Linux – The File System and VFS

Hard disk

Introduction

- Magnetic Disk Structure
  - Platter - 1.8-3.5 inches in diameter
  - Array of disk blocks
  - Cylinder, track, sector
  - Translation is difficult
    - Defective sectors
    - No. of sectors / track is not const.
  - Const. Linear Velocity (CLV)
  - Const. Angular Velocity (CAV)

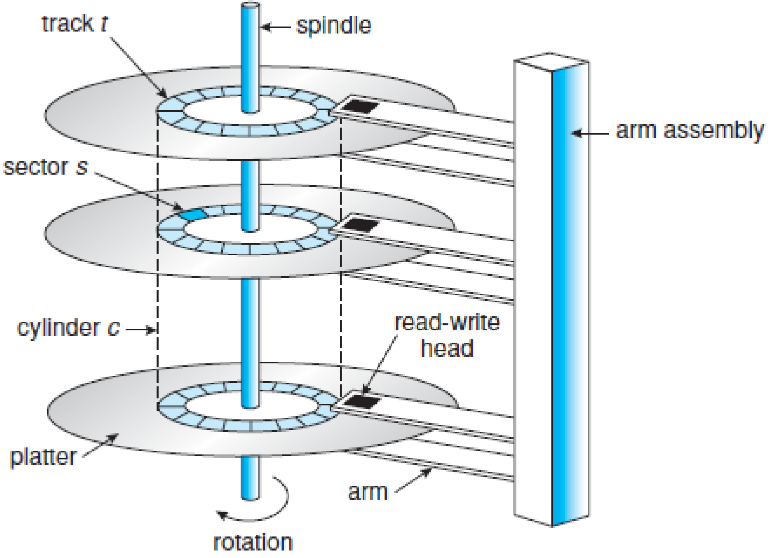
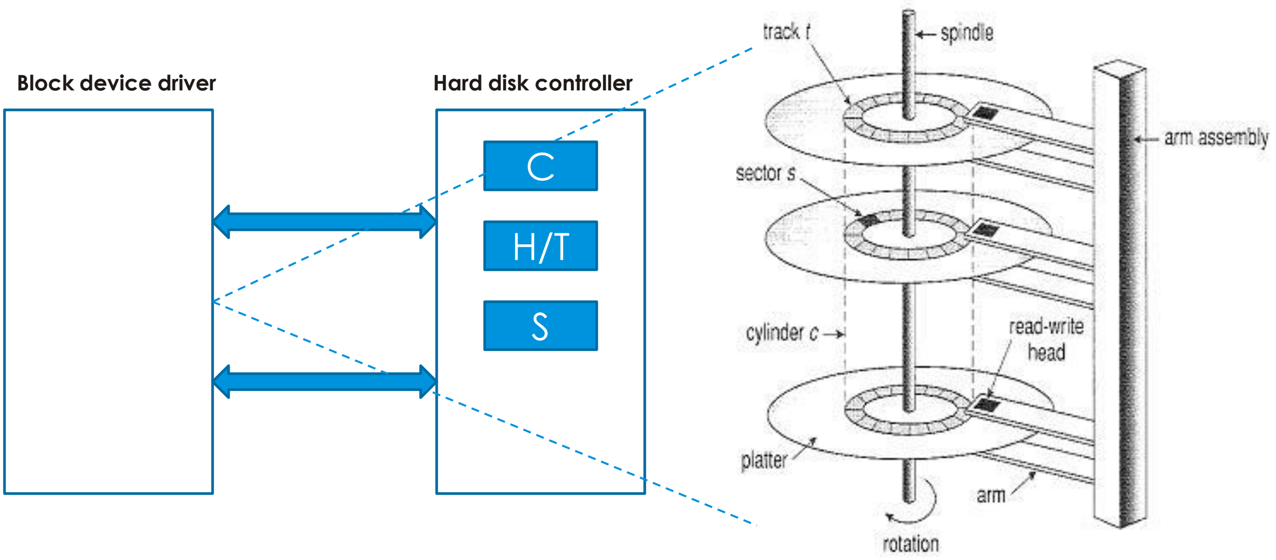


