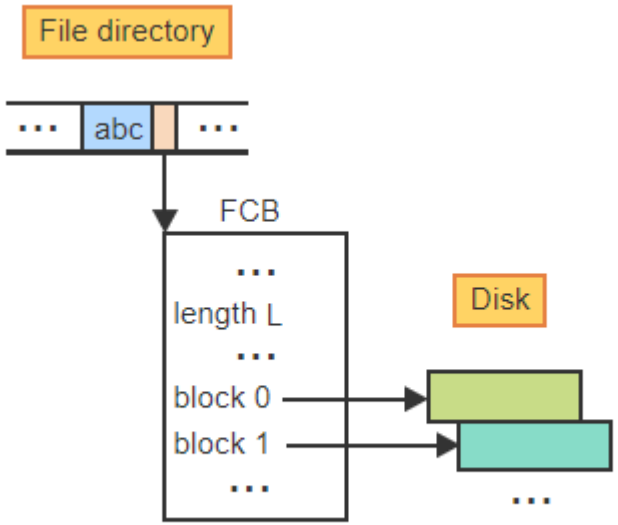
File System-2

Spring 2022

Operations on File

- Create and Destroy

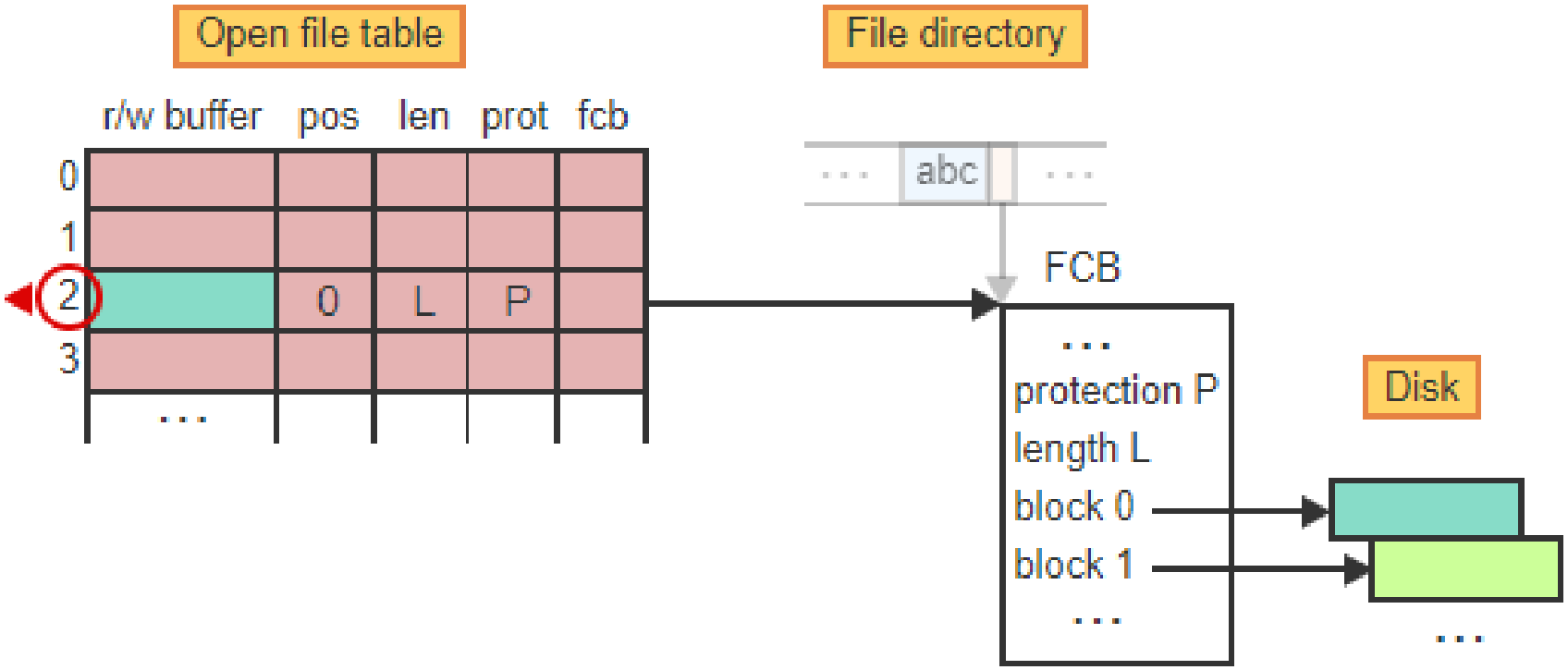


Operations on files

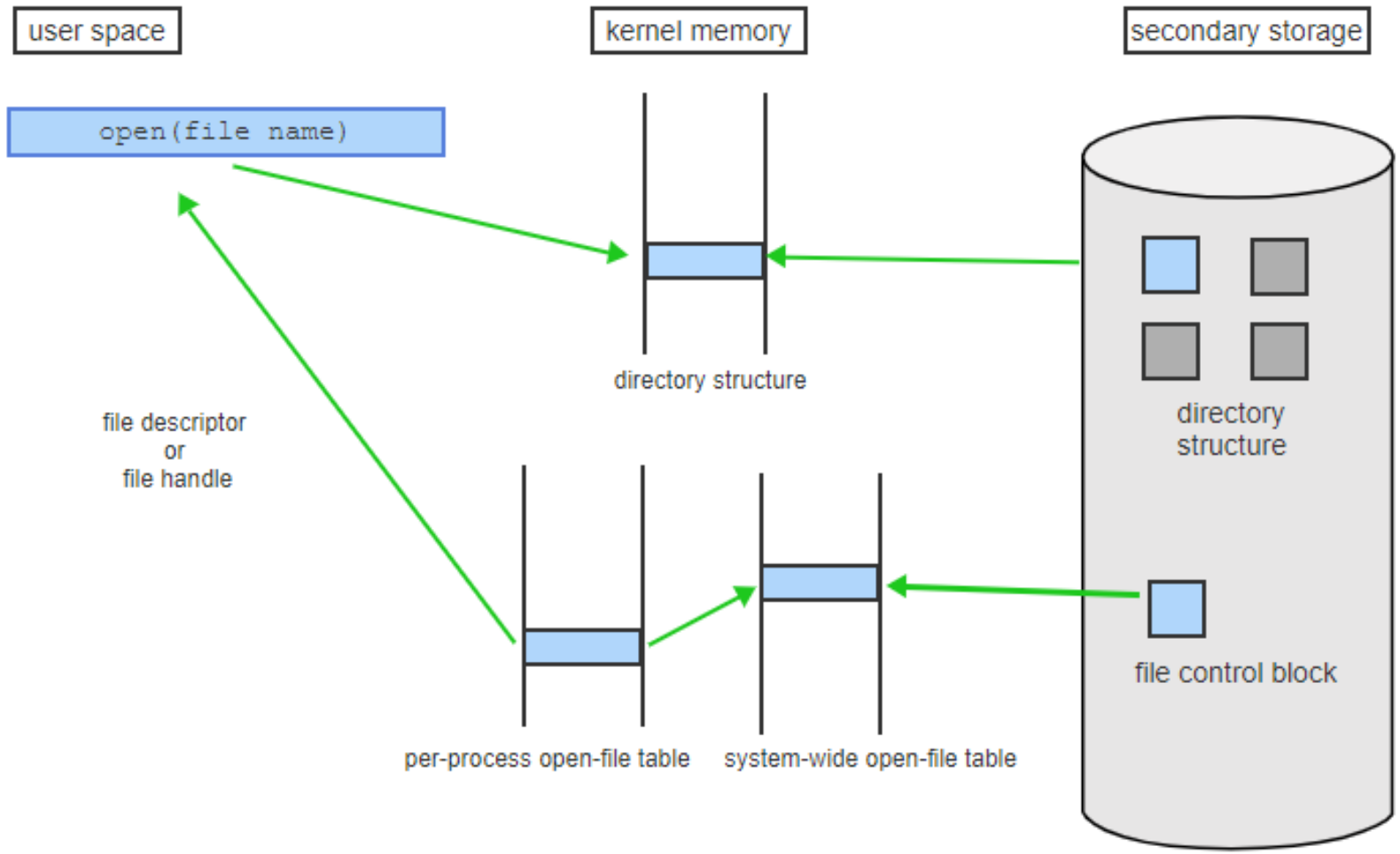
- Open a file
  - The ***open file table*** (***OFT***) is a data structure that keeps track of all files currently in use to facilitate efficient access to and manipulation of the files.

- The steps performed by the open operation include:
  - Verify access permission
  - Allocate a free entry in the OFT.
  - Allocate read/write buffers as necessary.
  - Copy relevant information from the FCB to the OFT entry(like file length and location on the disk).
  - Initialize required information in the OFT entry, (initial position of a sequentially accessed file).
  - Return the index of the allocated OFT entry to the calling process for subsequent access to the file.

