MC514

Sistemas Operacionais:

Teoria e Prática

1s2006

Sistema de Arquivos

Sistema de Arquivos

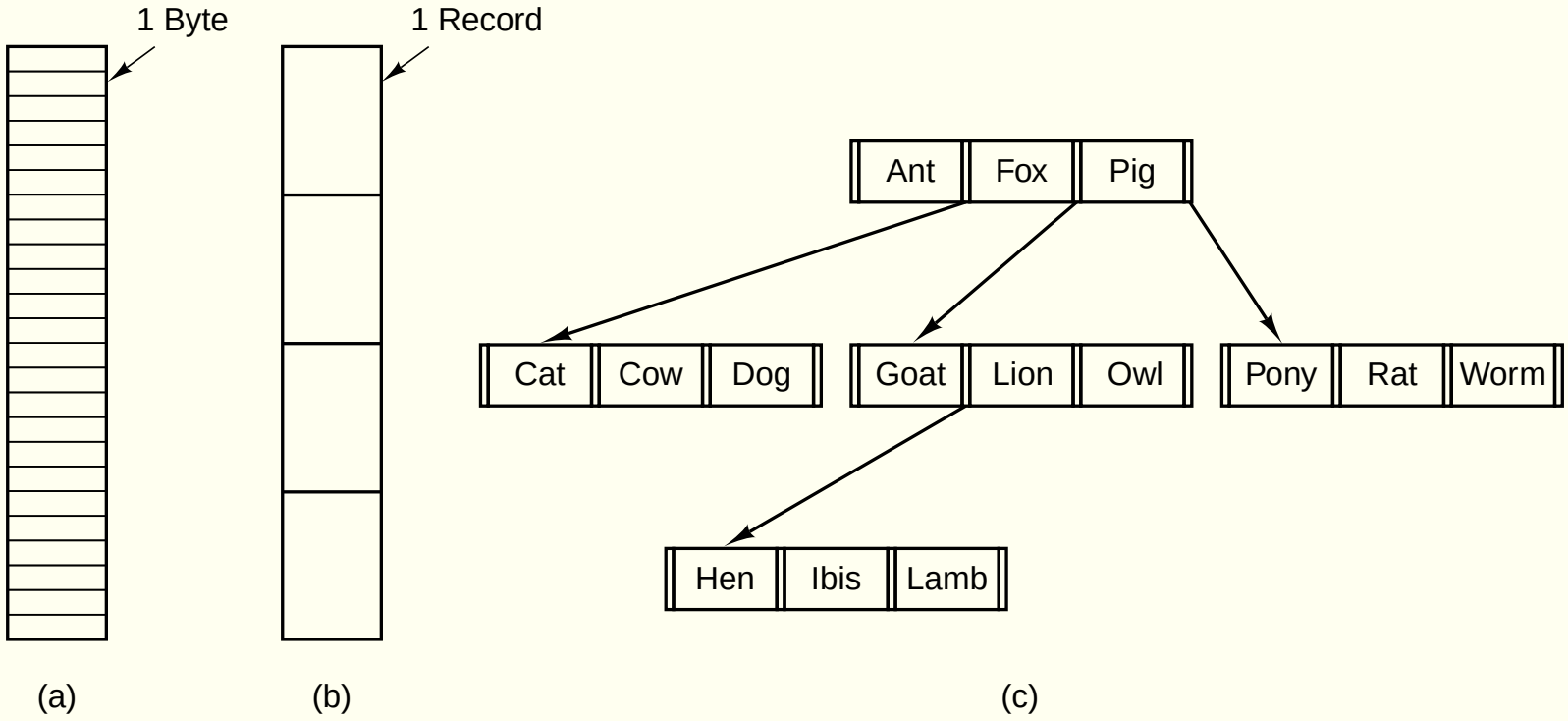
- Armazenam grande quantidade de informação
- Dados deve ser persistentes (não-voláteis)
- Acessso à informação pode ser concorrente

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

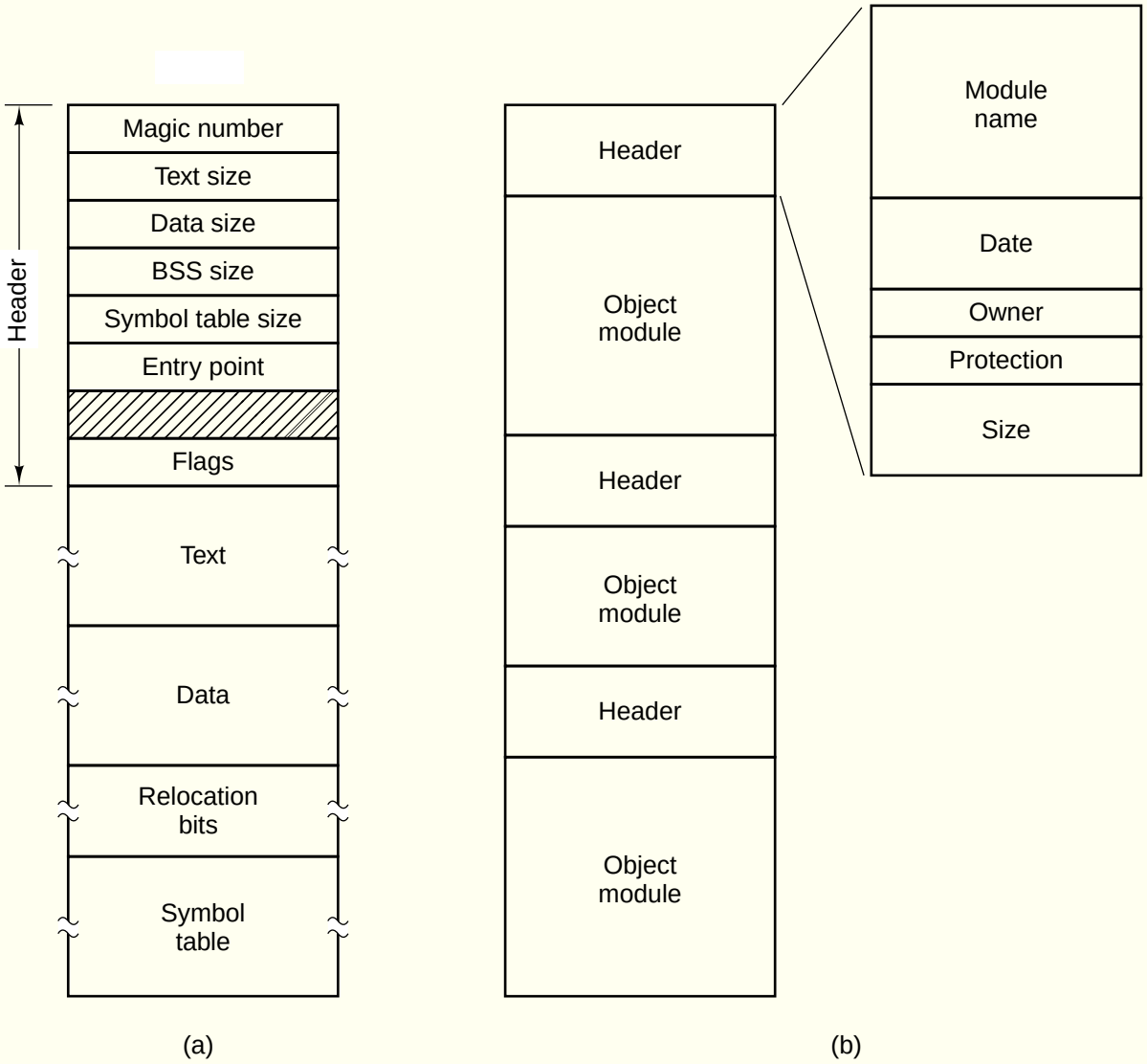
Estrutura de arquivos



Tipos de arquivos

- regulares
- diretórios
- caracter
  - terminais, impressoras e rede
- bloco
  - discos

Exemplos: executável e archive



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

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

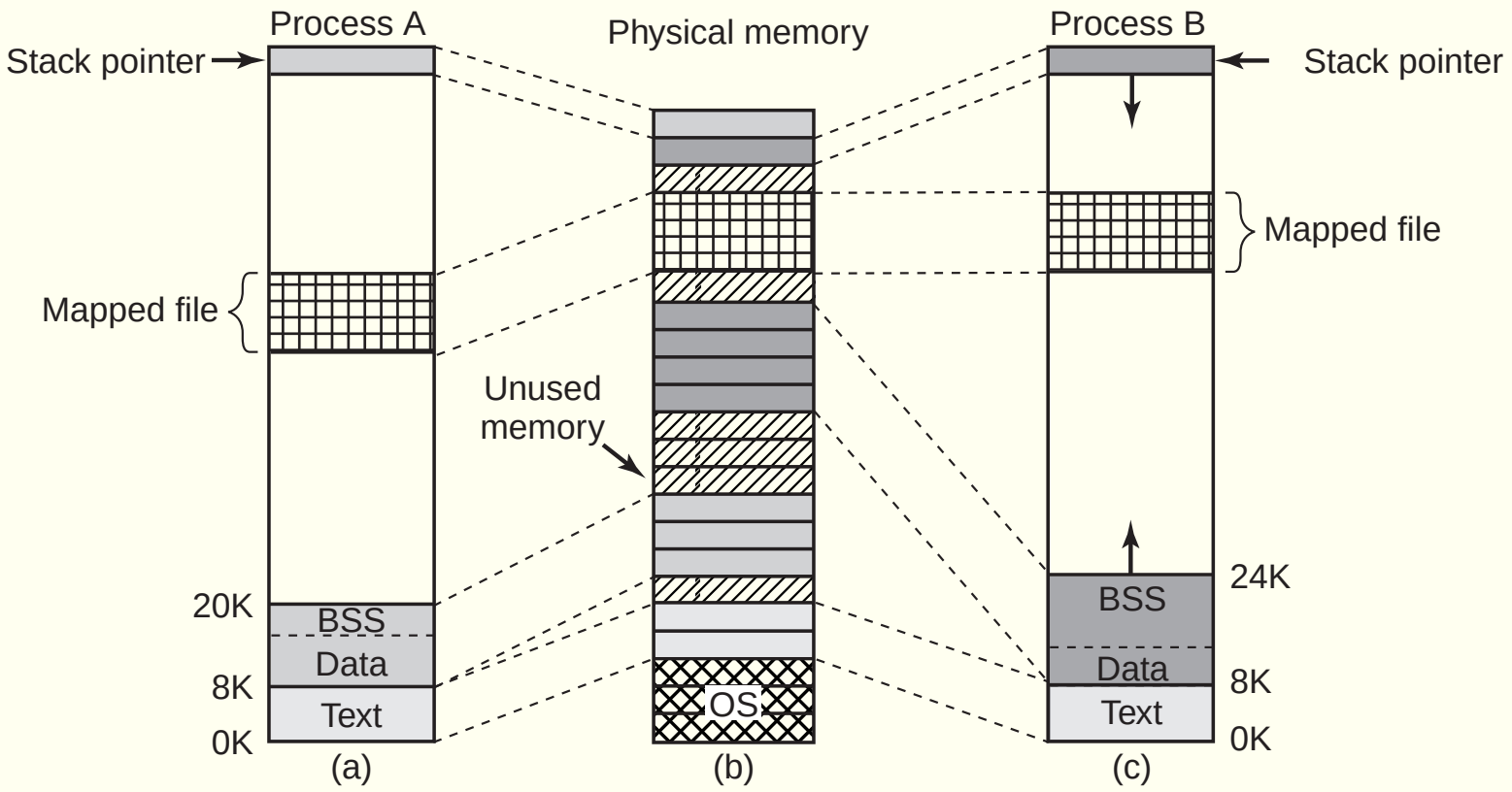
```
  while((rd_count = read(in_fd, buffer, BUF_SIZE)) > 0) {
    wt_count = write(out_fd, buffer, rd_count);
    if (wt_count <= 0) exit(4);
  }

  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



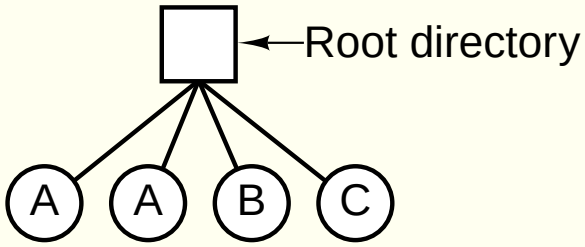
Mmap

