

MC514
Sistemas Operacionais:
Teoria e Prática
1s2008

Sistemas de Arquivos

Sistemas de Arquivos

- Grande quantidade de informação
- Dados persistentes (não-voláteis)
- Acesso concorrente

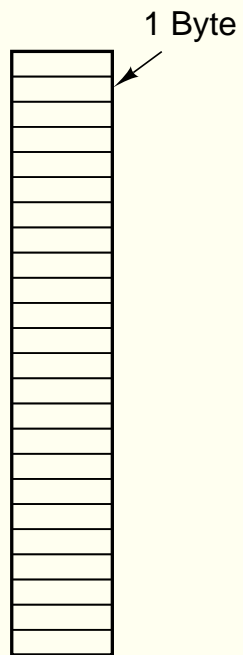
Nomes e extensões

Extension	Meaning
file.bak	Backup file
file.c	C source program
file.gif	Compuserve Graphical Interchange Format image
file.hlp	Help file
file.html	World Wide Web HyperText Markup Language document
file.jpg	Still picture encoded with the JPEG standard
file.mp3	Music encoded in MPEG layer 3 audio format
file.mpg	Movie encoded with the MPEG standard
file.o	Object file (compiler output, not yet linked)
file.pdf	Portable Document Format file
file.ps	PostScript file
file.tex	Input for the TEX formatting program
file.txt	General text file
file.zip	Compressed archive

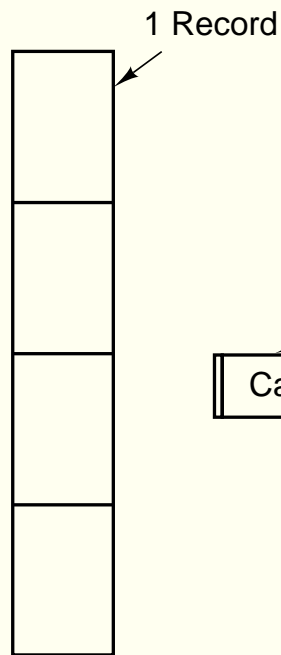
Arquivos podem ter mais de uma extensão: file.ps.gz

Comando **file** verifica o tipo dos arquivos

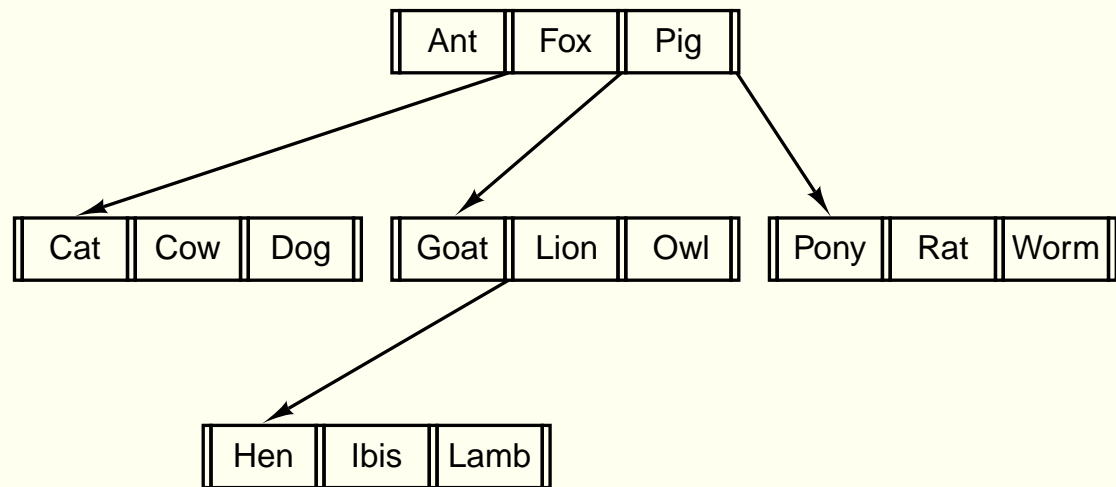
Estruturas de arquivos



(a)



(b)



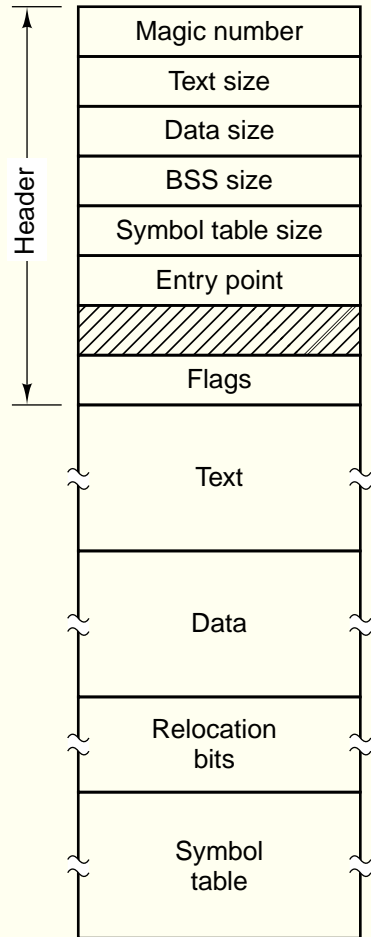
(c)

Tipos de arquivos

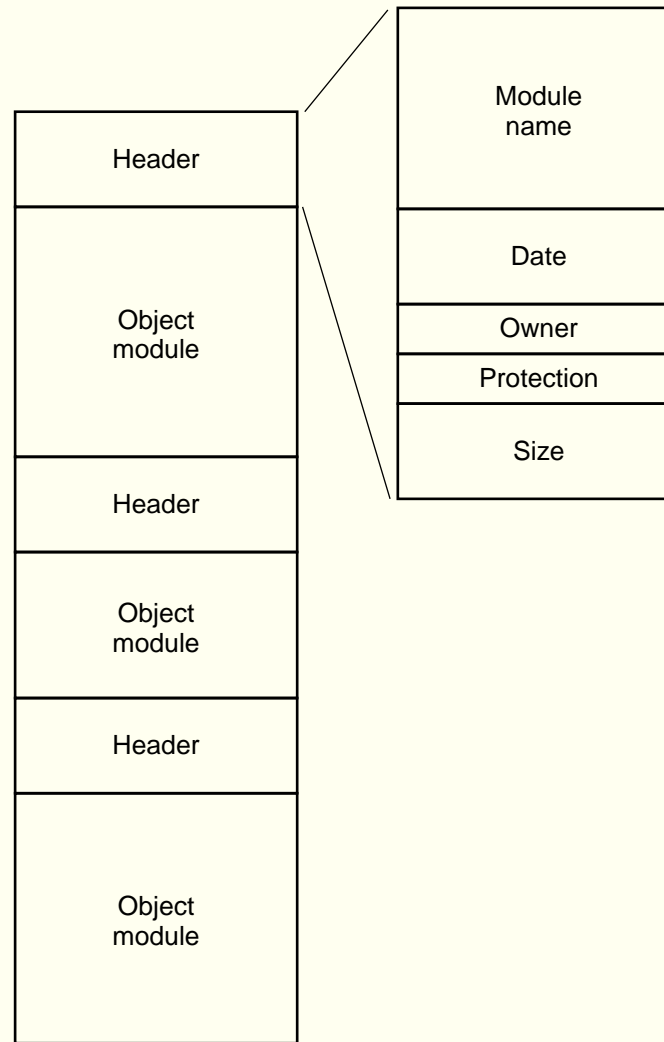
- regular
- diretório
- caracter
 - terminais, impressoras e rede
- bloco
 - discos

Use o comando stat

Exemplos: executável e archive



(a)



(b)

Acesso a arquivos

- Seqüencial
 - Lê todos os bytes a partir do início
 - Fitas magnéticas
- Aleatório
 - Bytes podem ser lidos em qualquer ordem
 - Bancos de dados

Atributos de arquivos

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

Veja os comandos stat e make

Operações sobre arquivos

- create
- delete
- open
- close
- read
- write
- append
- seek
- get attributes
- set attributes
- rename

Veja `linux-2.6.24.2/include/linux/fs.h`

Programa copy

```
#define BUF_SIZE 4096
#define OUTPUT_MODE 0700

int main(int argc, char *argv[]) {
    int in_fd, out_fd, rd_count, wt_count;
    char buffer[BUF_SIZE];

    if (argc!=3) exit(1);

    in_fd = open(argv[1], O_RDONLY);
    if (in_fd < 0) exit(2);

    out_fd = creat(argv[2], OUTPUT_MODE);
    if (out_fd < 0) exit(3);
```

Programa copy

```
while((rd_count = read(in_fd, buffer, BUF_SIZE)) > 0) {  
    wt_count = write(out_fd, buffer, rd_count);  
    if (wt_count <= 0) exit(4);  
}
```

```
close(in_fd);  
close(out_fd);
```

```
if (rd_count == 0) exit(0);  
else exit(5);  
}
```

