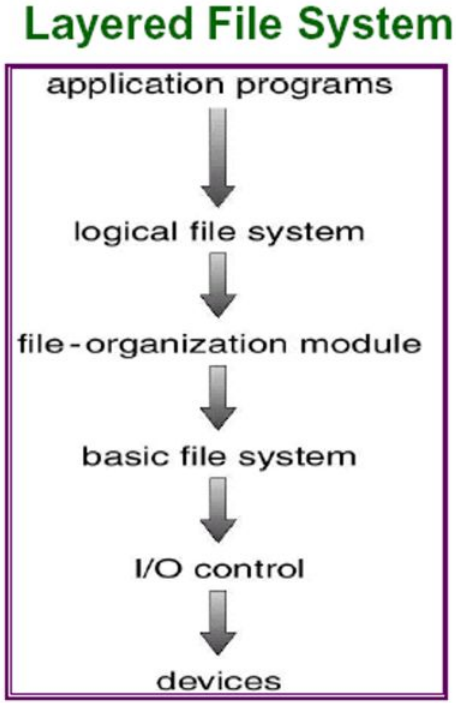
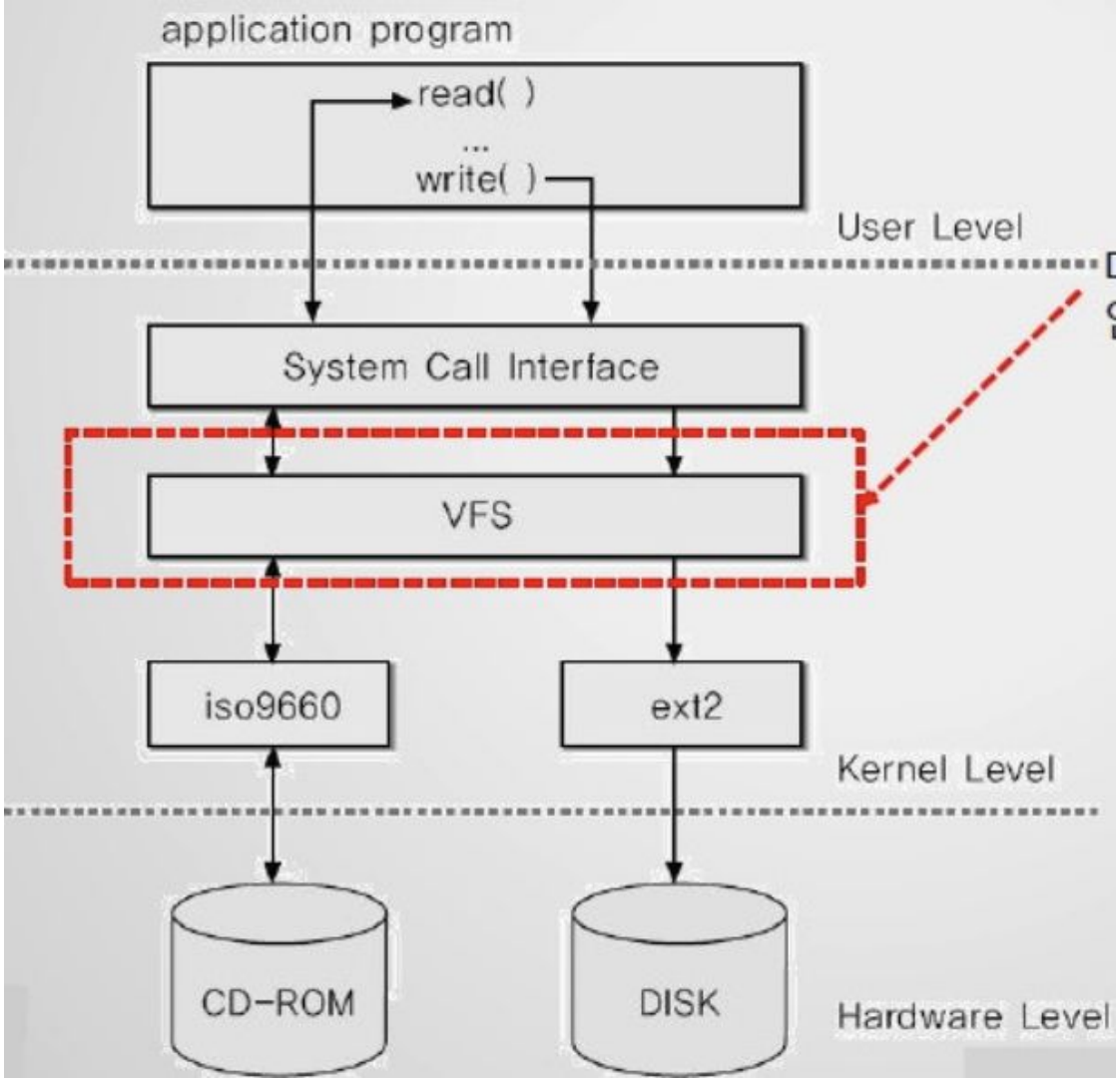
