Secondary Storage Devices

Instructor: Dr Tarunpreet Bhatia
Assistant Professor
CSED, TIET

Disclaimer

This is NOT A COPYRIGHT MATERIAL

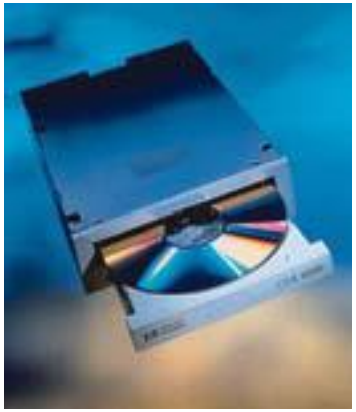
Content has been taken mainly from the following books:

Operating Systems Concepts By Silberschatz & Galvin,
Operating Systems: Internals and Design Principles By William Stallings

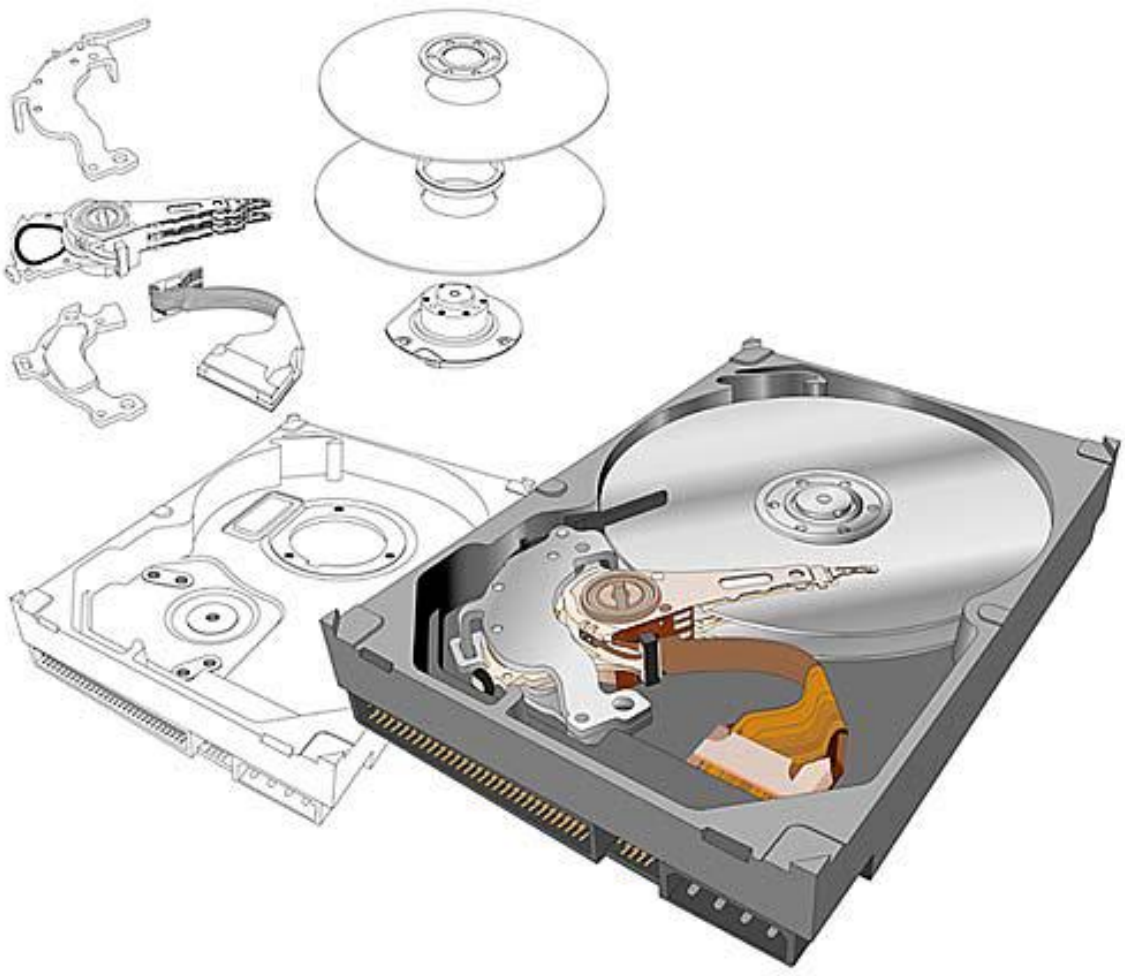
Topics

- Disk Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure

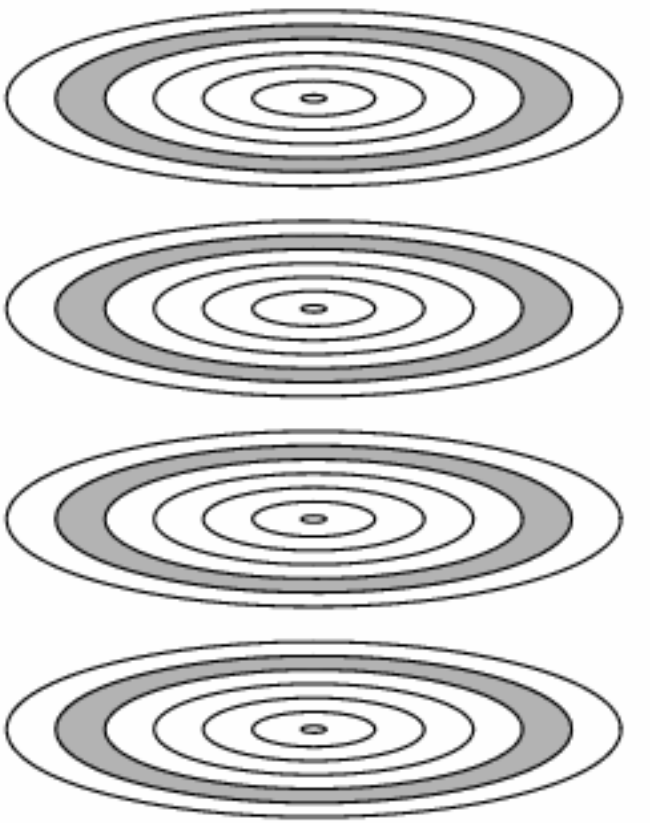
Secondary Storage Devices



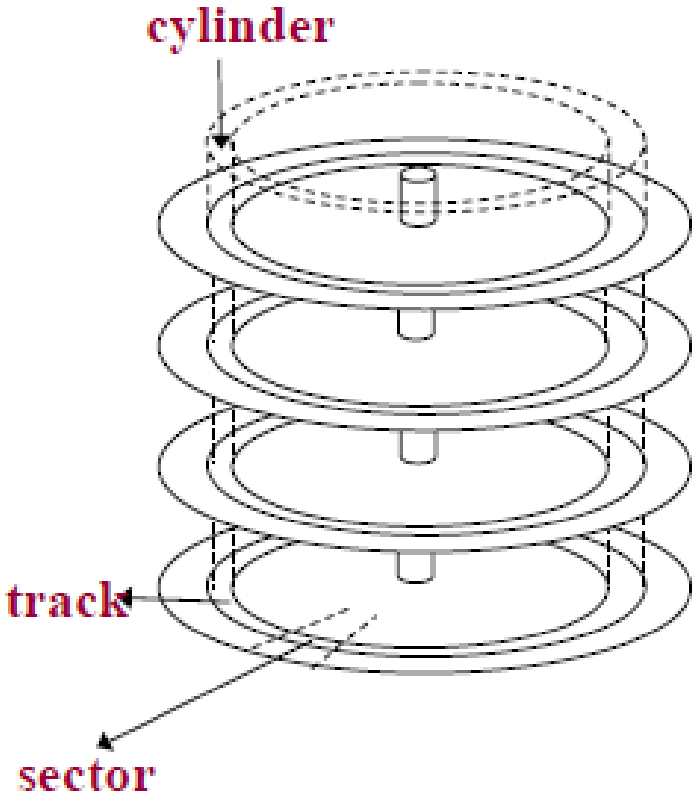
Hard Disk



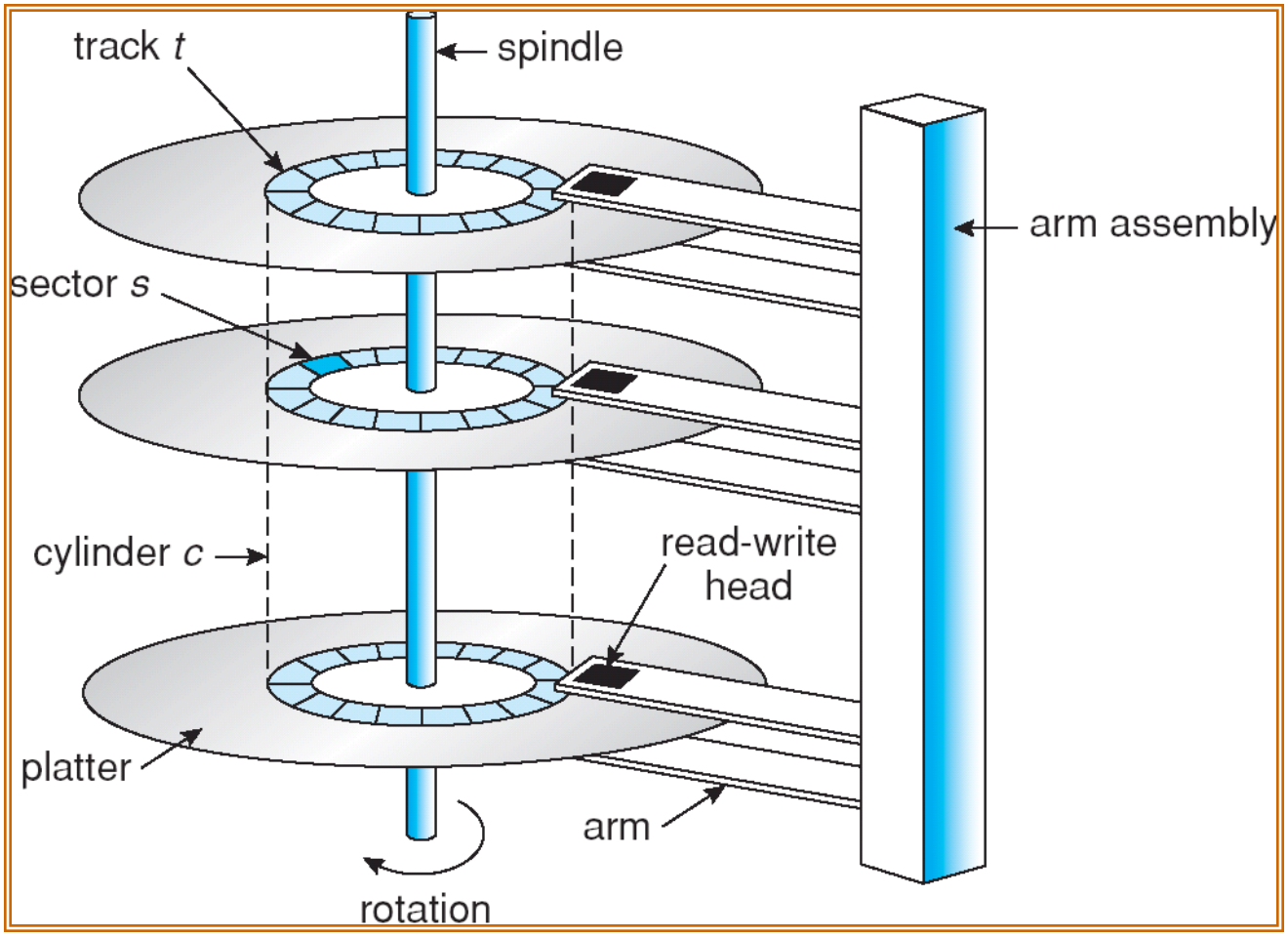
Group of Platters



Cylinder



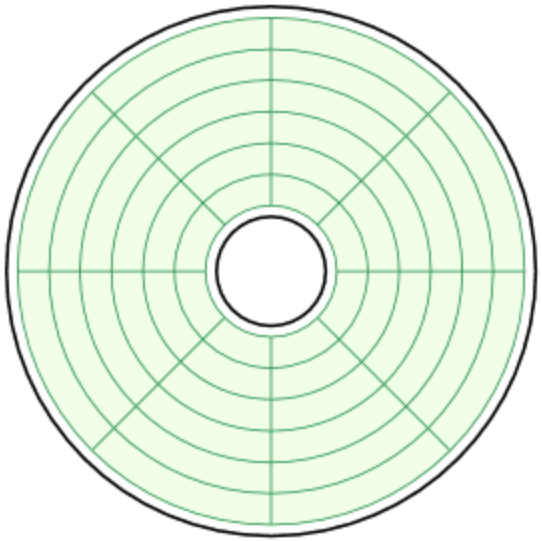
Moving Head Disk Mechanism



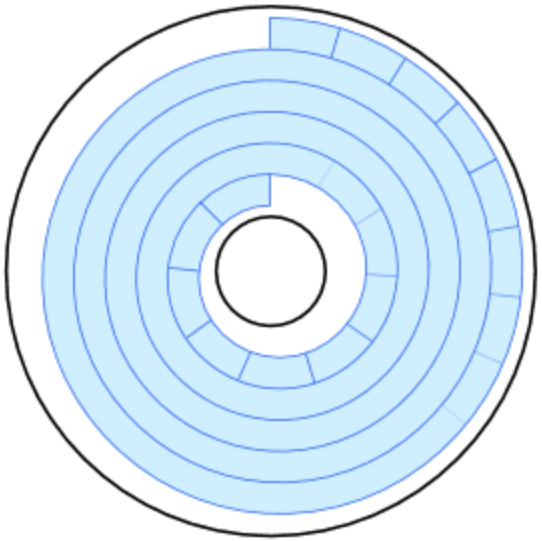
Disk Structure

- 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