Streams

```
int fprintf(FILE *stream, const char *format, ...);
```

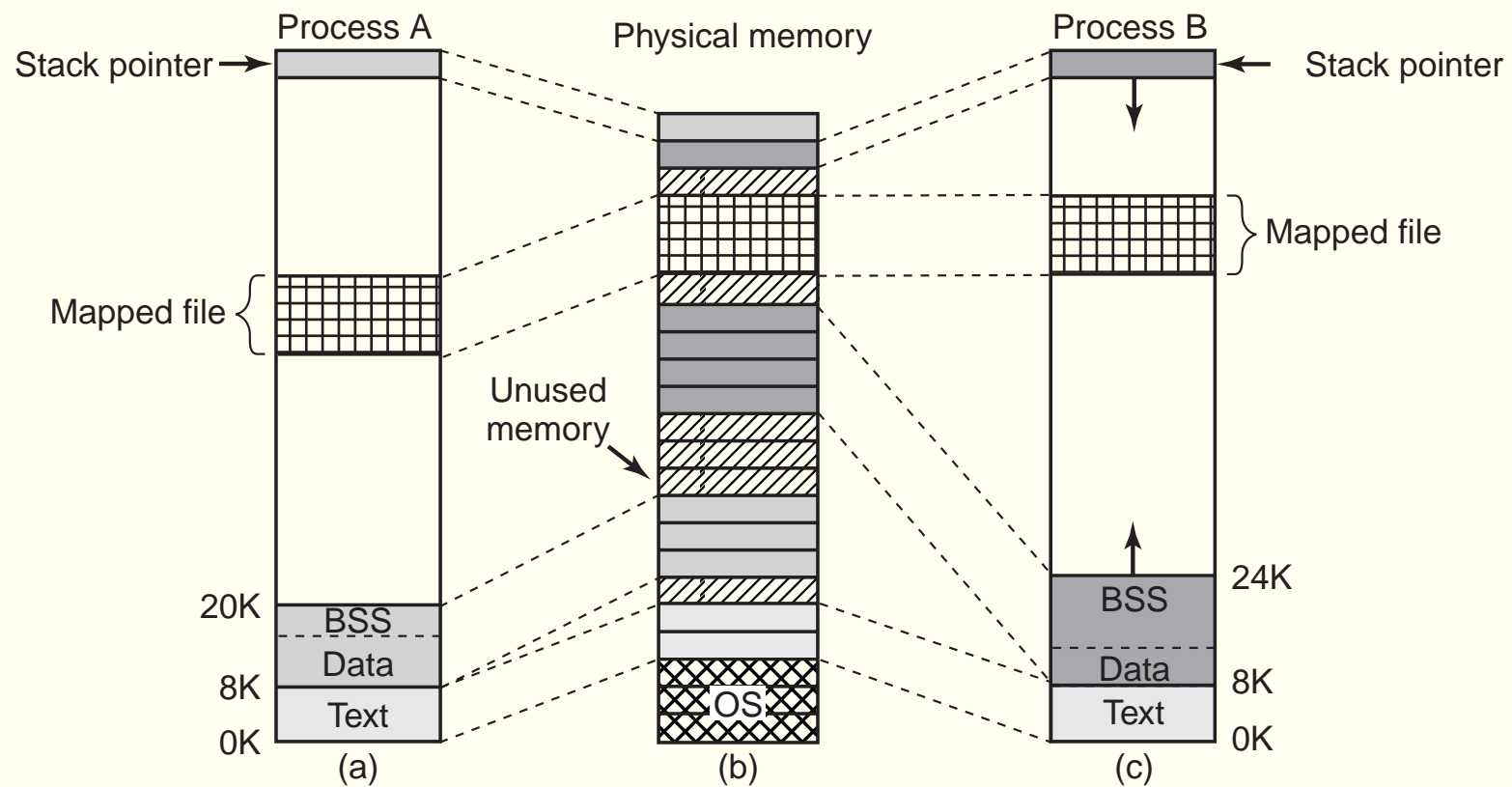
```
int fscanf(FILE *stream, const char *format, ...);
```

```
FILE *fopen(const char *path, const char *mode);
```

```
int fclose(FILE *stream);
```

Veja os exemplos `fscanf.c` e `fscanf2.c`

Arquivos mapeados em memória



Mmap

```
void *mmap(void *start, size_t length, int prot,  
int flags, int fd, off_t offset);
```

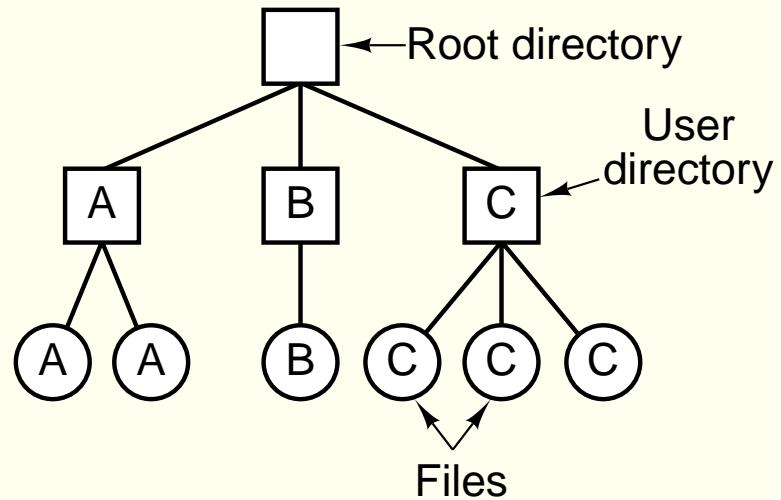
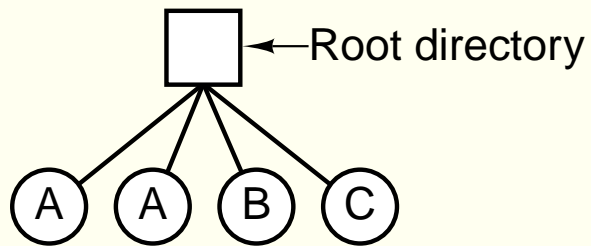
- start: endereço preferencial na memória onde o arquivo pode ficar; NULL caso o sistema deva escolher.
- length: tamanho do mapeamento
- prot: PROT_EXEC, PROT_READ, PROT_WRITE, PROT_NONE
- flags: MAP_SHARED ou MAP_PRIVATE
- fd: file descriptor do arquivo a ser mapeado
- offset: deslocamento em relação ao início do arquivo

Outros exemplos com mmap

- `map-size.c`: carrega um arquivo em memória e o escreva na saída padrão.
- `map-armadilha.c`: copia o código de uma função para o arquivo mapeado
- `map-loop.c`: executa trecho de código na área do arquivo mapeado.