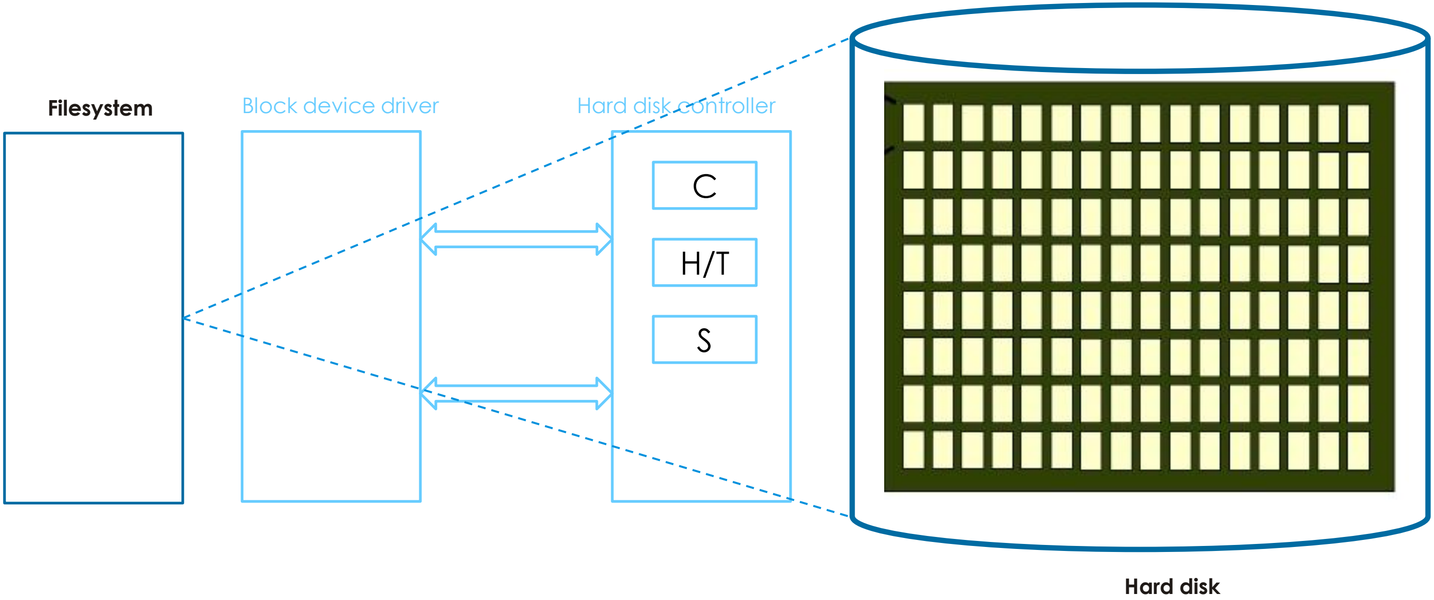
Figure 10.1 Moving-head disk mechanism.

