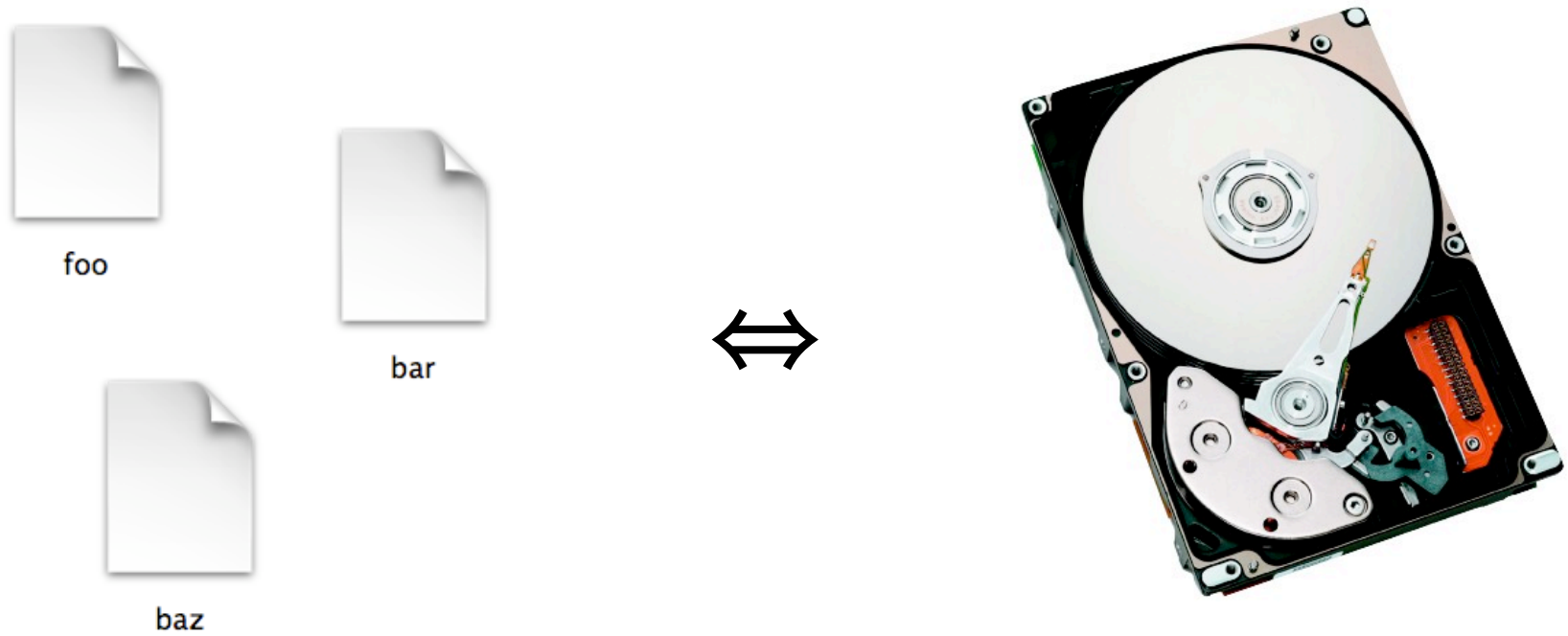
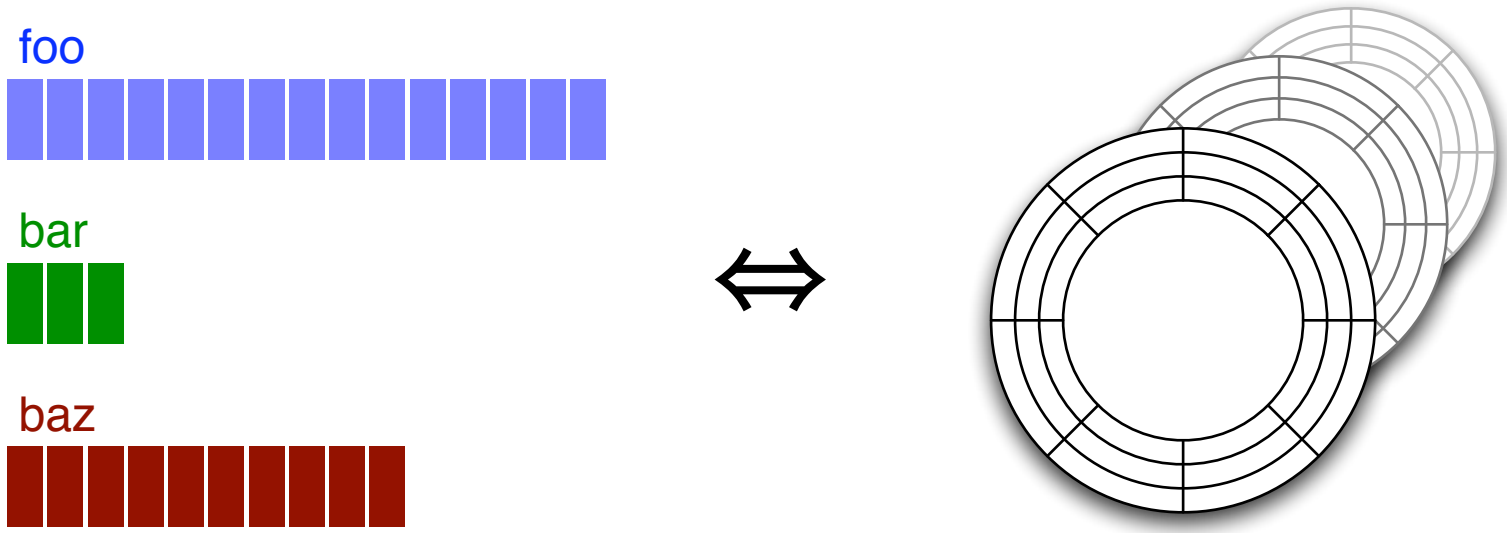
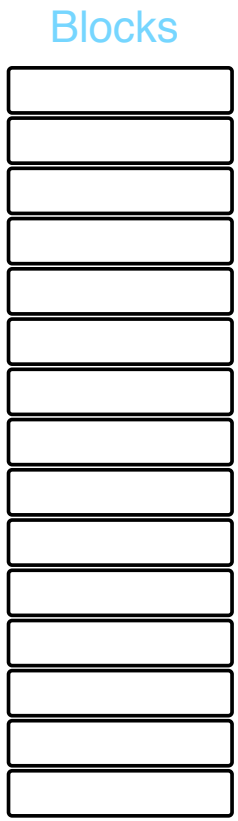
File System