Diretórios

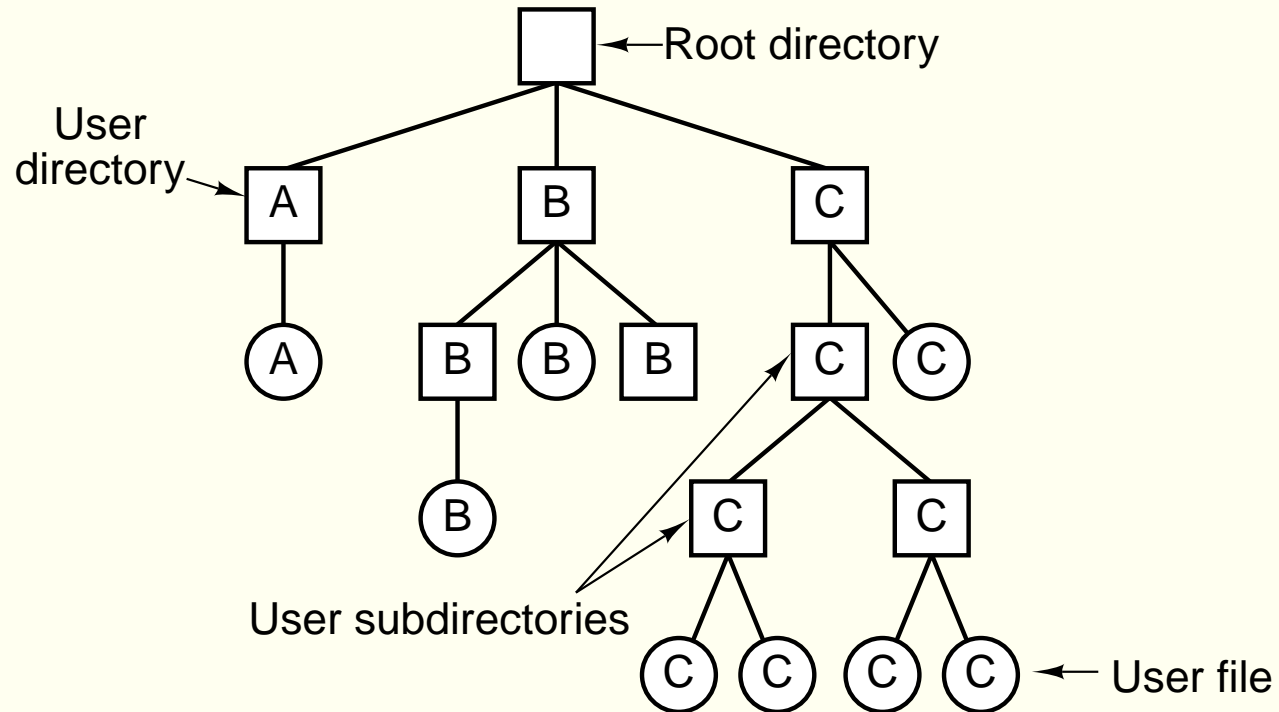
Nível único e dois níveis



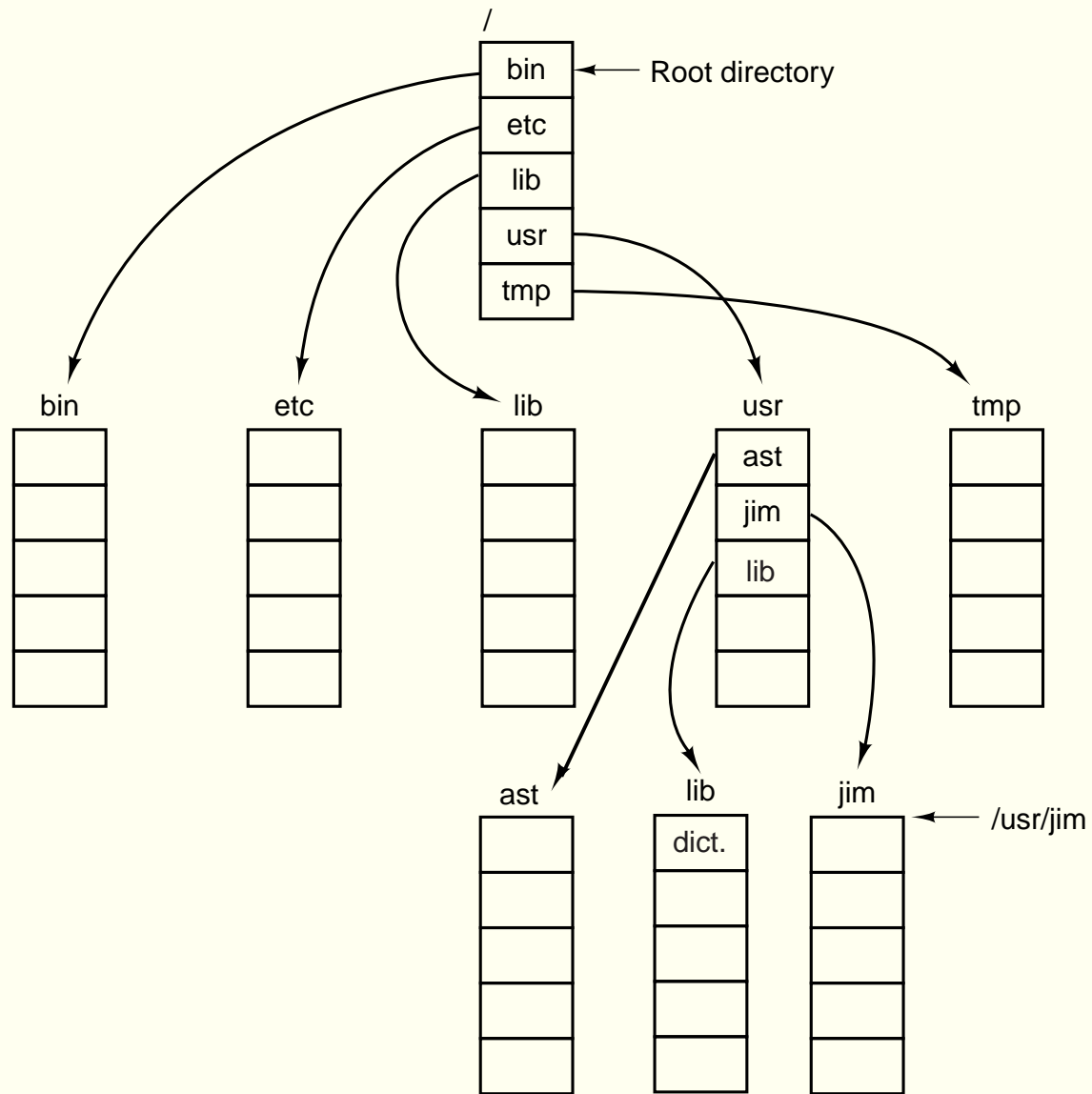
Daria para viver assim?

Diretórios

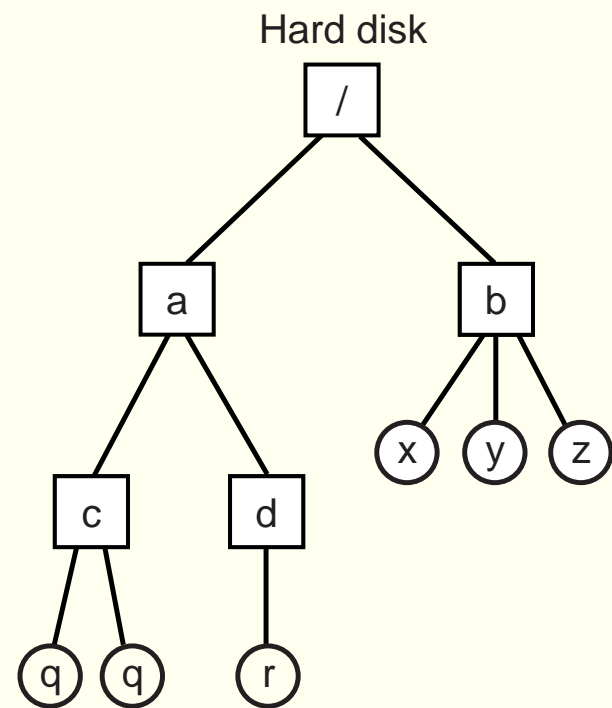
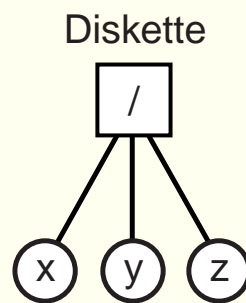
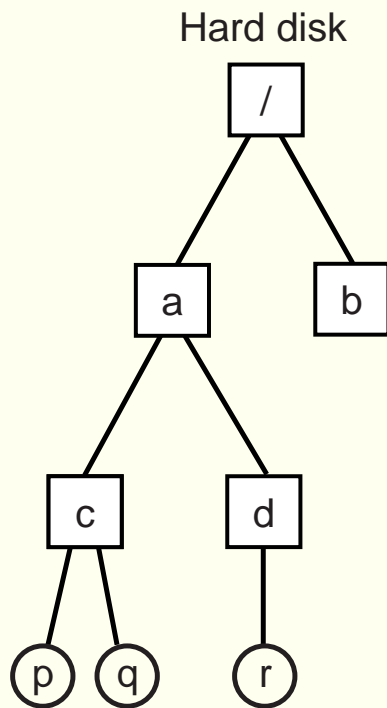
Estrutura hierárquica



Caminhos



Mount

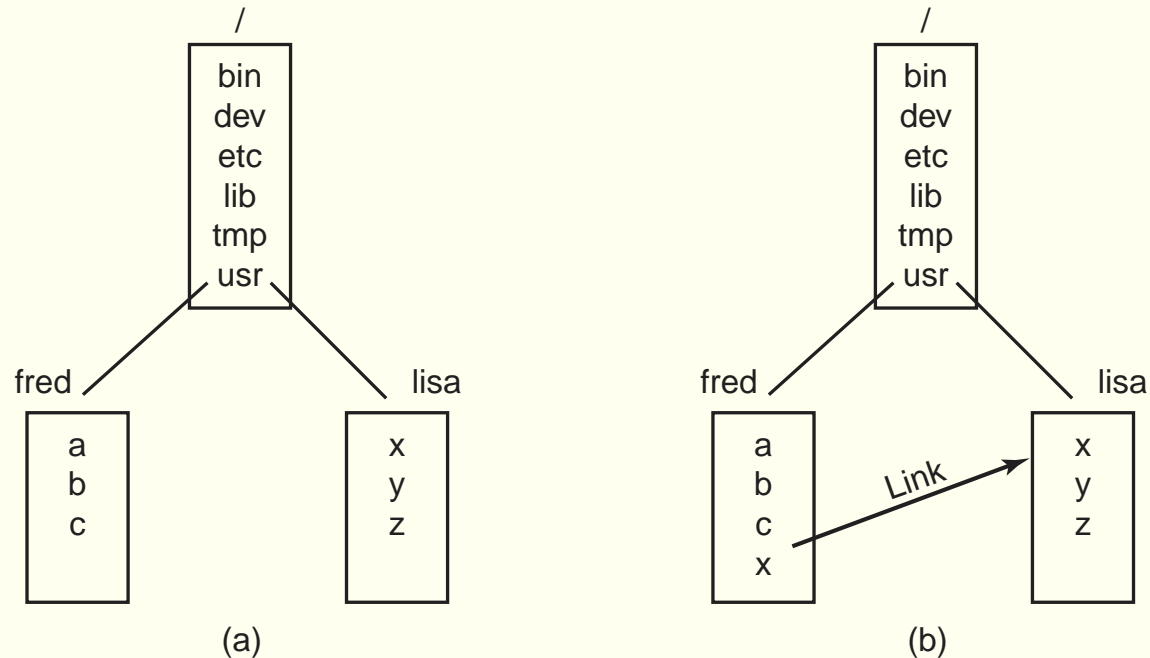


Operações sobre diretórios

- create
- delete
- opendir
- closedir
- readdir
- rename
- link
- unlink

Veja o código `dir.c`

Links

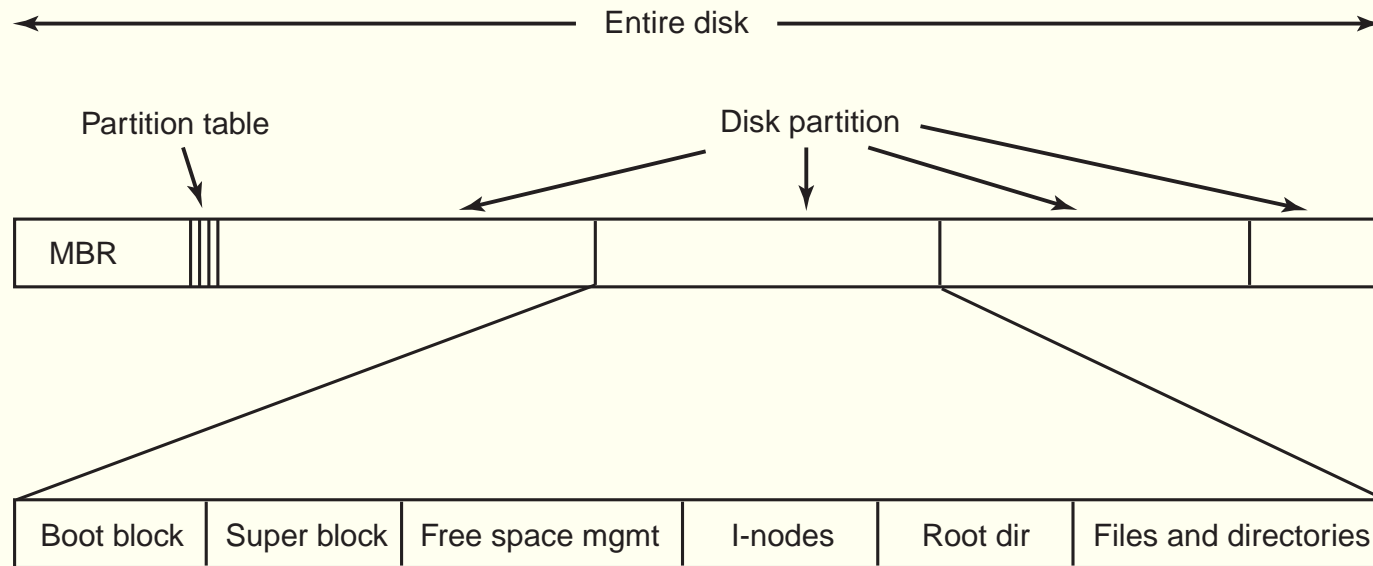


- links simbólicos ou hard links
- caminhos absolutos ou relativos
- como copiar?