Driver to Hard disk abstraction



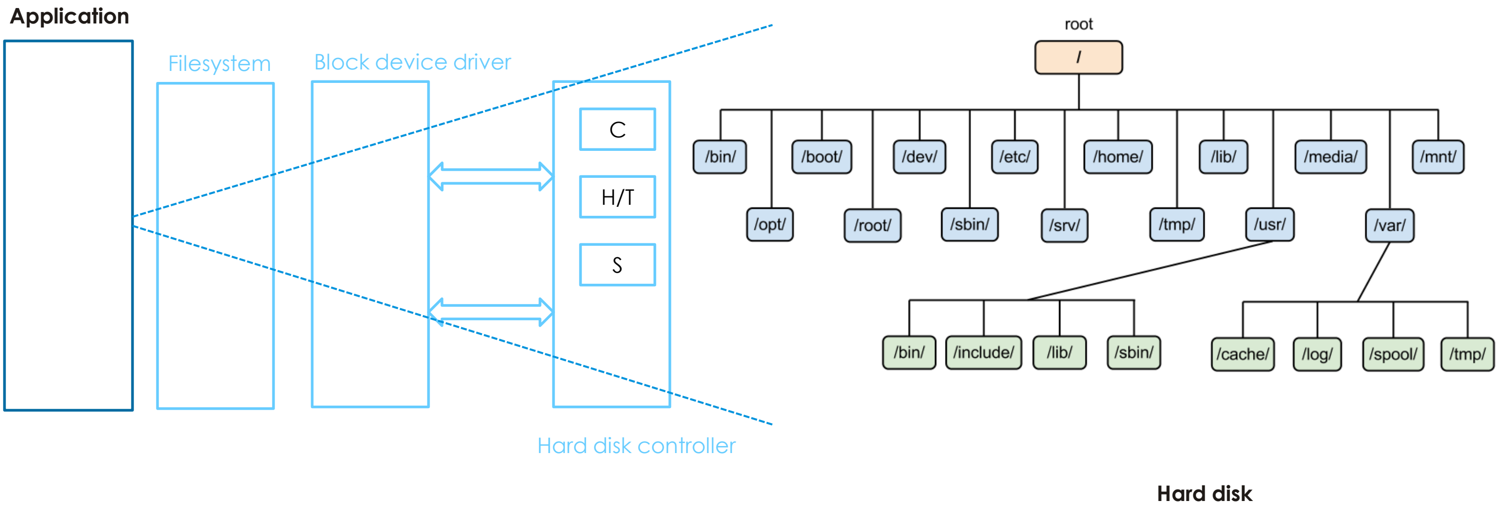
Hard disk

Filesystem to Hard disk abstraction

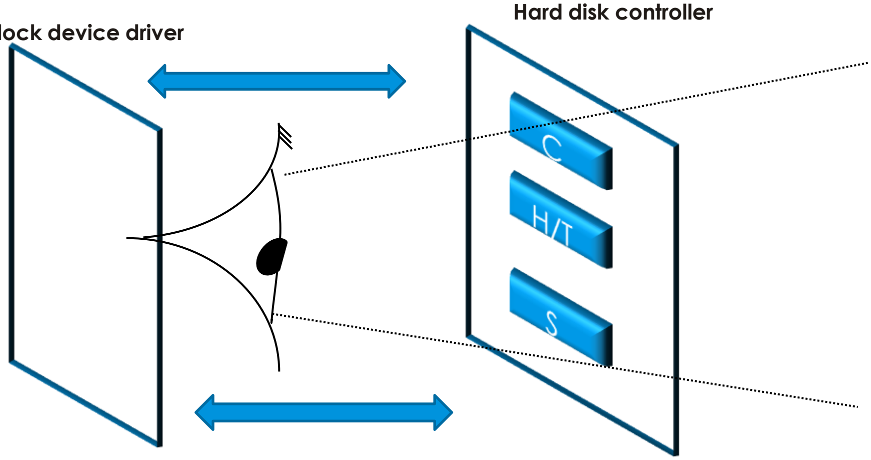


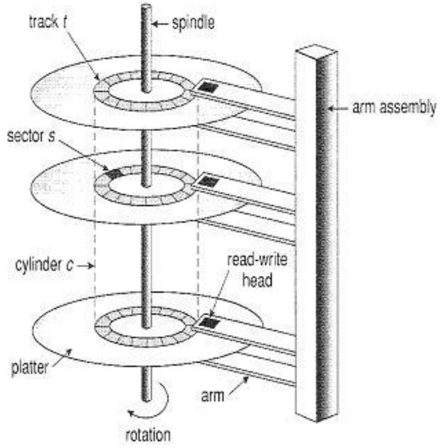
GLOBAL EDGE