```
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

Diretórios

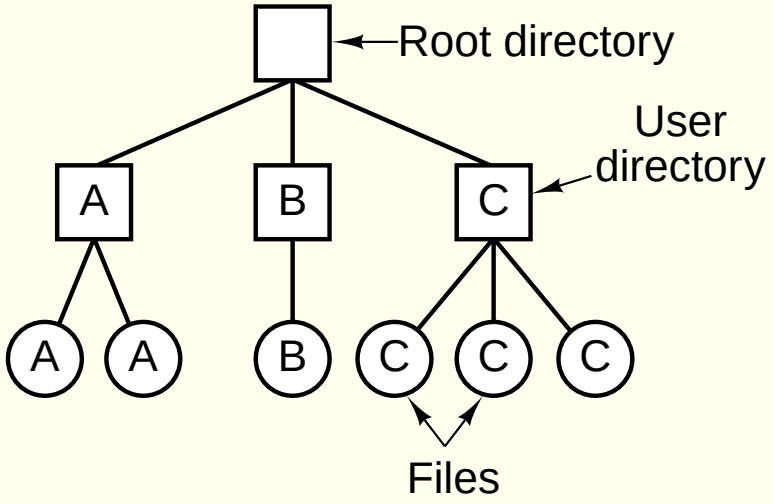
Nível único



- 4 arquivos
- 3 proprietários: A, B e C

Diretórios

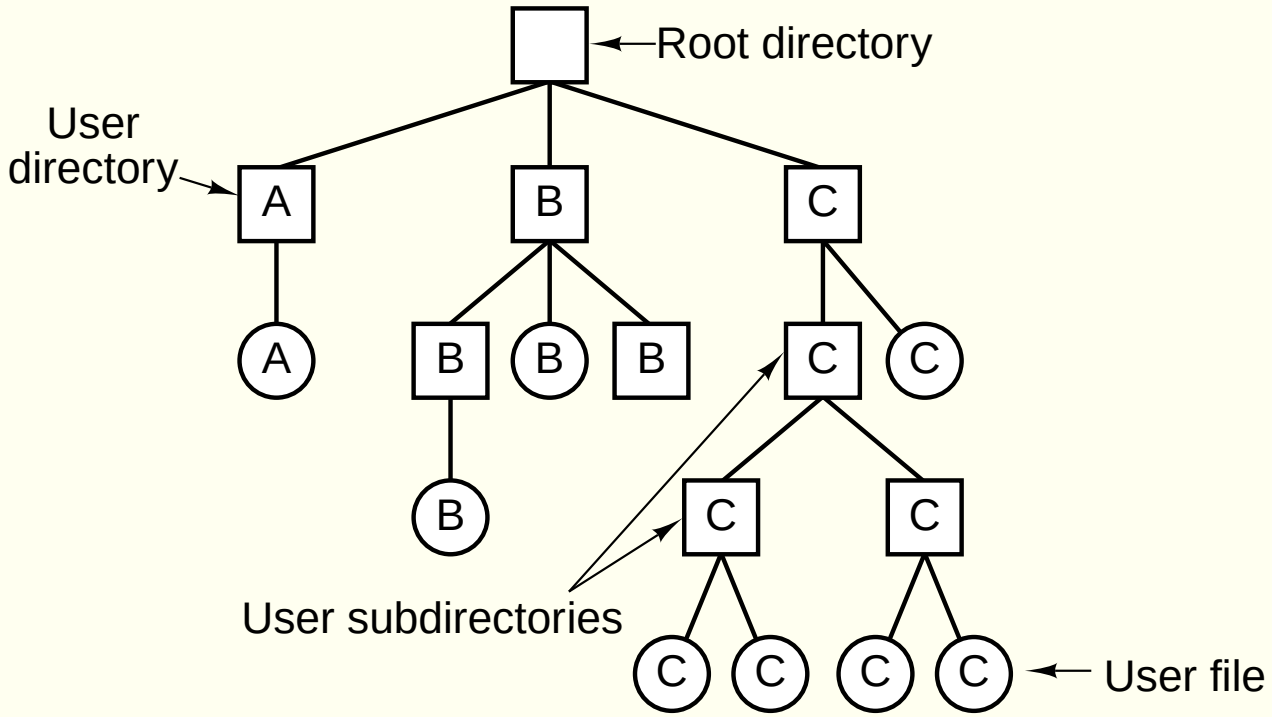
Dois níveis



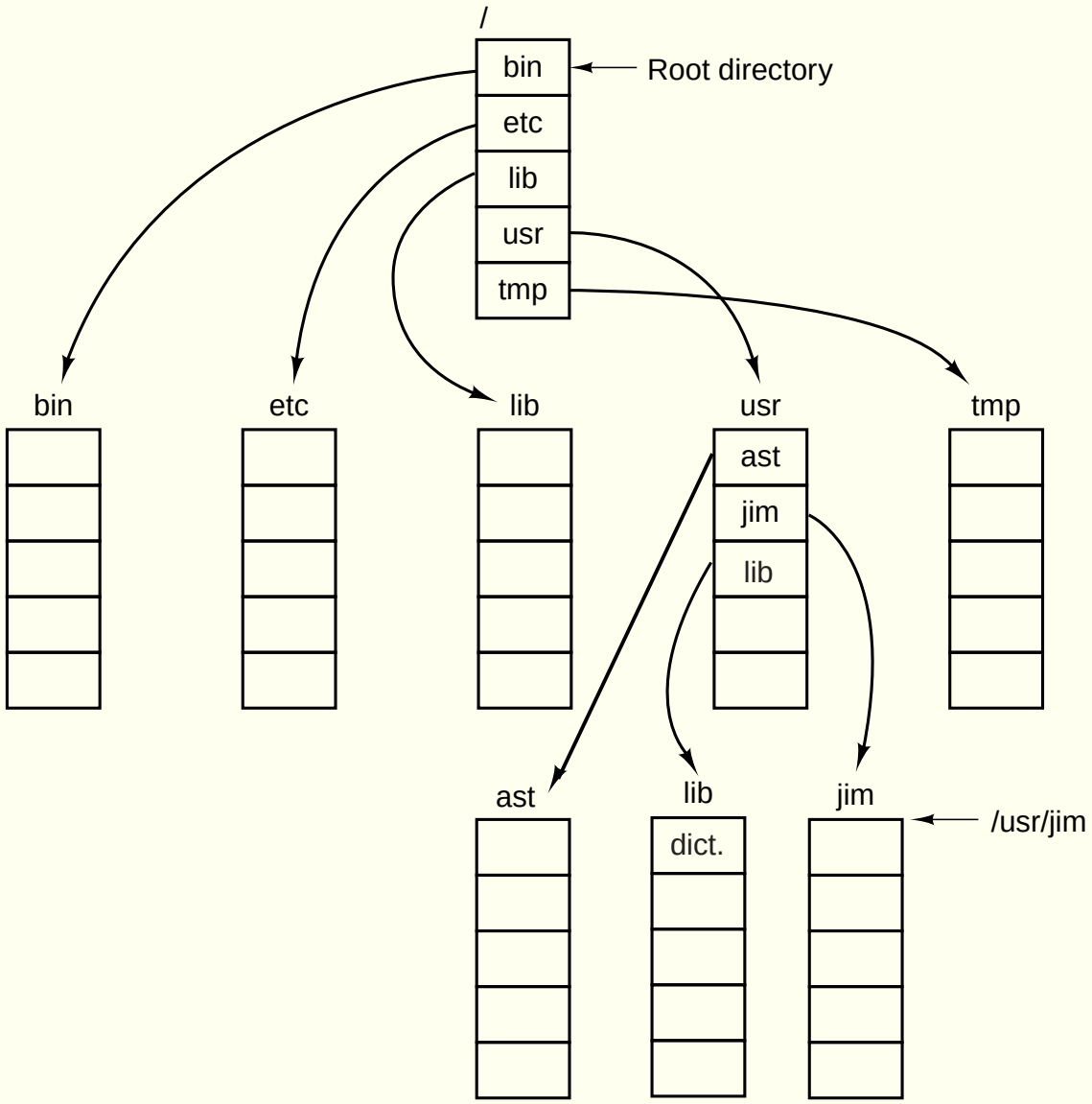
- Um diretório por usuário

Diretórios

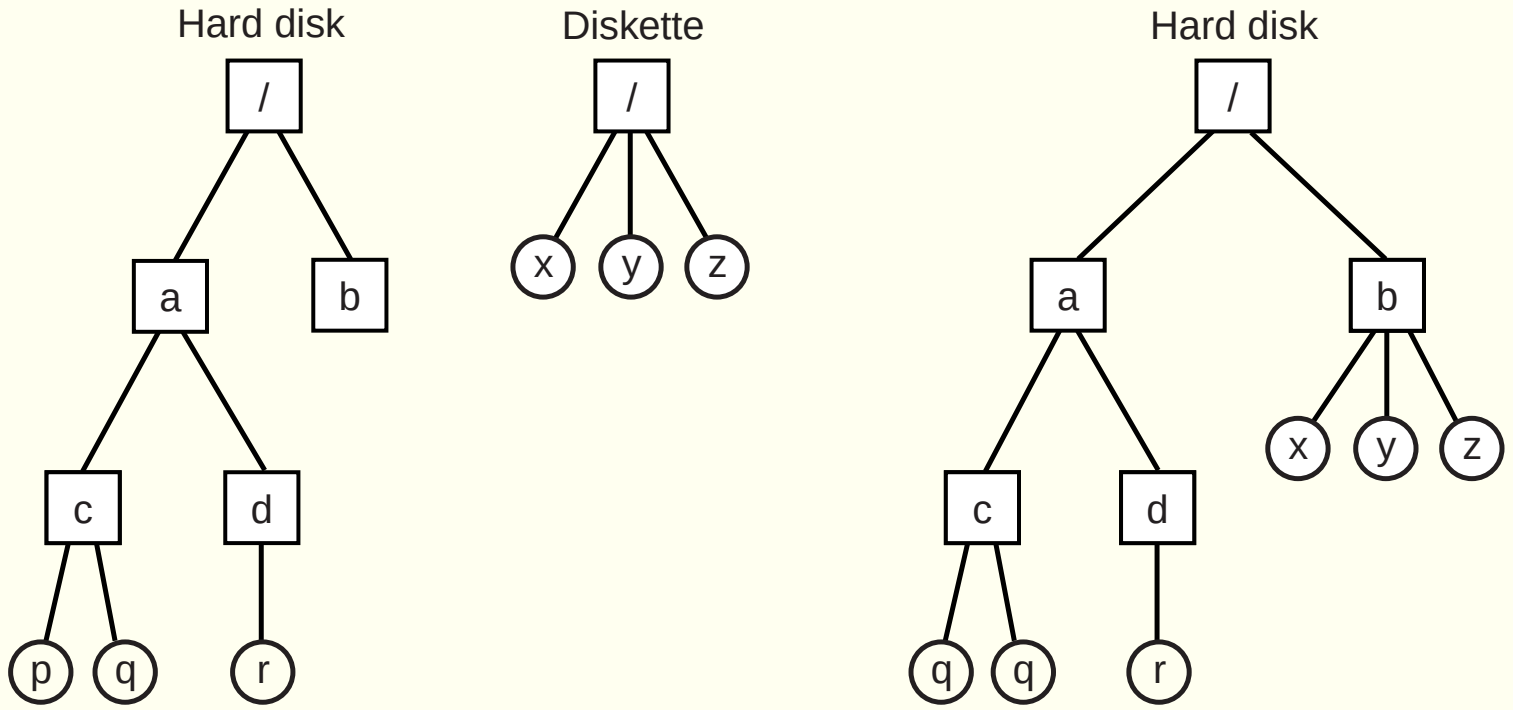
Estrutura hierárquica



Caminhos


/
bin
etc
lib
usr
tmp
Root directory
bin
etc
lib
usr
tmp
ast
jim
lib
ast
lib
jim
dict.
/usr/jim