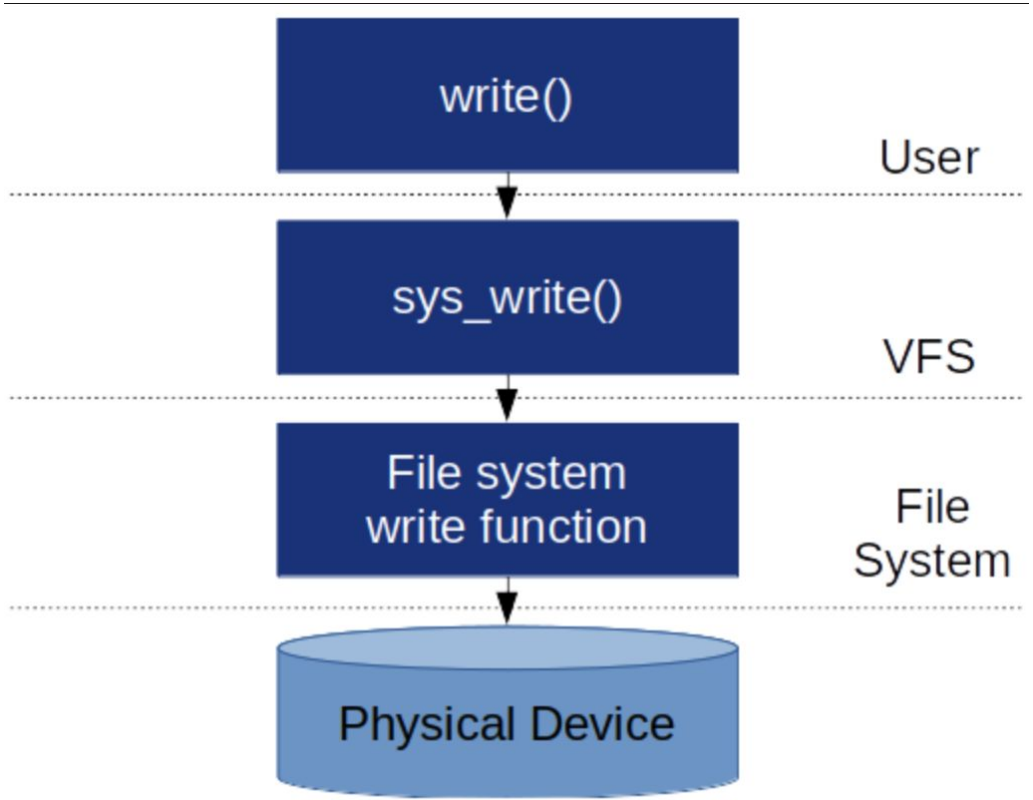
시스템프로그램 : Filesystem