Open a File


Open file table
File directory
r/w buffer
pos
len
prot
fcb
0
1
2
3
0
L
P
abc
FCB
protection P
length L
block 0
block 1
Disk

File Open

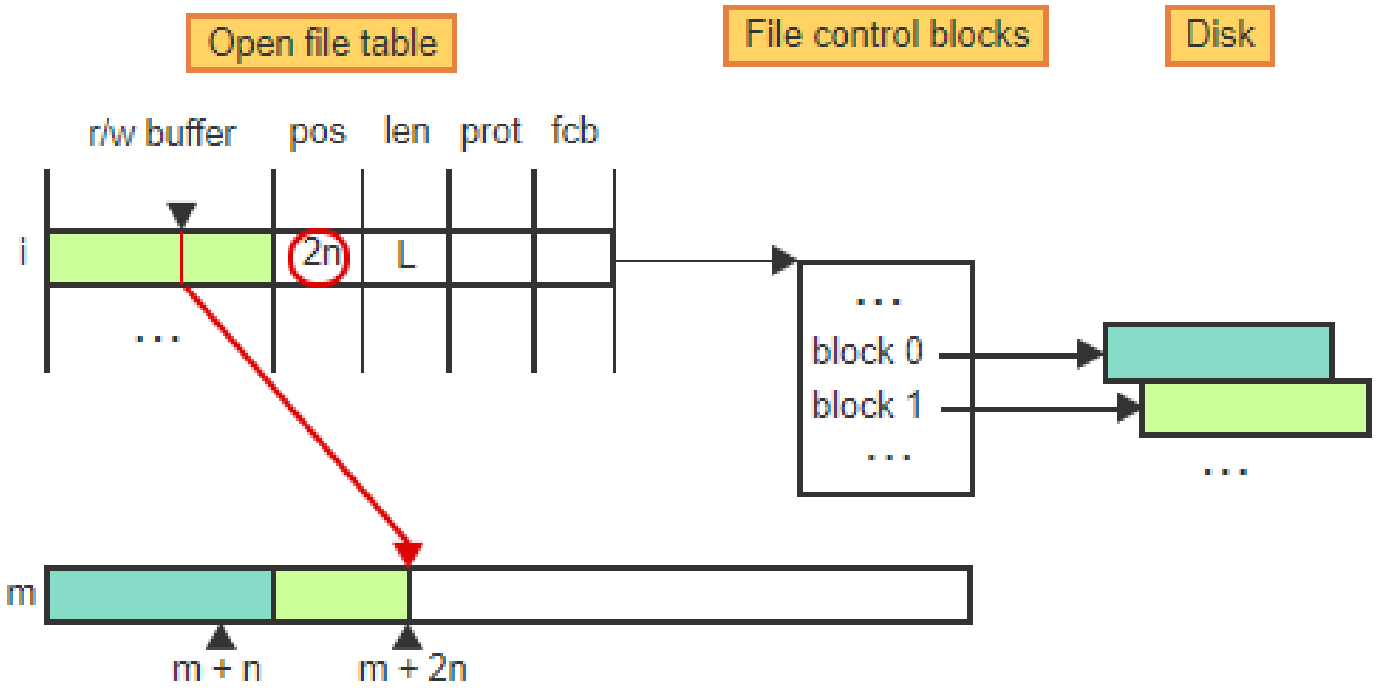

user space
kernel memory
secondary storage
open(file name)
directory structure
file descriptor
or
file handle
directory
structure
file control block
per-process open-file table
system-wide open-file table

Read a file

- A ***read file*** operation copies data from an open file to a specified area in main memory. The data may be accessed either directly, one record at a time, or sequentially, by specifying the number of bytes to be read next.