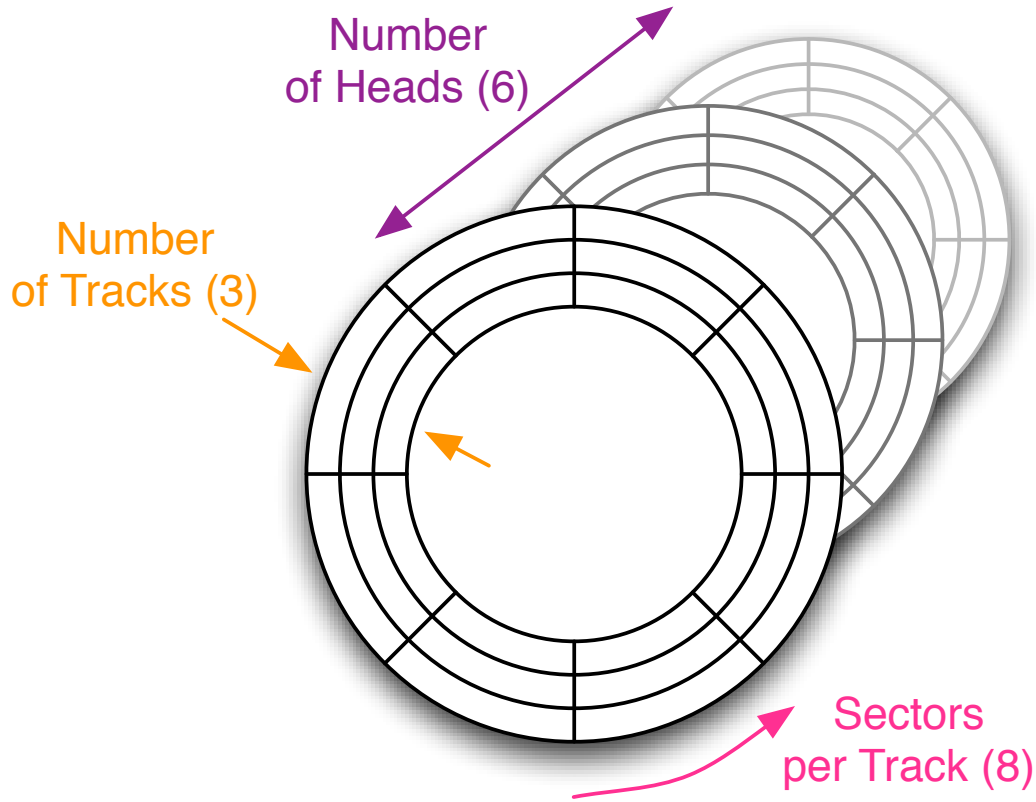
part II

CS170

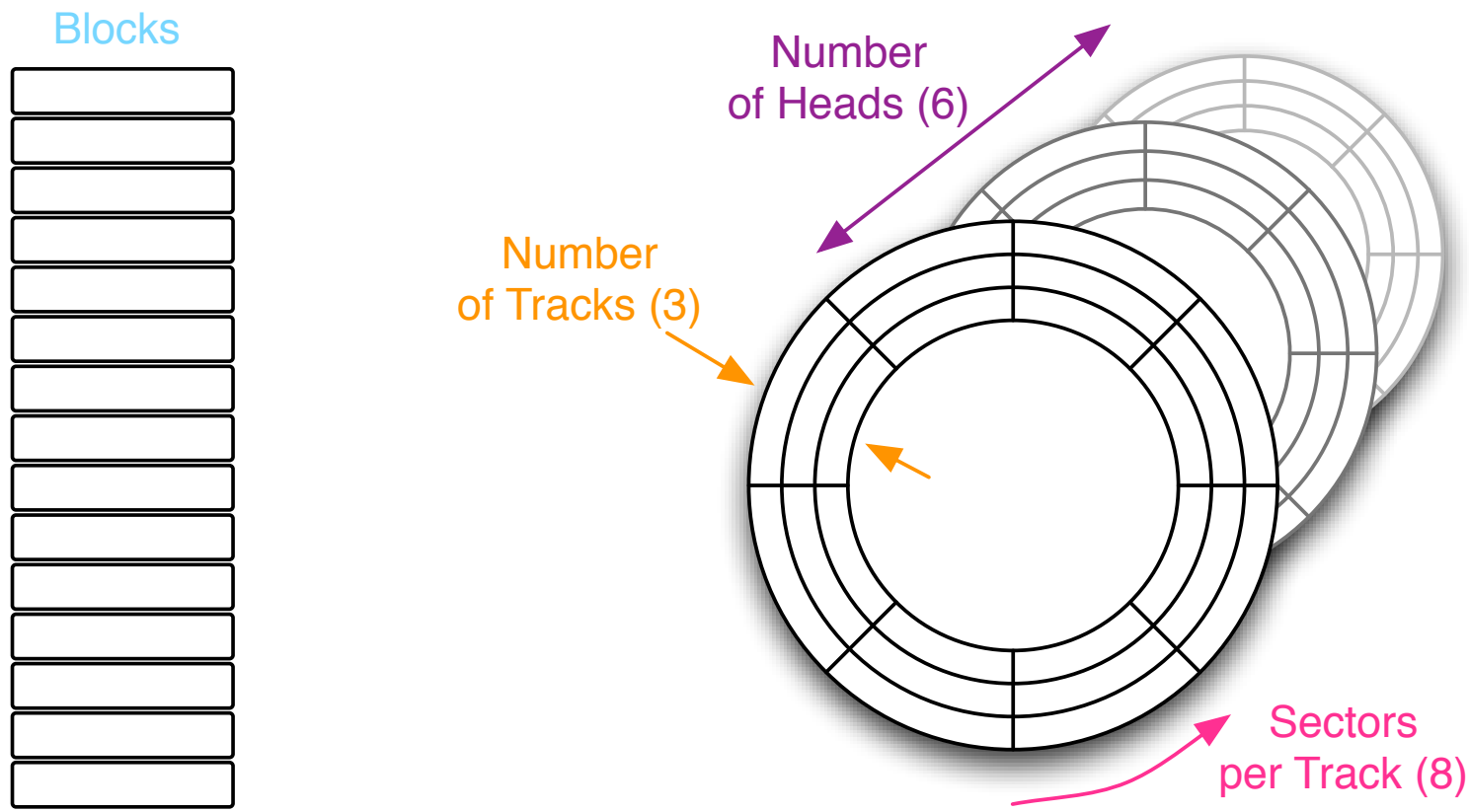

foo
bar
baz


foo
bar
baz



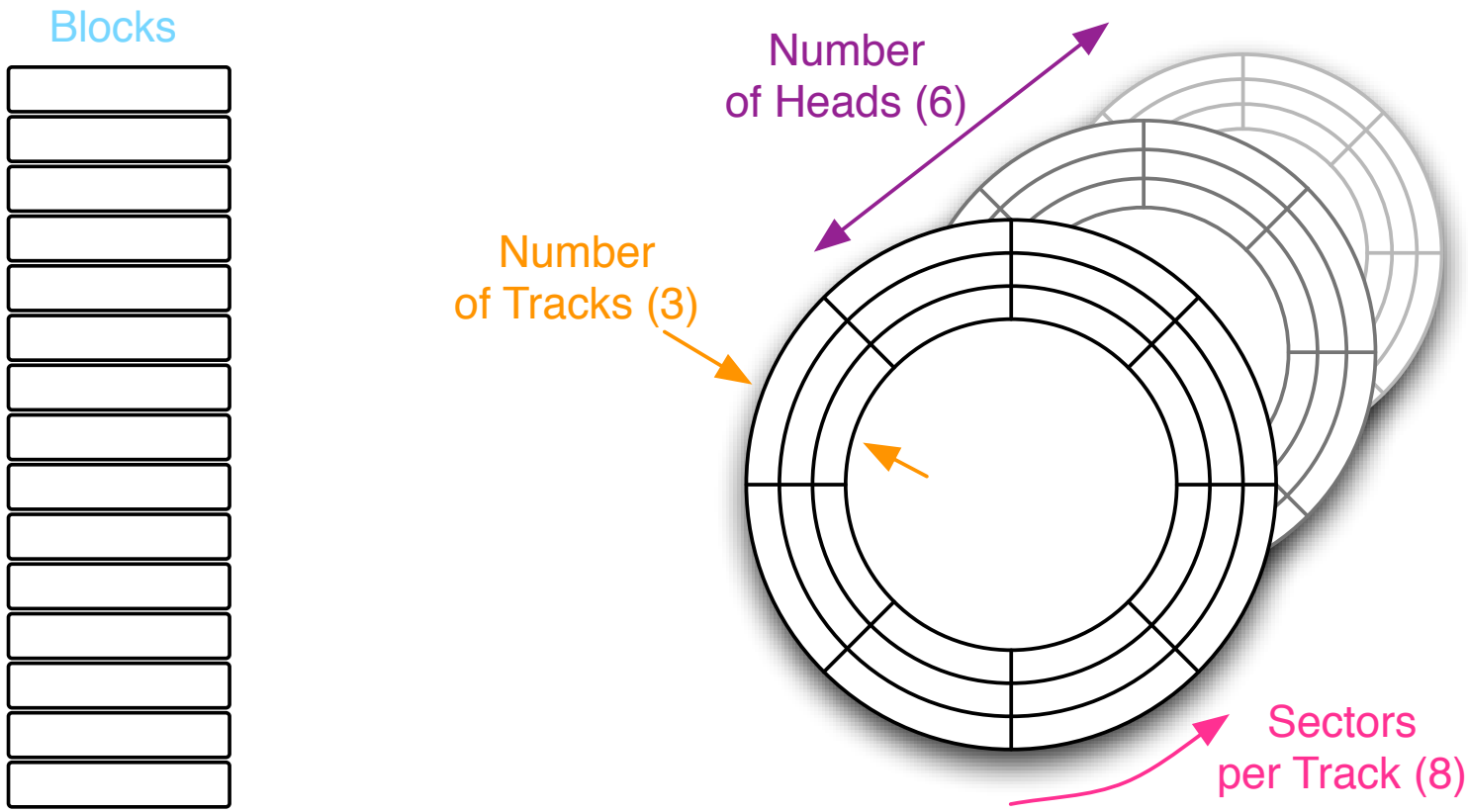


Total-Disk-Size = Number-of-Heads *
Sectors-per-Track *
Number-of-Tracks



Sectors-per-Cylinder = Number-of-Heads * Sectors-per-Track

Cylinder = Block / Sectors-per-Cylinder

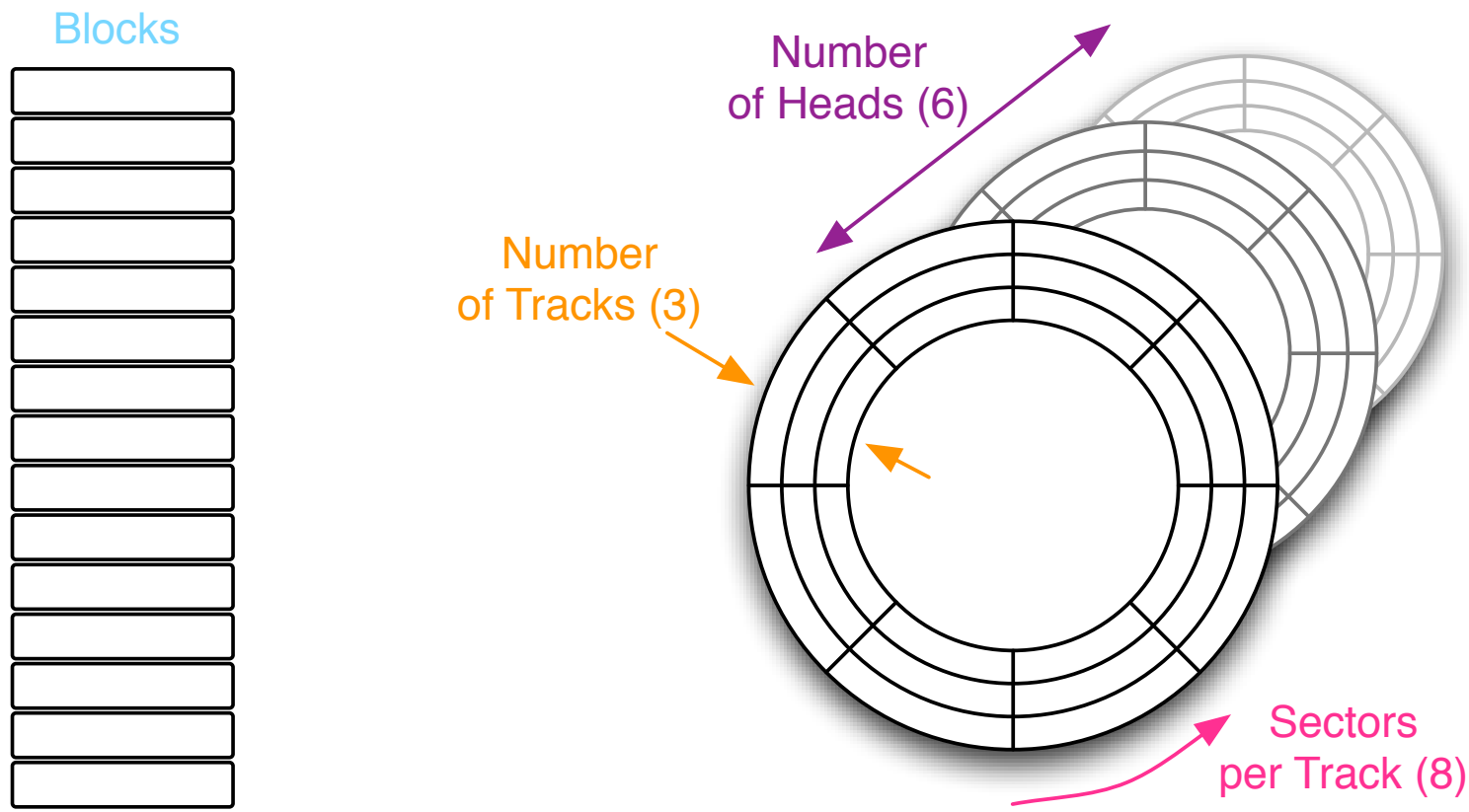