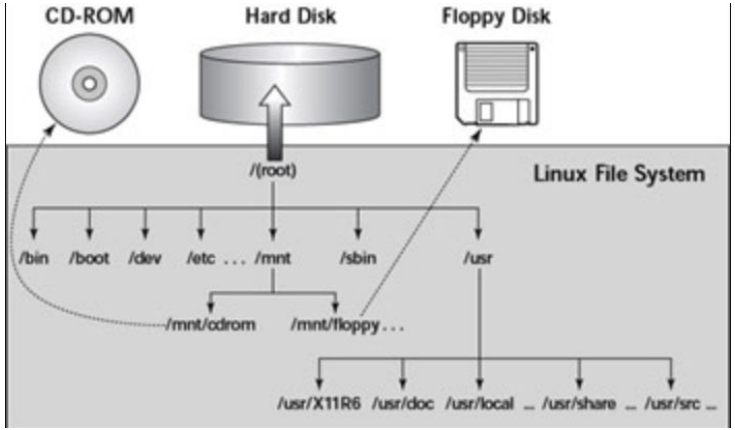
Layered File System



- Logical File System
  - Maintains file structure via FCB (file control block)
- File organization module
  - Translates logical block to physical block
- Basic File system
  - Converts physical block to disk parameters (drive 1, cylinder 73, track 2, sector 10 etc)
- I/O Control
  - Transfers data between memory and disk

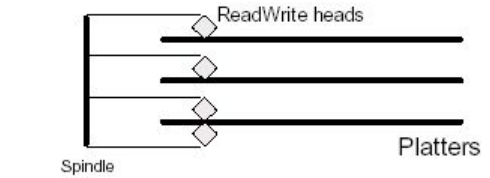

application program
read( )
...
write( )
User Level
System Call Interface
VFS
iso9660
ext2
Kernel Level
CD-ROM
DISK
Hardware Level

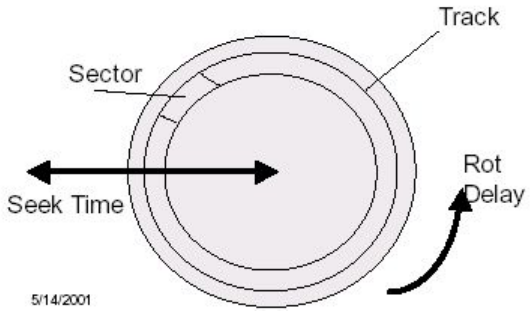

write()
sys_write()
File system
write function
Physical Device
User
VFS
File
System


CD-ROM
Hard Disk
Floppy Disk
/(root)
Linux File System
/bin
/boot
/dev
/etc
/mnt
/sbin
/usr
/mnt/cdrom
/mnt/floppy
/usr/X11R6
/usr/doc
/usr/local
/usr/share
/usr/src

Physical Disk Structure

- Parameters to read from disk:
  - cylinder(=track) #
  - platter(=surface) #
  - sector #
  - transfer size