Applicationto Hard disk abstraction



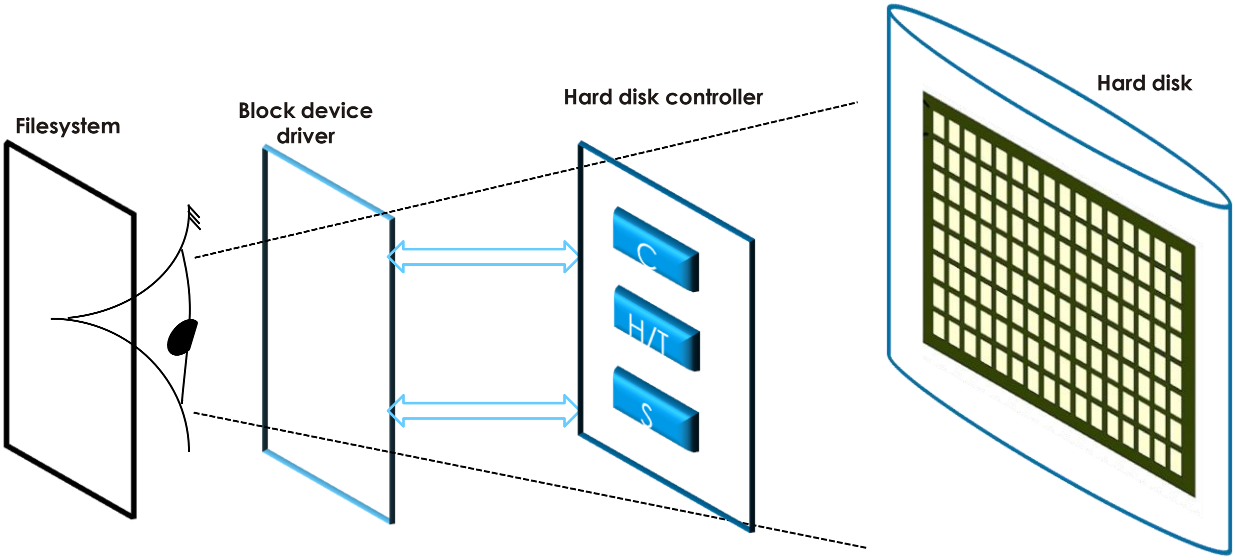
Driver to Hard disk abstraction



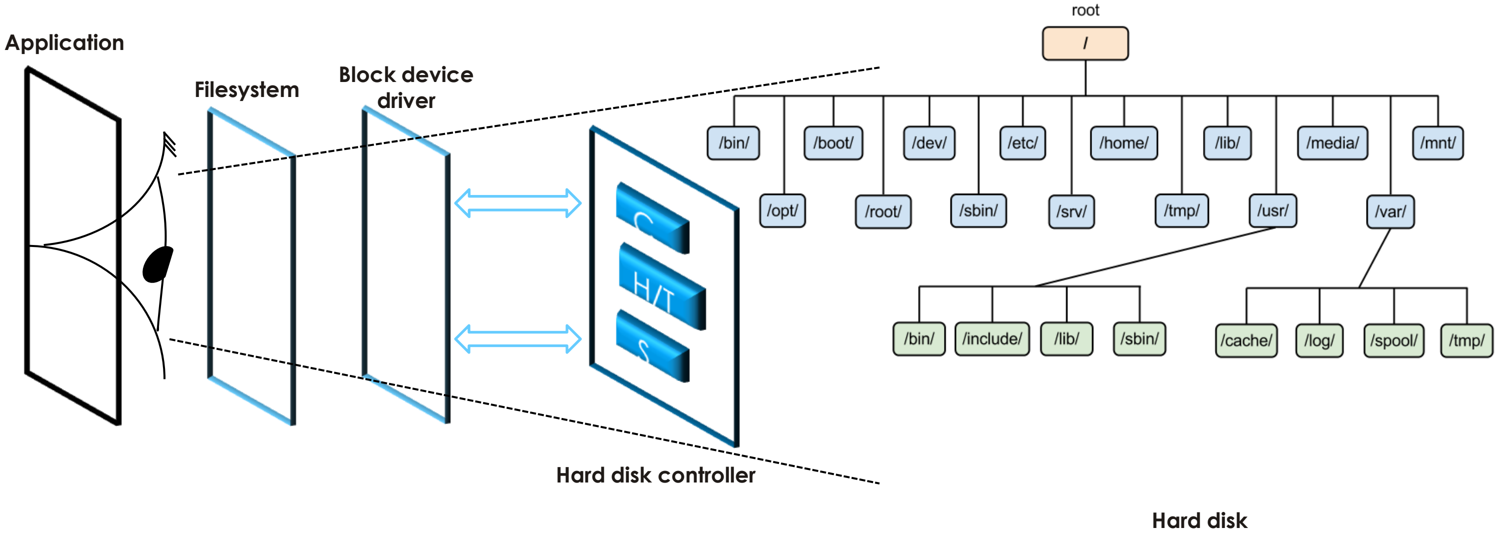


Hard disk

Filesystem to Hard disk abstraction

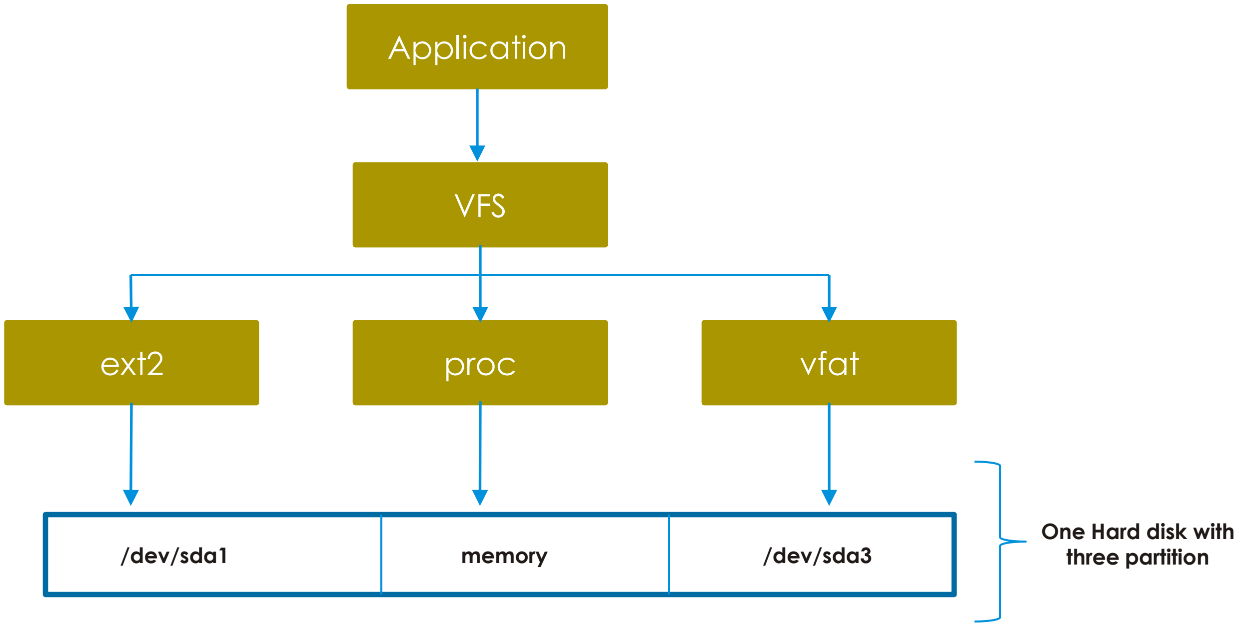


Filesystem to Hard disk abstraction



Question

Application	Filesystem	Hard disk
kmod.c →	block 2 →	2,48,05 2,48,06


Application
VFS
ext2
proc
vfat
/dev/sda1
memory
/dev/sda3
One Hard disk with
three partition

