CS343 - Operating Systems

Module-5A

Secondary Storage Systems Management

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

http://www.iitg.ac.in/johnjose/

Disk Storage Systems Management

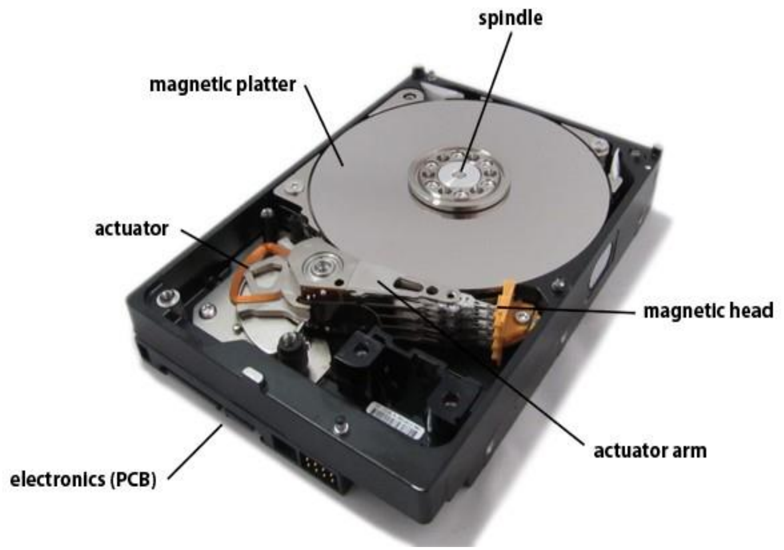
- Disk Organization & Structure
- Storage Characteristics
- Disk Management
- Disk Scheduling
- Disk Attachment
- RAID Structure
- Swap-Space Management

Objectives

- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms
- To discuss operating-system services provided for mass storage, including RAID

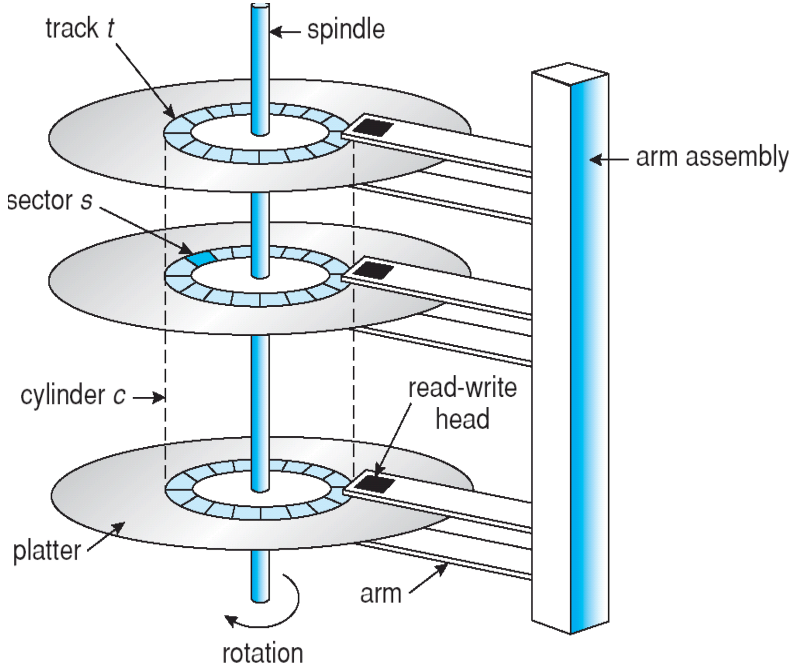
Mass Storage - Hard Disk Drive

- ❖ Systems today need to store many terabytes of data.
- ❖ Primary level of permanent storage is hard disk.
- ❖ Electromechanical
  - ❖ Rotating disks
  - ❖ Arm assembly
- ❖ Electronics
  - ❖ Disk controller
  - ❖ Cache
  - ❖ Interface controller



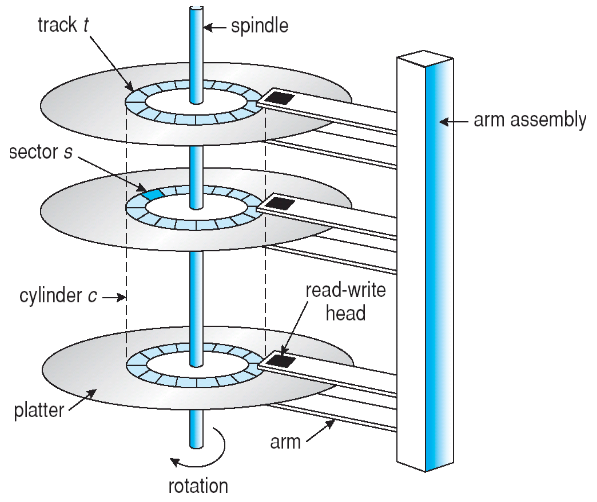
Hard Disk Drive Organization

- Hard disk drive consists of spinning disks with heads that move over the disks and store data in tracks and sectors.
- The heads read and write data in concentric rings called tracks.
- Tracks are divided into sectors, which normally store 512 bytes each.



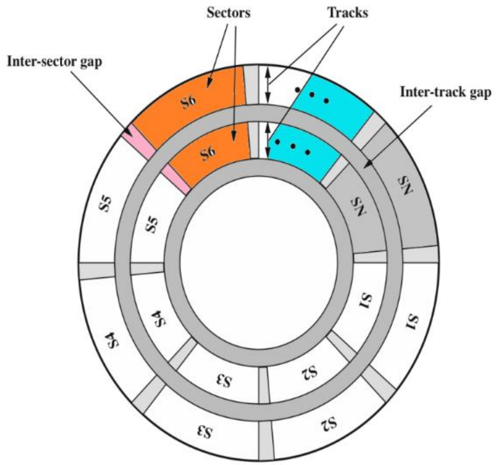
Hard Disk Drive Organization

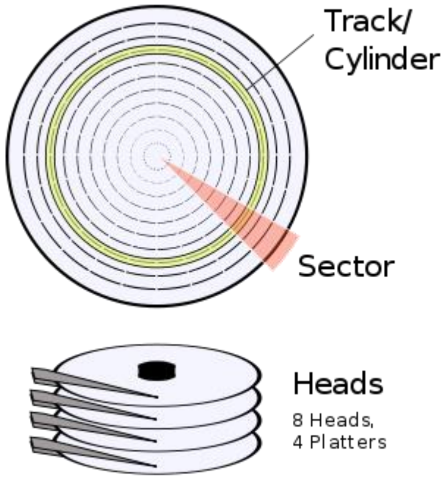
- ❖ Platter diameters: 3.7”, 3.3”, 2.6”
- ❖ RPMs: 5400, 7200, 10000, 15000 [0.5 to 1%variation]
- ❖ Number of platters: 1-5
- ❖ Power proportional to: $(\mathit{Platters})*(\mathit{RPM})^{2.8}(\mathit{Diameter})^{4.6}$
- ❖ Read/write head



Hard Disk Drive Operation

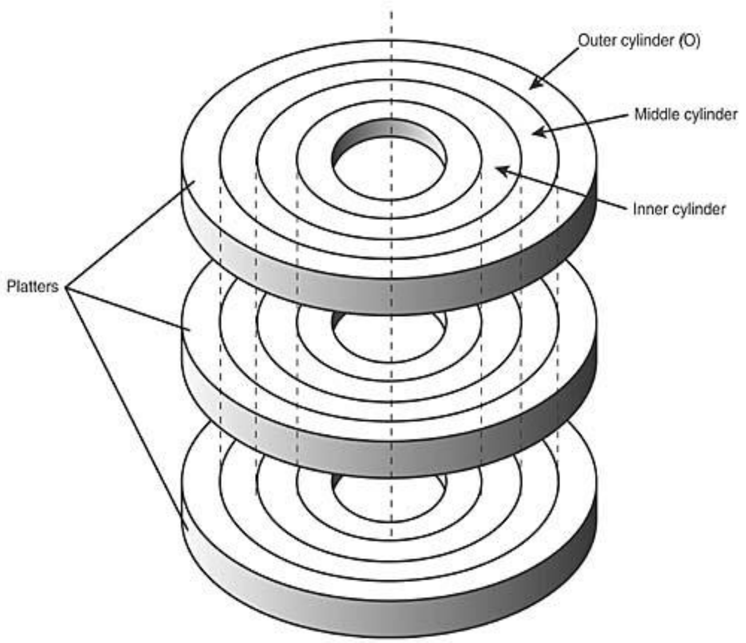
- ❖ One side of a platter is called a head.
- ❖ HDD can have multiple platters, depending on their design and storage capacity.
- ❖ On the heads, there are concentric rings (tracks) and pieces of rings (sectors)