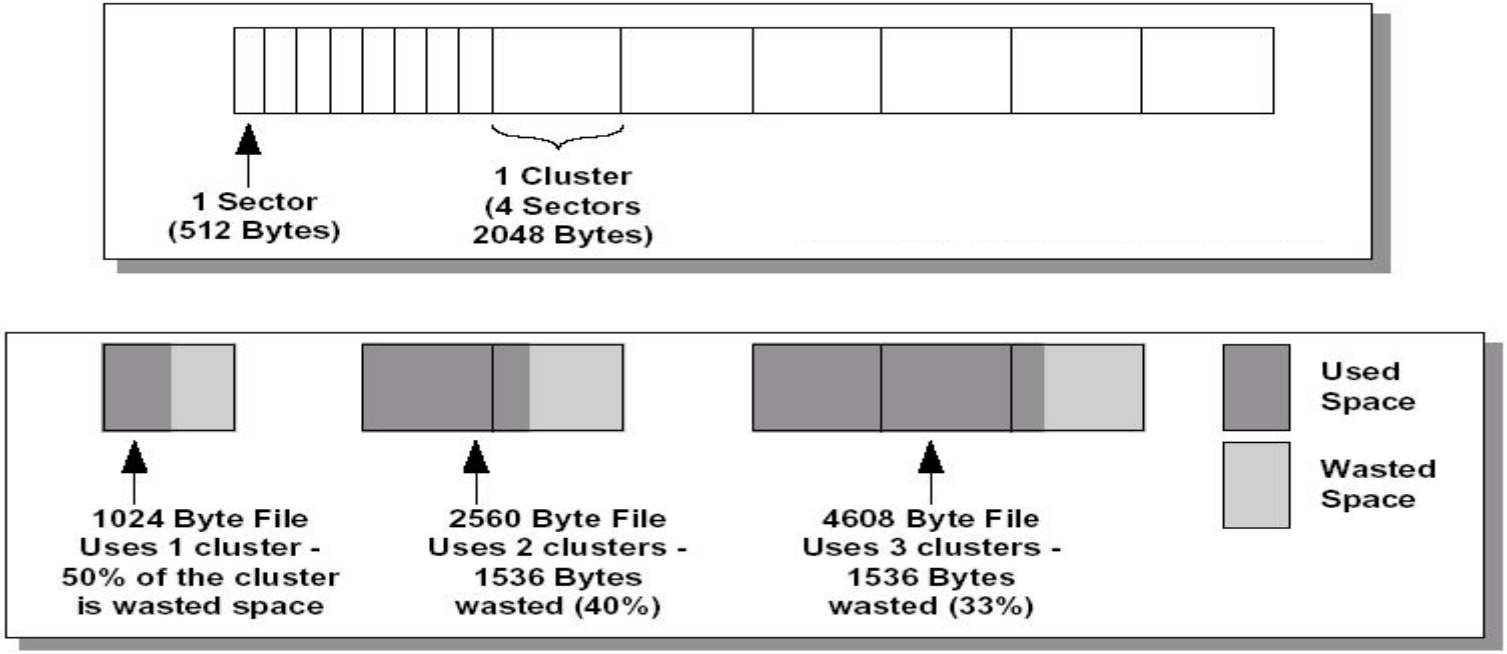




File system Units

- **Sector** – the smallest unit that can be accessed on a disk (typically 512 bytes)
- **Block(or Cluster)** – the smallest unit that can be allocated to construct a file
- What's the actual size of 1 byte file on disk?
  - takes at least one cluster,
  - which may consist of 1~8 sectors,
  - thus 1byte file may require ~4KB disk space.

Sector~Cluster~File layout



FCB – File Control Block

- Contains file attributes + block locations
  - Permissions
  - Dates (create, access, write)
  - Owner, group, ACL (Access Control List)
  - File size
  - Location of file contents
- UNIX File System I-node
- FAT/FAT32 part of FAT (File Alloc. Table)
- NTFS part of MFT (Master File Table)

Partitions

- Disks are broken into one or more partitions.
- Each partition can have its own file system method (UFS, FAT, NTFS, …).

A Disk Layout for A File System

- Super block defines a file system
  - size of the file system
  - size of the file descriptor area
  - start of the list of free blocks
  - location of the FCB of the root directory
  - other meta-data such as permission and times
- Where should we put the boot image?