Questões de Implementação

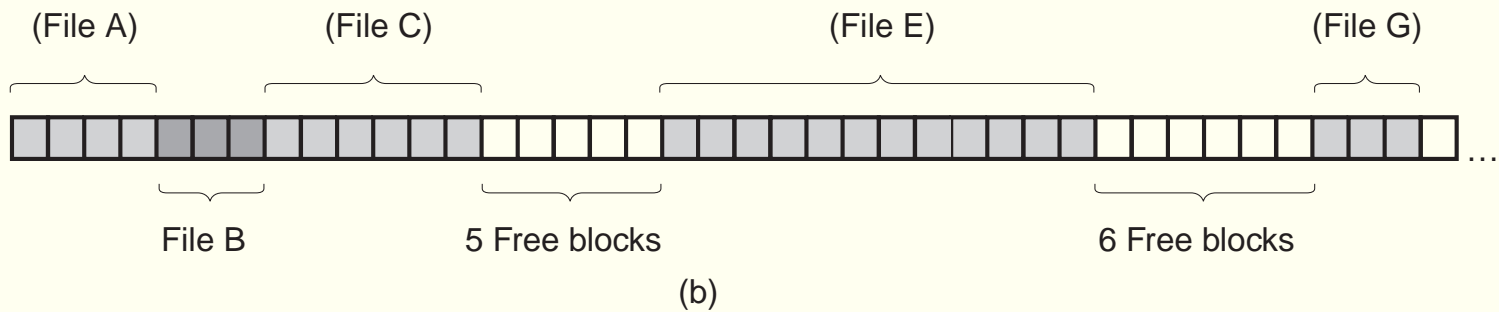
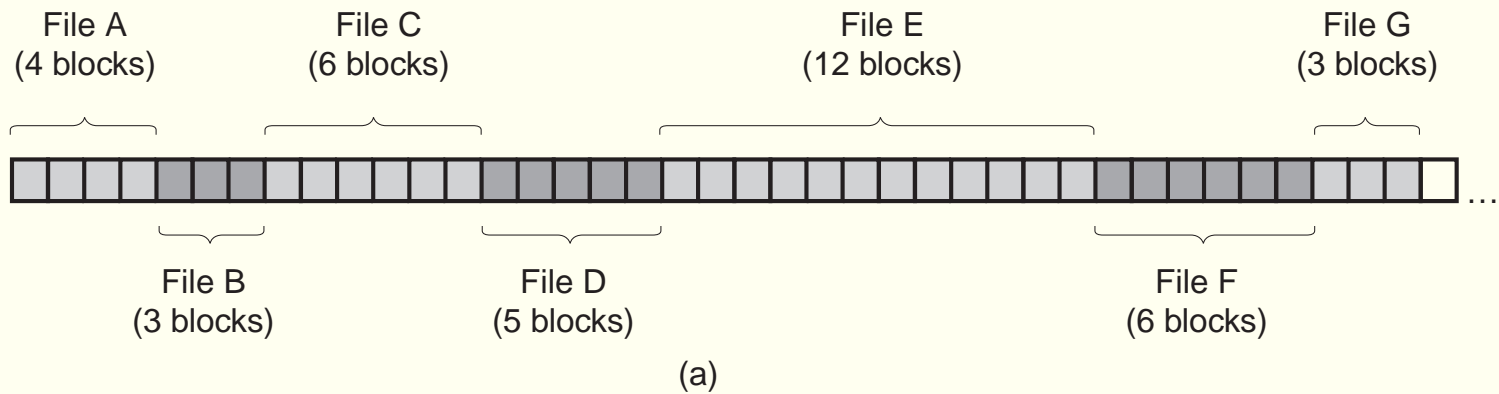
- Como os arquivos são armazenados
- Como o espaço livre é gerenciado
- Eficiência
- Confiabilidade

Layout

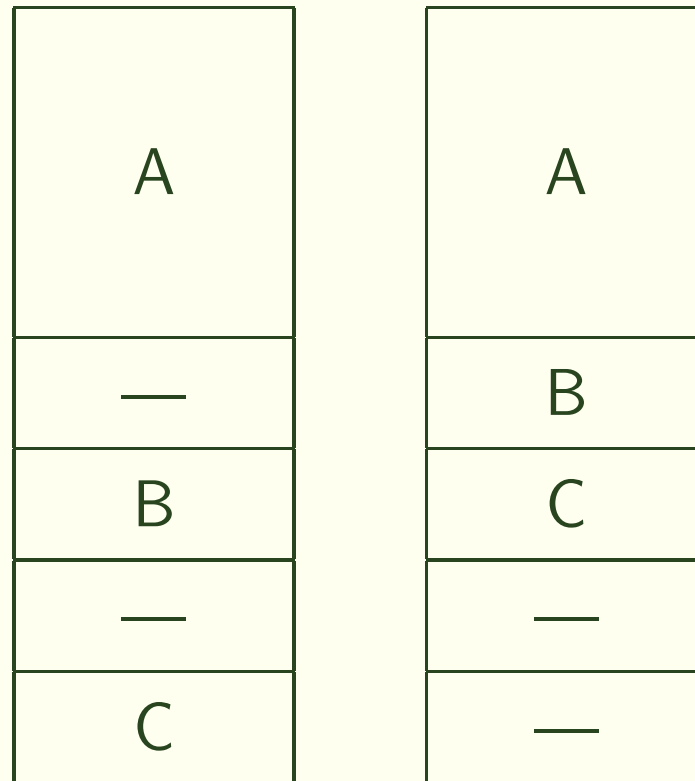


- MBR (Master Boot Record)
- Tabela de partições
- Boot block

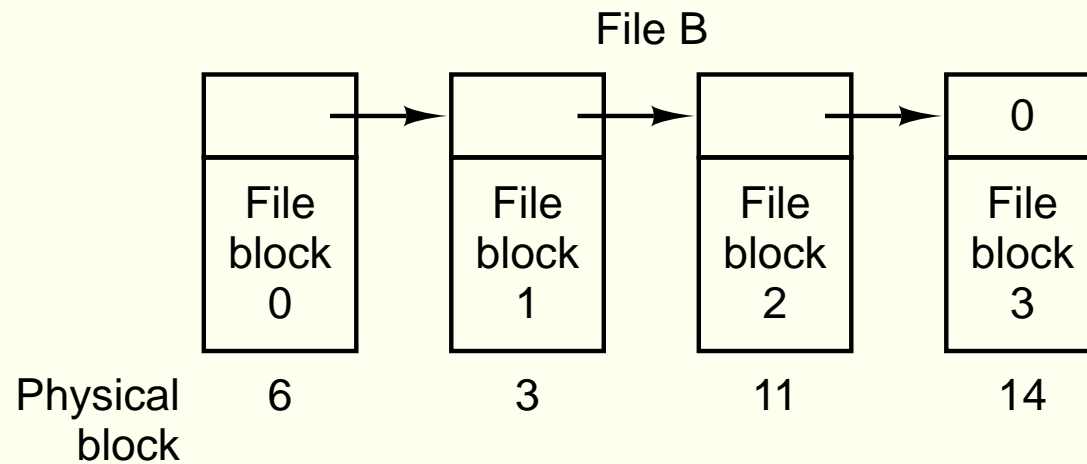
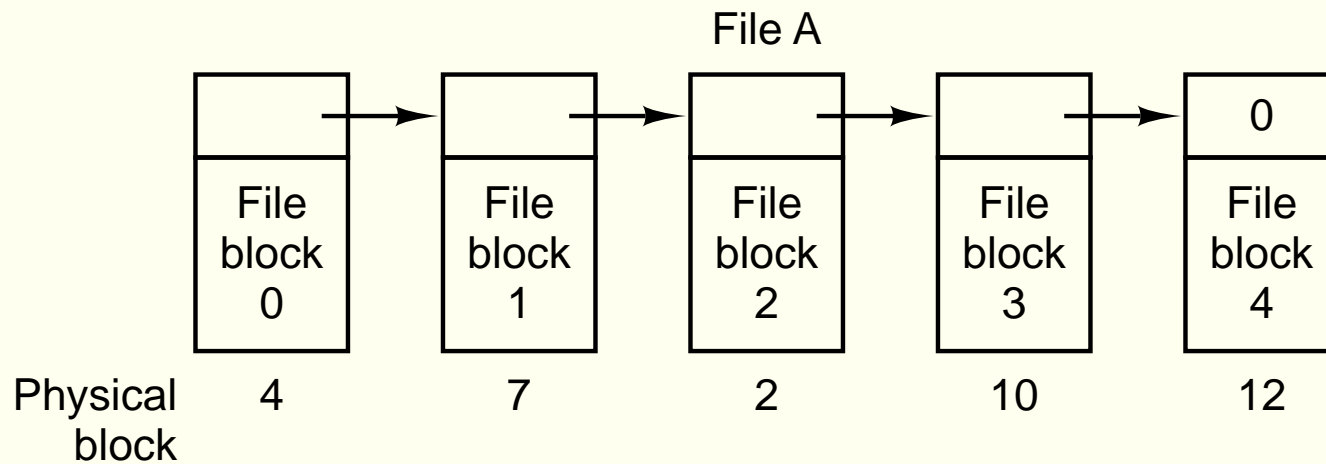
Alocação contínua



Compactação do disco



Lista ligada de blocos



File Allocation Table (FAT)