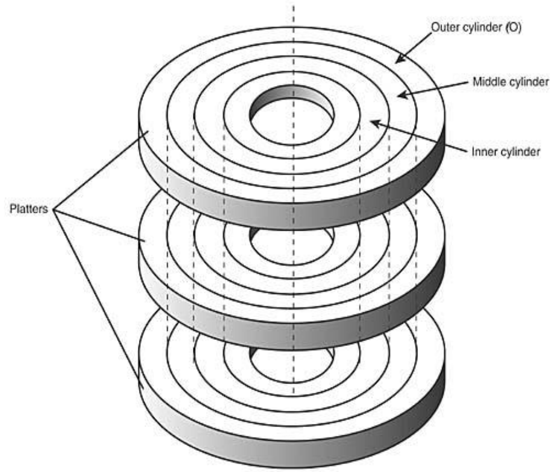
Hard Disk Drive Organization

❖ Cylinder: 3D collection of track ‘n’of each surface of all platters.



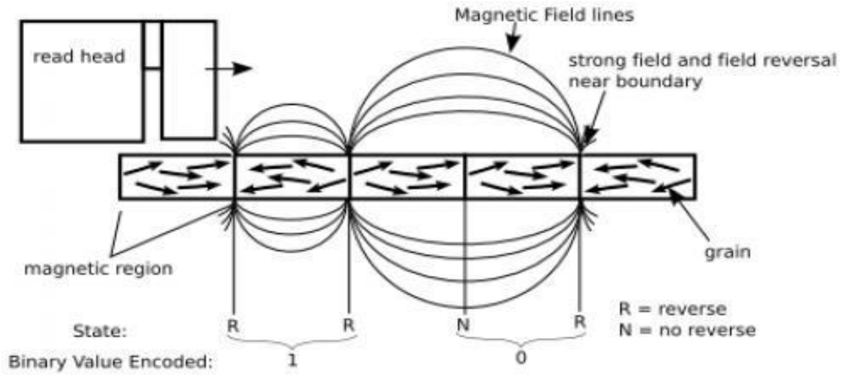
Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer
  - Low-level formatting creates **logical blocks** on physical media
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost

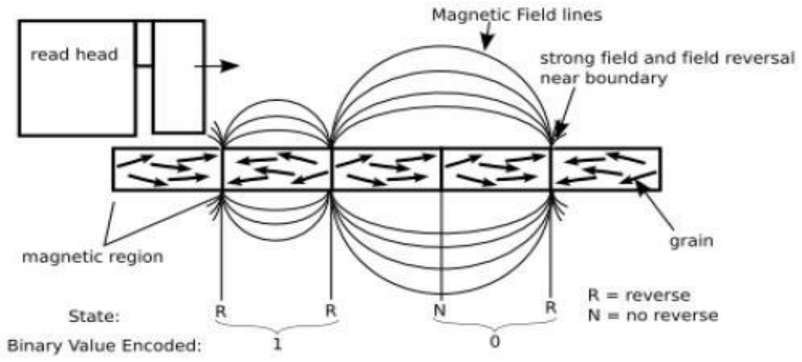


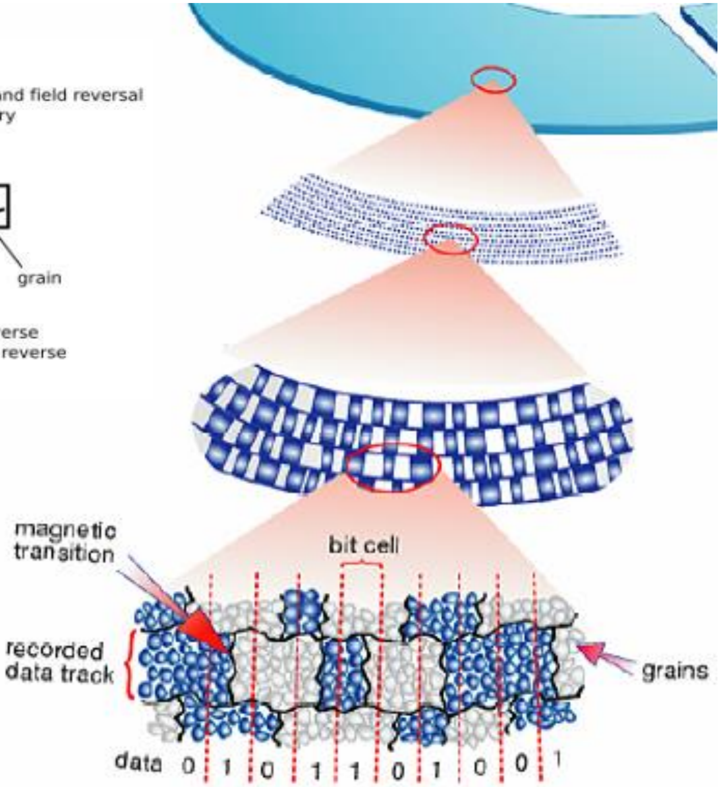
Logic Storage in Hard Disk Drive

- ❖ Bit-cell composed of magnetic grains : 50-100 grains/bit
- ❖ Size of grains is order of 10 nm.
- ❖ **‘0’** Region of grains of uniform magnetic polarity
- ❖ **‘1’** Boundary between regions of opposite magnetization
- ❖ The read-and-write head is used to detect and modify the magnetization of the material immediately under it.



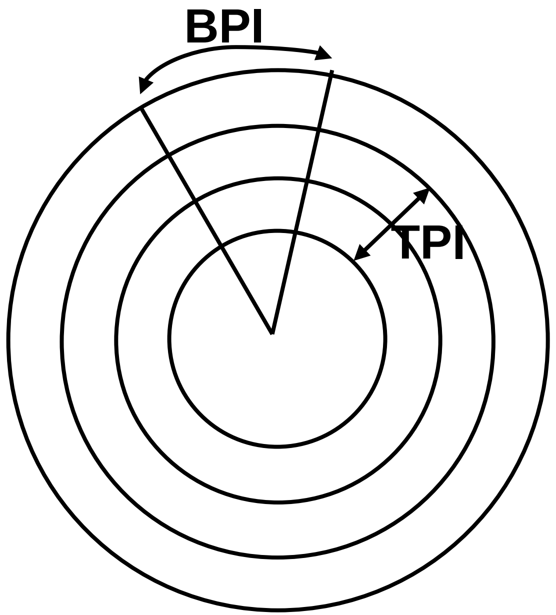
Logic Storage in Hard Disk Drive





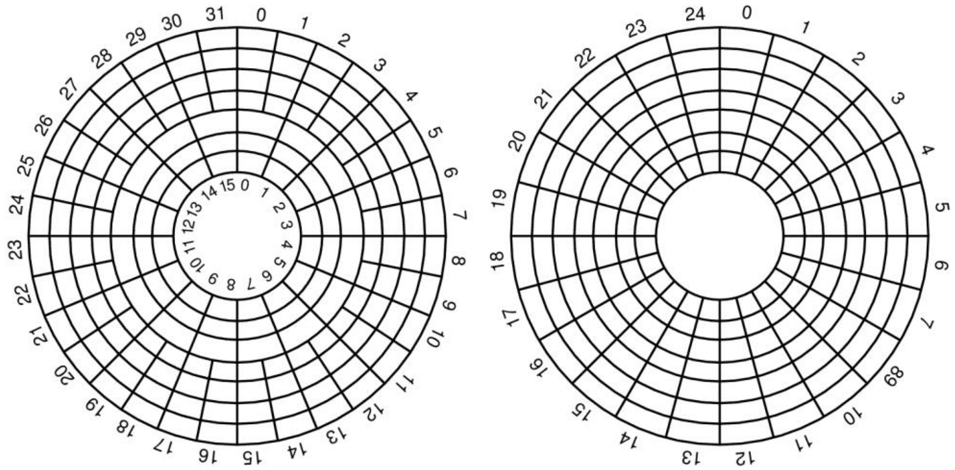
Storage Density

- ❖ Determines both capacity and performance
- ❖ Density Metrics
    - ❖ Linear density (Bits/inch or BPI)
    - ❖ Track density (Tracks/inch or TPI)
    - ❖ Areal Density = BPI x TPI