Mount

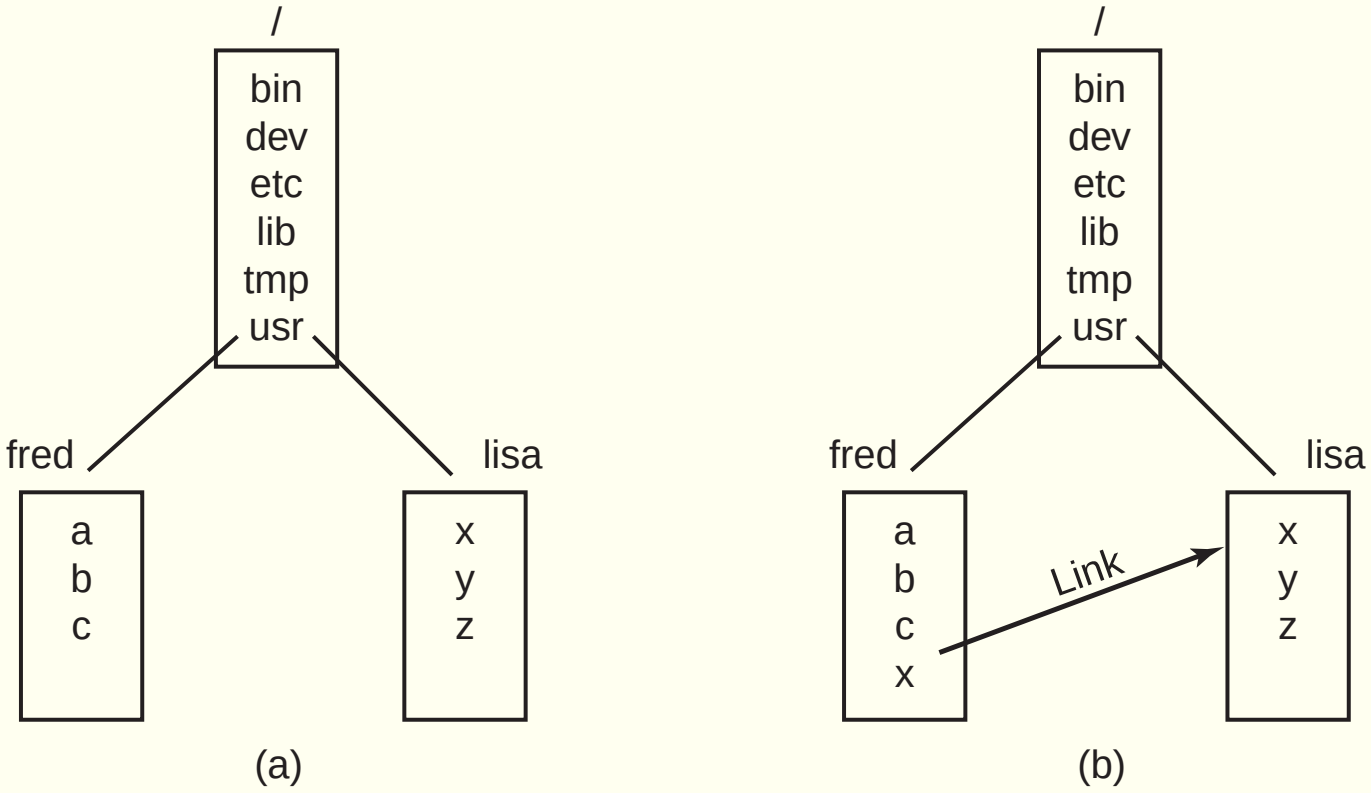


Operações sobre diretórios

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

Veja o código dir.c

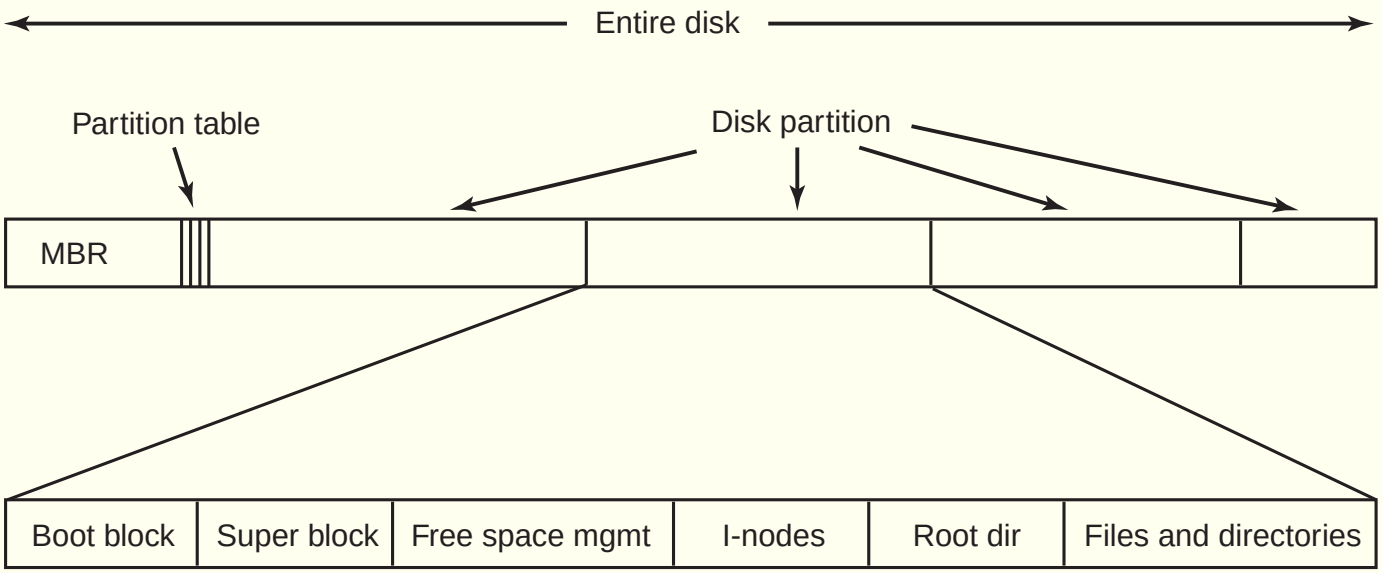
Links


/
bin
dev
etc
lib
tmp
usr
fred
lisa
a
b
c
x
y
z
(a)
/
bin
dev
etc
lib
tmp
usr
fred
lisa
a
b
c
x
Link
x
y
z
(b)

Questões de Implementação

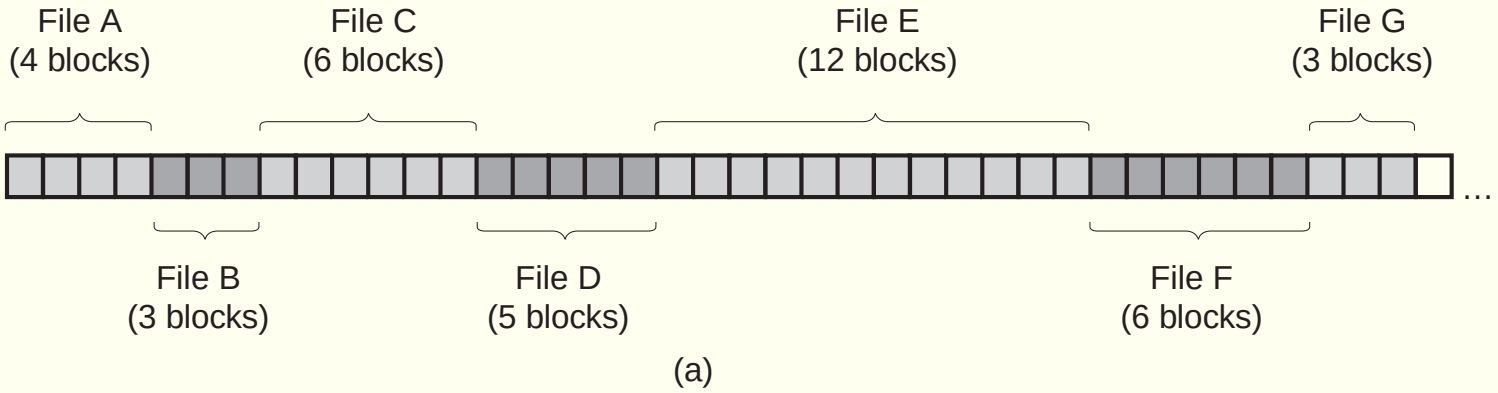
- Como os arquivos são armazenados
- Como o espaço livre é gerenciado
- Eficiência
- Confiabilidade