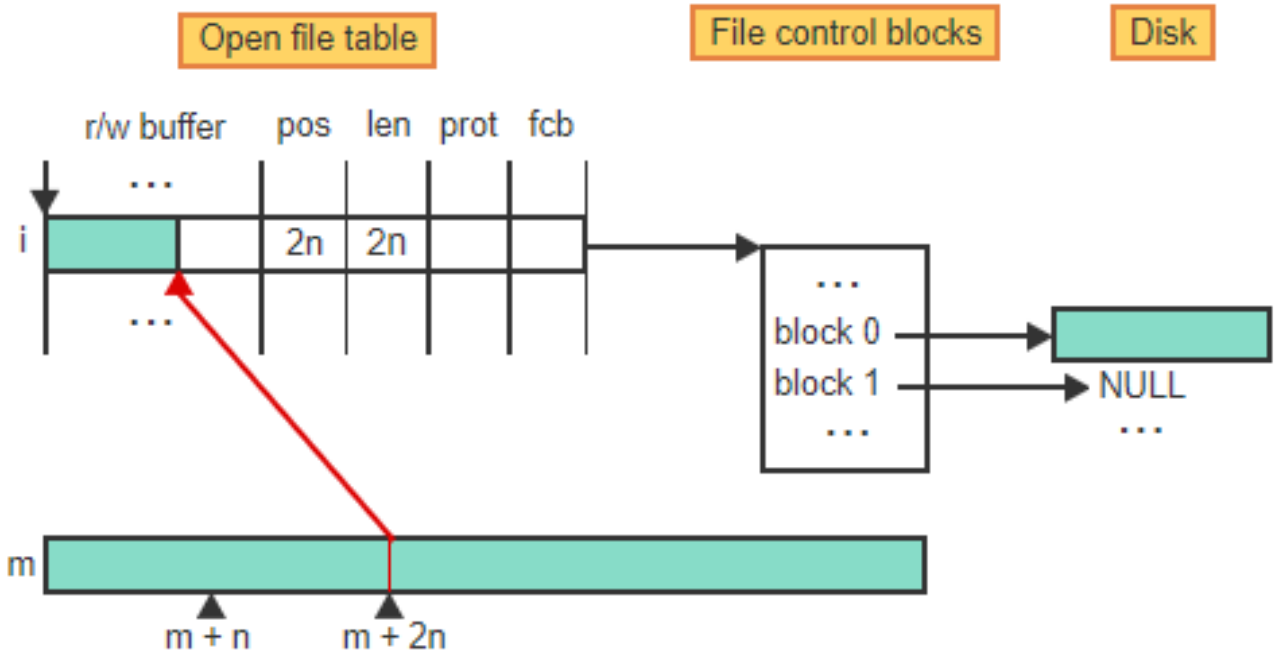
- A generic sequential read instruction has the form:
- read i, m, n ,
  - i is an OFT index corresponding to an open file,
  - n specifies the number of bytes to be read,
  - m is a location in memory

Read a file



read i, m, n
read i, m + n, n

Write a file



write i, m, n
write i, m + n, n

File Operations

- A ***seek*** operation moves the current position of an open file to a new specified position. A generic seek operation has the form:
  - seek i, k
- where i is an OFT index corresponding to an open file and k specifies the new position within the file.

- The ***close file*** operation reverses the effects of the open operation by saving the current state of the file in the FCB and freeing the OFT entry.
- The steps performed by the close operation include:
  - If the current content of the r/w buffer is modified, then save the buffer in the corresponding block on the disk.
  - Update the file length in the FCB.
  - Free the OFT entry.
  - Return the status of the close operation to the calling process.

Disk Allocation

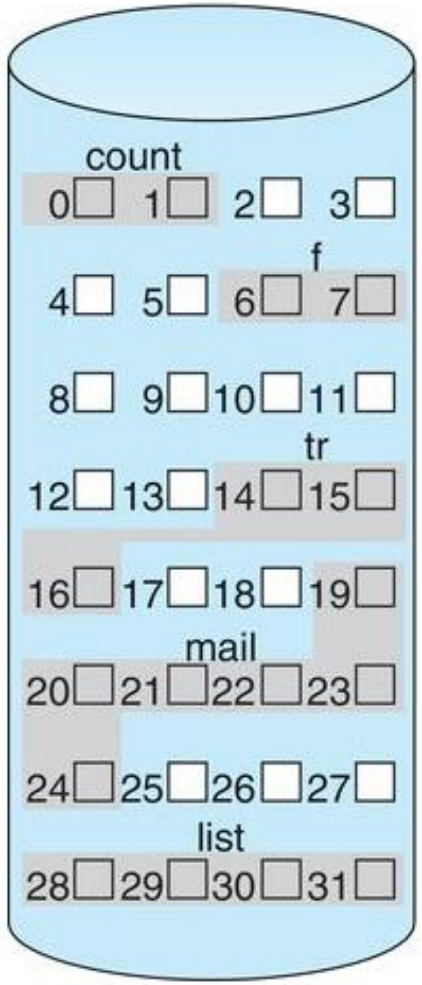
- Disk block
  - FS views the disk as a fixed sequence of bytes called blocks
  - numbered consecutively 0 through D - 1,
  - where D is the total number of blocks on the disk.