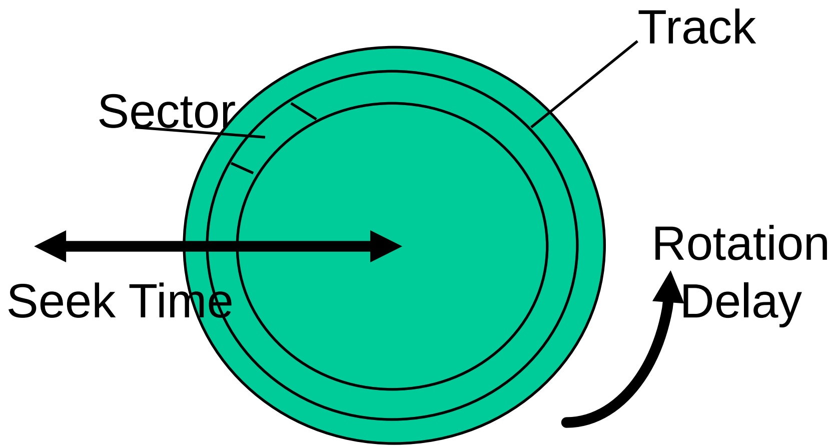
HDD- Bit Density

- Reduce bit density per track for outer layers. Constant Linear Velocity.
- Have more sectors per track on the outer layers, and increase rotational speed when reading from outer tracks. Constant Angular Velocity.



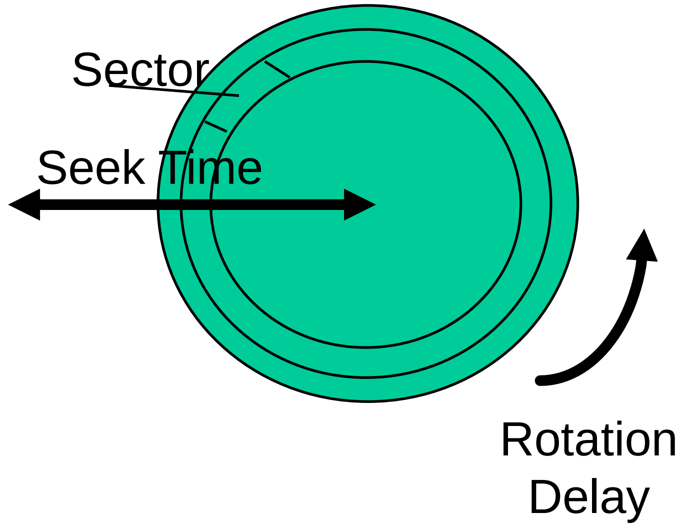
Disk Access Time

- To read from disk, we must specify:
  - cylinder #, surface #, sector #, size, memory address
- Transfer time includes:
  - Seek time: to get to the track
  - Rotational Latency: to get to the sector
  - Transfer time: get bits off the disk



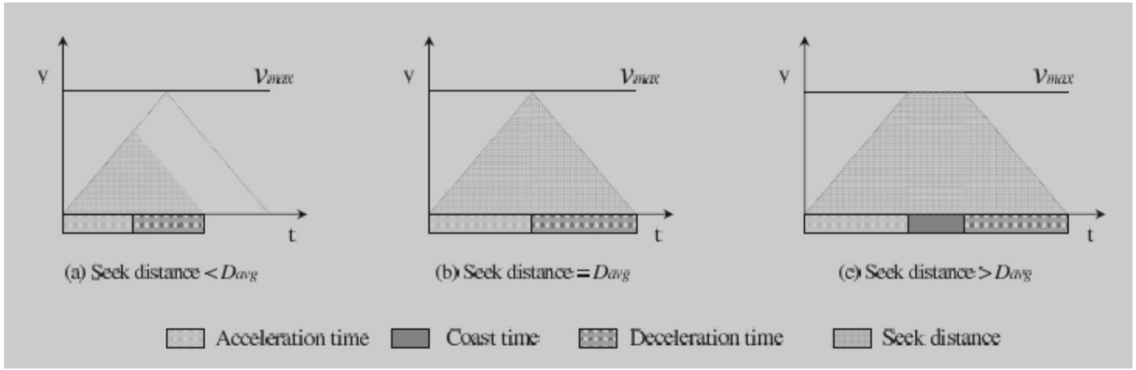
Seek Time

- ❖ **Seek time depends on:**
  - ❖Inertial power of the arm actuator motor
  - ❖Distance between outer-disk recording radius and inner-disk recording radius (data-band)
  - ❖Depends on platter-size



- ❖ **Components of a seek:**
  - ❖ Speedup: Arm accelerates
  - ❖ Coast: Arm moving at maximum velocity (long seeks)
  - ❖ Slowdown: Arm brought to rest near desired track
  - ❖ Settle: Head is adjusted to reach the access the desired location

Variations in Seek Time



- ❖Very short seeks (2-4 cylinders)
    - ❖ Settle-time dominates
- ❖Short seeks (100-200 cylinders)
    - ❖ Speedup/Slowdown-time dominates
- ❖Longer seeks (> 200 cylinders)
    - ❖ Coast-time dominates
- ❖With smaller platter-sizes and higher TPI
    - ❖ Settle-time becoming more important

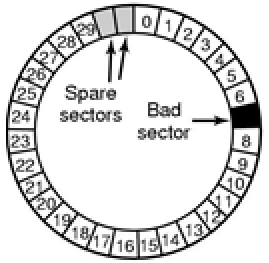
Disk Formatting

- **Low-Level Formatting (LLF)** is the process of outlining the positions of the tracks and sectors on the hard disk, and writing the control structures that define where the tracks and sectors. Latest hard disks are LLF at factory.

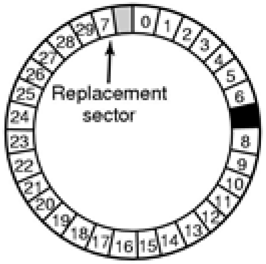
- **High-Level Formatting (HLF)** is the process of initializing portions of the hard disk and creates the file system structures on the disk, such as the master boot record and the file allocation tables. High-level formatting is typically done to erase the hard disk and reinstall the operating system back onto the disk drive.

Bad sector management in disks

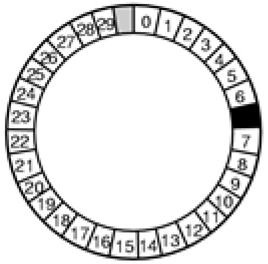
- **Bad sector** is a sector on disk that is either inaccessible or un-writeable due to permanent damage.
- Bad sectors are usually detected by LLF or HLF or by utility software such as CHKDSK or SCANDISK .
- The sectors unusable are not used for storage.
- If a file uses a sector which is marked as bad then the bad sector of the file is remapped to a free sector.



