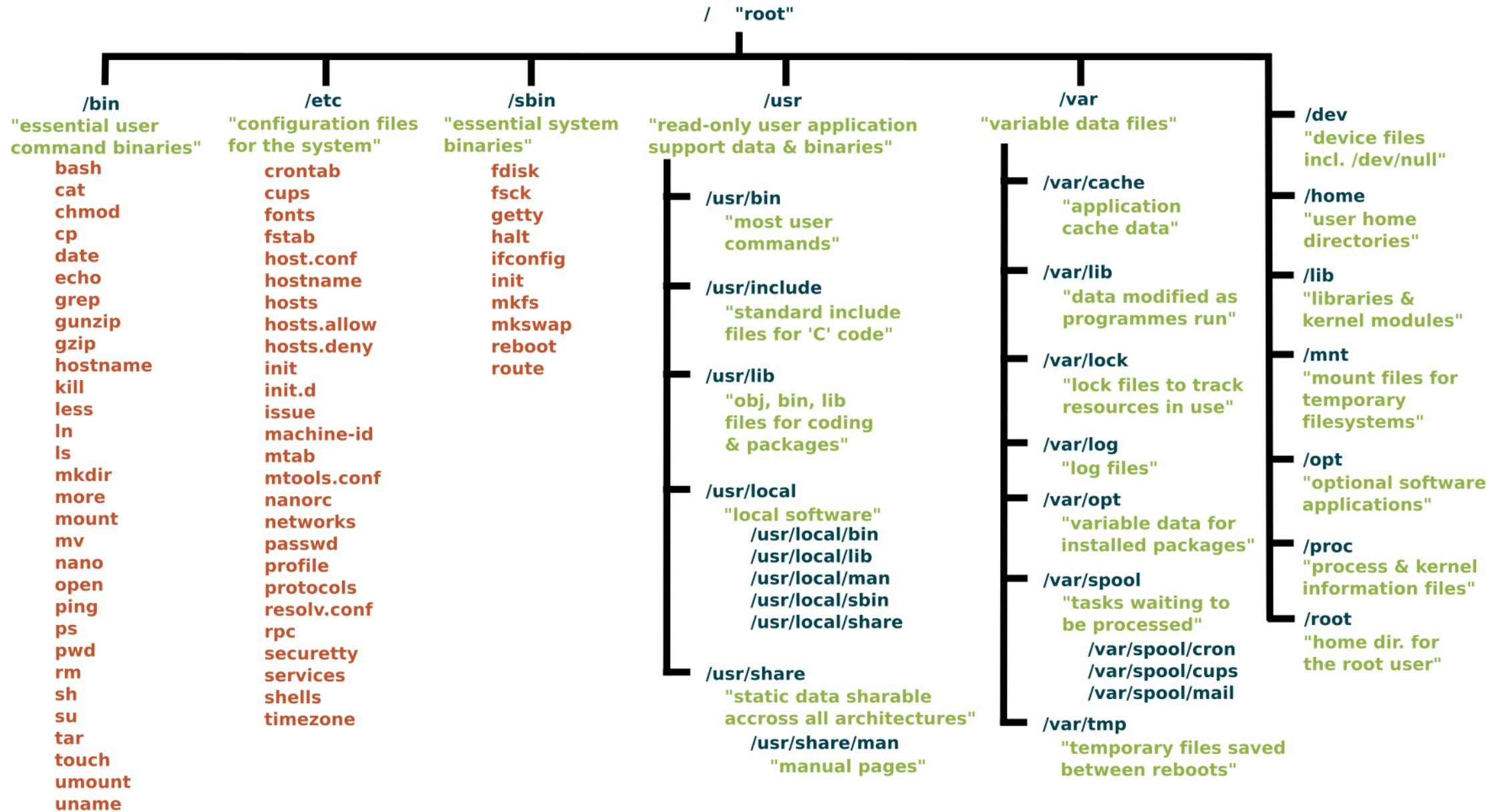
(a)



(b)