Sectors-per-Cylinder = Number-of-Heads * Sectors-per-Track

Cylinder = Block / Sectors-per-Cylinder

Offset-in-Cylinder = Block % Sectors-per-Cylinder

Head = Offset-in-Cylinder / Sectors-per-Track



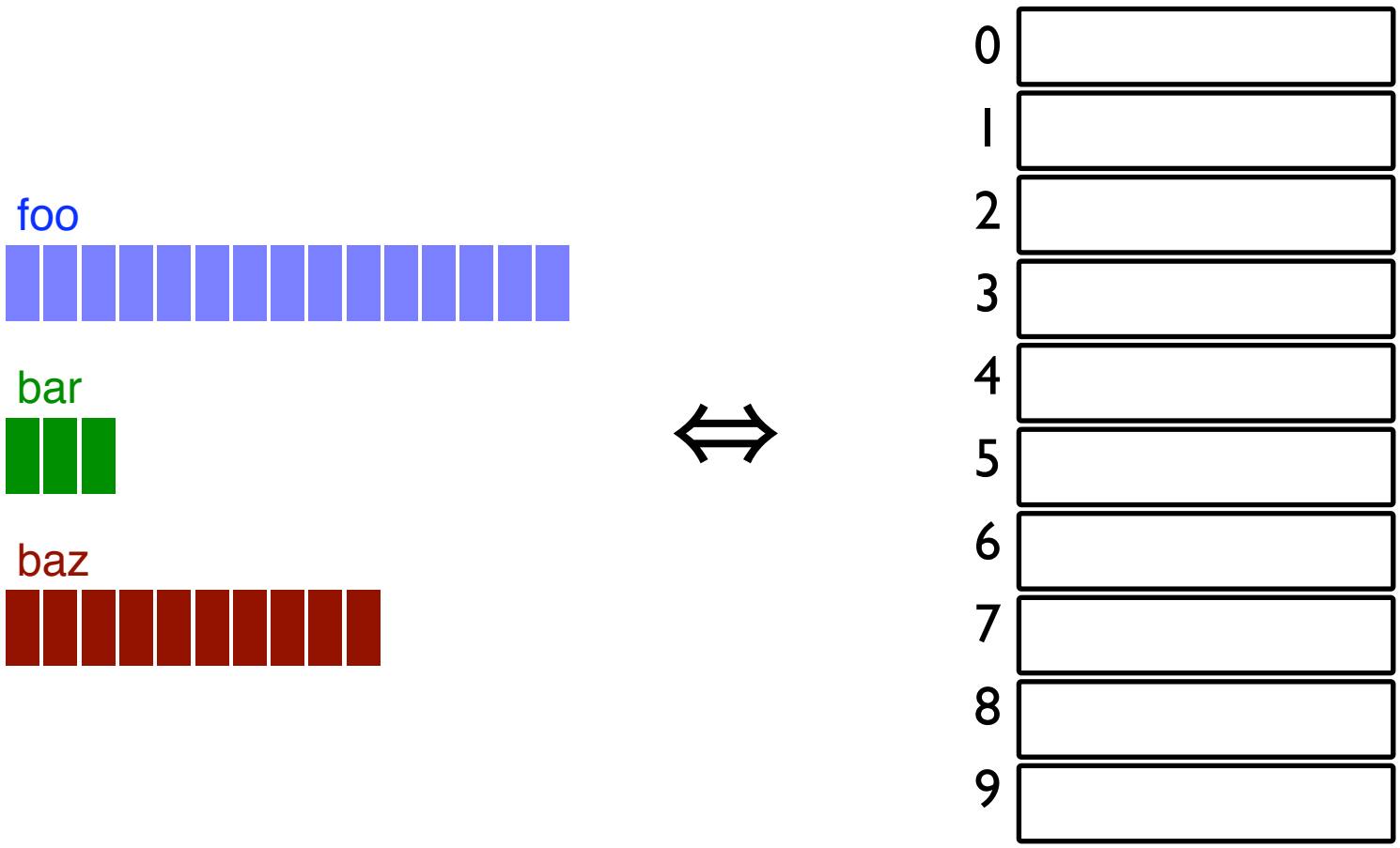
Sectors-per-Cylinder = Number-of-Heads * Sectors-per-Track

Cylinder = Block / Sectors-per-Cylinder

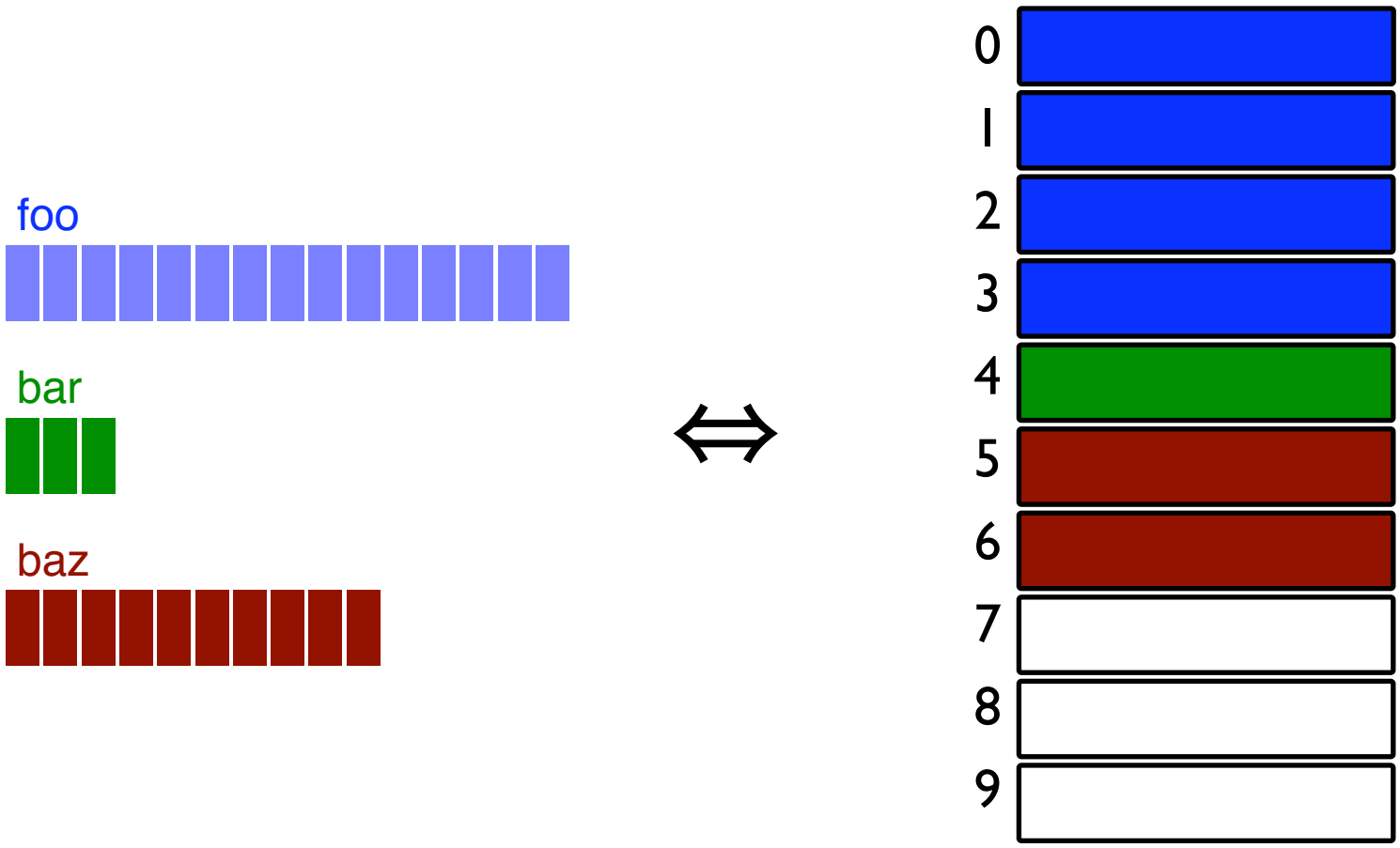
Offset-in-Cylinder = Block % Sectors-per-Cylinder