(a)



Sector forwarding



Sector slipping

Thank you

johnjose@iitg.ac.in

http://www.iitg.ac.in/johnjose/

CS343 - Operating Systems

Module-5B

Disk Scheduling

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

http://www.iitg.ac.in/johnjose/

Disk Storage Systems Management

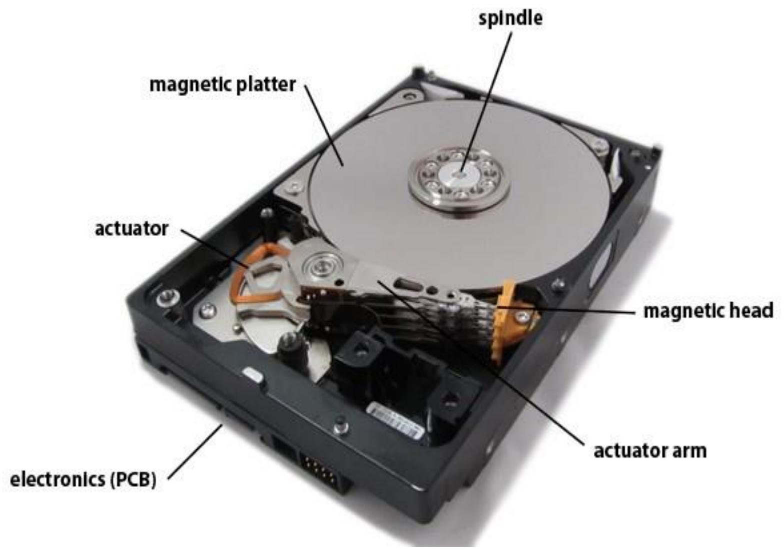
- ❖ Disk Organization & Structure
- ❖ Disk Attachment
- ❖ Disk Scheduling
- ❖ Disk Management
- ❖ Swap-Space Management
- ❖ RAID Structure
- ❖ Stable-Storage Implementation

Objectives

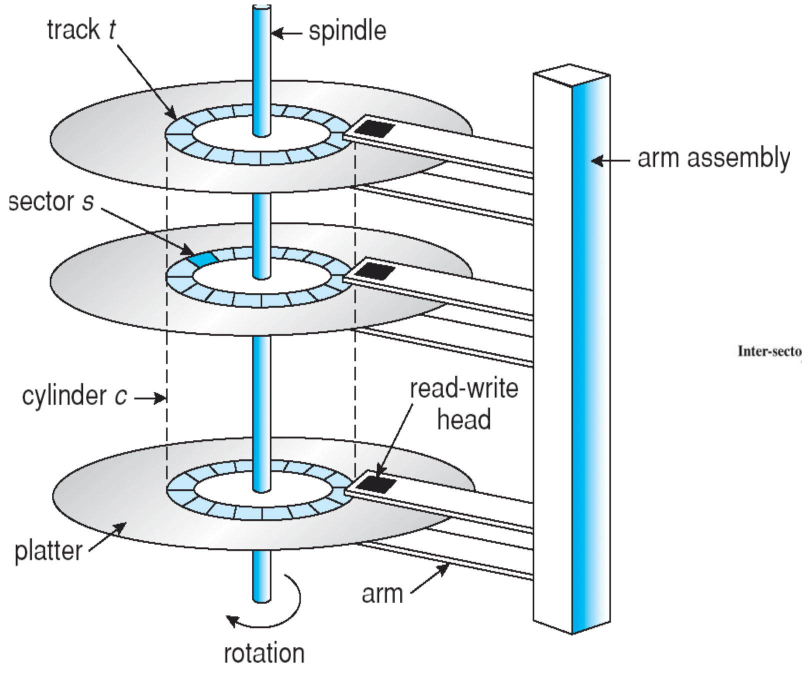
- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms
- To discuss operating-system services provided for mass storage, including RAID

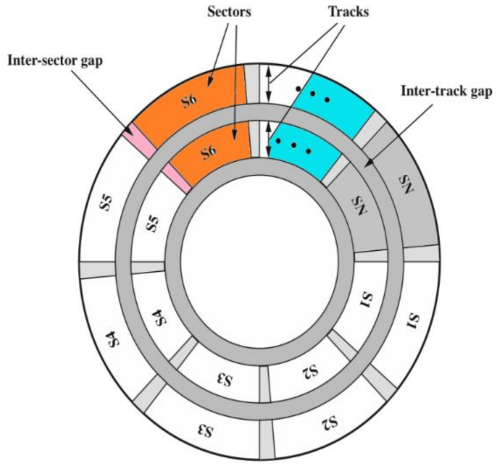
Mass Storage - Hard Disk Drive

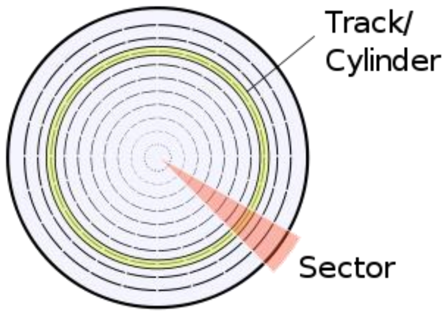
- Systems today need to store many terabytes of data.
- Primary level of permanent storage is hard disk.
- Electromechanical
  - Rotating disks
  - Arm assembly
- Electronics
  - Disk controller
  - Cache
  - Interface controller

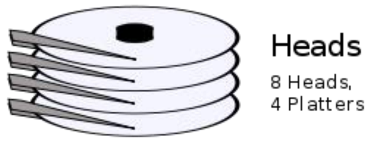


Hard Disk Drive Organization



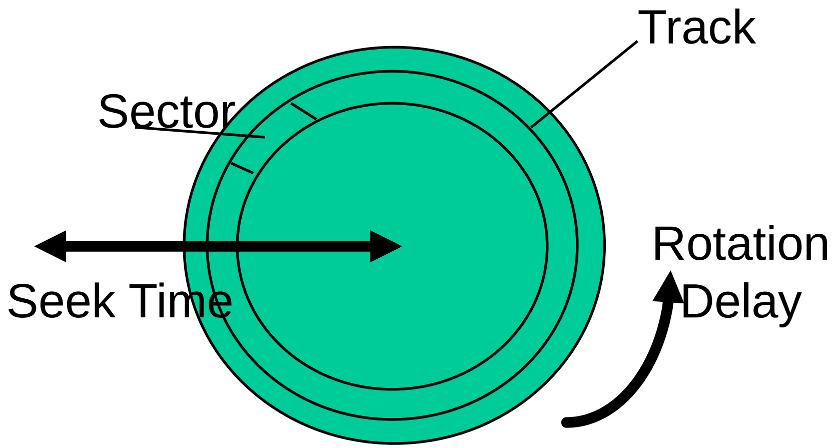






Disk Access Time

- To read from disk, we must specify:
  - cylinder #, surface #, sector #, size, memory address
- Transfer time includes:
  - Seek time: to get to the track
  - Rotational Latency: to get to the sector and
  - Transfer time: get bits off the disk



Disk Scheduling

- ❖Access time has two major components
  - ❖ **Seek time** is time to move the heads to the cylinder containing the desired sector
  - ❖ **Rotational latency** is additional time waiting to rotate the desired sector to the disk head.
- ❖ Minimize seek time
- ❖ Disk bandwidth is total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

Disk Scheduling

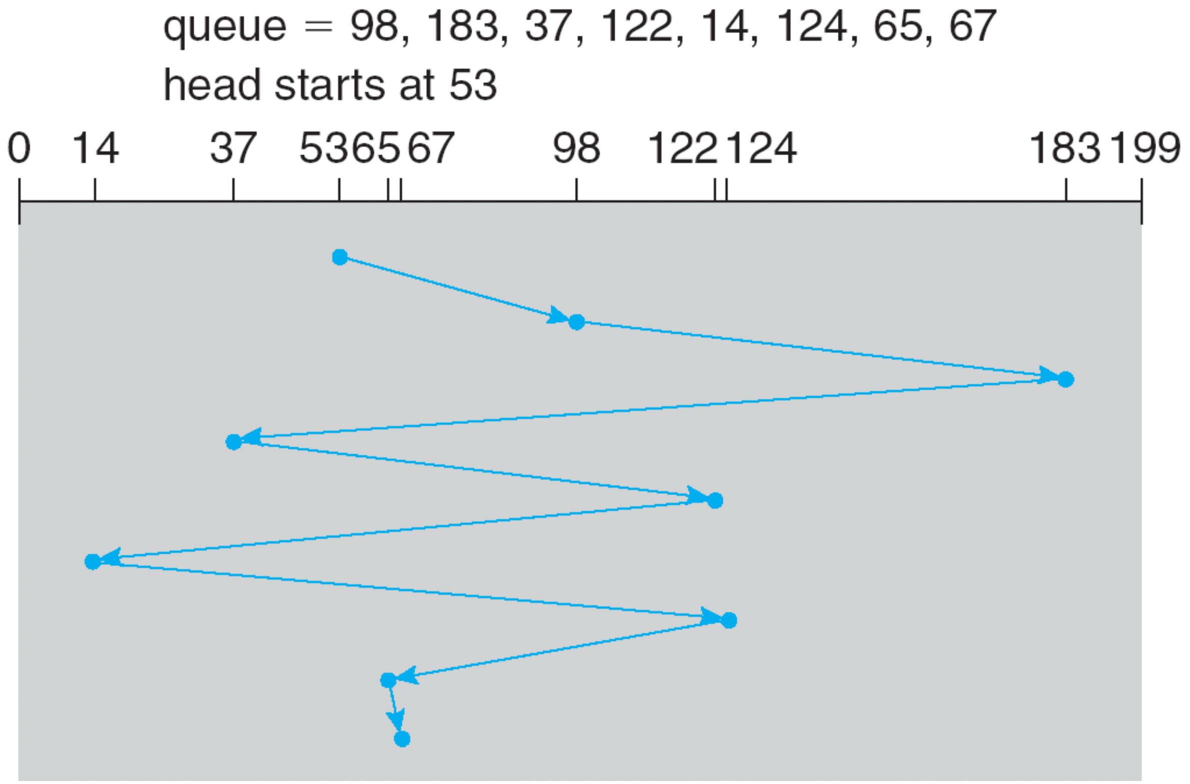
- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists

Disk Scheduling

- Drive controllers have small buffers to manage a queue of I/O requests.
- Several algorithms exist to schedule the servicing of disk I/O requests
- Disk Scheduling - The order in which disk cylinder request are serviced so as to optimize average seek time.
    - **FCFS**
    - **SSTF**
    - **SCAN**
    - **C-SCAN**
    - **C-LOOK**
- **Example:** 98, 183, 37, 122, 14, 124, 65, 67 (0-199 cylinders, Head pointer @ 53)

Disk Scheduling Algorithm: FCFS

Total head movements = 640



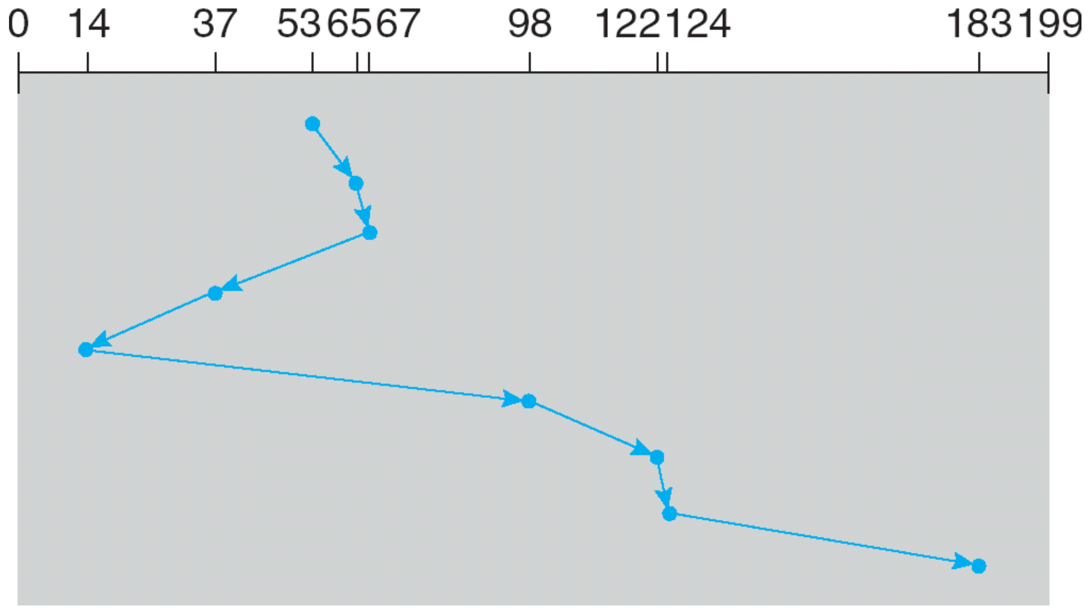
Disk Scheduling Algorithm: SSTF

- Selects request with minimum seek time from current head position,
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests

Total head movement of 236 cylinders

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



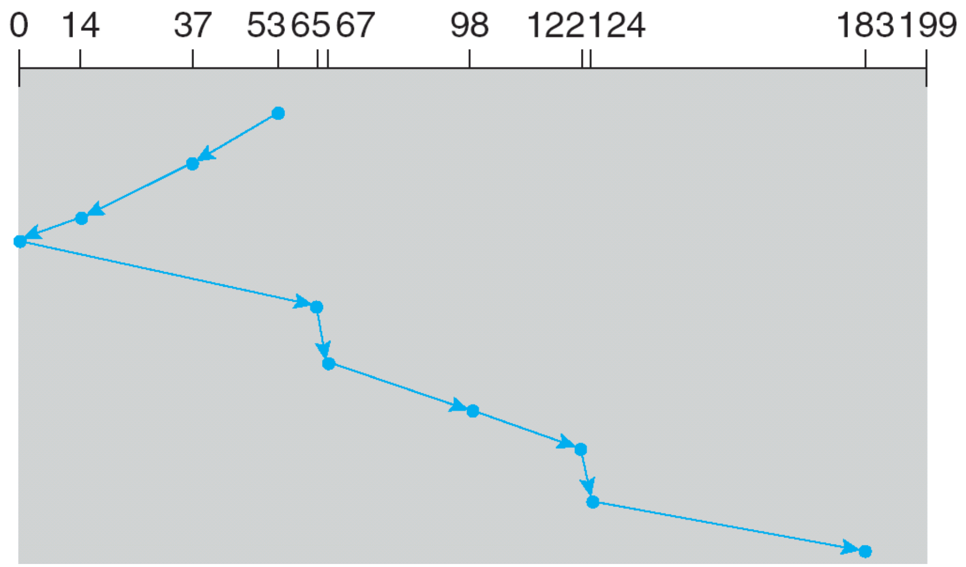
Disk Scheduling Algorithm: SCAN

- The disk arm moves toward one end servicing requests
- Head movement is reversed when it reach the end and servicing continues. [Also known as elevator algorithm]

Total head movement of 208 cylinders

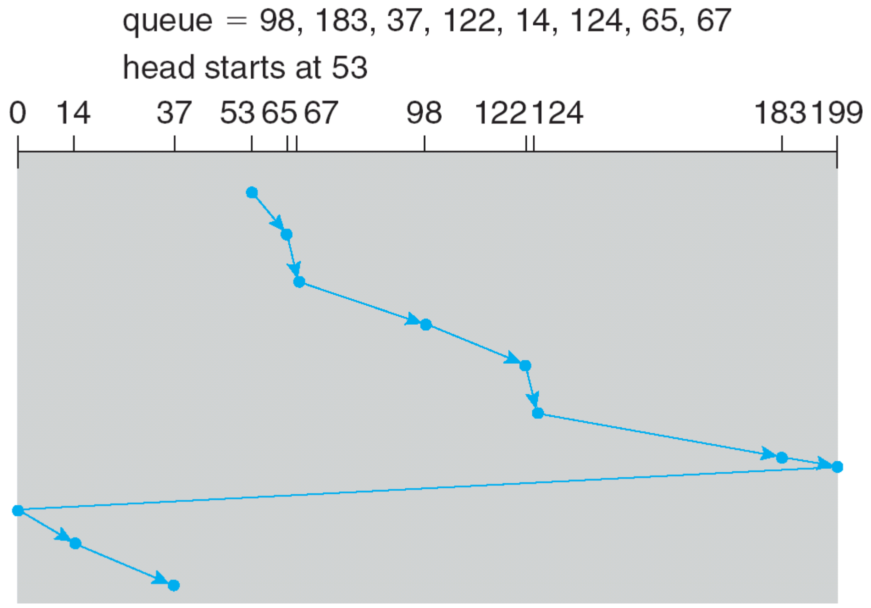
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