Layout

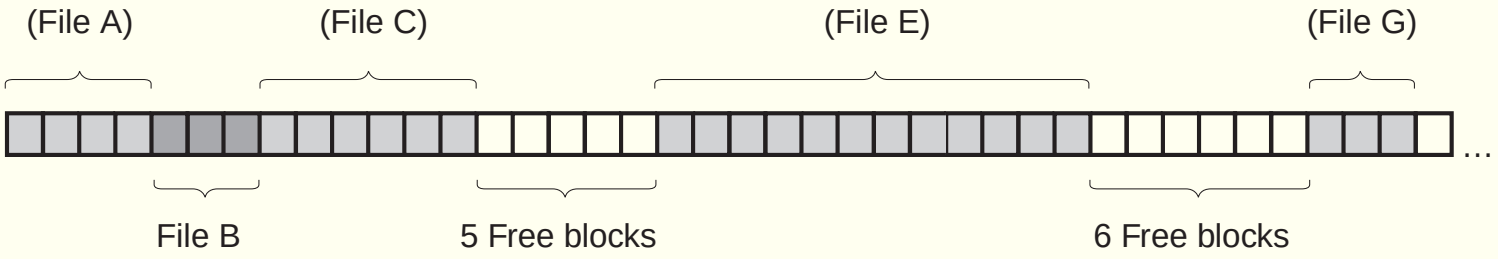


- MBR (Master Boot Record)
- Tabela de partições
- Boot block

Alocação contínua

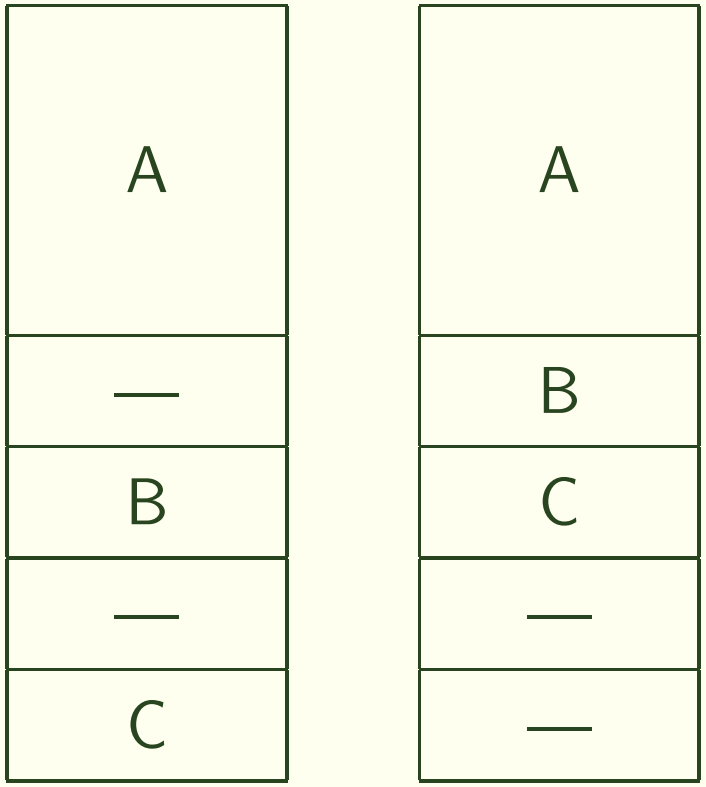


(a)

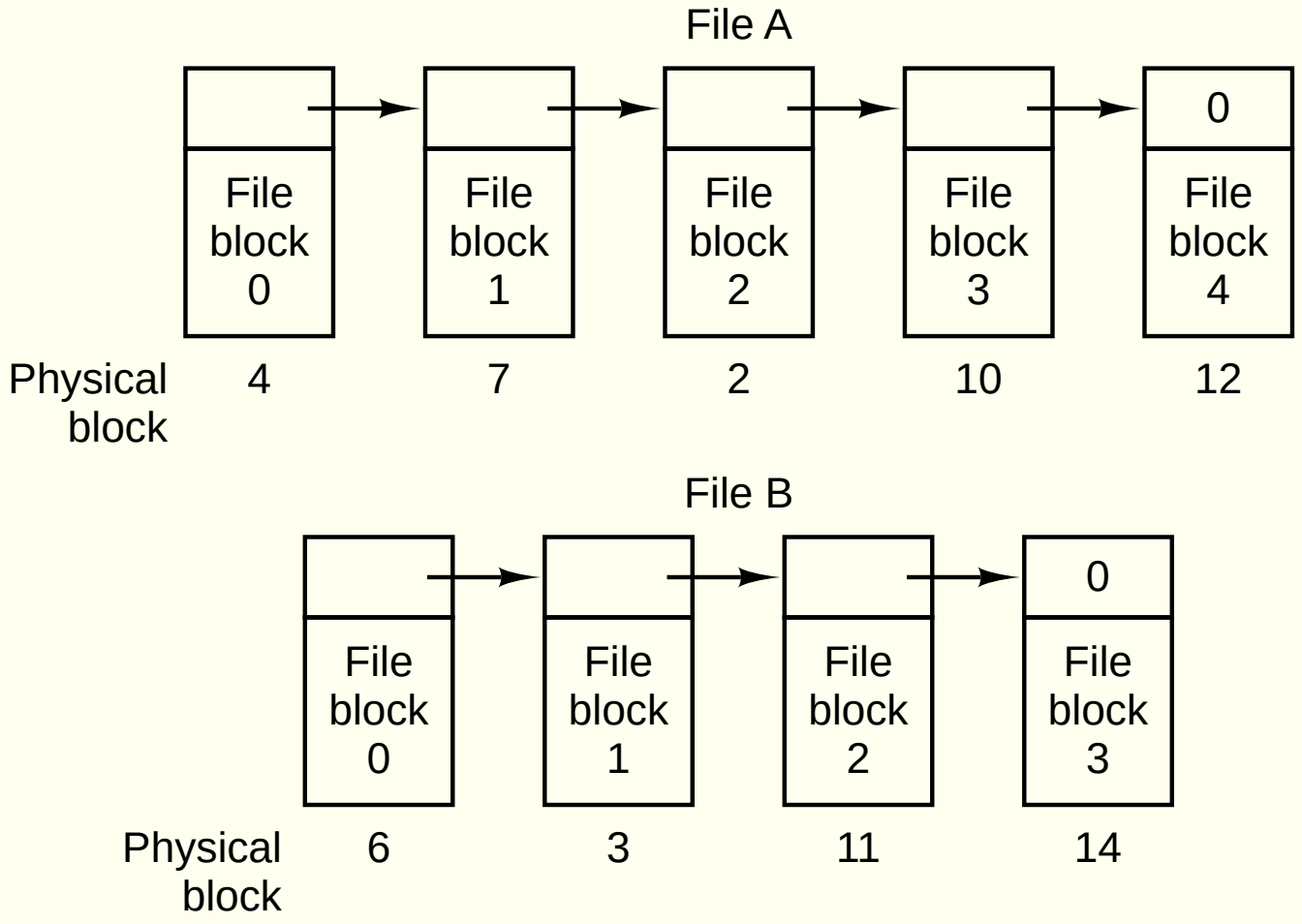


(b)