Head = Offset-in-Cylinder / Sectors-per-Track

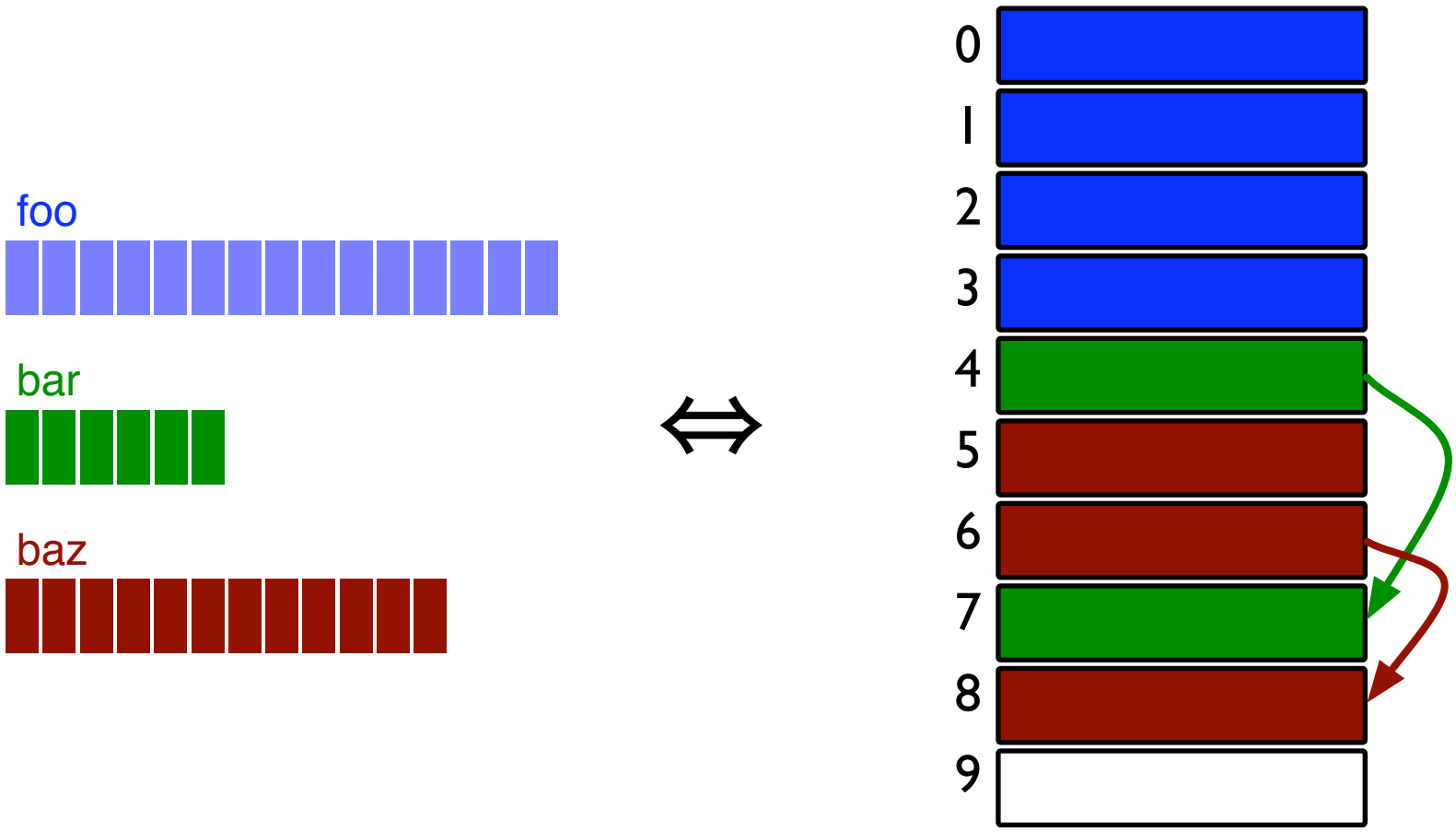
Sector = Offset-in-Cylinder % Sectors-per-Track