  - Logical to physical address should be easy
    - Except for bad sectors
    - Non-constant # of sectors per track via constant angular velocity

CAV and CLV



Constant angular velocity disc



Constant linear velocity disc

Important Terms

- **Seek time:** The time taken by the R-W head to reach the desired track from it's current position.
- **Rotational latency:** Time taken by the sector to come under the R-W head.
- **Data transfer time:** Time taken to transfer the required amount of data. It depends upon the rotational speed.
- **Controller time:** The processing time taken by the controller.
- **Average Access time:** Average seek time + Average Rotational latency + data transfer time + controller time.

Rotational latency

- Max latency = time of a full revolution = sec / rev
- Average latency = time of ½ revolution = ½ of max latency

- Disk rotation speed is measured in Revolutions Per Minute (RPMs).
- Latency is the inverse of speed.
- Time of revolution can be computed as the inverse of revolutions per time.
- For example, assume we have a 3600RPM drive,
- So, the max latency = 16.67ms and the average latency = 8.33ms.

Transfer Time

- The transfer time for a disk operation is the time required to transfer the data from (or to) the disk surface to (or from) the computer once the start of the data is under the R/W head.
- Transfer time is based on:
  1 - speed of rotation
  2 - density of data on the track
  3 - amount of data to be transferred
- The rotational speed of the drive and the track capacity, or track density, can be combined into a single value which is the transfer rate.
- Remembering that one track = 1 revolution, we find ...

  Transfer Rate = Track capacity / Maximum Rotational latency

  and so, the time to transfer a set of data is

  Transfer Time = Amount to Transfer/Transfer Rate

Question 1

Consider a typical disk that rotates at 15000 rotations per minute (RPM) and has a transfer rate of 50 × 10^6bytes/sec. If the average seek time of the disk is twice the average rotational delay and the controller's transfer time is 10 times the disk transfer time, the average time (in milliseconds) to read or write a 512 byte sector of the disk is _____

Question 2

- Disk parameters:
  - Transfer size is 8K bytes
  - Advertised average seek time is 12 ms
  - Disk spins at 7200 RPM
  - Transfer rate is 4 MB/sec
- Controller overhead is 2 ms
- Assume that disk is idle so no queuing delay
- What is average disk access time for a sector?