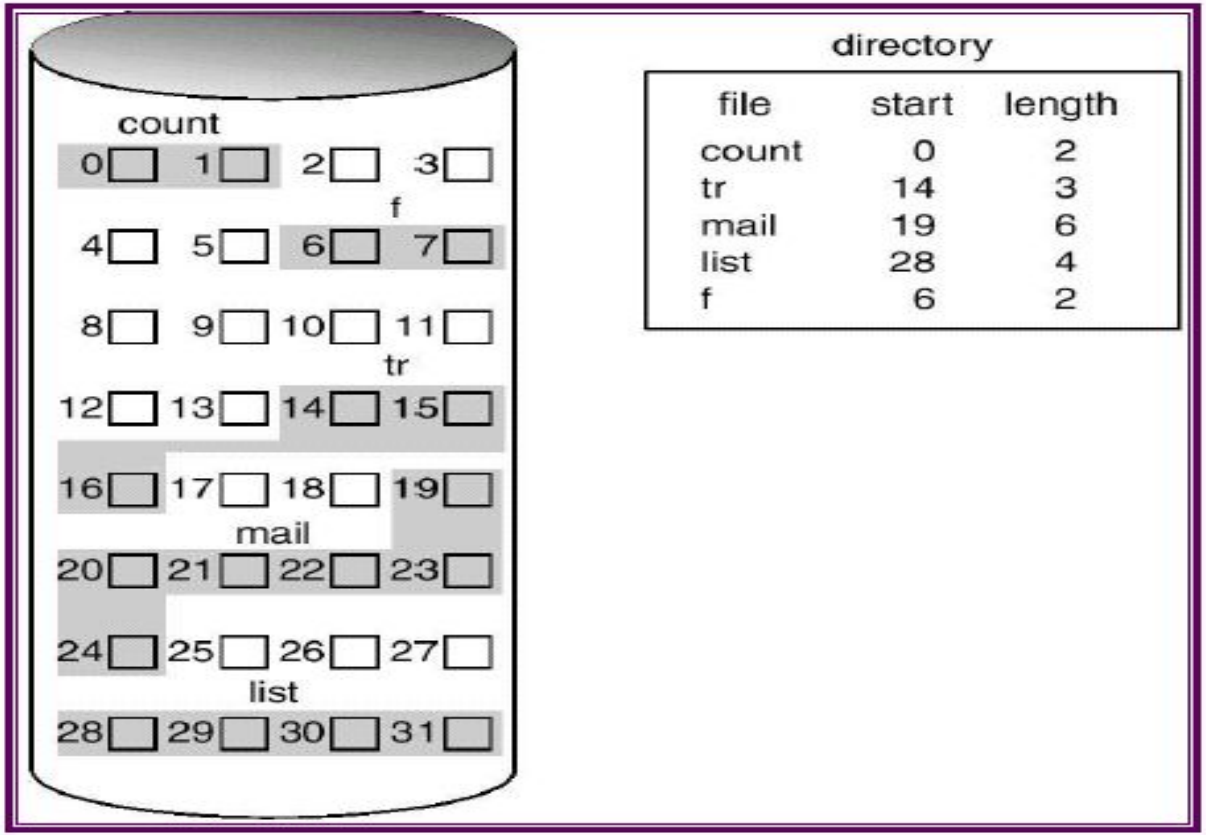
Boot block

- Dual Boot
  - Multiple OS can be installed in one machine.
  - How system knows what/how to boot?
- Boot Loader
  - Understands different OS and file systems.
  - Reside in a particular location in disk.
  - Read Boot Block to find boot image.

Block Allocation

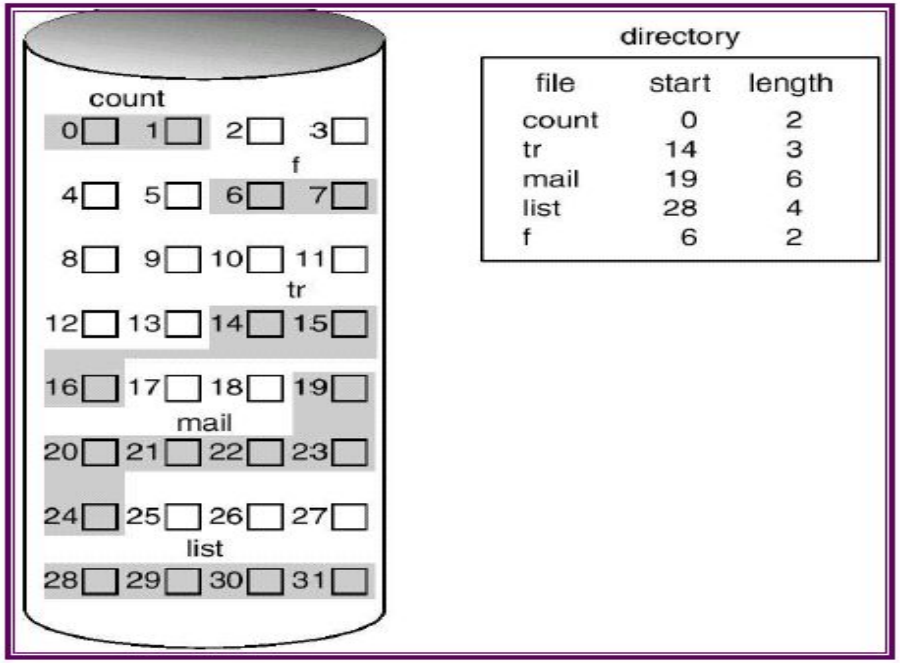
- Contiguous allocation
- Linked allocation
- Indexed allocation