foo
bar
baz
⇔
0
1
2
3
4
5
6
7
8
9

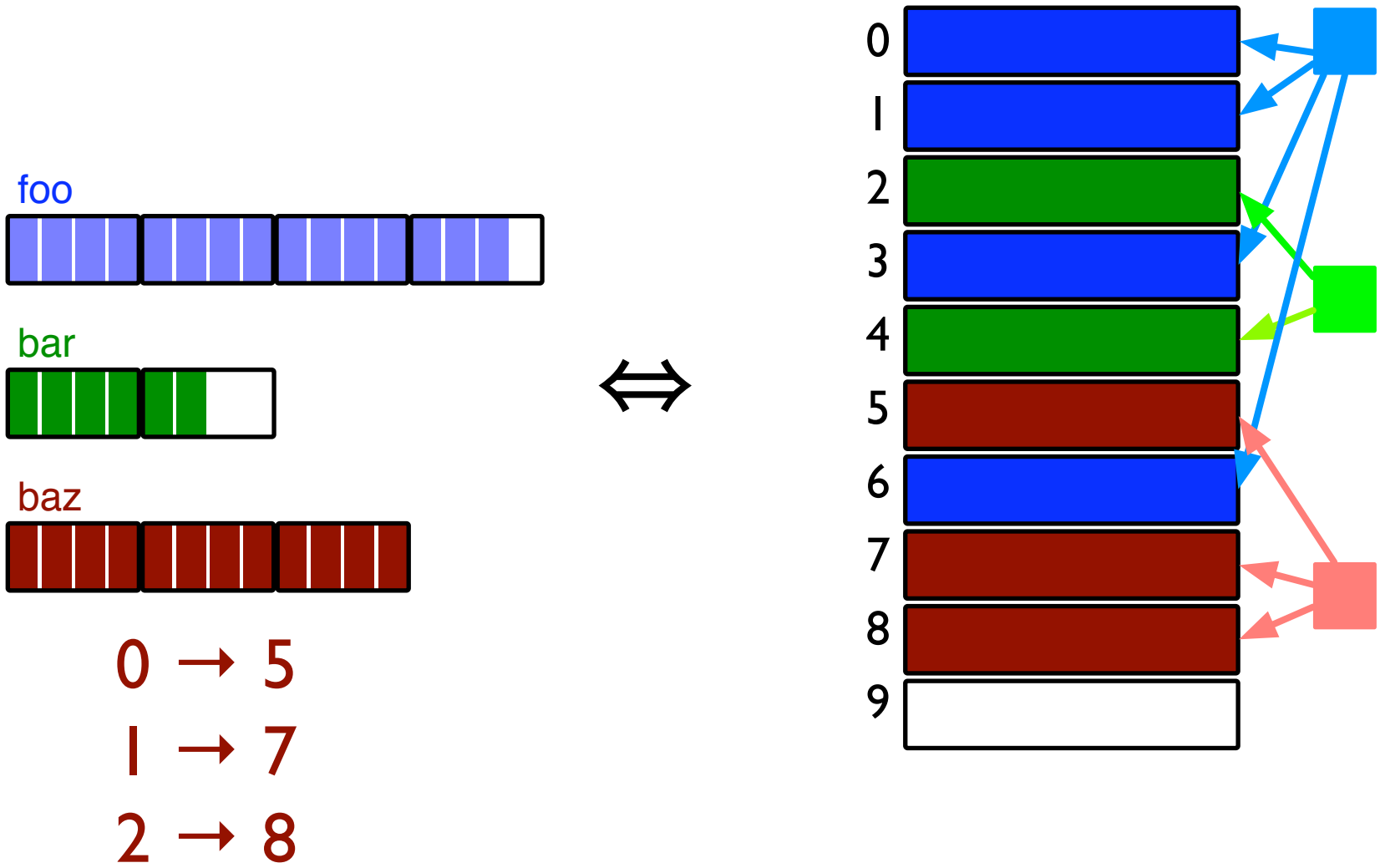
Contiguous allocation



Linked allocation



Indexed allocation

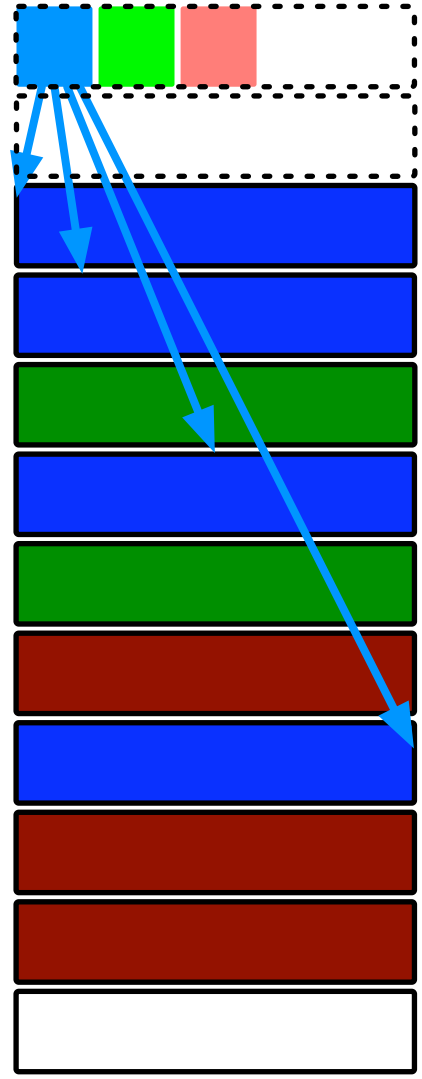


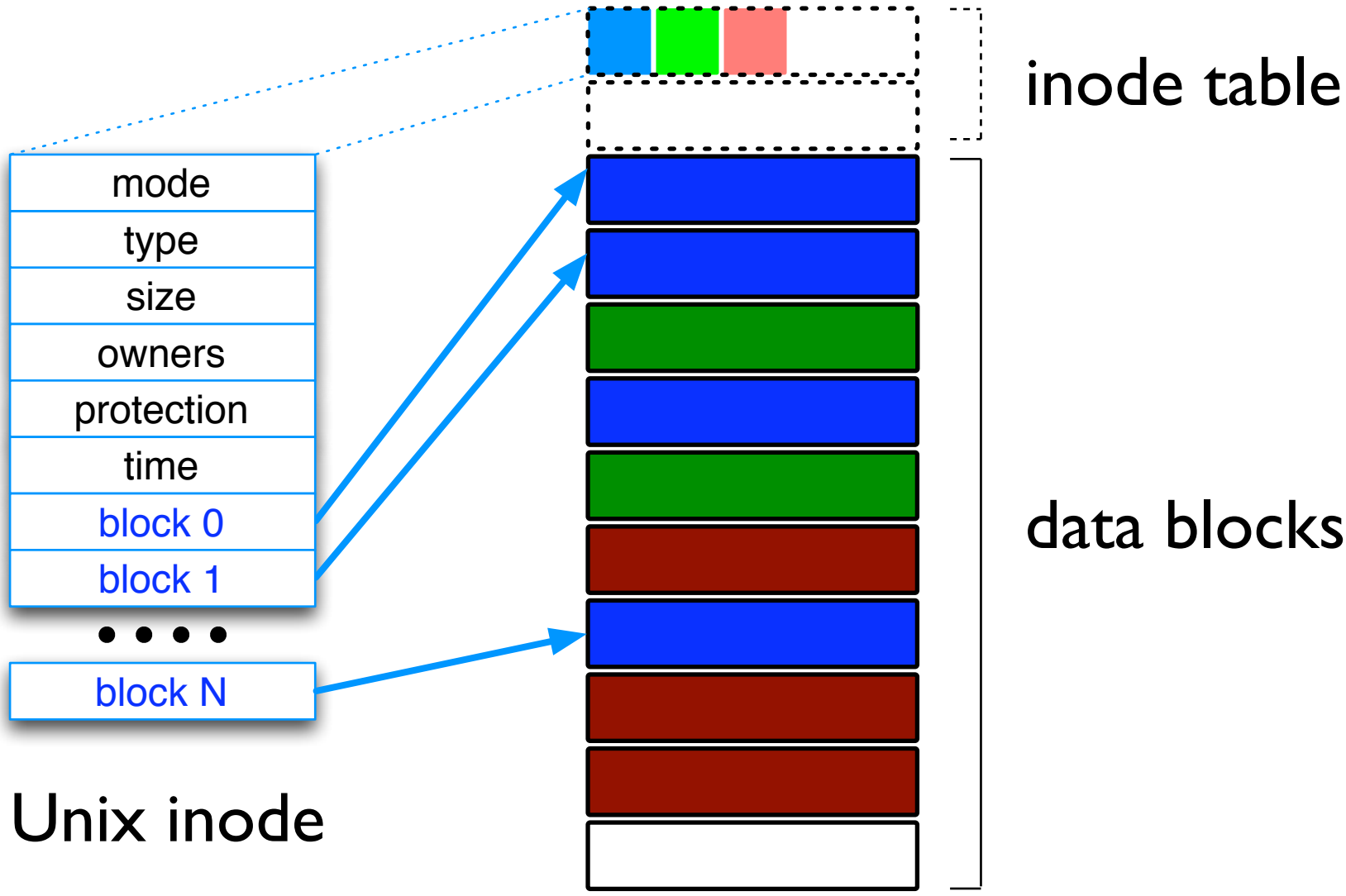
Example: 93-GB disk

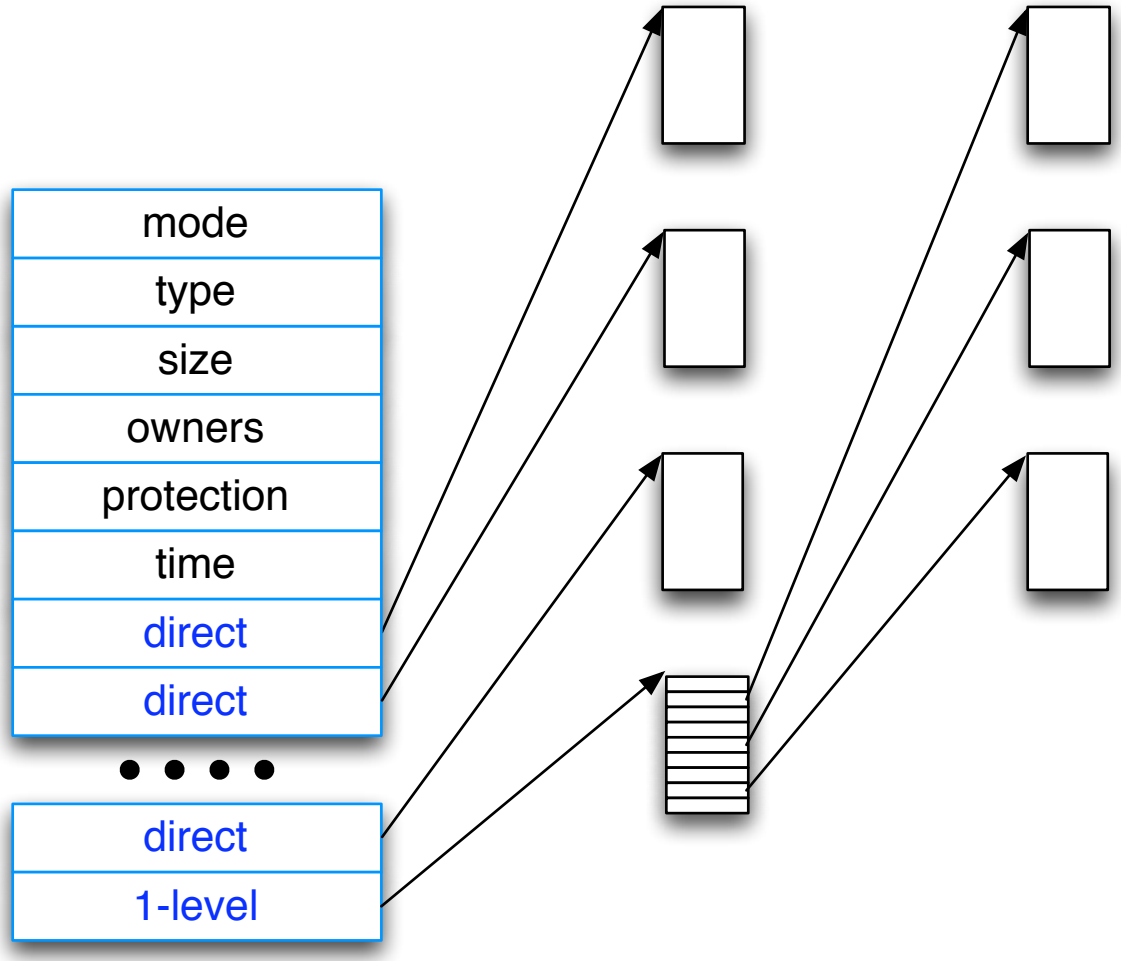
- 194,699,744 512-byte sectors
- 97,349,872 1-Kb blocks
- 371MB / 380,272 blocks [0.39%] for the index