- The disk block allocation can vary the efficiency of file operations

Continuous Allocation

- Fast sequential access
- Fast Direct access

- Disk fragmentation
- Potential expansion



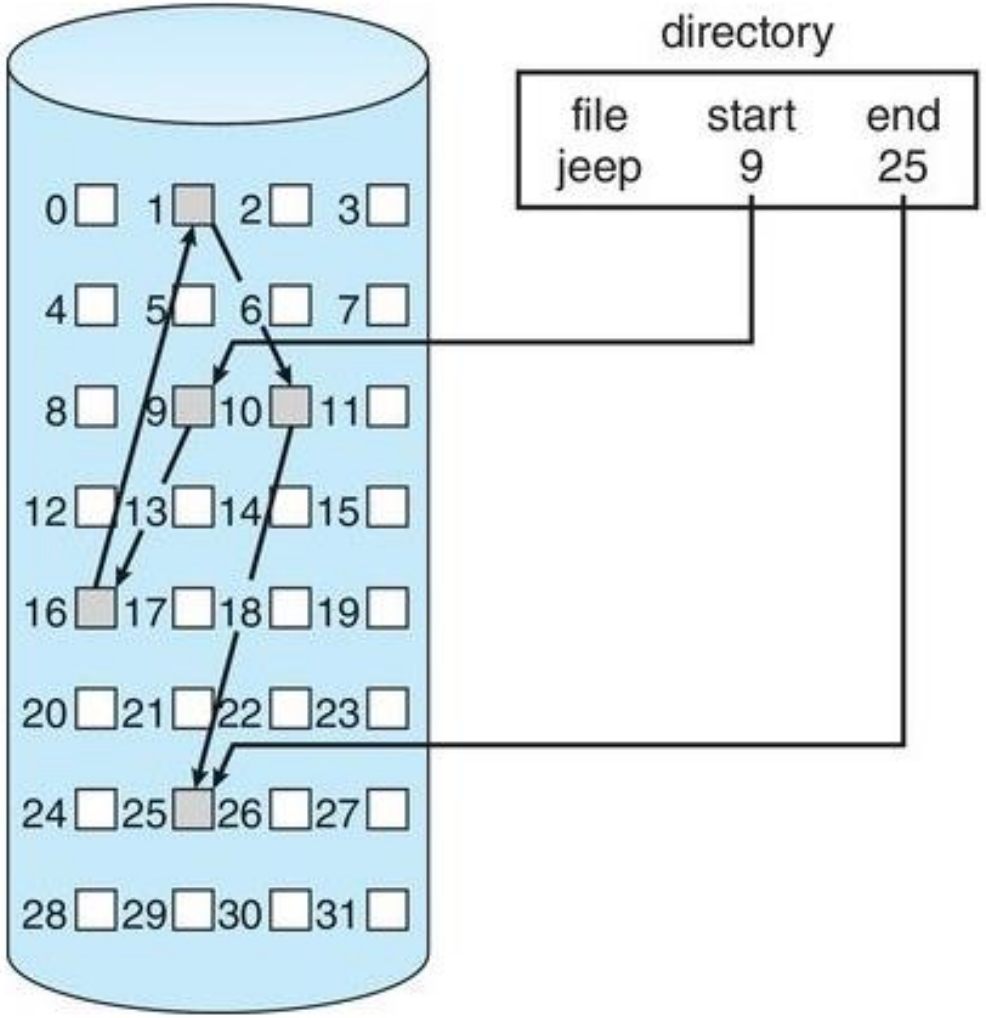
directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