VFS

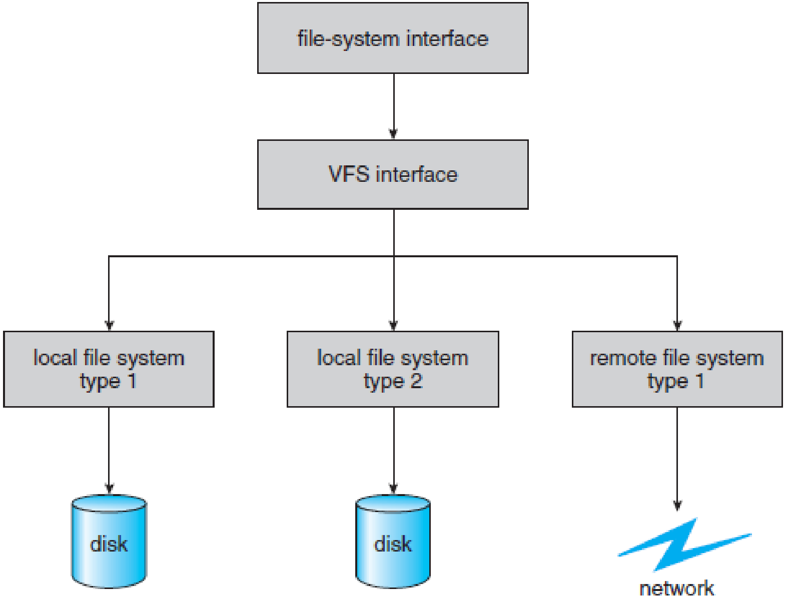


Figure 12.4 Schematic view of a virtual file system.

VFS

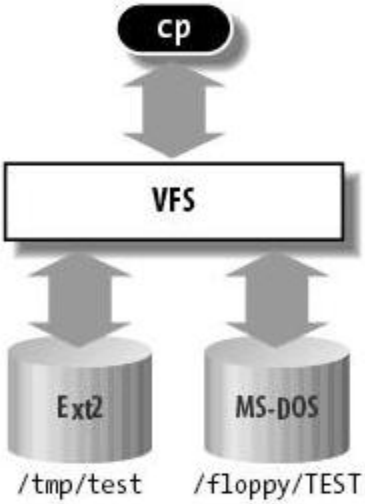
- put a wide range of information in the kernel to represent many different types of filesystems
- there is a field or function to support each operation provided by all real filesystems supported by Linux
- For each read, write, or other function called, the kernel substitutes the actual function that supports a native Linux filesystem, the NTFS filesystem, or whatever other filesystem the file is on.
- is a kernel software layer that handles all system calls related to a standard Unix filesystem
- Its main strength is providing a common interface to several kinds of filesystems.
- in the common file model, each directory is regarded as a file, which contains a list of files and other directories.

VFS

- However, several non-Unix disk-based filesystems use a File Allocation Table (FAT), which stores the position of each file in the directory tree.
- In these filesystems, directories are not files.
- To stick to the VFS's common file model, the Linux implementations of such FAT-based filesystems must be able to construct on the fly, when needed, the files corresponding to the directories.
- Such files exist only as objects in kernel memory.
- More essentially, the Linux kernel cannot hardcode a particular function to handle an operation such as read( ) or ioctl( ).
- Instead, it must use a pointer for each operation; the pointer is made to point to the proper function for the particular filesystem being accessed.
- read( ) would be translated by the kernel into a call specific to the MS-DOS filesystem.

VFS

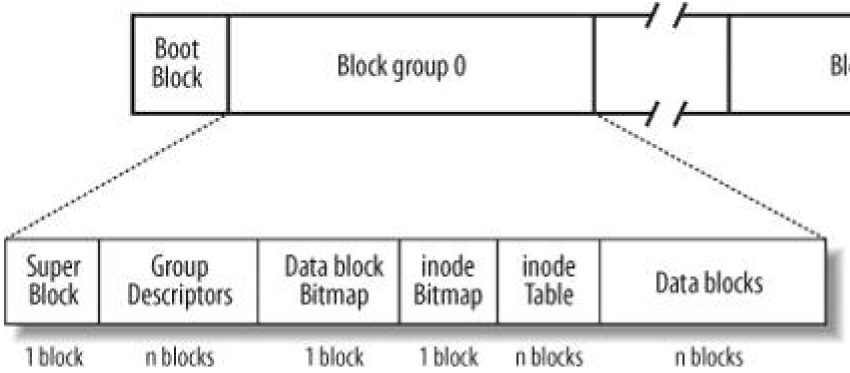
Figure 12-1. VFS role in a simple file copy operation




```
inf = open("/floppy/TEST", O_RDONLY, 0);
outf = open("/tmp/test",
        O_WRONLY|O_CREAT|O_TRUNC, 0600);
do {
    i = read(inf, buf, 4096);
    write(outf, buf, i);
} while (i);
close(outf);
close(inf);
```

Ext2 File System

Figure 18-1. Layouts of an Ext2 partition and of an Ext2 block group



Ext2 File System

- How many inodes will be there in one block group?