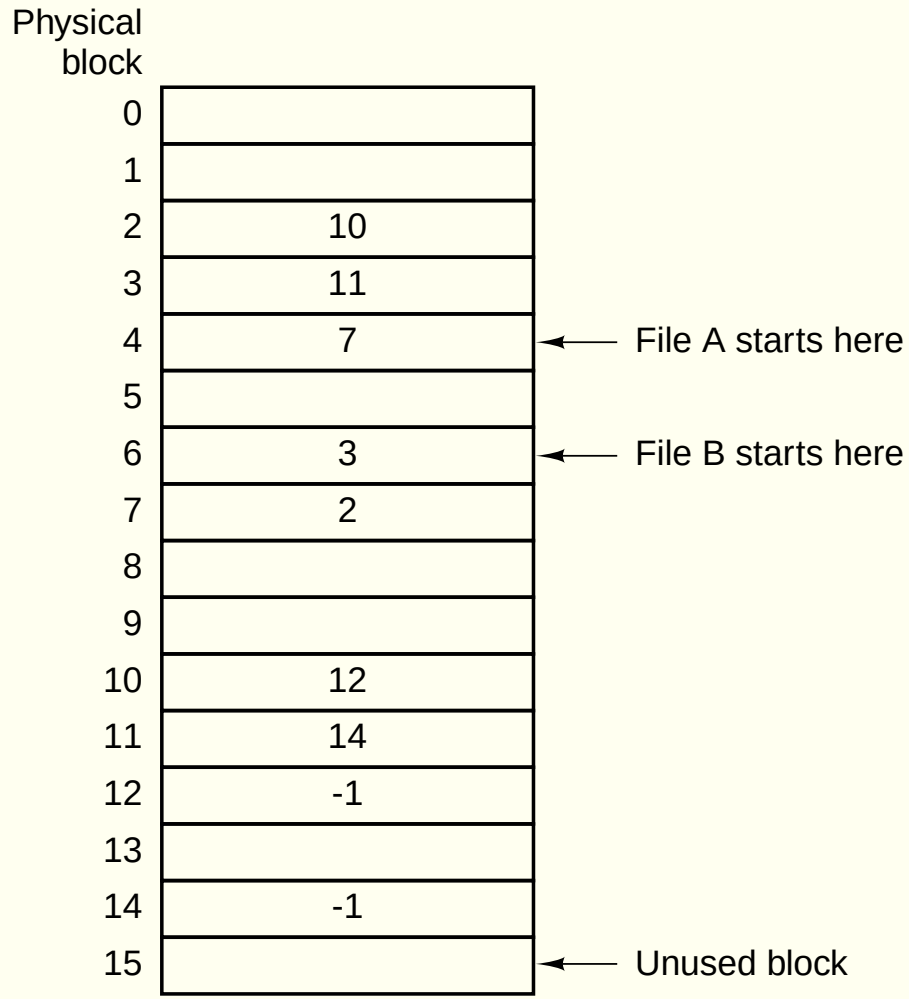
Compactação do disco



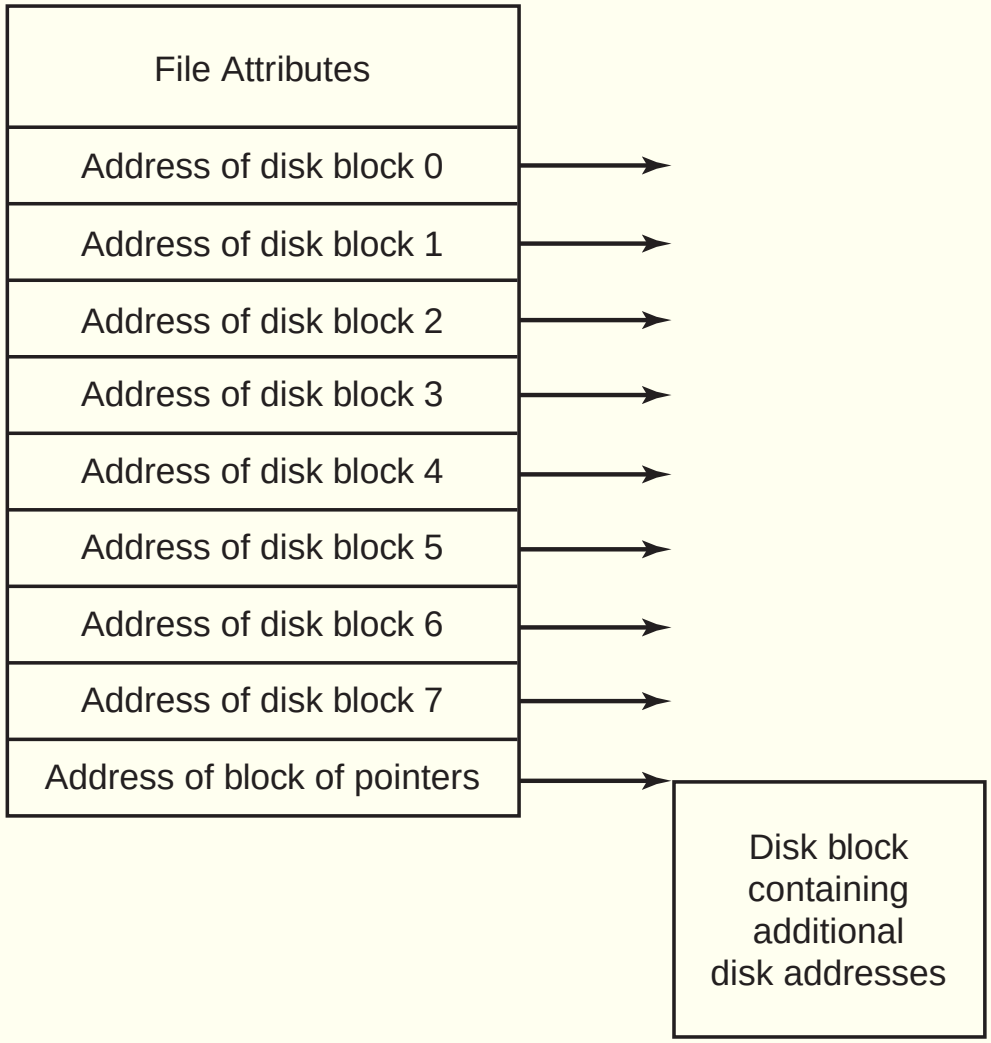
Lista ligada de blocos



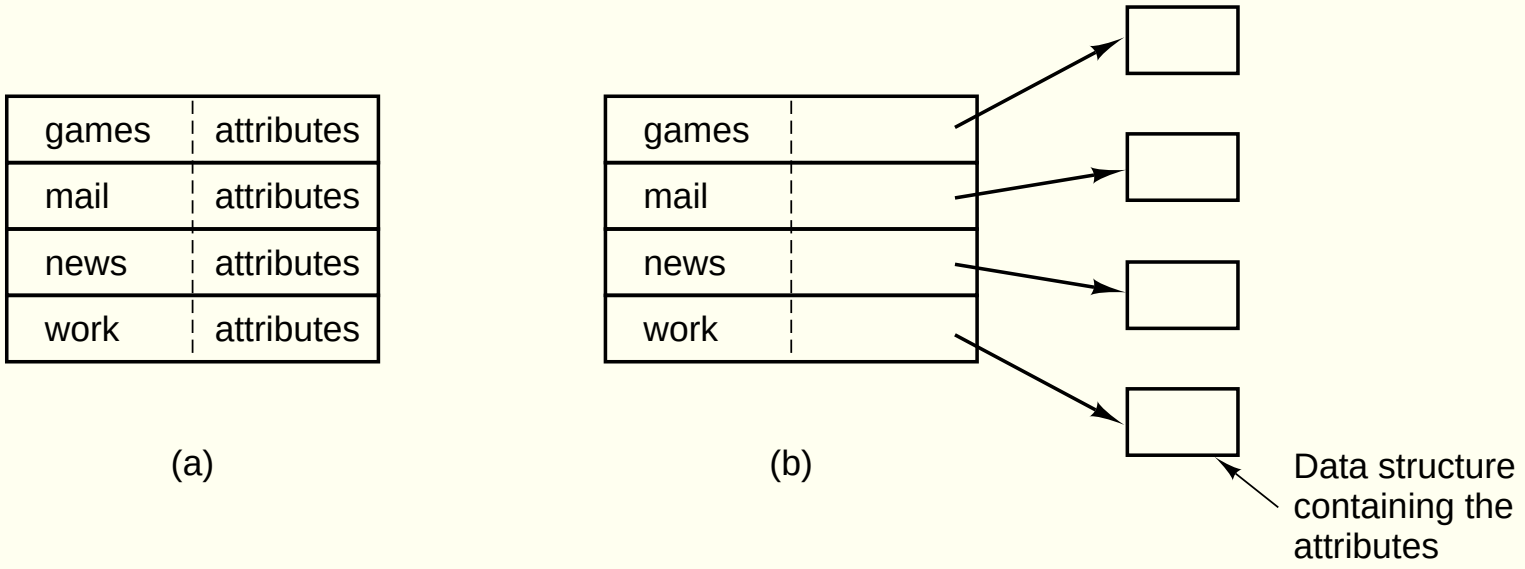
File Allocation Table (FAT)



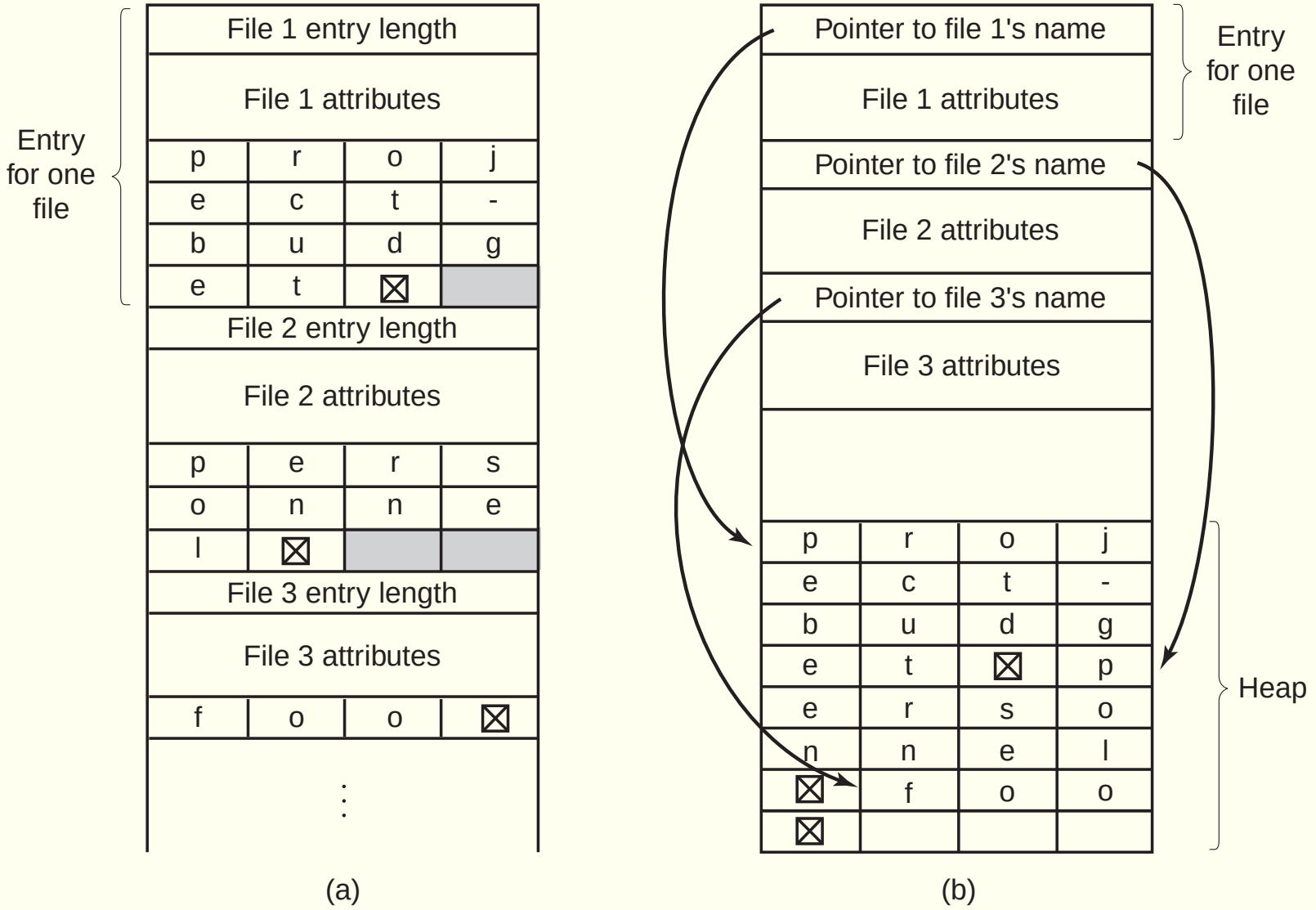
I-node


File Attributes
Address of disk block 0
Address of disk block 1
Address of disk block 2
Address of disk block 3
Address of disk block 4
Address of disk block 5
Address of disk block 6
Address of disk block 7
Address of block of pointers
Disk block containing additional disk addresses