Physical
block

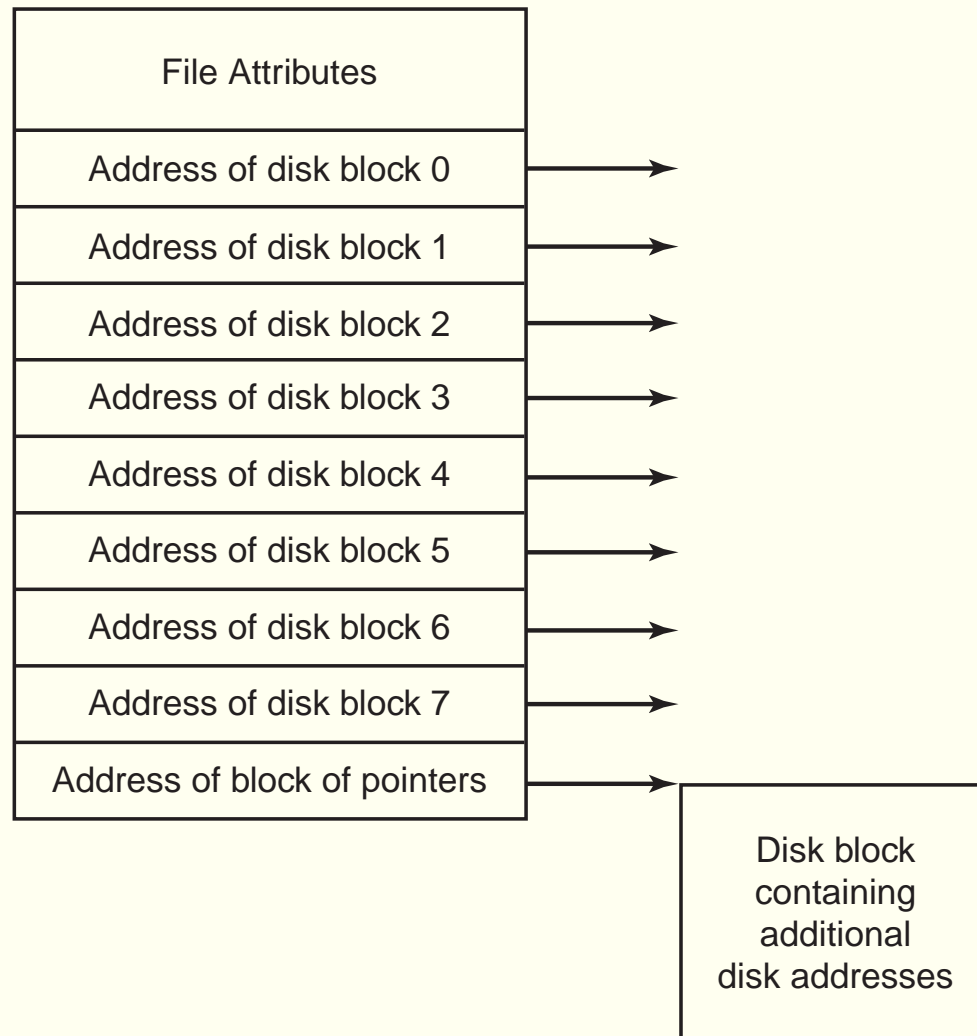
0	
1	
2	10
3	11
4	7
5	
6	3
7	2
8	
9	
10	12
11	14
12	-1
13	
14	-1
15	

← File A starts here

← File B starts here

← Unused block

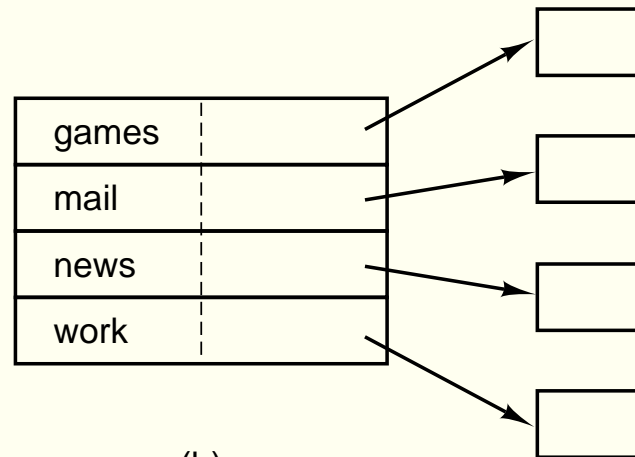
I-node



Implementação de diretórios

games	attributes
mail	attributes
news	attributes
work	attributes

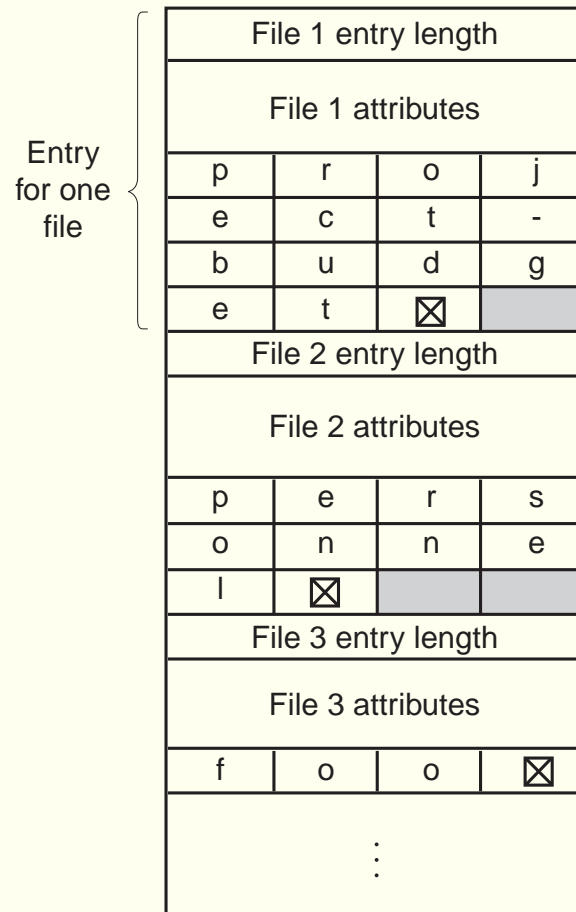
(a)



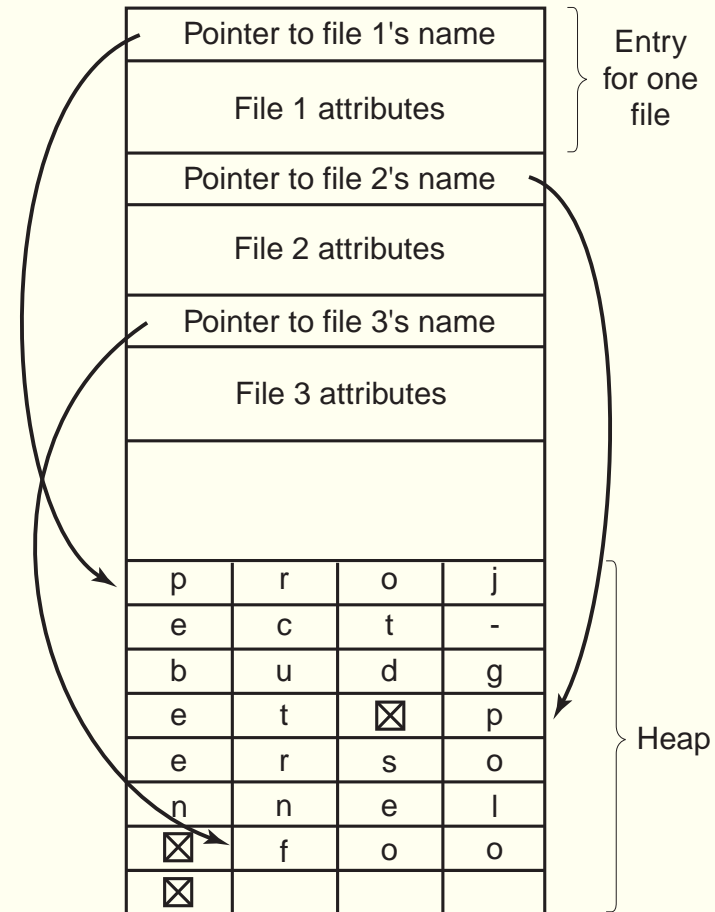
(b)

Data structure
containing the
attributes

Nomes de tamanho variável



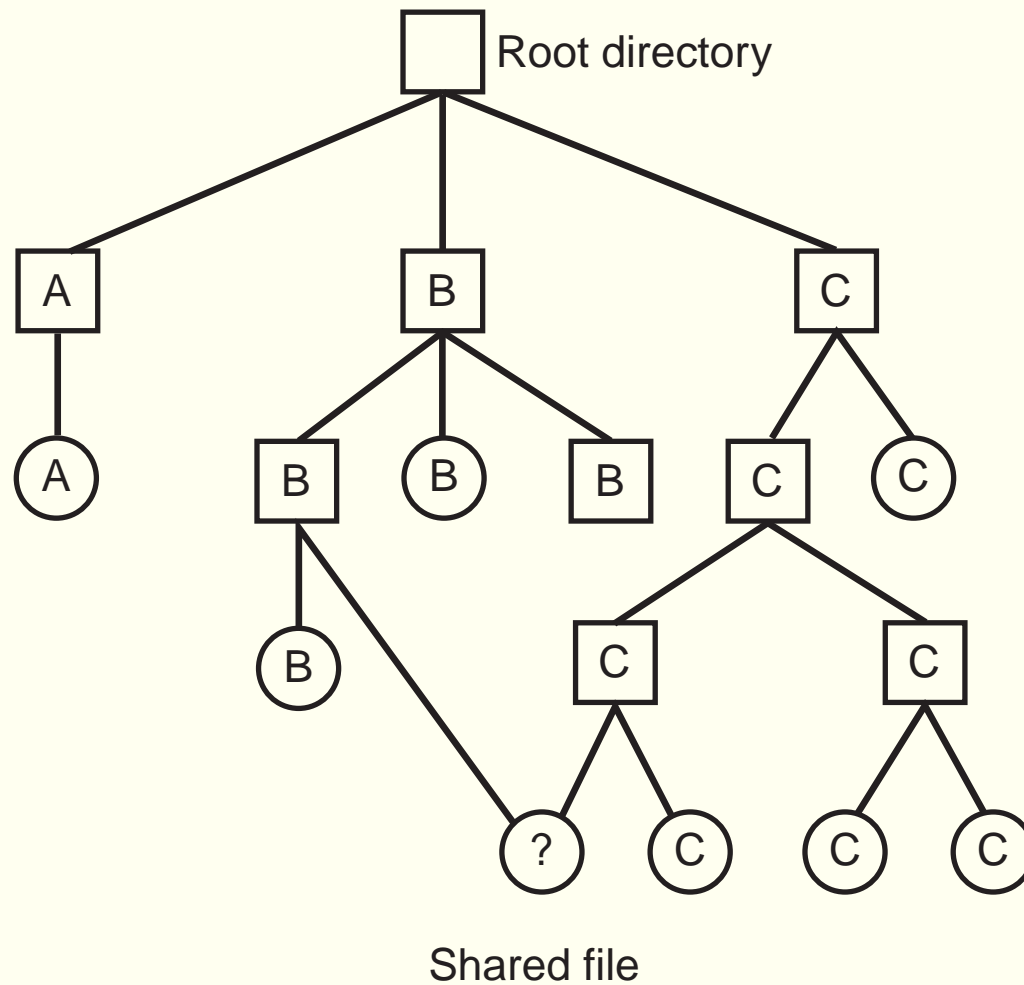
(a)