  (lower bound, assuming 32-bit block pointers)

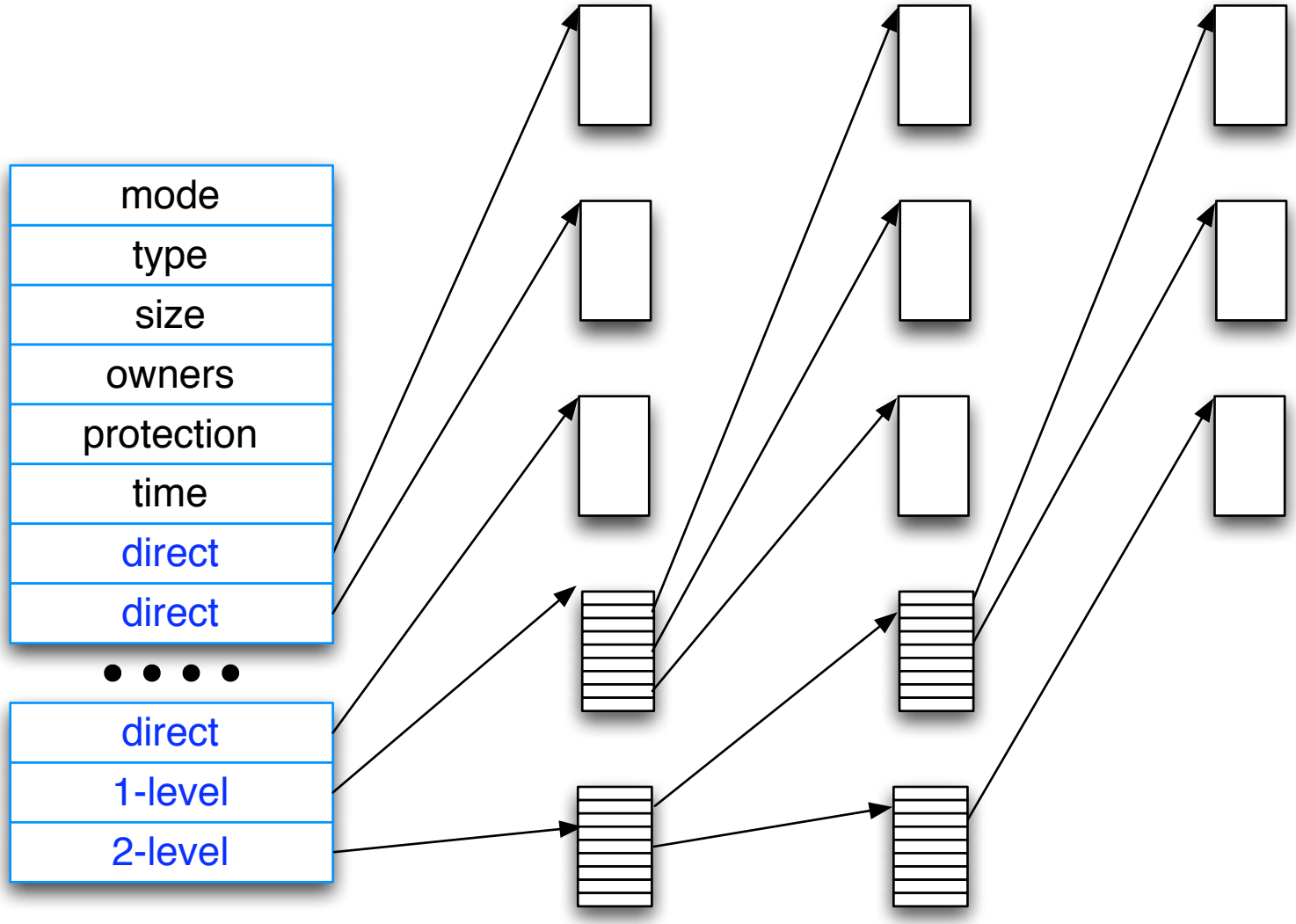
- Where to put index nodes?
- How big should they be?
- What if they fill up?
  - Chain
  - Hierarchy
  - Both



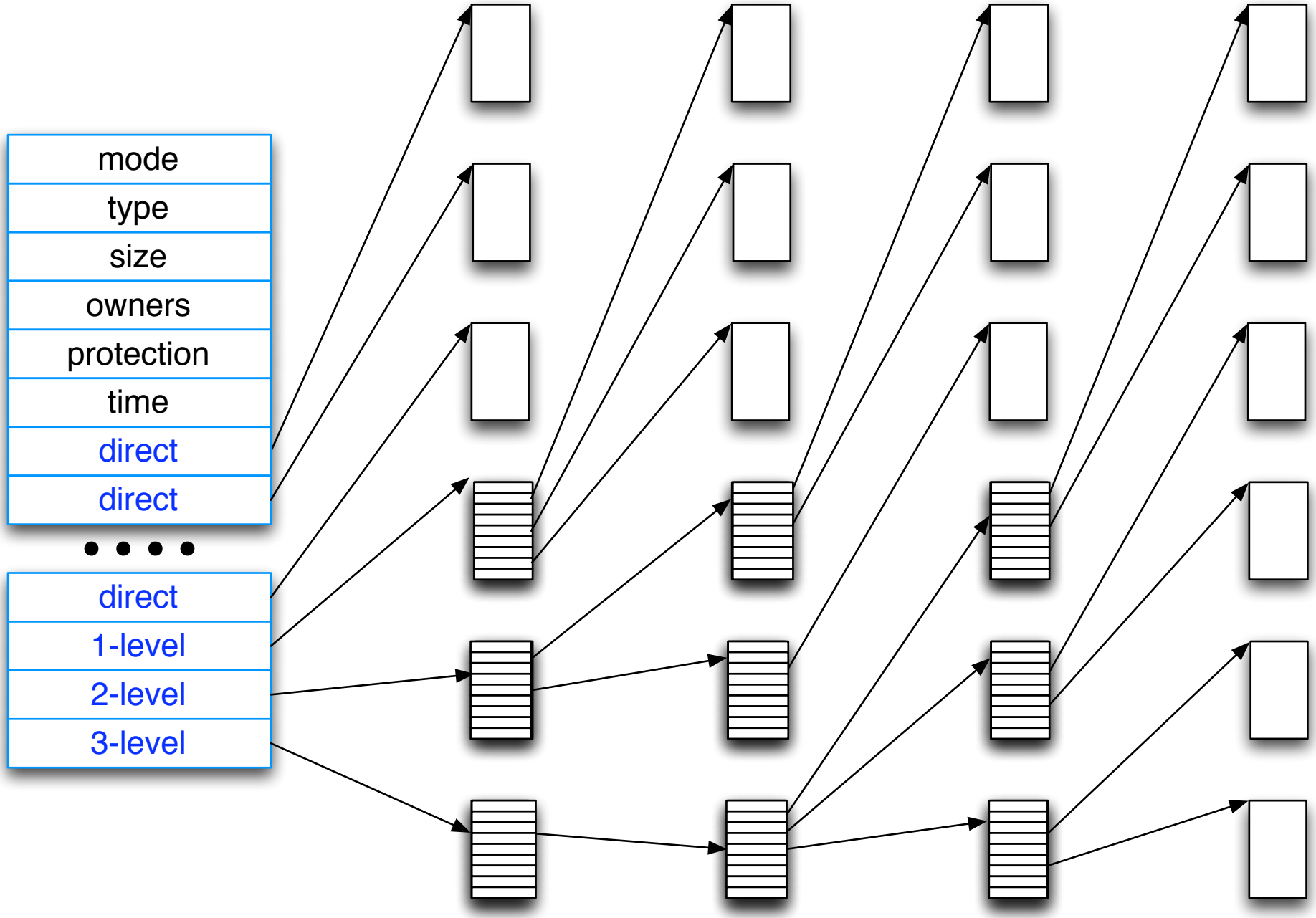

inode table
mode
type
size
owners
protection
time
block 0
block 1
block N
data blocks
Unix inode


mode
type
size
owners
protection
time
direct
direct
direct
1-level

Unix inode


mode
type
size
owners
protection
time
direct
direct
direct
1-level
2-level

Unix inode


mode
type
size
owners
protection
time
direct
direct
direct
1-level
2-level
3-level