Contiguous Block Allocation



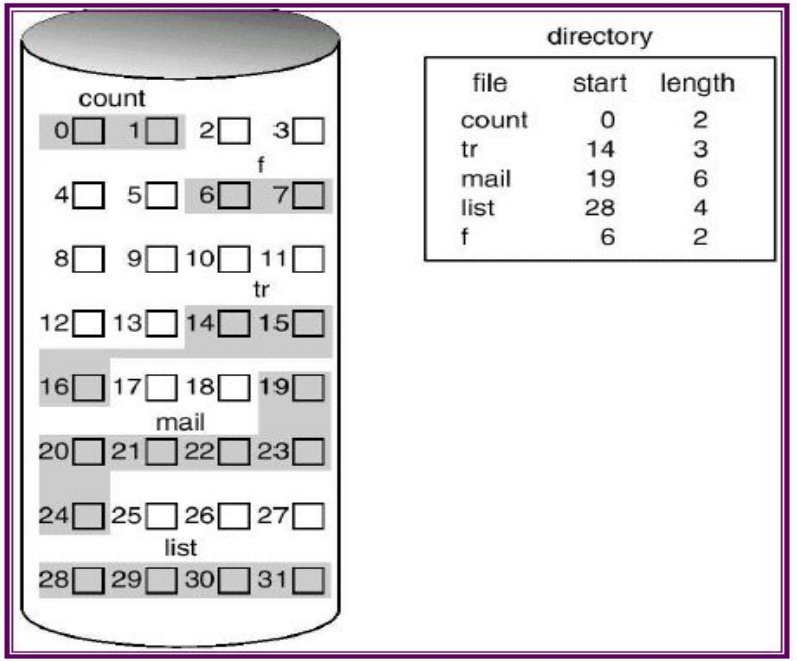
Contiguous Block Allocation

- Pros:
  - Efficient read/seek. Why?
    - disk location for both sequential & random access can be obtained instantly.
    - Spatial locality in disk

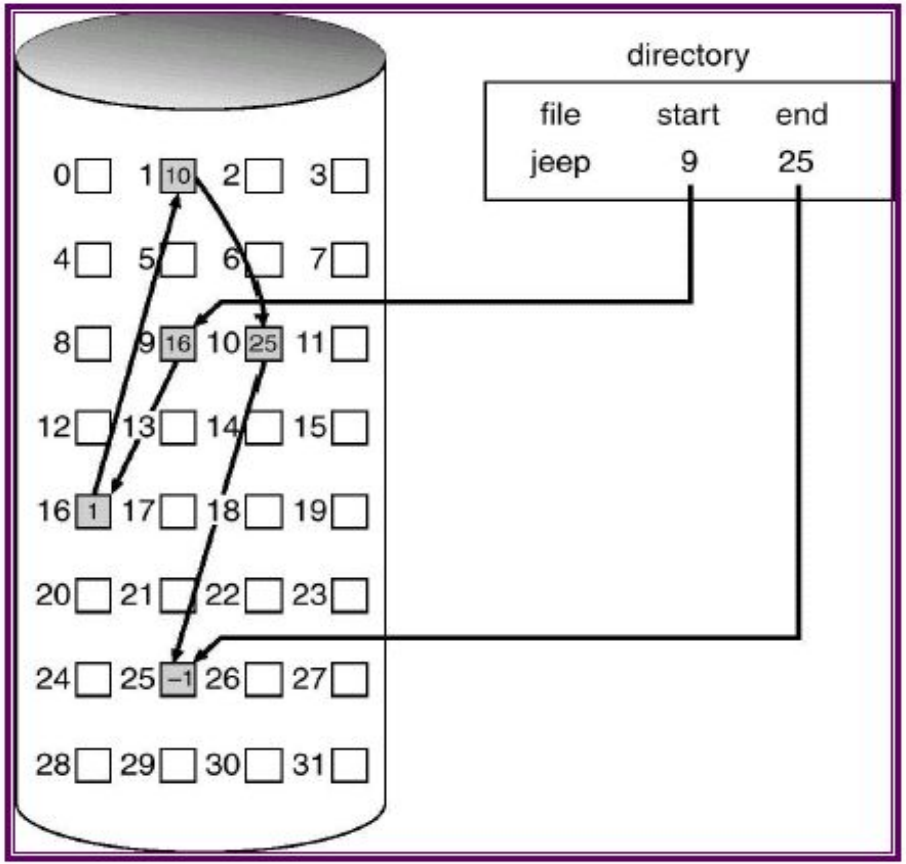


Contiguous Block Allocation

- Pros:
  - Efficient read/seek. Why?
    - disk location for both sequential & random access can be obtained instantly.
    - Spatial locality in disk
- Cons:
  - When creating a file, we don't know how many blocks may be required…
    - what happens if we run out of contiguous blocks?
  - Disk fragmentation!