Linked Allocation

- Easy file expansion

- Slow sequential access
- No direct access possible
- Pointer takes extra space

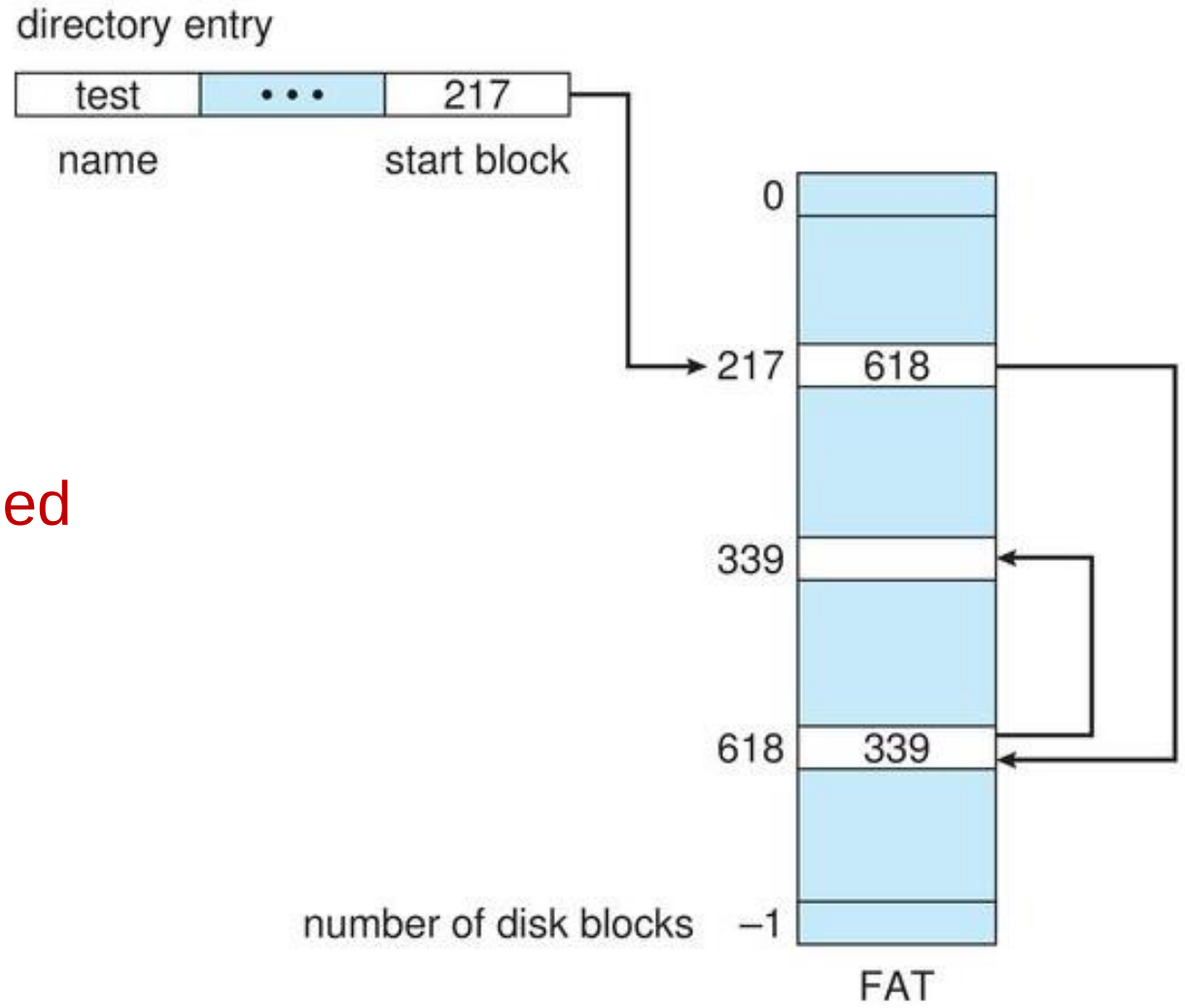


The usual solution to this problem is to collect blocks into multiples, called ***clusters***

Allocation Using a File Allocation Table (FAT)