Unix inode

Maximum file sizes

with 1-KB blocks

Level	Blocks	Bytes
Direct	10	10K
1-level	256	256K
2-level	65K	65M
3-level	16M	16G

(With 8-KB blocks, files can go up to 64TB)

Example: 93-Gb disk

- 97,349,872 1-Kb blocks
- 24,337,468 128-byte **inodes** (1 per 4 blocks)
- **inode table:** 3GB / 3,042,183 blocks [3.1%]

Example: df

```
$ df
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda2             10154020   6757468   2872432  71% /
/dev/sda1               256666     19396    224018   8% /boot
/dev/sda7             14503268   2122812  11631832  16% /local
/dev/sda5              4061540     74024   3777872   2% /tmp
/dev/sda3              4061572    243200   3608724   7% /var

$ df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/sda2            2621440  278819 2342621   11% /
/dev/sda1              66264      37   66227    1% /boot
/dev/sda7            3746240      26 3746214    1% /local
/dev/sda5            1048576     102 1048474    1% /tmp
/dev/sda3            1048576    2526 1046050    1% /var
```

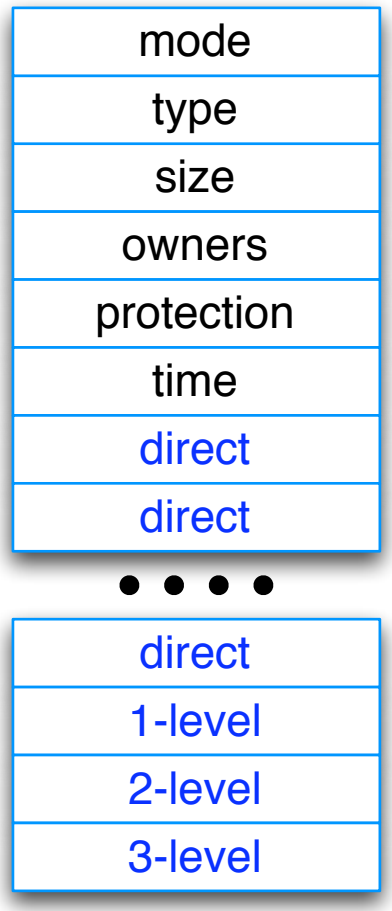
Free Space Management

- Free-space "list"
  - Shrinks upon allocation, grows upon deletion
- Implemented as
  - Chained free portions
  - Free block list in superblock
  - Bit map (NTFS, ext2, HFS+)

Example: 93-Gb disk

- 97,349,872 1-Kb blocks
- 24,337,468 inodes (1 per 4 blocks)
- inode table: 3,042,183 blocks [3.1%]
- **data block bitmap:** 11,512 blocks [0.01%]
- **inode bitmap:** 371 blocks [0.000004%]

What is missing?



Directories

- "Regular" files carrying name and inode
- Identified by type in the inode
- Can be kept sorted for faster access:
  - B-Tree or Hash table

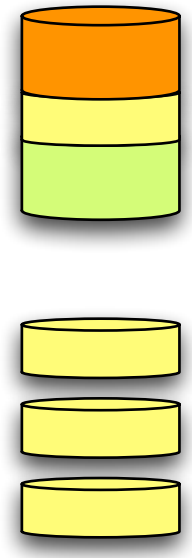
inode	len	name
0021	04	home
0123	07	dmitrii
0431	03	foo
0432	03	bar
0434	03	baz

Example: ls -i

```
$ ls -lai /
total 128
      2 drwxr-xr-x  24 root root  4096 2007-07-09 16:39 .
      2 drwxr-xr-x  24 root root  4096 2007-07-09 16:39 ..
  32705 drwxr-xr-x   2 root root  4096 2007-05-25 14:38 bin
1340865 drwxr-xr-x   3 root root  4096 2007-08-16 07:15 boot
   1180 drwxr-xr-x  14 root root 13920 2007-11-14 09:15 dev
  81761 drwxr-xr-x 136 root root  8192 2007-11-29 08:14 etc
  11209 drwxr-xr-x  12 root root     0 2007-12-04 07:35 home
1586145 drwxr-xr-x  16 root root  8192 2007-08-16 07:14 lib
      1 dr-xr-xr-x 113 root root     0 2007-07-09 04:22 proc
      1 drwxr-xr-x  11 root root     0 2007-07-09 04:22 sys
 915713 drwxrwxrwt  10 root root  4096 2007-12-05 04:14 tmp
 866657 drwxr-xr-x  12 root root  4096 2006-09-15 06:31 usr
1749665 drwxr-xr-x  15 root root  4096 2006-09-15 06:50 var
```

Big picture

- Disks can consist of **partitions**
  - each with its own file system
- Disks make up a **logical volume**
  - that can be used by one file system
- Multiple file systems can be **mounted**



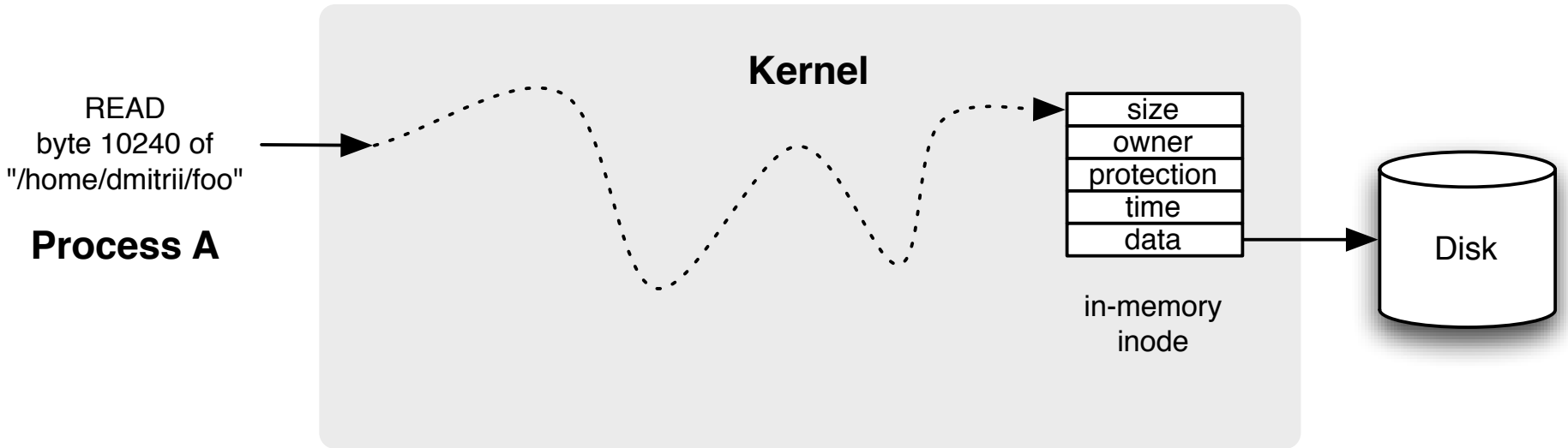
Inside a partition or logical volume

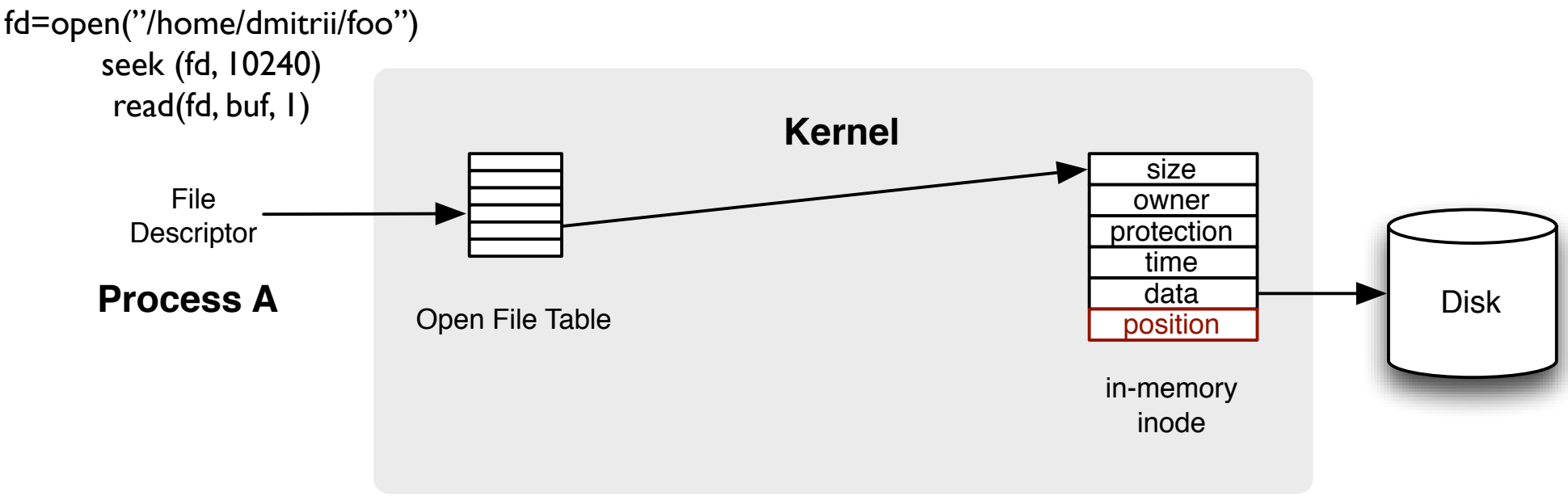
- **Boot block**
- **Superblock**
  - Size of file system, of inode table, of blocks
  - Number of free blocks, inodes
  - (Root inode)
- Free data block list
- Free inodes list
- Inode table
- Data blocks