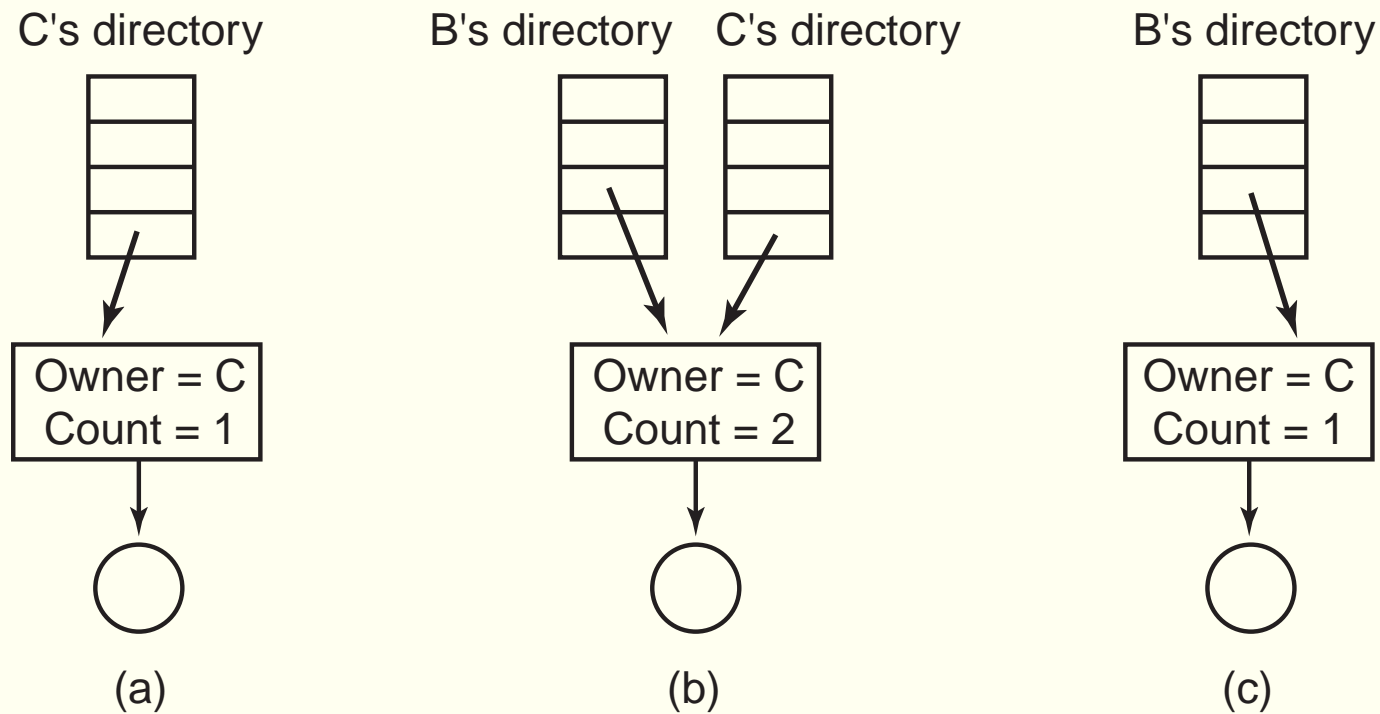
(b)

Execute man readdir

Arquivos compartilhados

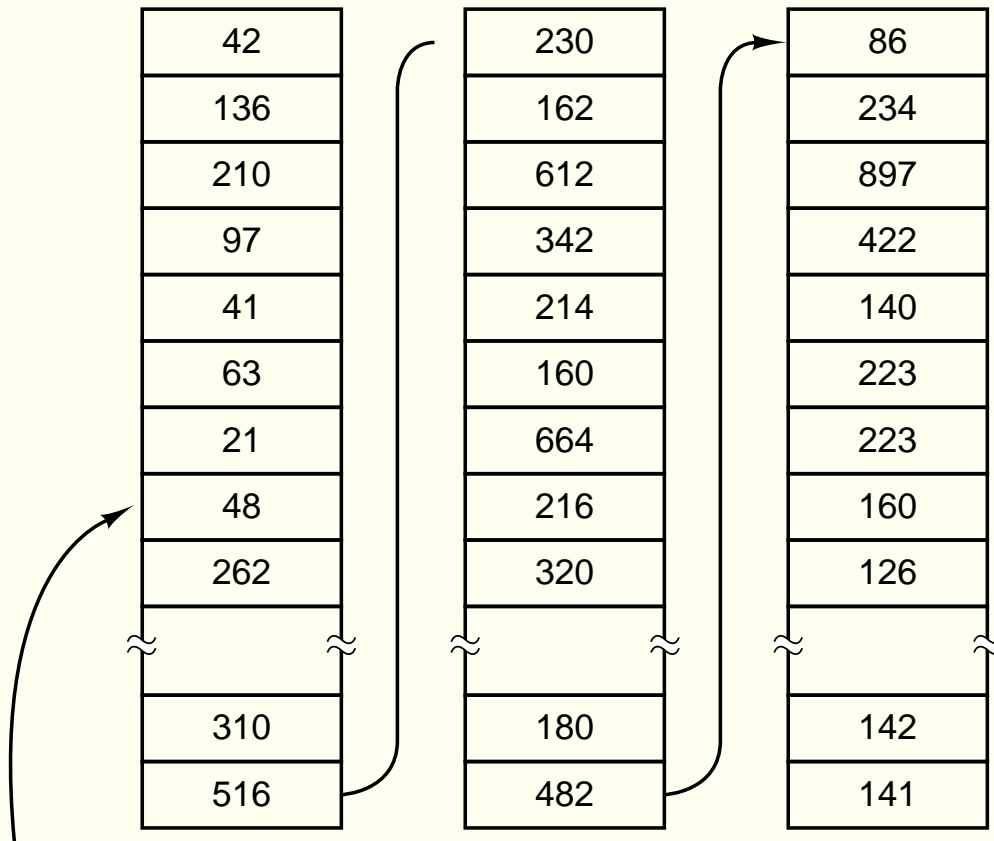


Arquivos compartilhados



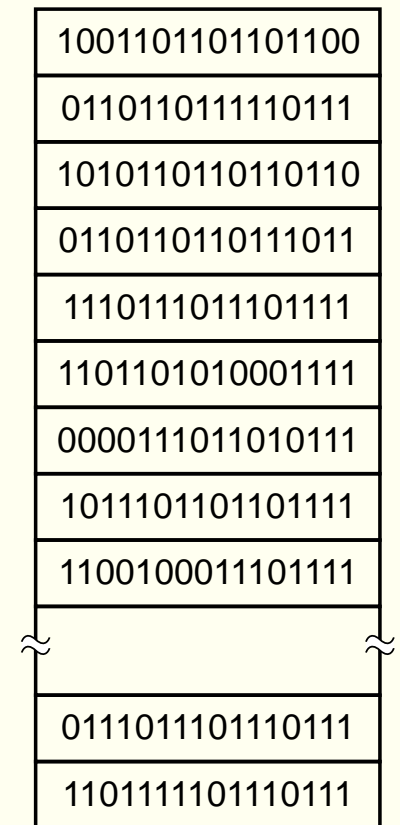
Lista de livres e bitmaps

Free disk blocks: 16, 17, 18



A 1-KB disk block can hold 256
32-bit disk block numbers

(a)

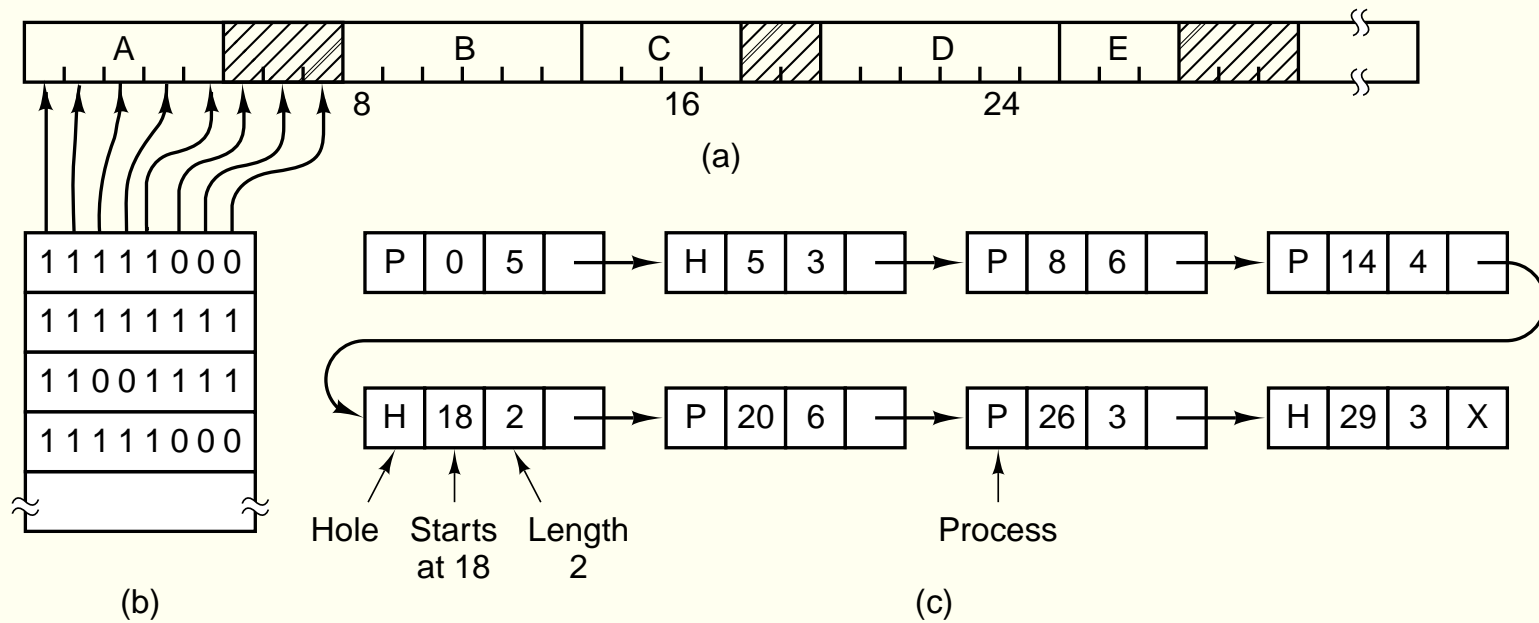


A bitmap

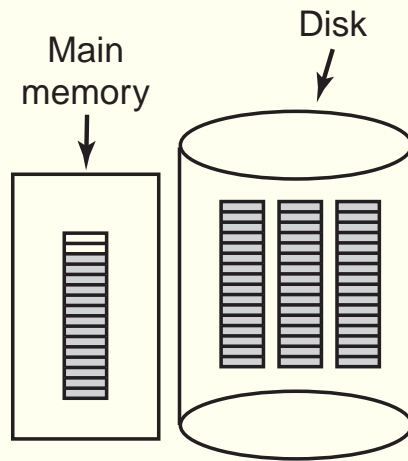
(b)