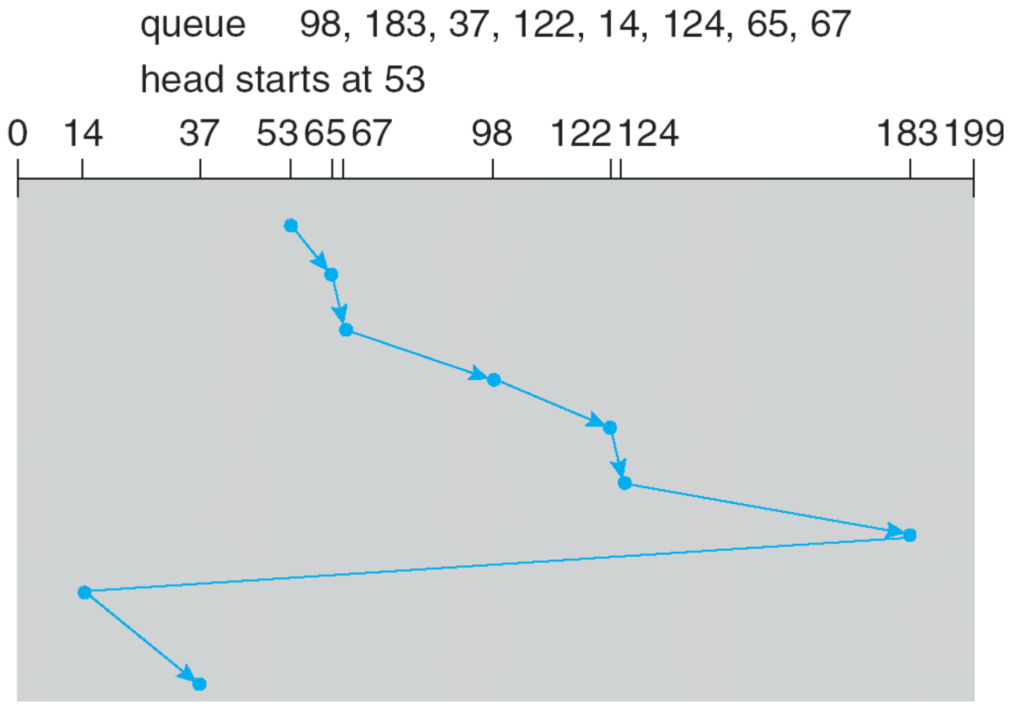
Disk Scheduling Algorithm : C-SCAN

- The head moves from one end of the disk to the other and service the requests as it goes.
- When it reaches the other end it immediately returns to beginning of the disk, No servicing on the return trip.



Disk Scheduling Algorithm : C-LOOK

- Version of C-SCAN
- Arm only goes as far as last request in each direction, then reverses direction immediately,

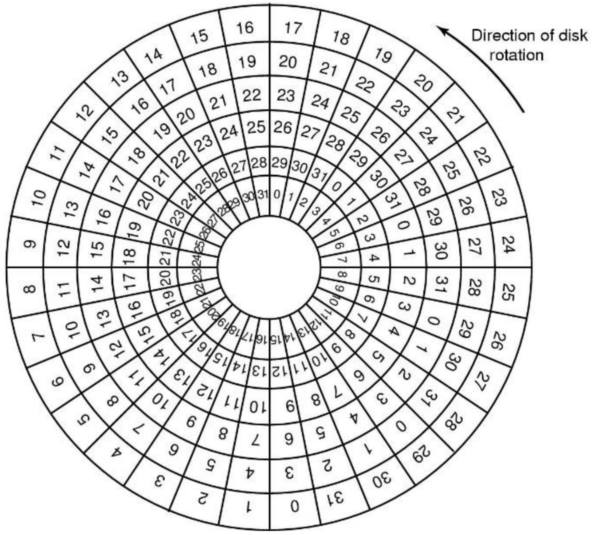


Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk : Less starvation
- Performance depends on the number and types of requests
- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm

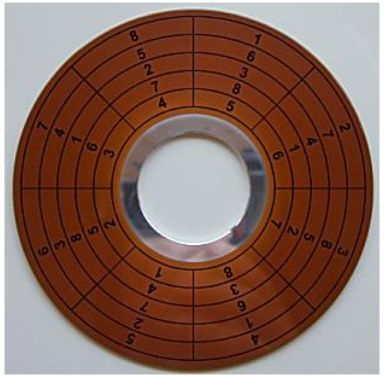
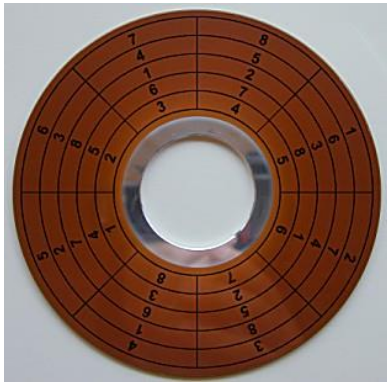
Cylinder Skew

- Why cylinder skew?
- Offsetting the start sector of adjacent tracks to minimize the likely wait time (rotational latency) when switching tracks
- How much skew?
- Example, if 10000 rpm disk drive rotates in 6 ms.
  - Track has 300 sectors
  - New sector every 20 µs
  - If track seek time 800 µs
    - 40 sectors pass on seek
- Cylinder skew: 40 sectors



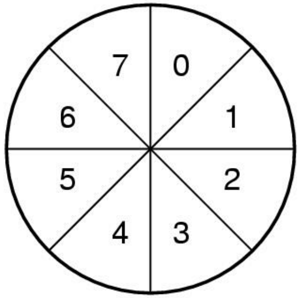
Head Skew

- Occurs when we change heads within a cylinder, but different platter surfaces.
- Here there is no physical movement of arm assembly.
- But it still takes time for the switch from reading one head to reading another.
- Head skew is the offsetting done on the start sector of tracks of adjacent platters (heads) of same cylinder.

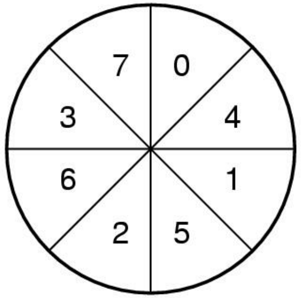
Sector Interleaving

❖ To ensure that sector #n+1 didn't rotate past the head while sector #n was being processed.



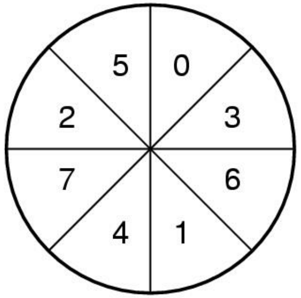
(a)

No interleaving



(b)

Single interleaving



(c)

Double interleaving

Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management in various OS.
  - Allocates swap space when process starts; holds text segment (the program) and data segment
  - Uses **swap maps** to track swap-space use
  - Some allocate swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first created.

Thank you

johnjose@iitg.ac.in

http://www.iitg.ac.in/johnjose/

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी • Indian Institute of Technology Guwahati •

CS343 - Operating Systems

Module-5C

Storage Arrays & RAID Implementation

Dr. John Jose

Assistant Professor

Department of Computer Science & Engineering

Indian Institute of Technology Guwahati, Assam.

http://www.iitg.ac.in/johnjose/

Disk Storage Systems Management

- ❖ Disk Organization & Structure
- ❖ Disk Attachment
- ❖ Disk Scheduling
- ❖ Disk Management
- ❖ Swap-Space Management
- ❖ Storage Arrays
- ❖ RAID Structure
- ❖ Stable-Storage Implementation