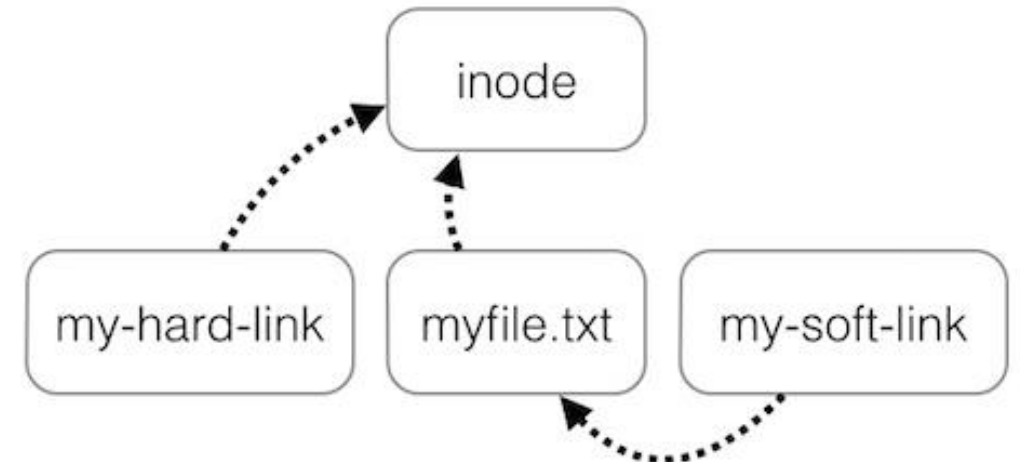
Data structure  
containing the  
attributes

# Sistema de Arquivos



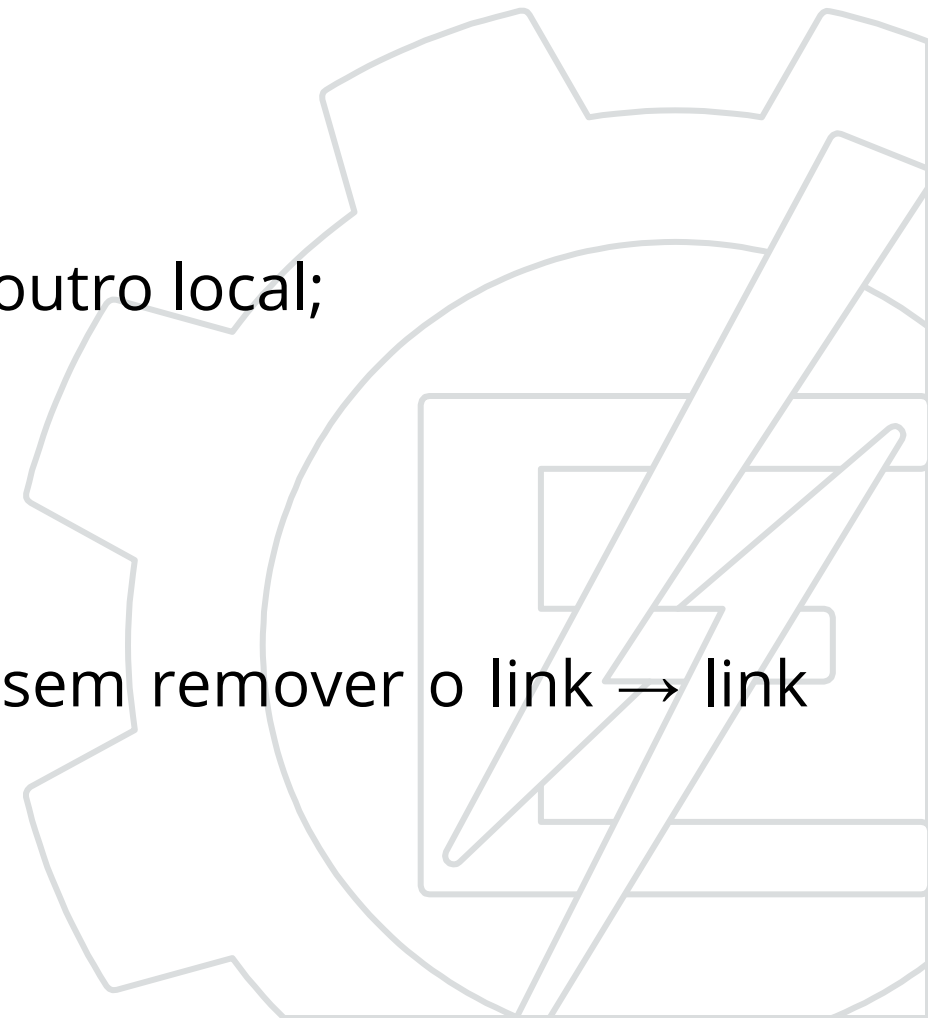
## Diretórios - *Links*:

- Permite que um arquivo apareça em mais de um diretório.
- ***Hard Link***:
  - Incrementa o contador de *links* no *i-node*;
  - Nenhum arquivo é realmente removido enquanto seu contador for maior que zero.
- ***Soft Link***:
  - Ponteiro para um arquivo em outro diretório;
  - Não incrementa o contador.



## Diretórios - *Links*:

- ***Soft Link***:
  - Vantagem:
    - Não precisa copiar o mesmo arquivo para outro local;
    - Economia de espaço.
  - Desvantagem:
    - Se ocorrer a remoção do arquivo original sem remover o link → link quebrado.

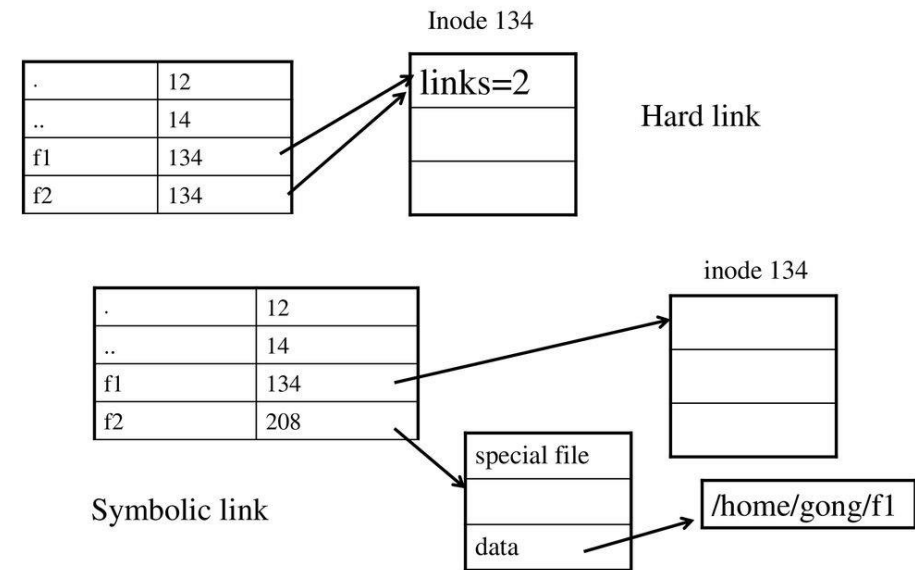


# Sistema de Arquivos

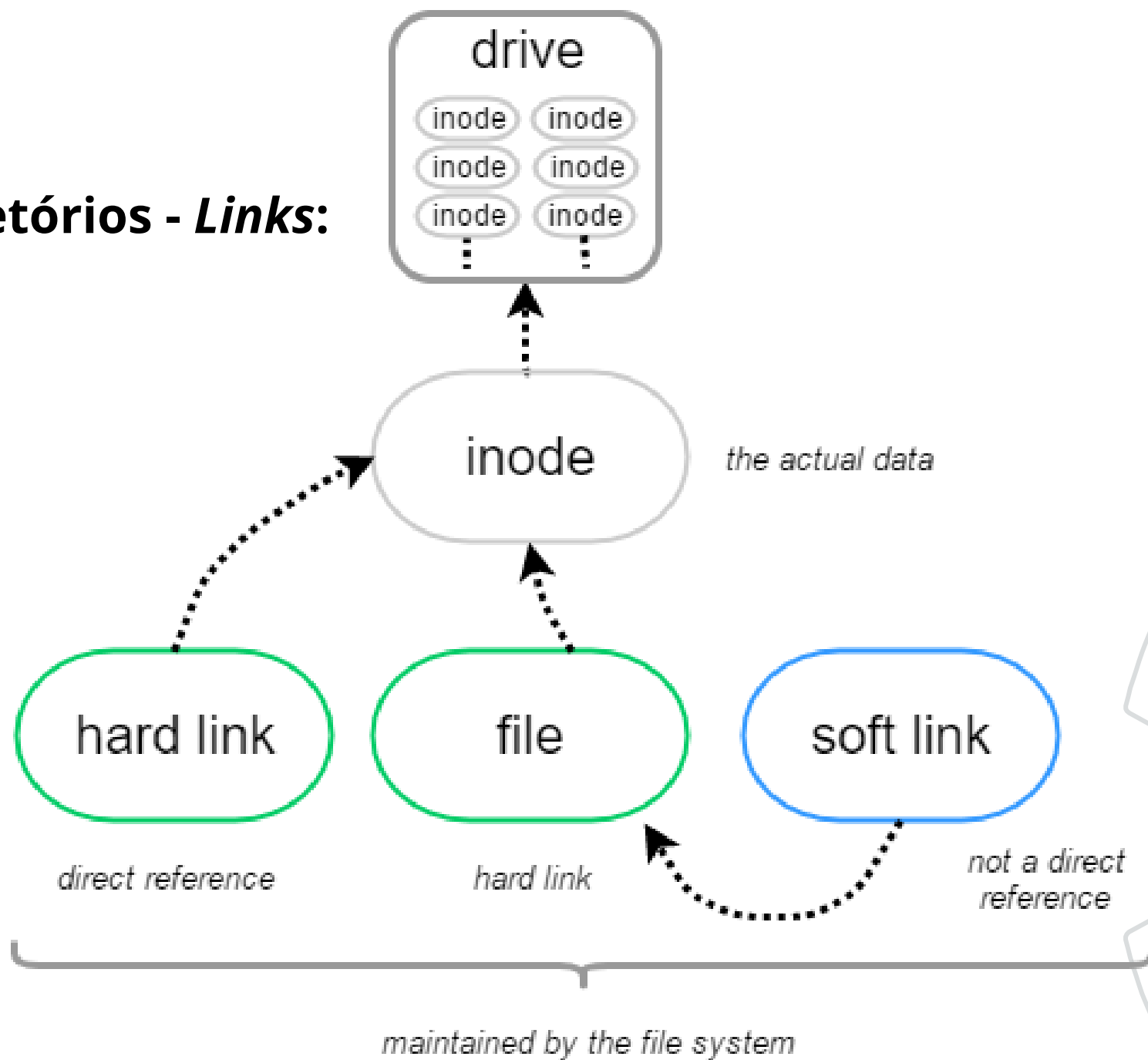
## Diretórios - *Links*:

- ***Unlink*** (*hard link*):
  - Remove uma entrada (arquivo) no diretório.
  - Decrementa o contador de *links* do arquivo.
  - Se após a remoção o contador for igual a zero, o arquivo correspondente estava somente neste diretório e será removido do sistema de arquivos.

### Hard links vs. Symbolic links



## Diretórios - *Links*:



# Bibliografia

- TANENBAUM, Andrew S; BOS, Herbert. Sistemas operacionais modernos. 4a ed. São Paulo: Pearson Education do Brasil, 2016.

## **Capítulo 4.**

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/1233>

- DEITEL, H.M; DEITEL, P.J; CHOFFNES,D.R. Sistemas Operacionais. 3a ed. São Paulo: Pearson Prentice Hall, 2005. **Capítulos 12-13.**

<https://plataforma.bvirtual.com.br/Acervo/Publicacao/315>





# Sistemas Operacionais

Prof. Otávio Gomes

[otavio.gomes@unifei.edu.br](mailto:otavio.gomes@unifei.edu.br)