Bitmaps e lista de livres

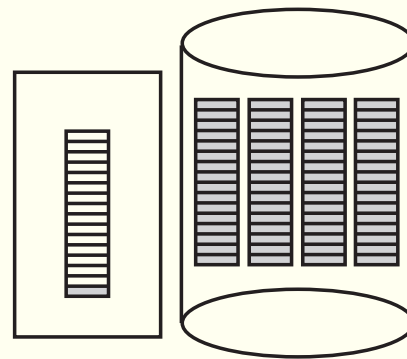
Gerência de memória



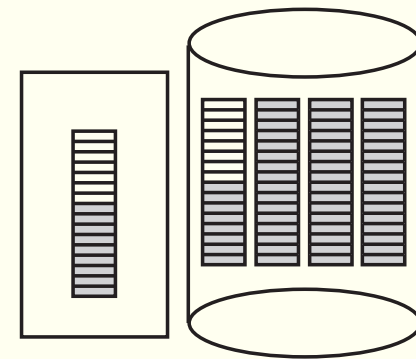
Lista de livres em memória



(a)

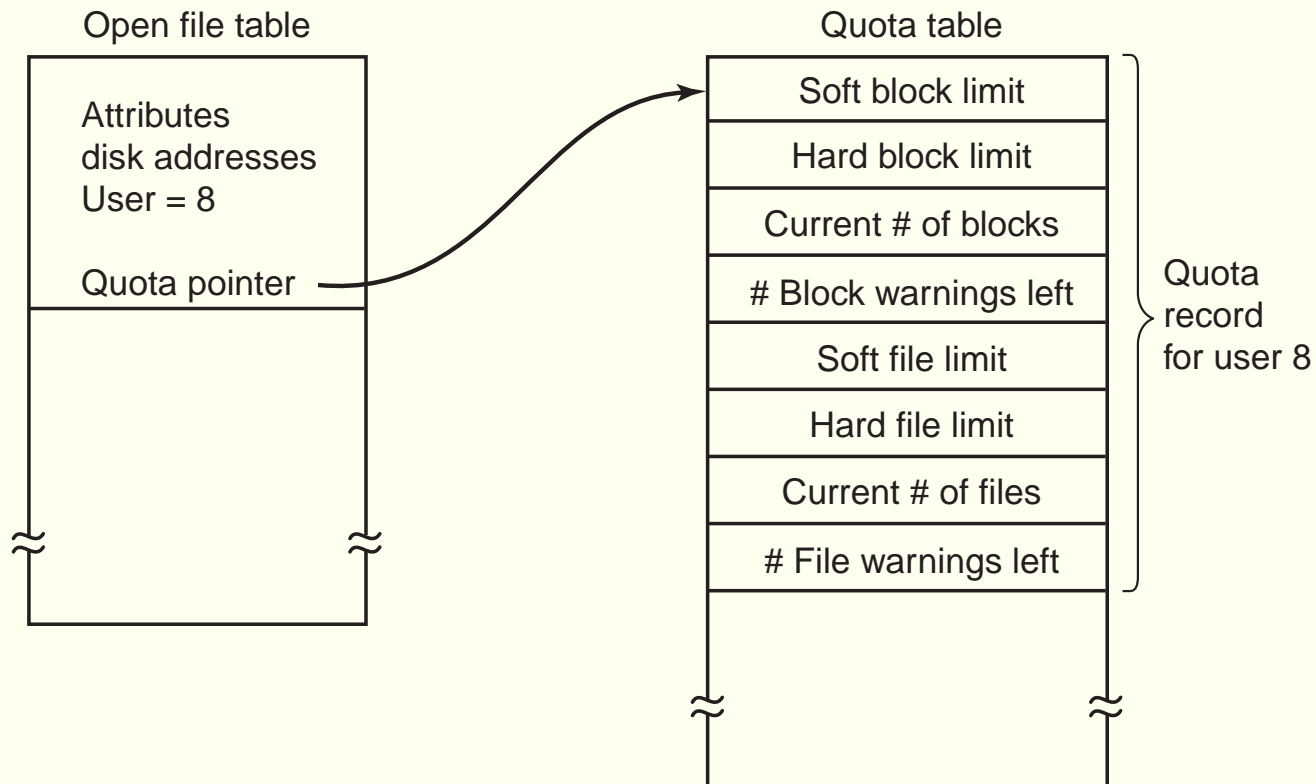


(b)



(c)

Gerência de quotas



Consistência do sistema de arquivos

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(a)

(a) consistente

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	2	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(c)

(c) duplicação na
lista de livres

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	1	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	0	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(b)

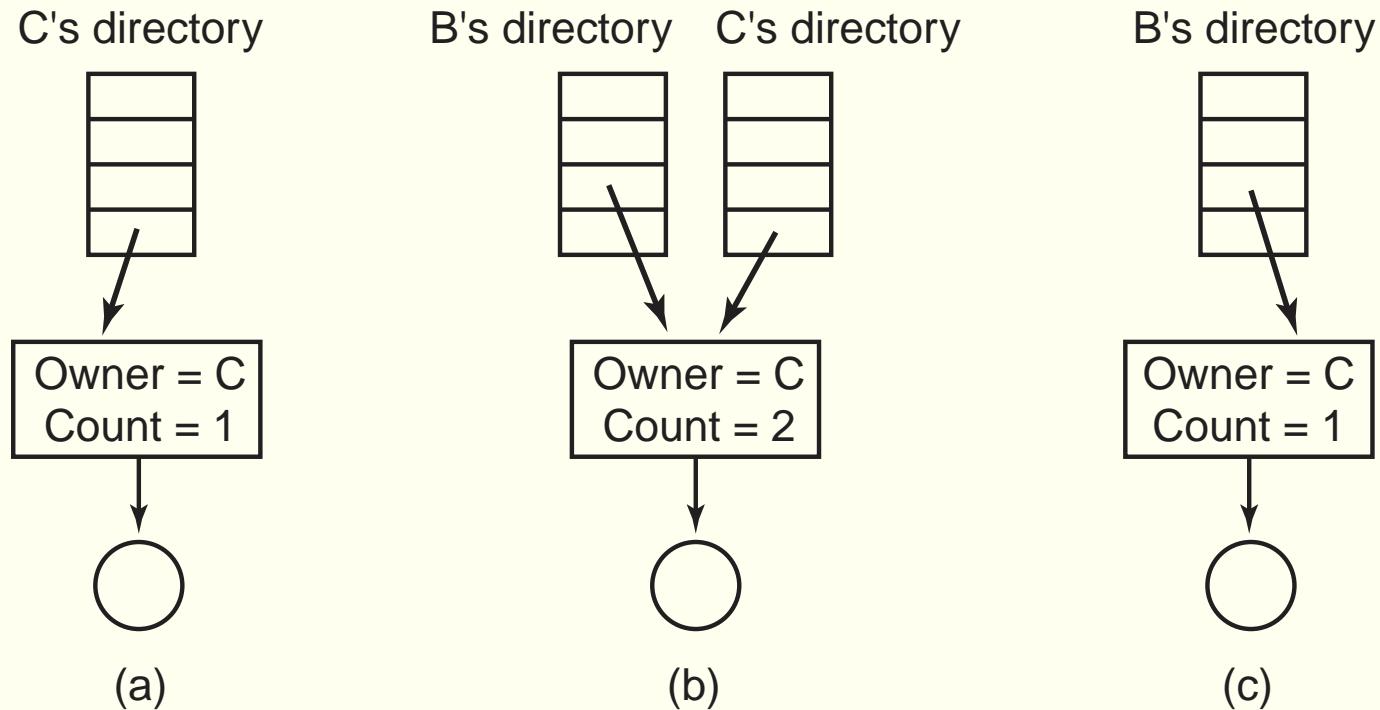
(b) bloco faltando

Block number															
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	0	1	0	2	1	1	1	0	0	1	1	1	0	0
Blocks in use															
0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	1
Free blocks															

(d)