  Hint: Inode bitmap

  inode bitmap – 1 block

  1 bloc

Ext2 File System

- How many data blocks will be there in one block group?

Ext2 File System

- Given the number of data blocks in a block group, how many block groups we have in a partition?

Ext2 File System

- Given the size of per inode as 128 bytes, how many blocks will the inode table occupy?

Ext2 File System

- Given the size of one block group descriptor as 32 bytes, how many blocks will the block group descriptor occupy?

Ext2 File System

- How many blocks occupy Meta information?

References

- Operating System Concepts, Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, 9th Edition
- Understanding the Linux Kernel, 3rd Edition, Daniel P. Bovet, Marco Cesati

Disk Scheduling

- Disk Scheduling
  - Fast access time - for magnetic disks ,Access time consists
    - Seek time - time for the disk arm to move the heads to the cylinder containing the desired sector.
    - Rotational latency - is the additional time for the disk to rotate the desired sector to the disk head.
  - Disk bandwidth - the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.
  - Process gives I/O request through system calls
    - Whether this operation is input or output
    - What the disk address for the transfer is
    - What the memory address for the transfer is
    - What the number of sectors to be transferred is

Disk Scheduling

- first-come, first-served(FCFS) Scheduling
  - intrinsically fair, but it generally
  - does not provide the fastest service
  - for example, a disk queue with requests for I/O to blocks on cylinders
    98, 183, 37, 122, 14, 124, 65, 67
  - Total head movement of 640 cylinders
  - Problem
    wild swing from 122 to 14 and then back to 124

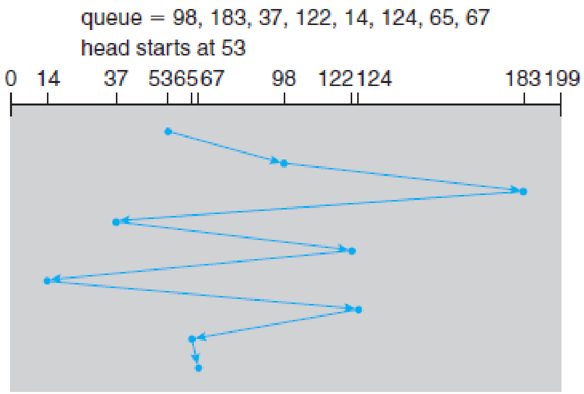


Figure 10.4 FCFS disk scheduling.

Disk Scheduling

- shortest-seek-time-first (SSTF) Scheduling
  - service all the requests close to the current head position
  - selects the request with the least seek time from the current head position
  - for example, a disk queue with requests for I/O to blocks on cylinders 98, 183, 37, 122, 14, 124, 65, 67
  - Total head movement of 236 cylinders
  - Problem
    - Starvation of some requests
    - scenario becomes increasingly likely as the pending-request queue grows longer.
  - Not optimal

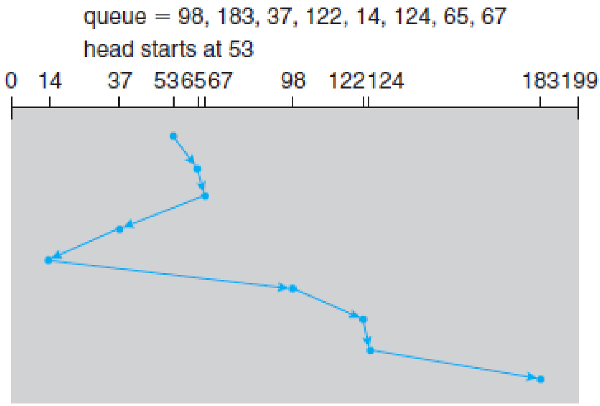


Figure 10.5 SSTF disk scheduling.