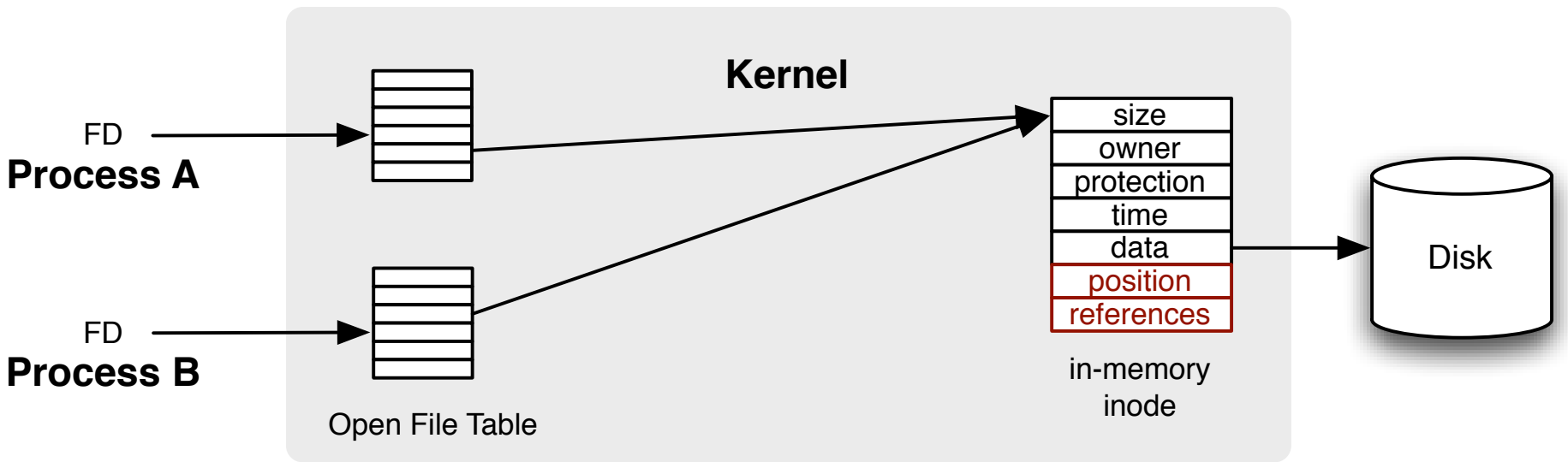
READ

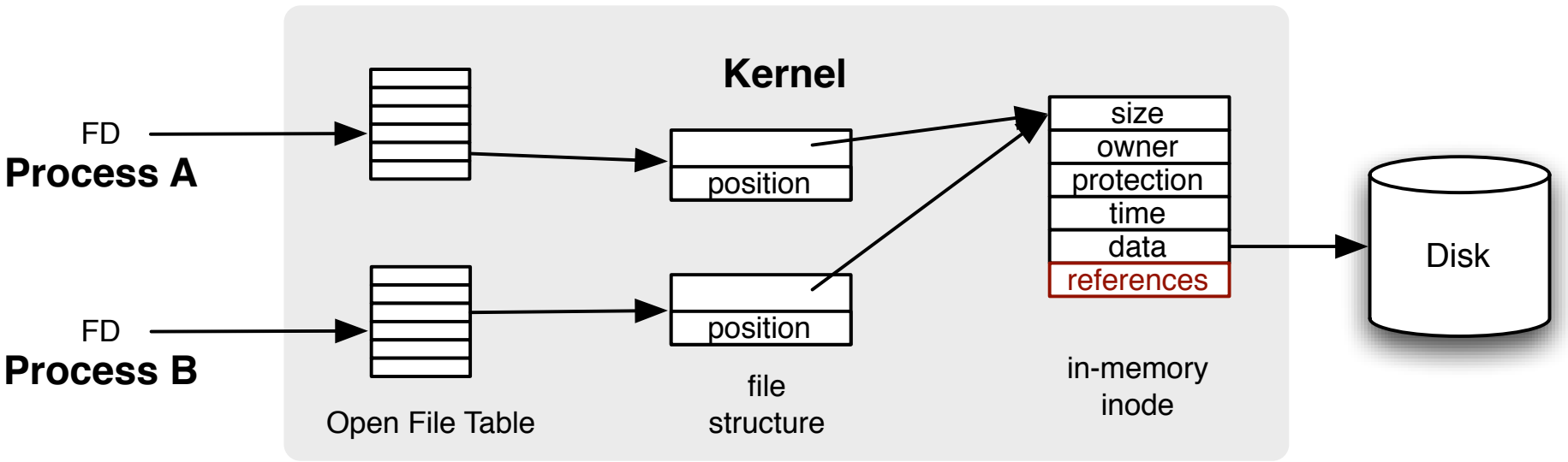
byte **10240** of

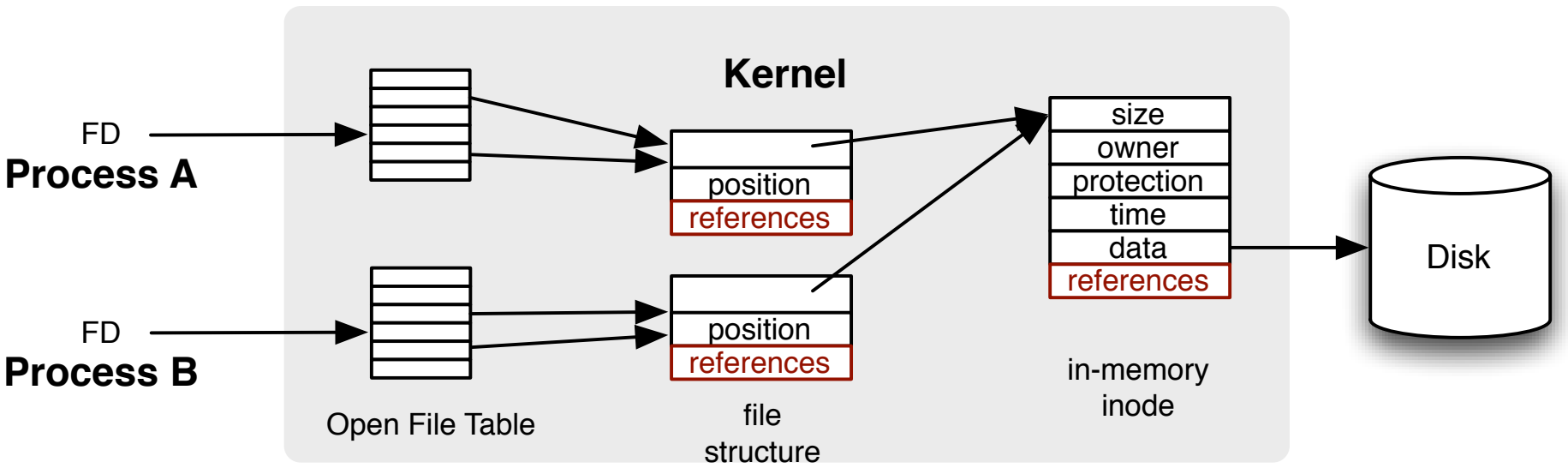
“/home/dmitrii/foo”


READ
byte 10240 of
"/home/dmitrii/foo"
Process A
Kernel
size
owner
protection
time
data
in-memory
inode
Disk


fd=open("/home/dmitrii/foo")
seek (fd, 10240)
read(fd, buf, 1)
File
Descriptor
Process A
Kernel
Open File Table
size
owner
protection
time
data
position
in-memory
inode
Disk


Kernel
FD
Process A
FD
Process B
Open File Table
size
owner
protection
time
data
position
references
in-memory
inode
Disk


Kernel
FD
Process A
FD
Process B
position
position
size
owner
protection
time
data
references
Disk
Open File Table
file
structure
in-memory
inode


Kernel
FD
Process A
FD
Process B
position
references
position
references
size
owner
protection
time
data
references
Disk
Open File Table
file
structure
in-memory
inode

Kernel memory objects

- Structs:
  - superblock (always cached)
  - inodes, files, dentry (partially cached)
- Buffer cache:
  - a unified buffer+cache for completed I/O

Anything can be a file

- Resources as files
  - Devices (/dev), processes (/proc), network interface (/net), display,...
- Hierarchical namespace
- Communication protocol