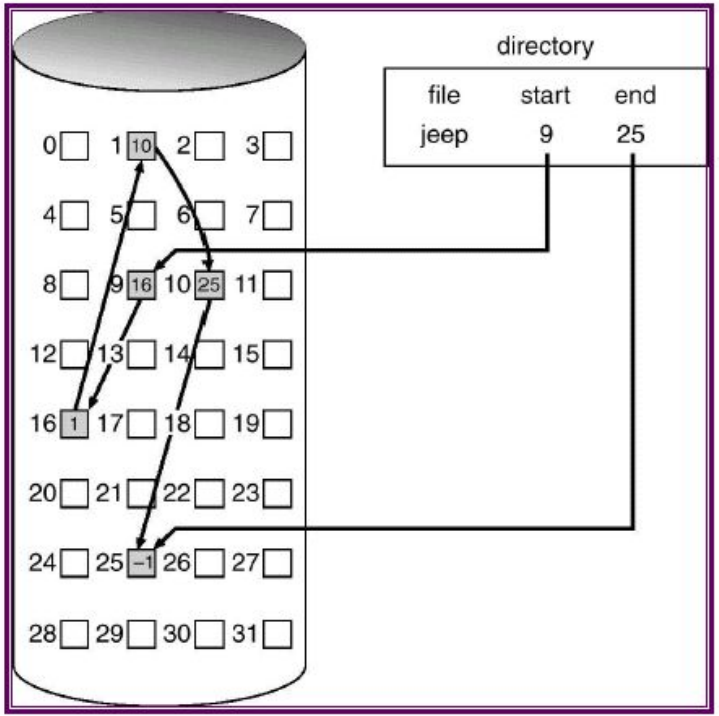


Linked Block Allocation



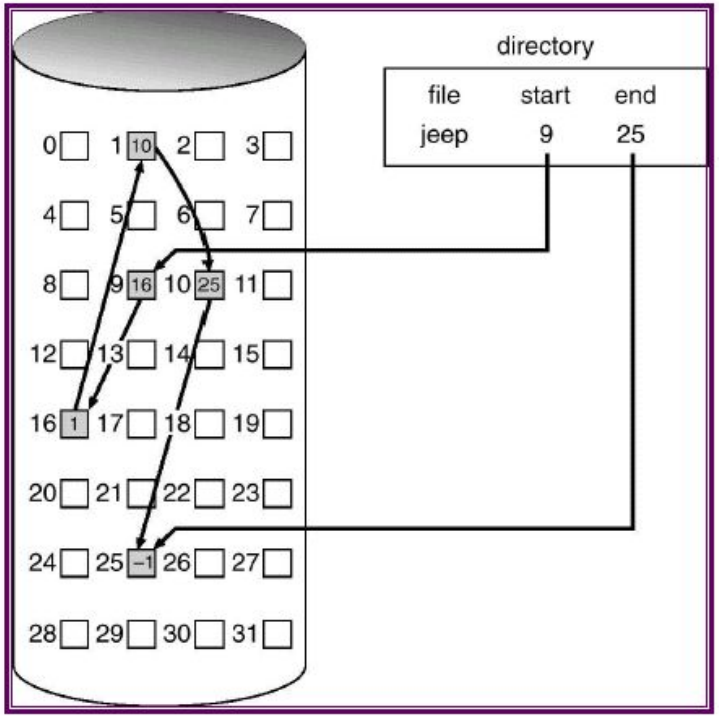
Linked Block Allocation

- Pros:
  - Less fragmentation
  - Flexible file allocation



Linked Block Allocation

- Pros:
  - Less fragmentation
  - Flexible file allocation
- Cons:
  - Sequential read requires disk seek to jump to the next block. (Still not too bad…)
  - Random read will be very inefficient!!
  - O(n) time seek operation (n = # of blocks in the file)



Indexed Block Allocation

- Maintain an array of pointers to blocks.
- Random access becomes as easy as sequential access!
- UNIX File System