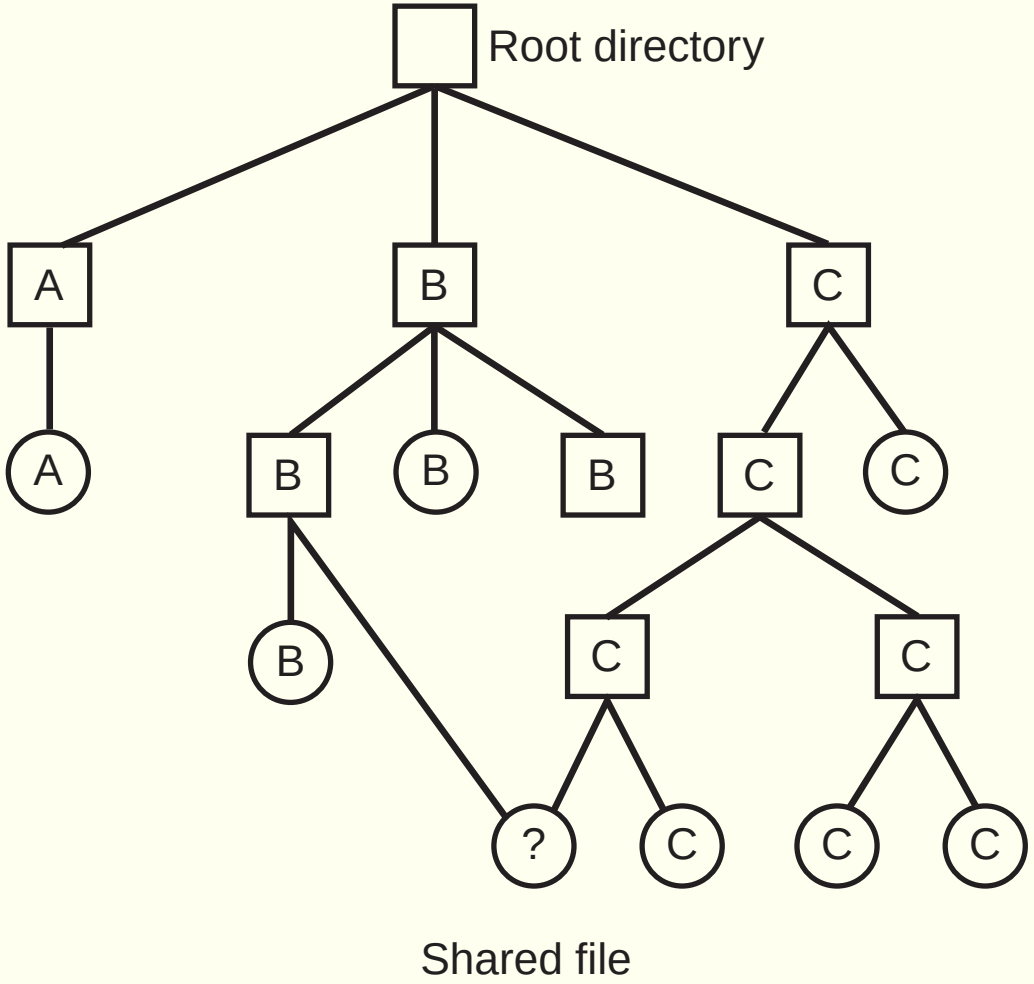
Implementação de diretórios



Nomes de tamanho variável

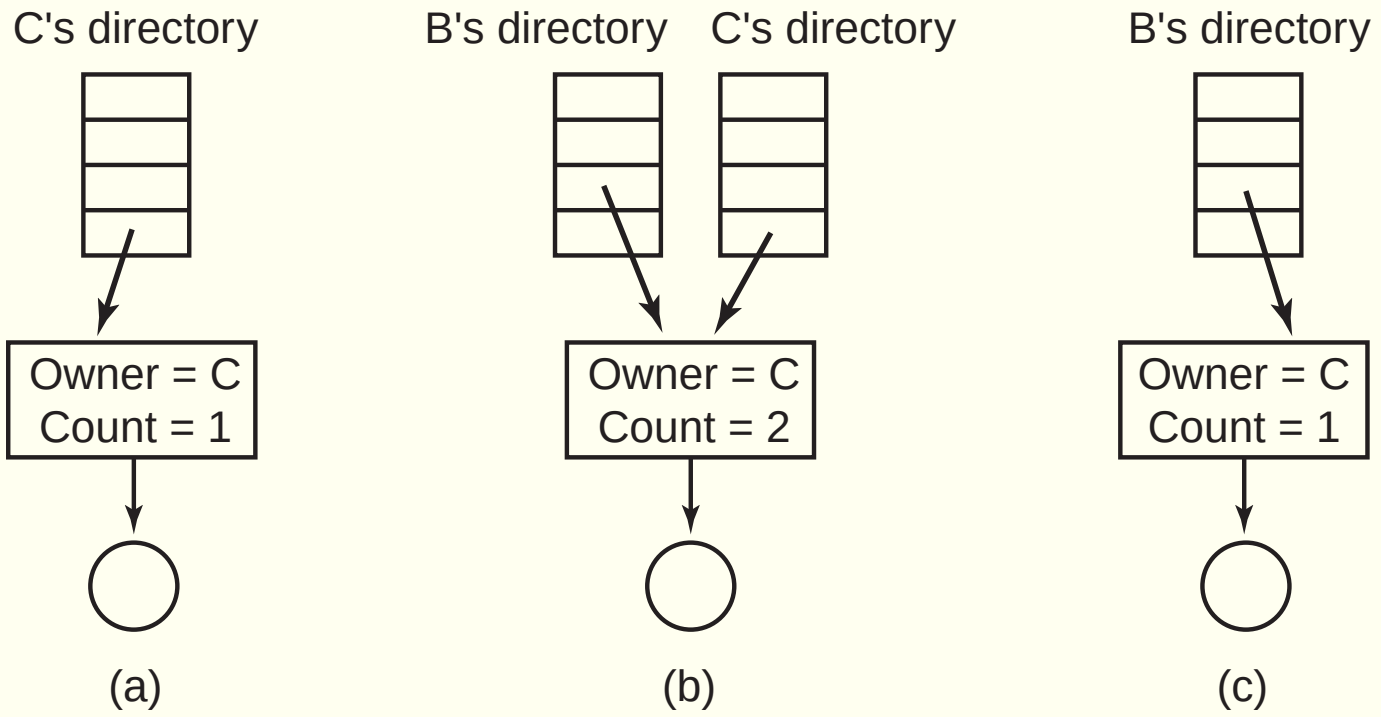


Arquivos compartilhados

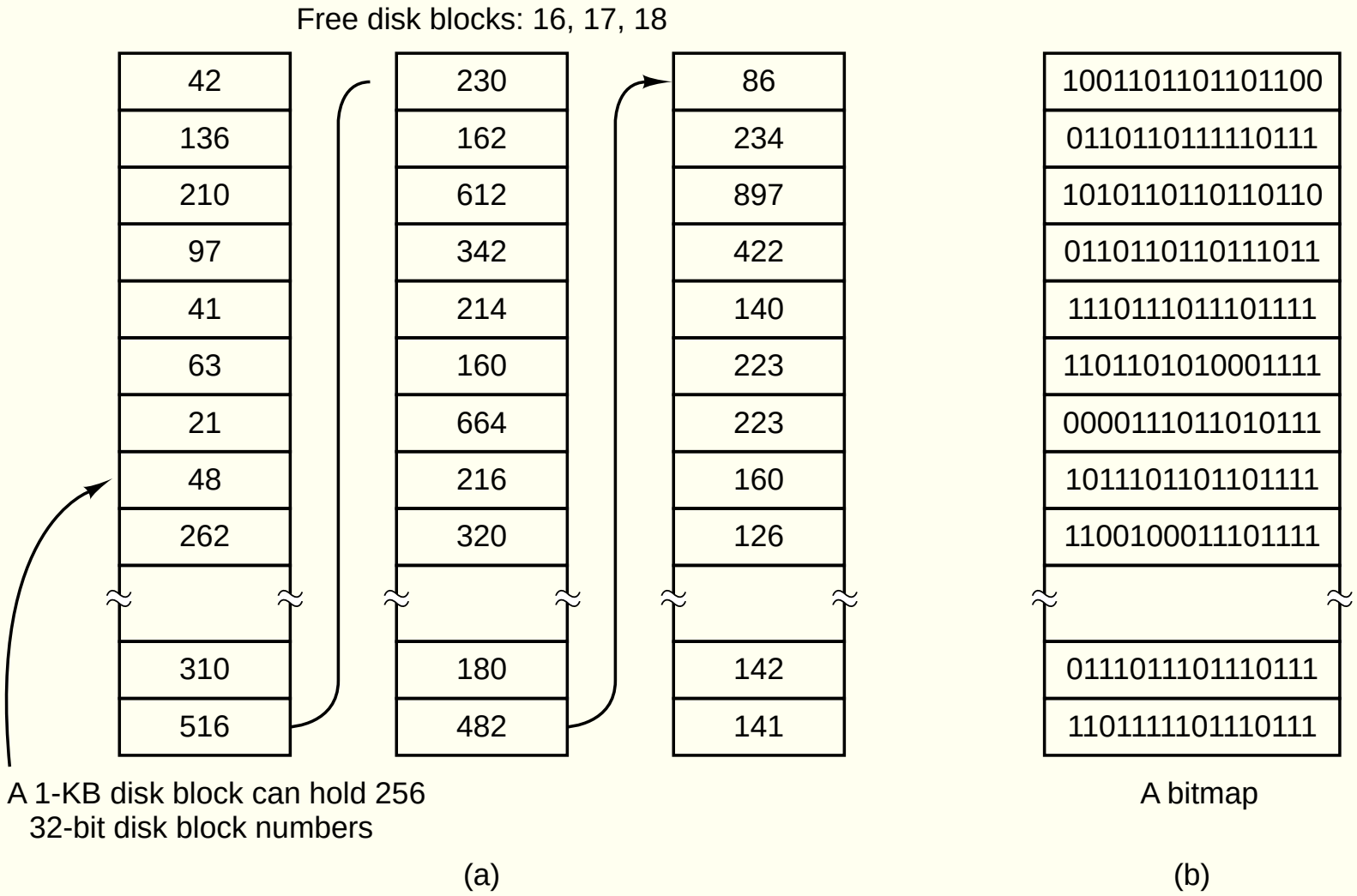


Shared file

Arquivos compartilhados

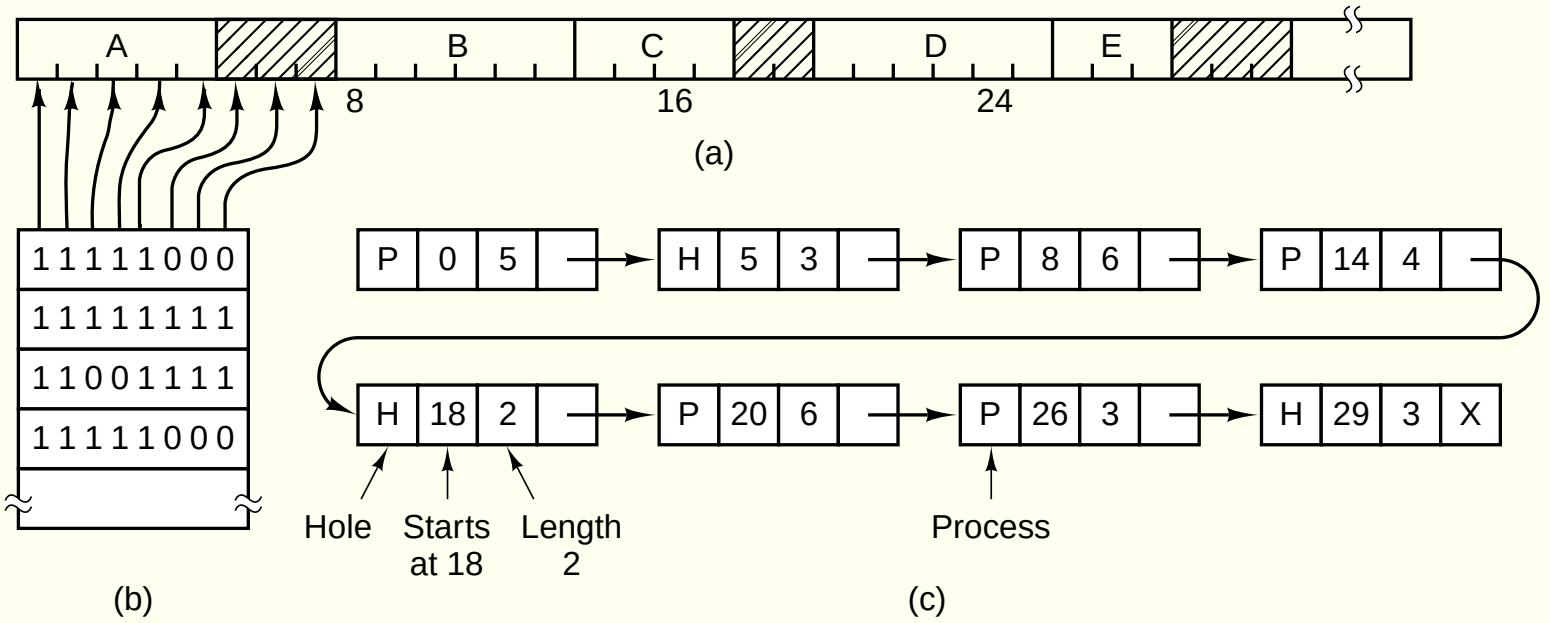


Lista de livres e bitmaps

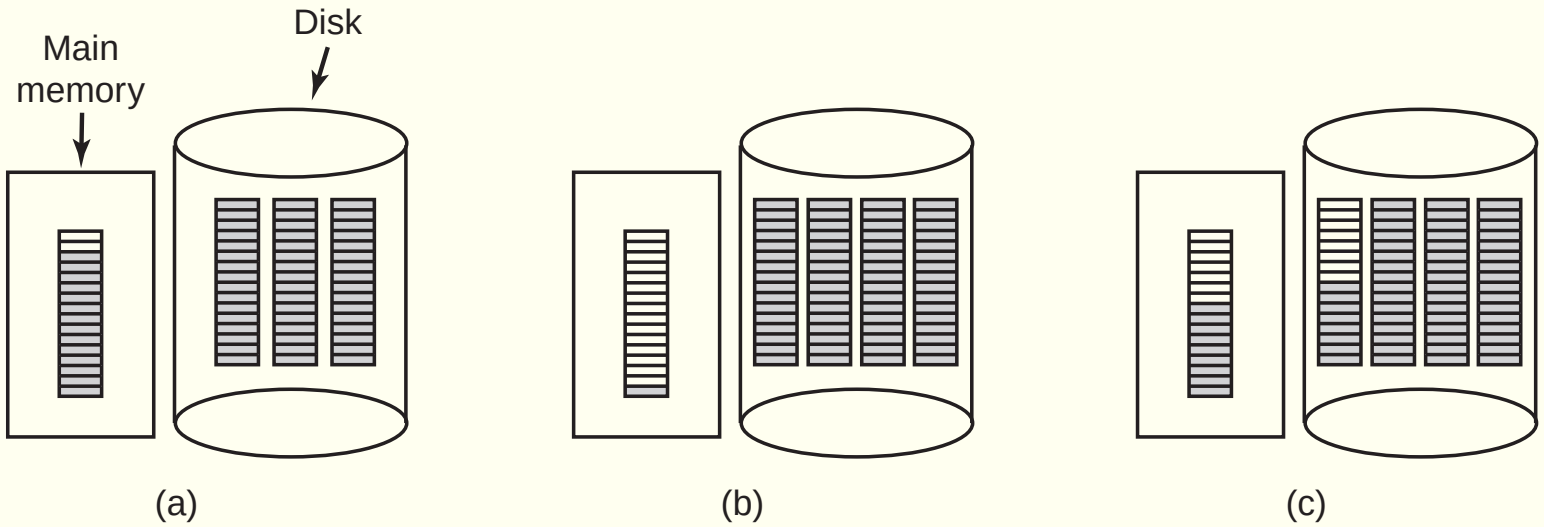


Bitmaps e lista de livres