Objectives

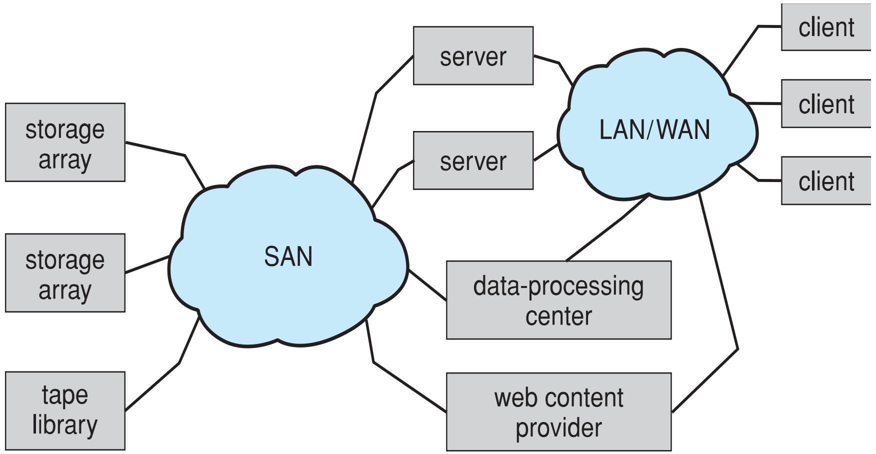
- To describe the physical structure of secondary storage devices and its effects on the uses of the devices
- To explain the performance characteristics of mass-storage devices
- To evaluate disk scheduling algorithms
- To discuss operating-system services provided for mass storage, including RAID

Storage Array

- Attach multiple disks - arrays of disks
- Storage array has controller that provides features to attached hosts
  - A few to thousands of disks
  - Ports to connect hosts to array
  - Memory, controlling software
  - Support RAID, hot spares, hot swap
  - Shared storage → more efficiency

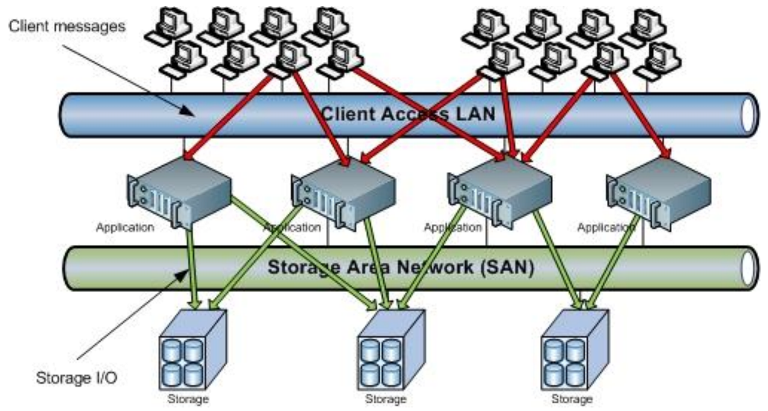
Storage Area Network

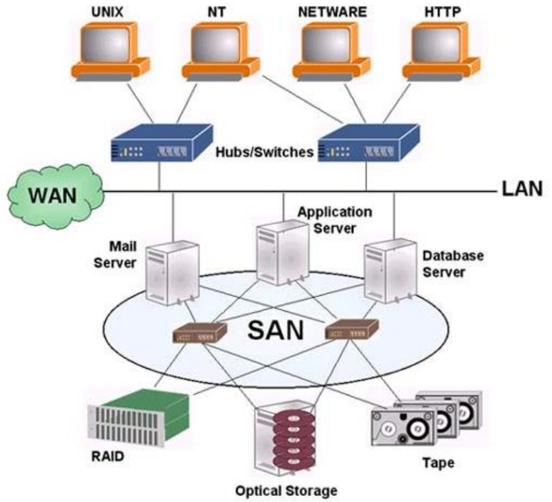
- Common in large storage environments
- Multiple hosts attached to multiple storage arrays - flexible



Storage Area Network

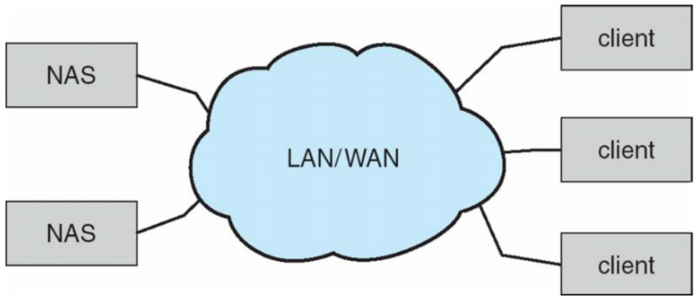
- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts also attach to the switches
- Easy to add or remove storage, add new host and allocate it storage



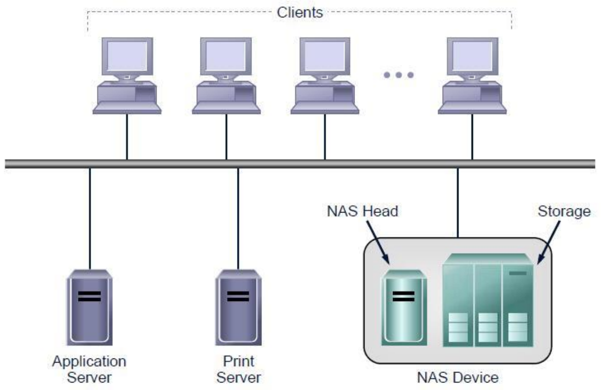


Network-Attached Storage

- Network-attached storage (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
  - Remotely attaching to file systems
- Implemented via remote procedure calls (RPCs) between host and storage over typically standard computer network protocols.



NAS is shared storage on a network infrastructure



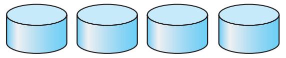
RAID Structure

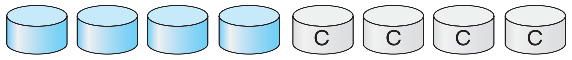
- RAID – redundant array of inexpensive disks
- Multiple disk drives provides reliability via **redundancy**
- Use of multiple disks working cooperatively
- Increases the **mean time to failure**
- Frequently combined with **NVRAM** to improve write performance
- Disk **striping** (**RAID 0**) uses a group of disks as one storage unit
- RAID is arranged into six different levels

RAID Structure

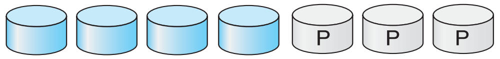
- RAID schemes improve performance and improve the reliability of the storage system by storing redundant data
  - **Mirroring** or **shadowing** (**RAID 1**) keeps duplicate of each disk
  - Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability
  - **Block interleaved parity** (**RAID 4, 5, 6**) uses much less redundancy
- RAID within a storage array can still fail if the array fails, so automatic **replication** of the data between arrays is common
- Frequently, a small number of **hot-spare** disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them

RAID Levels

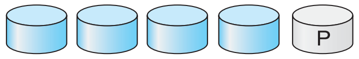

(a) RAID 0: non-redundant striping.



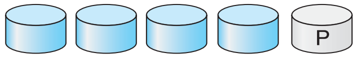
(b) RAID 1: mirrored disks.



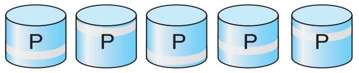
(c) RAID 2: memory-style error-correcting codes.



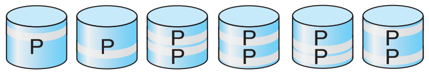
(d) RAID 3: bit-interleaved parity.



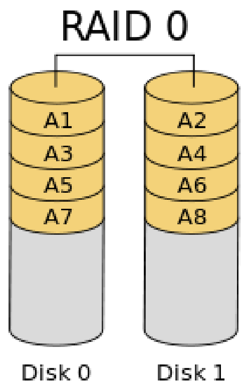
(e) RAID 4: block-interleaved parity.

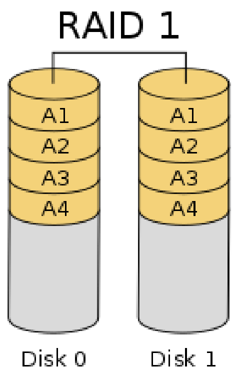


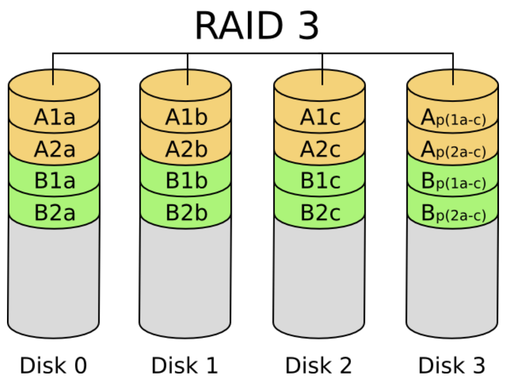
(f) RAID 5: block-interleaved distributed parity.