Disk Scheduling

- SCAN Scheduling/ elevator algorithm
  - disk arm starts at one end of the disk and moves toward the other end,
  - servicing requests as it reaches each cylinder, until it gets to the other end of the disk.
  - At the other end, the direction of head movement is reversed, and servicing continues.
  - for example, a disk queue with requests for I/O to blocks on cylinders 98, 183, 37, 122, 14, 124, 65, 67
  - Total head movement of 208 cylinders
  - Problem
    - request arriving just behind the head will have to wait longer

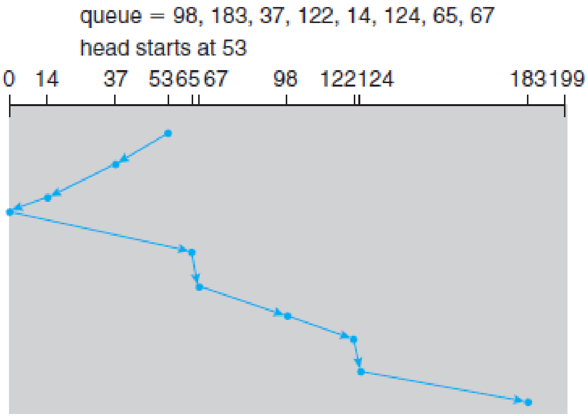


Figure 10.6 SCAN disk scheduling.

Disk Scheduling

- C-SCAN Scheduling
  - designed to provide a more uniform wait time
  - moves the head from one end of the disk to the other, servicing requests along the way
  - When the head reaches the other end, however, it immediately returns to the beginning of the disk without servicing any requests on the return trip
  - for example, a disk queue with requests for I/O to blocks on cylinders
    98, 183, 37, 122, 14, 124, 65, 67

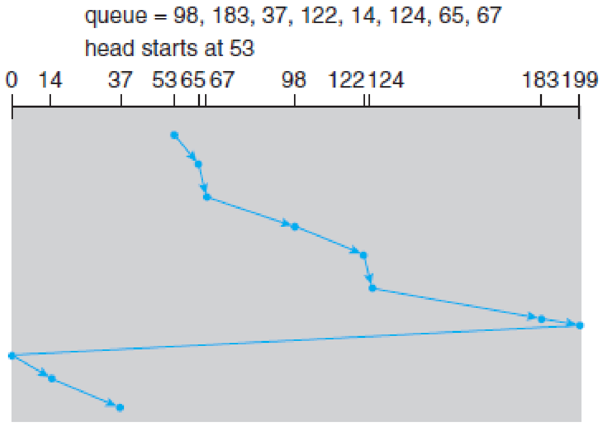


Figure 10.7 C-SCAN disk scheduling.

Disk Scheduling

- LOOK and LOOK C-SCAN Scheduling
  - look for a request before continuing to move in a given direction
  - arm goes only as far as the final request in each direction
  - Then, it reverses direction immediately, without going all the way to the end of the disk.

Disk Scheduling

- Summary
  - SSTF is common
  - SCAN and C-SCAN for heavy load on disk, no starvation
  - Performance depends on number and type of requests
  - File allocation method plays a major role disk service
    - Contiguous
    - Linked or indexed
  - location of directories and index blocks is also important

Large enough to Deliver, Small enough to Care

Global Village
IT SEZ
Bangalore

South Main Street
Milpitas
California

Raheja Mindspace
IT Park
Hyderabad

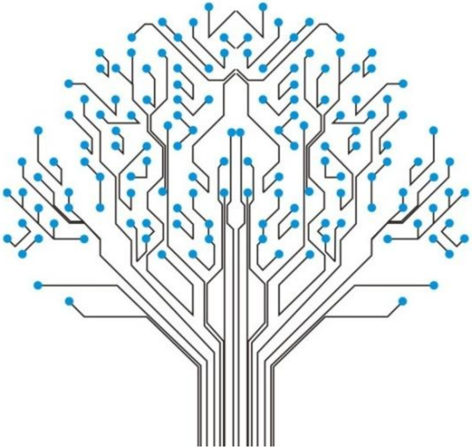
www.globaledgesoft.com

Intelligence Of Things®

GLOBAL EDGE

•ALTRAN GROUP

Thank you



Fairness

Learning

Responsibility

Innovation

Respect