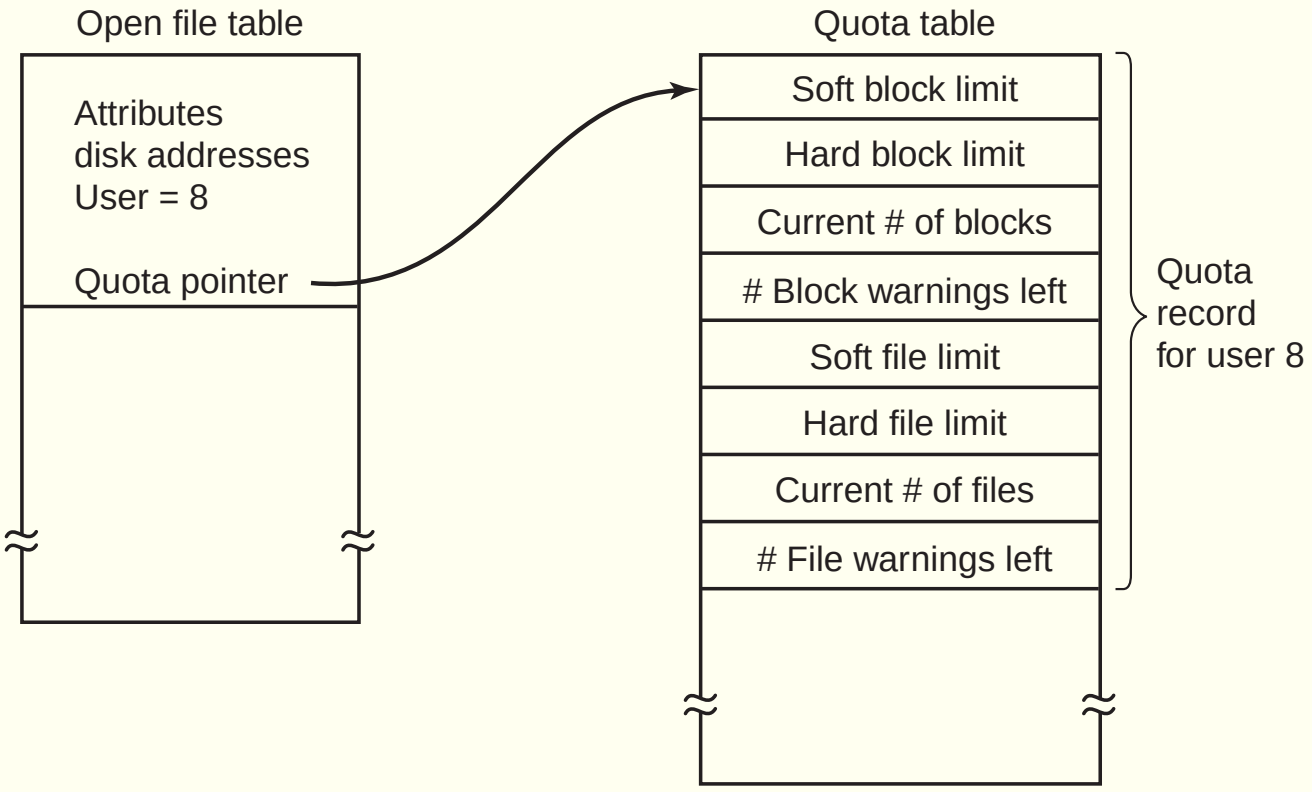
Gerência de memória



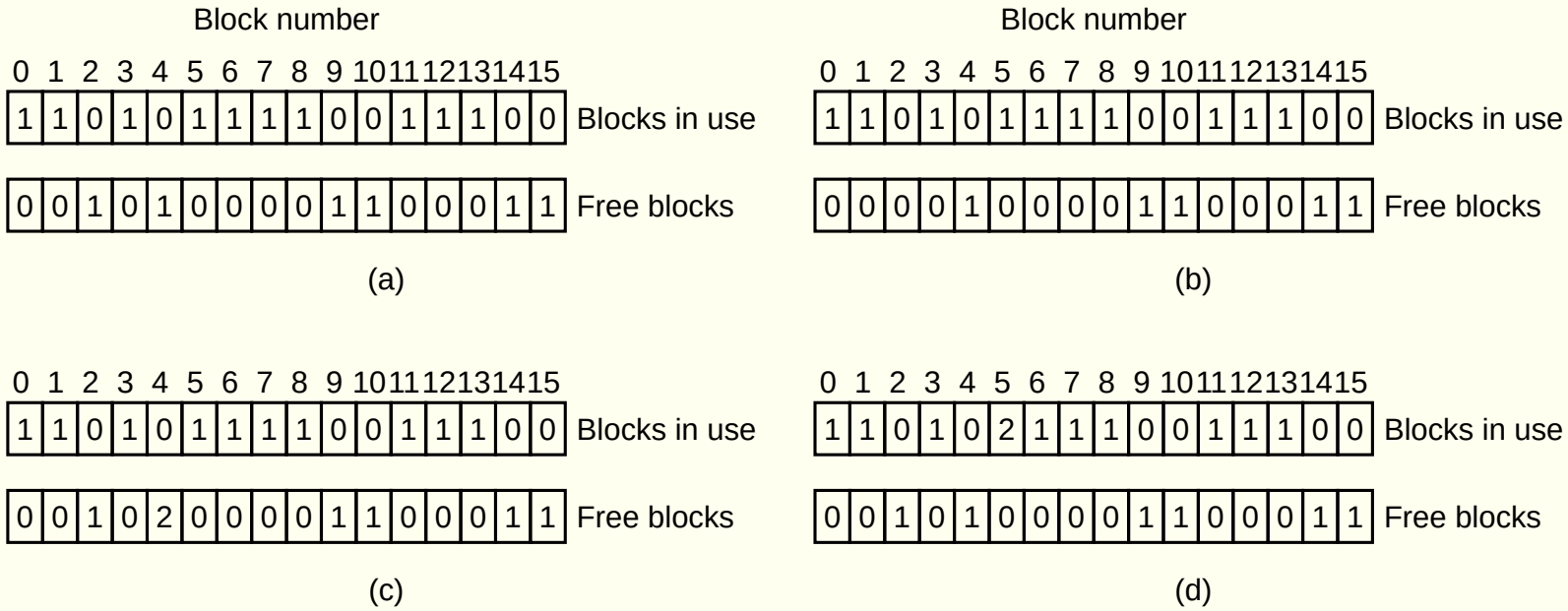
Lista de livres em memória



Gerência de quotas



Consistência do sistema de arquivos



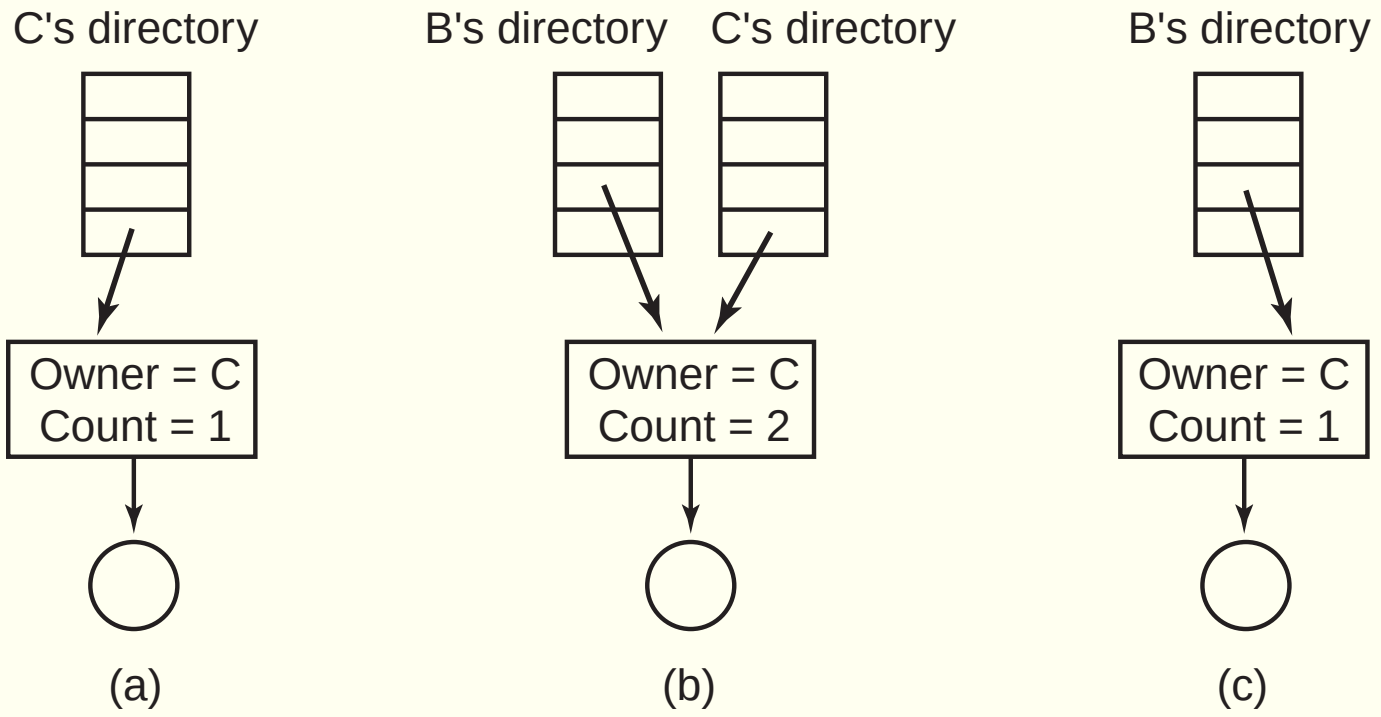
(a) consistente

(b) bloco faltando

(c) duplicação na lista de livres

(d) duplicação nos dados

Consistência do sistema de arquivos