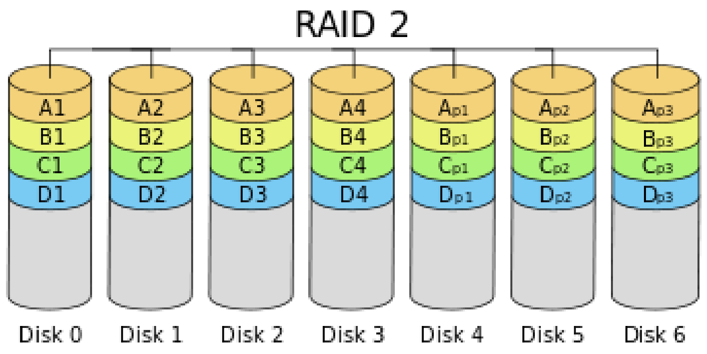


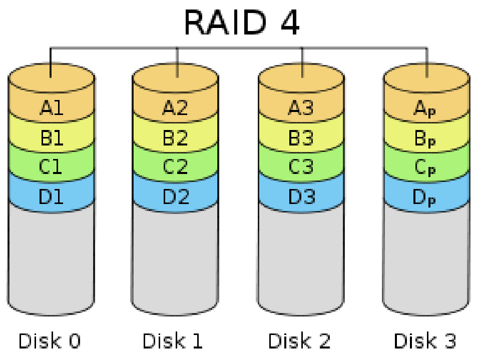
(g) RAID 6: P + Q redundancy.



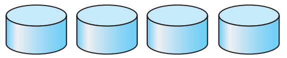


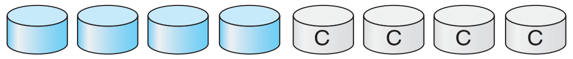




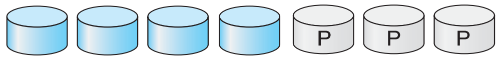


RAID Levels

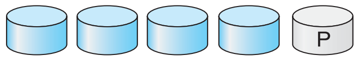

(a) RAID 0: non-redundant striping.



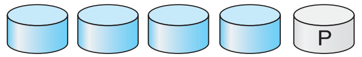
(b) RAID 1: mirrored disks.



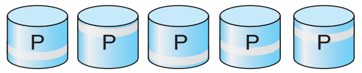
(c) RAID 2: memory-style error-correcting codes.



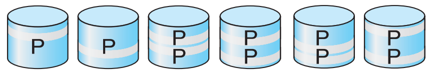
(d) RAID 3: bit-interleaved parity.



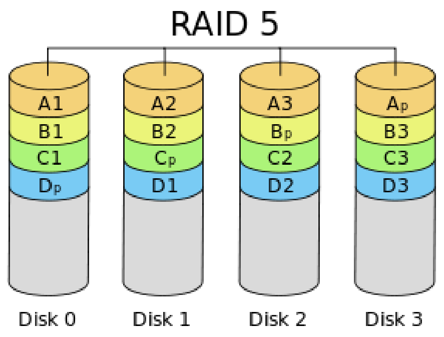
(e) RAID 4: block-interleaved parity.

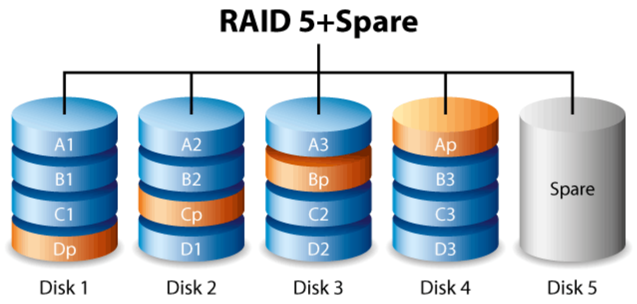


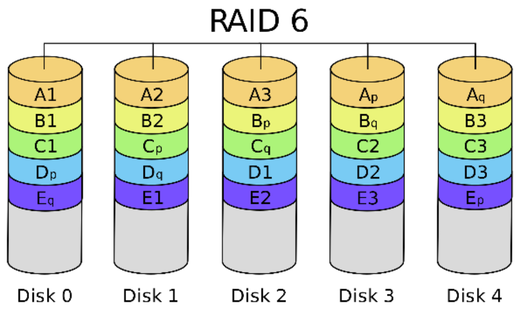
(f) RAID 5: block-interleaved distributed parity.



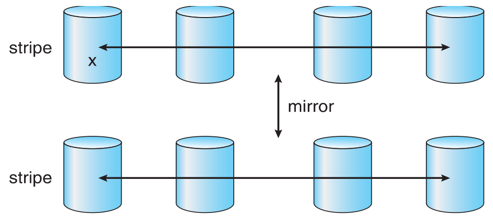
(g) RAID 6: P + Q redundancy.



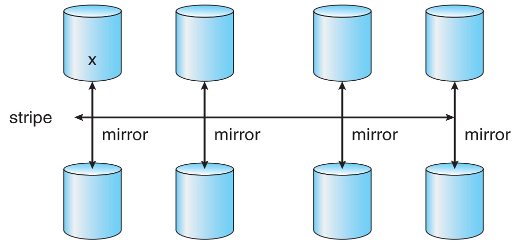




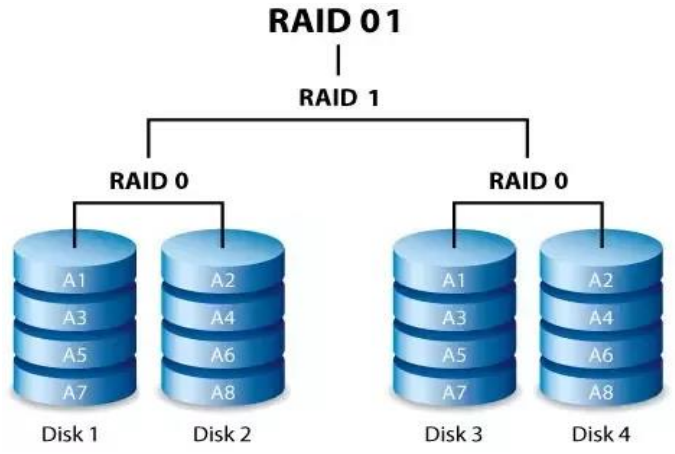
RAID Levels

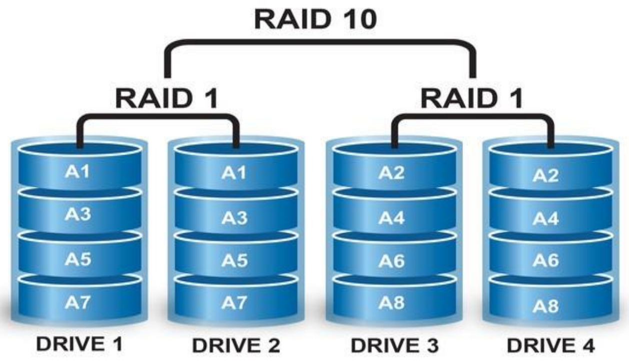


a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.





Stable-Storage Implementation

- Write-ahead log scheme requires stable storage
- Stable storage means data is never lost (due to failure, etc)
- To implement stable storage:
    - Replicate information on more than one nonvolatile storage media with independent failure modes
    - Update information in a controlled manner to ensure that we can recover the stable data after any failure during data transfer or recovery

Stable-Storage Implementation

- Disk write has 1 of 3 outcomes
  - **Successful completion -** The data were written correctly on disk
  - **Partial failure -** A failure occurred in the midst of transfer, so only some of the sectors were written with the new data, and the sector being written during the failure may have been corrupted
  - **Total failure -** The failure occurred before the disk write started, so the previous data values on the disk remain intact
- If failure occurs during block write, recovery procedure restores block to consistent state
  - System maintains 2 physical blocks per logical block
    - Write to 1st physical, When successful, write to 2nd physical
    - Declare complete only after second write completes successfully

Thank you

johnjose@iitg.ac.in

http://www.iitg.ac.in/johnjose/