(d) duplicação nos
dados

Consistência do sistema de arquivos



Verificar se todos os contadores estão corretos

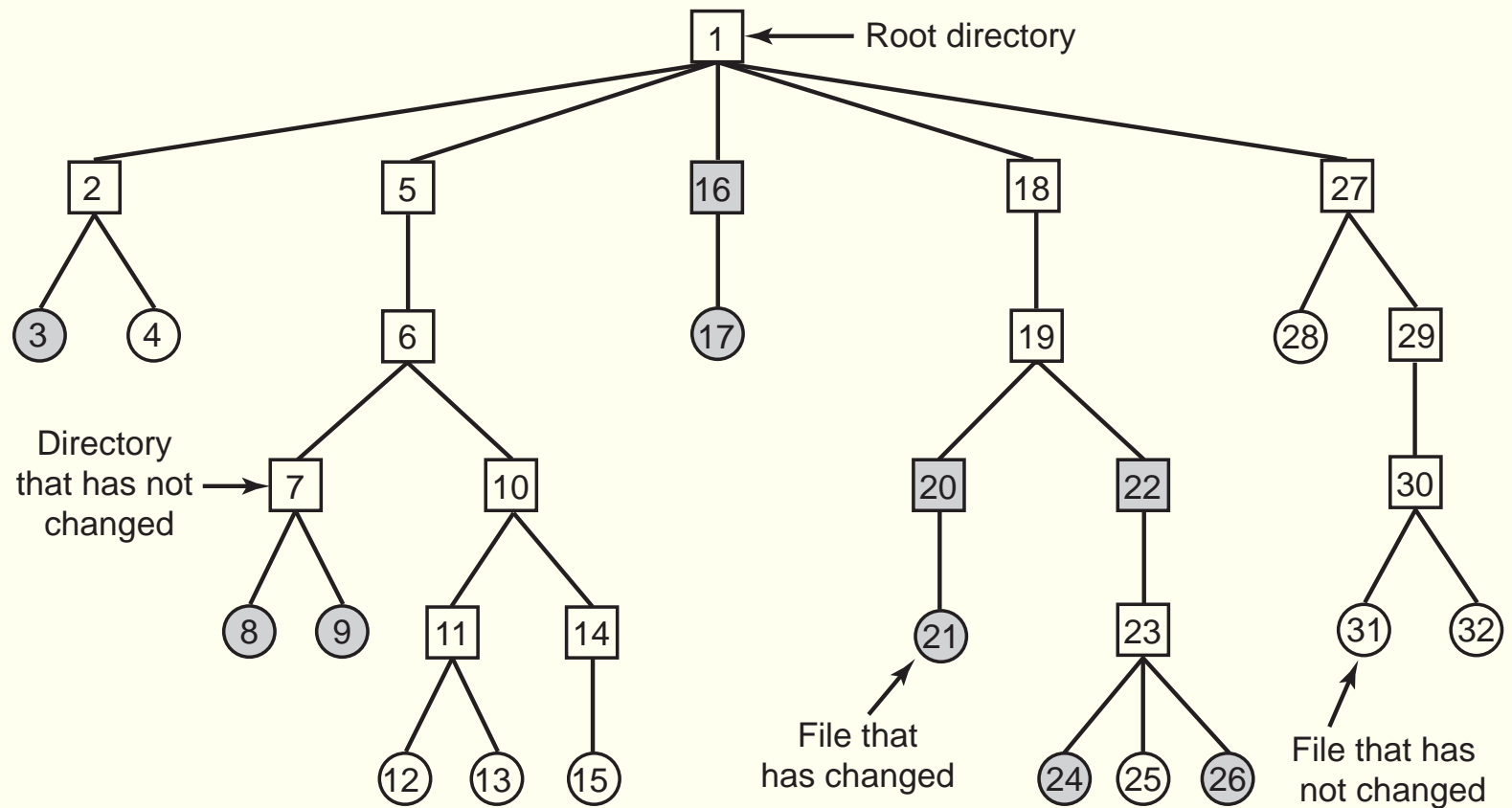
Cópias de segurança

- Dump físico
 - Cópia “total” disco
 - Simples e rápida
 - Blocos livre são copiados?
 - Gerência de blocos defeituosos

Cópias de segurança

- Cópias lógicas
- O que não copiar?
 - Arquivos de instalação do sistema
 - Arquivos /dev
 - Arquivos temporários
- Cópias incrementais

Cópia incremental



Cópia incremental

- Mapa de bits representando i-nodes
- Fase 1: marca todos os arquivos modificados e todos os diretórios.
- Fase 2: desmarca todos os diretórios sem arquivos ou sub-diretórios modificados
- Fase 3: varre os i-nodes e copia os diretórios e seus atributos
- Fase 4: copia os arquivos

Mapas de i-nodes e fases

(a)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

(b)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

(c)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

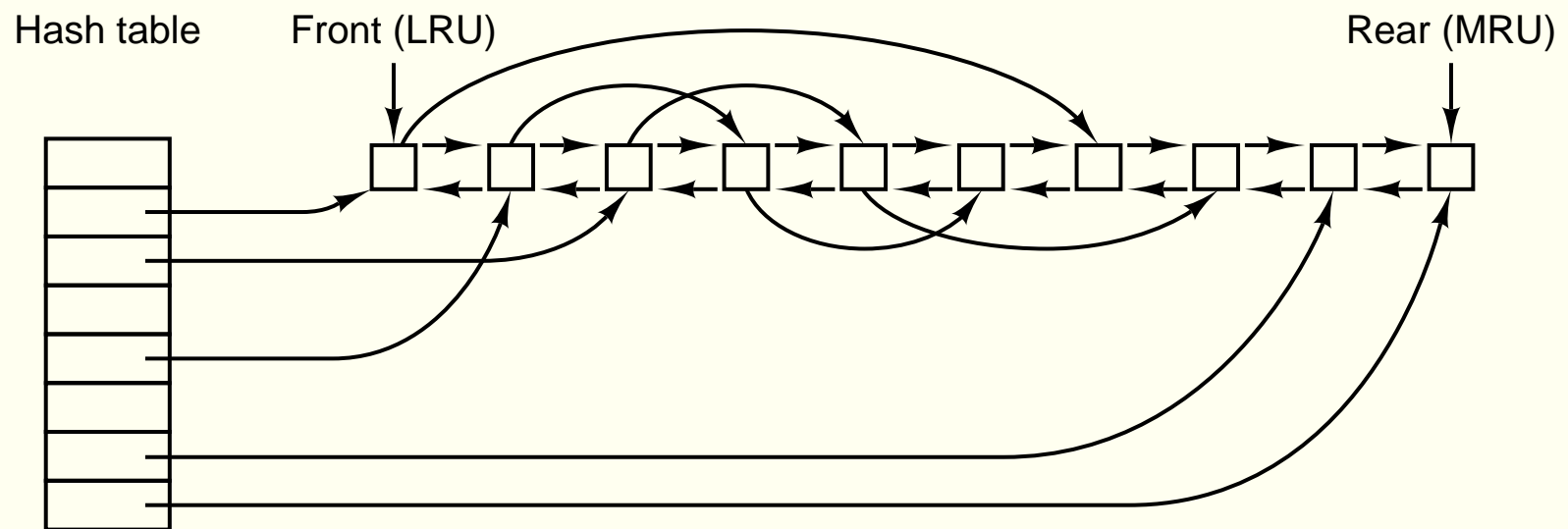
(d)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Restauração de arquivos

- Cria-se um sistema de arquivos vazio
- Cópia completa mais antiga é restaurada
- Cópias incrementais são restauradas
- Complicações
 - Hard links
 - Arquivos com lacunas (e.g., core)

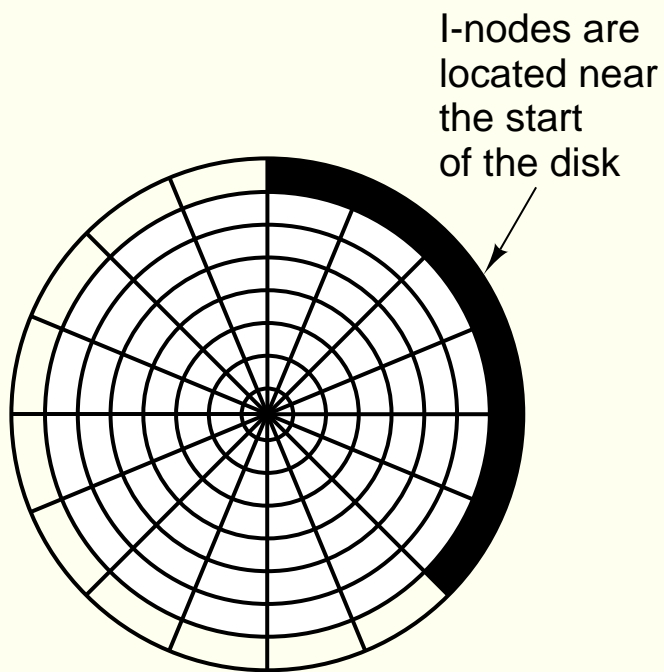
Caching



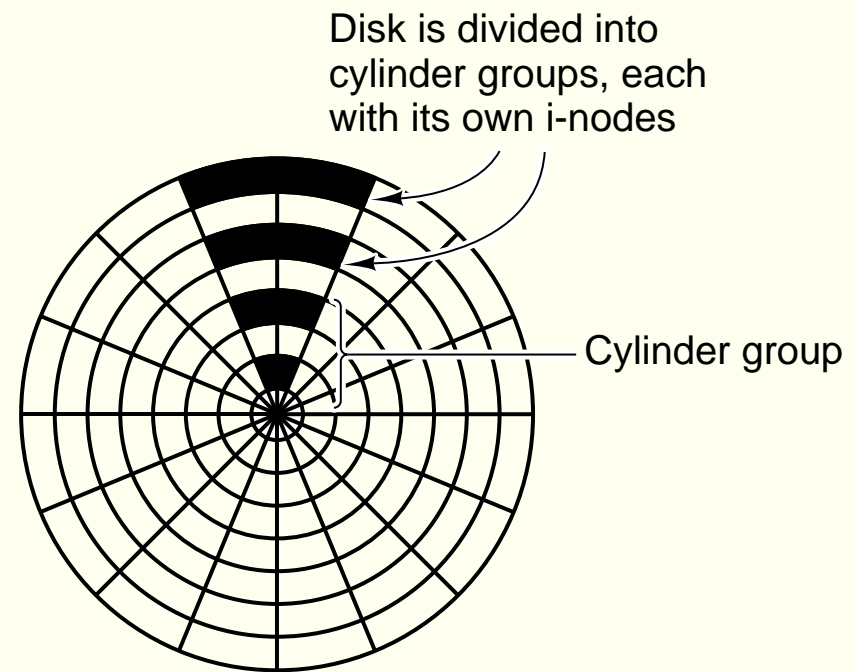
Block Read Ahead

- Lê um bloco antes de ele ser solicitado
- Acesso seqüencial
- Acesso aleatório

Distribuição da informação no disco



(a)



(b)