- Efficient sequential access
- Can do Direct access

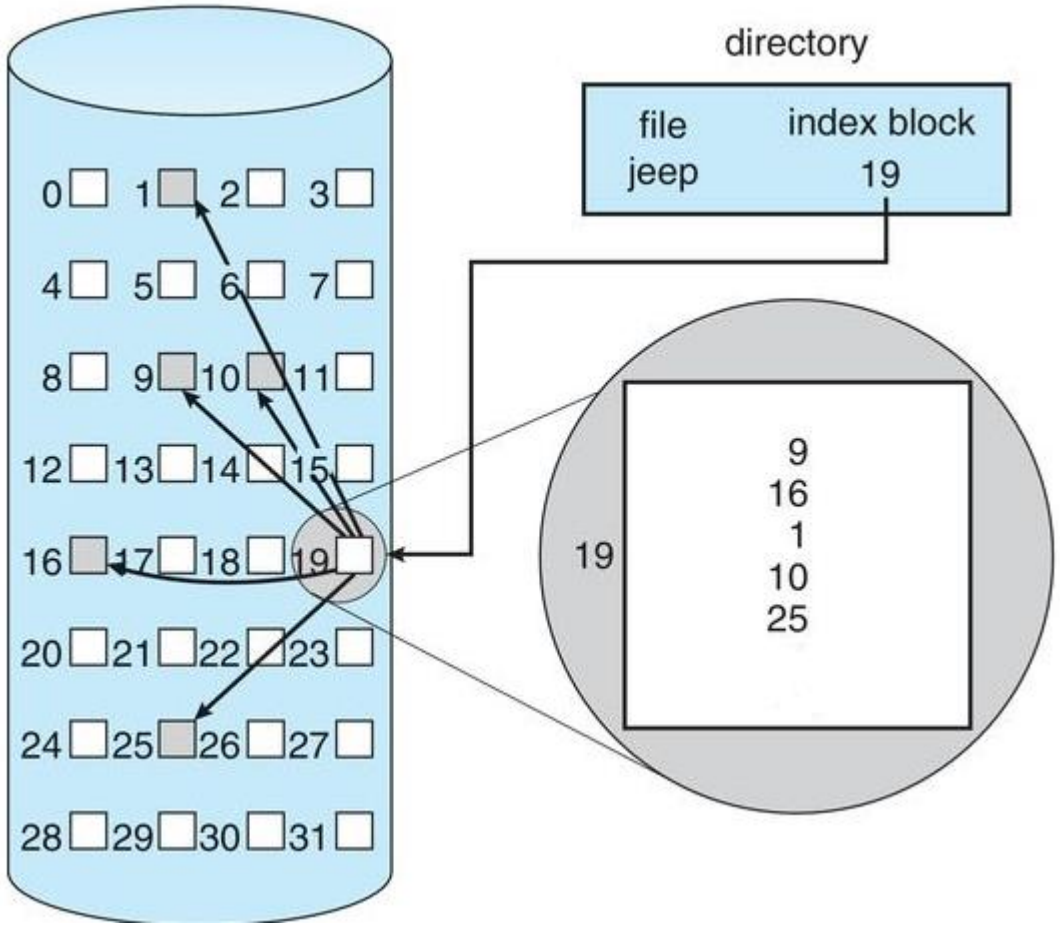
- Disk head seeks if not cached



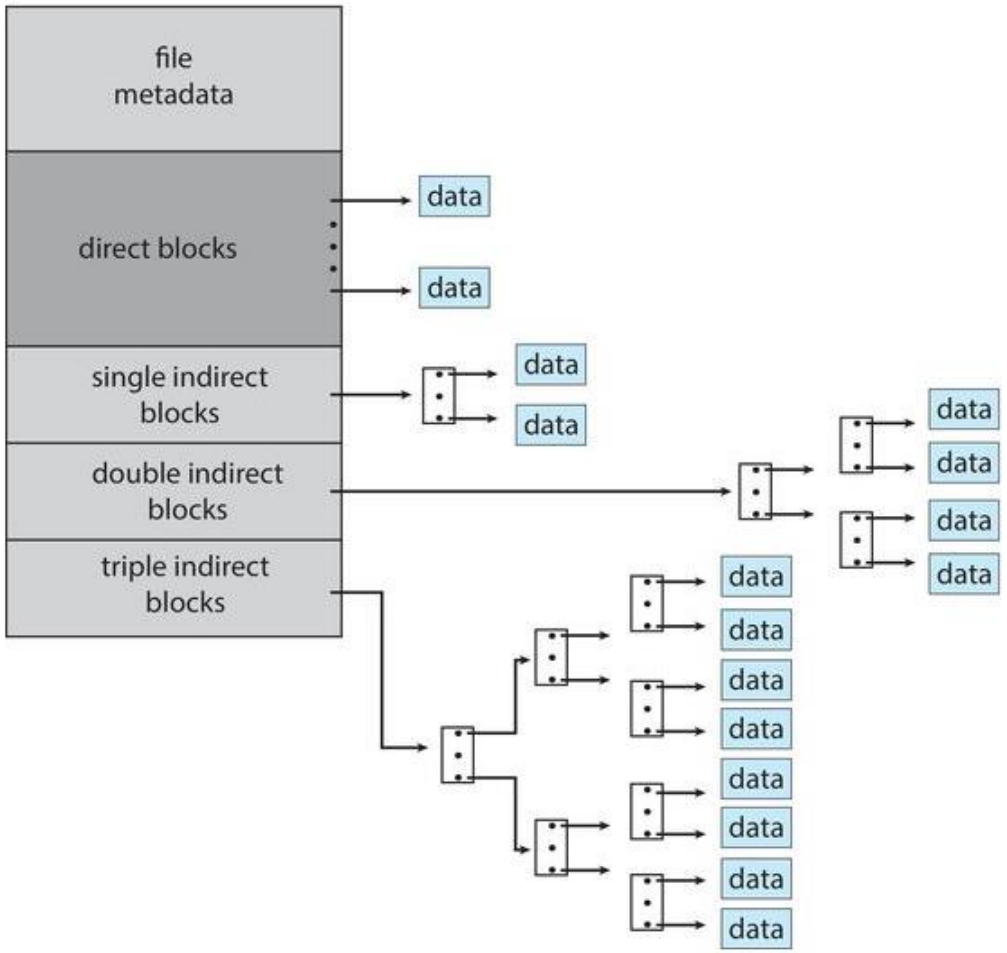
Indexed Allocation

- Fast sequential access
- Expansion by adding new blocks

- Decision on size of index table



Unix inode Example

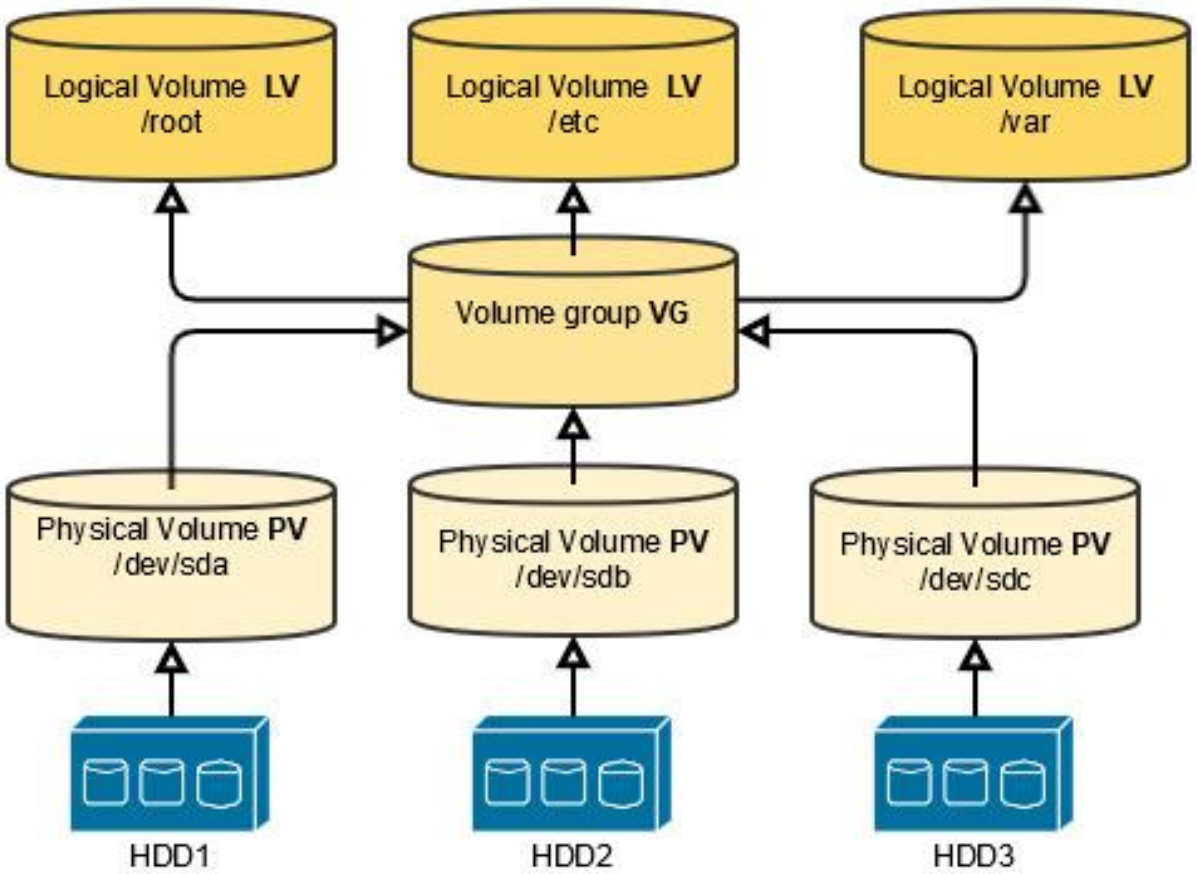


Free Space Management

- A ***bitmap*** is a data structure where each bit represents one disk block.
  - 1 indicates that the block is allocated
  - 0 indicates that the block is free
- Consider a disk where blocks 2, 5, 8 are free and the rest of the blocks are allocated

  110110110

Volume Management



To Do as on (04/19)

- Complete all participation activity in 8.4 and 8.5 in Weekly Quiz 6.
  - Please check your weekly quizzes if the points are not reflecting let me know.

- All weekly quizzes will be closing on April 30th so those who missed complete those.

- Programming Project 2– to do file operations using c language.
  - Question will be posted on (04/22)

- Finals study guide will be provided on (04/25)

- Extra credit (up to 3 grade points improvement and max up to 91%)
  - question will be posted on (04/22)