Verificar se todos os contadores estão corretos

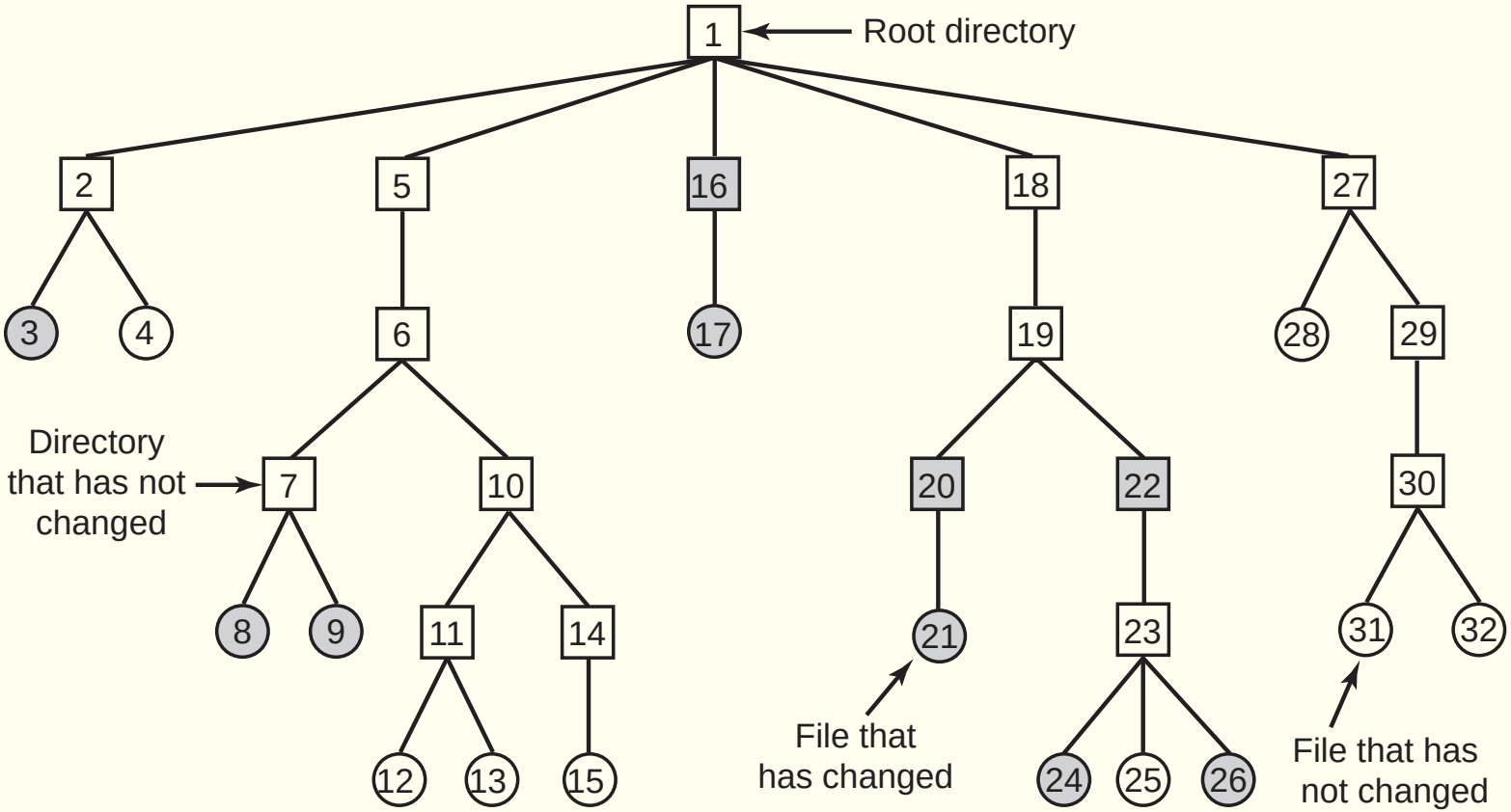
Cópias de segurança

- 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

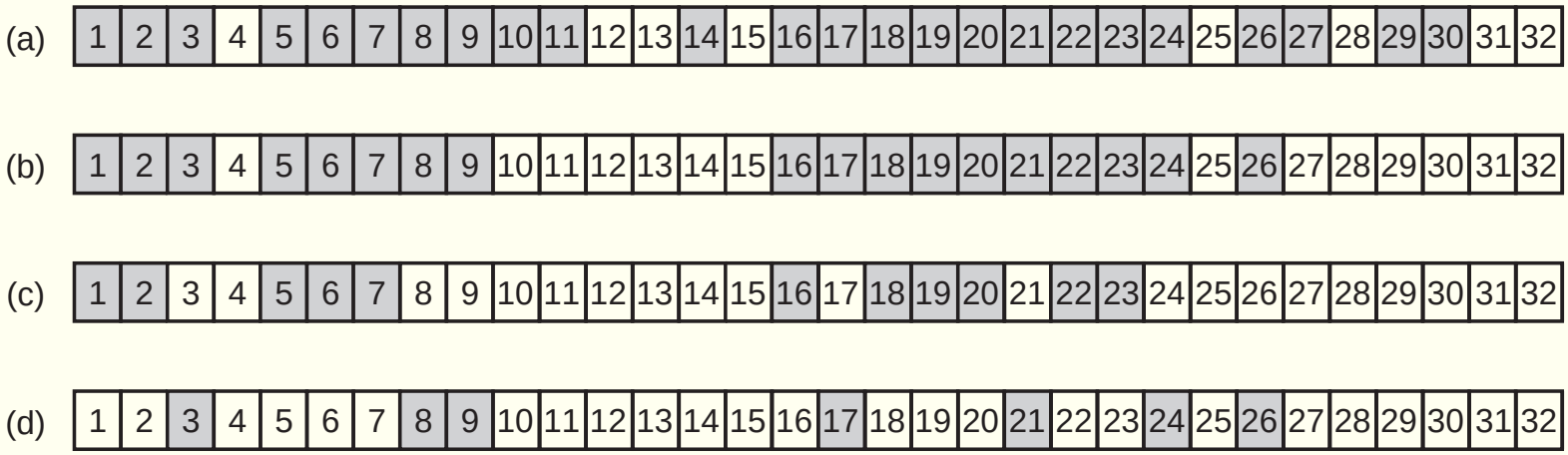
Cópia incremental



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

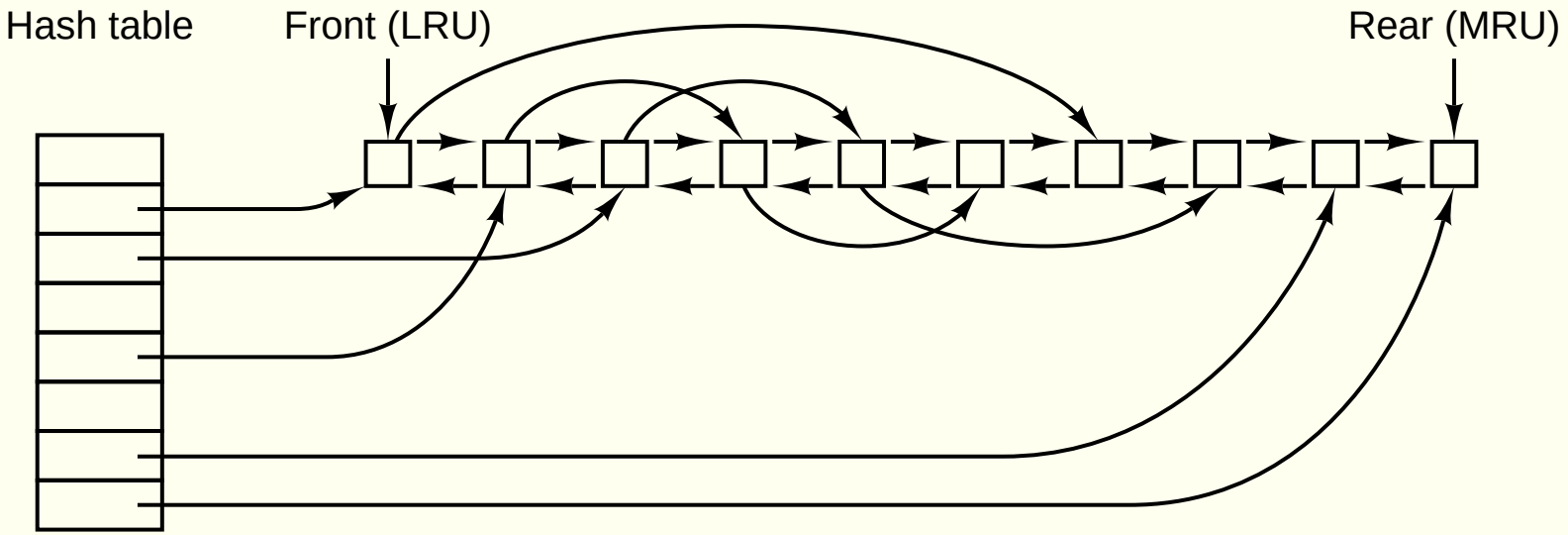
Mapas de i-nodes e fases



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