Question 3

Consider a hard disk with: 4 surfaces, 64 tracks/surface,128 sectors/track and 256 bytes/sector and the disk is rotating at 3600 RPM.

a) What is the capacity of the hard disk?
b) What is the data transfer rate?
c) What is the average access time?

Question 4

A hard disk system has the following parameters :

- Number of tracks = 500
- Number of sectors/track = 100
- Number of bytes /sector = 500
- Time taken by the head to move from one track to adjacent track = 1 ms
- Rotation speed = 600 rpm.

What is the average time taken for transferring 250 bytes from the disk ?

Question 5

Consider a disk pack with 16 surfaces, 128 tracks per surface and 256 sectors per track. 512 bytes of data are stored in a bit serial manner in a sector. The capacity of the disk pack and the number of bits required to specify a particular sector in the disk are respectively:

A. 256 Mbyte, 19 bits
B. 256 Mbyte, 28 bits
C. 512 Mbyte, 20 bits
D. 64 Gbyte, 28 bits

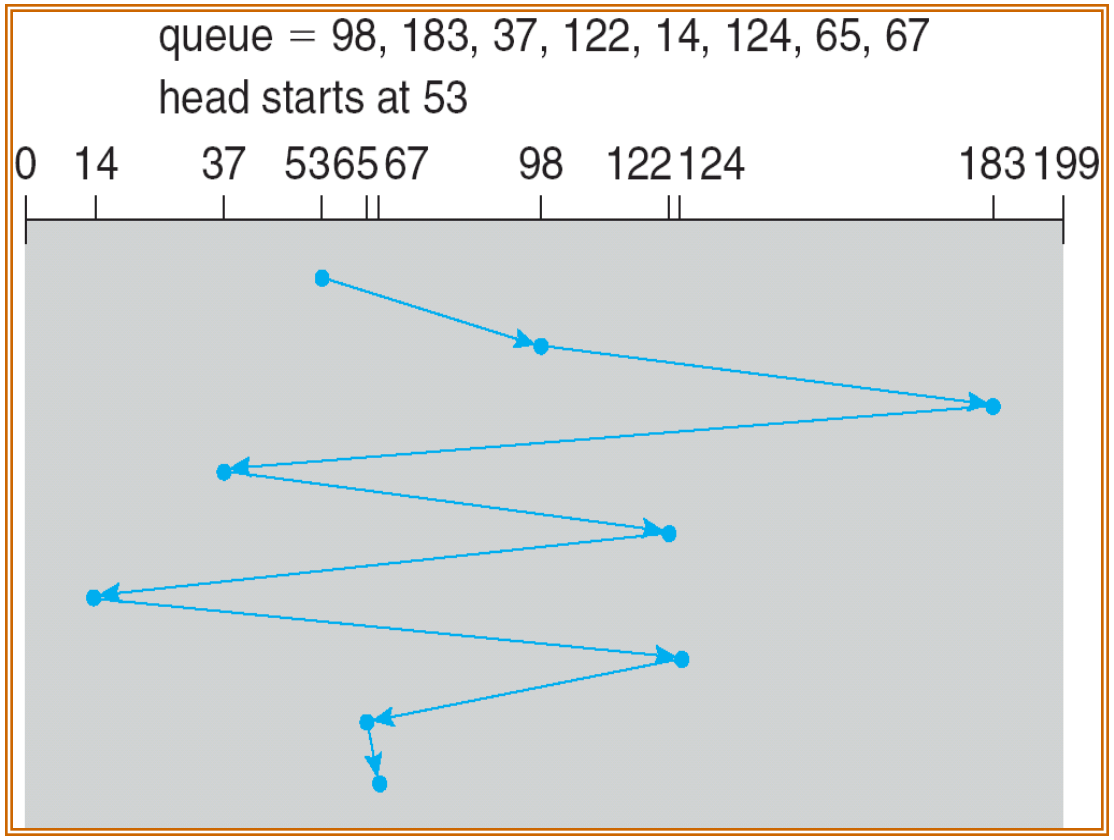
Disk scheduling Algorithms

- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

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

Head pointer 53

FCFS (640 moves)



Total head movement =
(98-53) + (183-98)+
(183- 37) + (122-37)+
(122-14) +(124-14)+
(124-65) + (67-65)

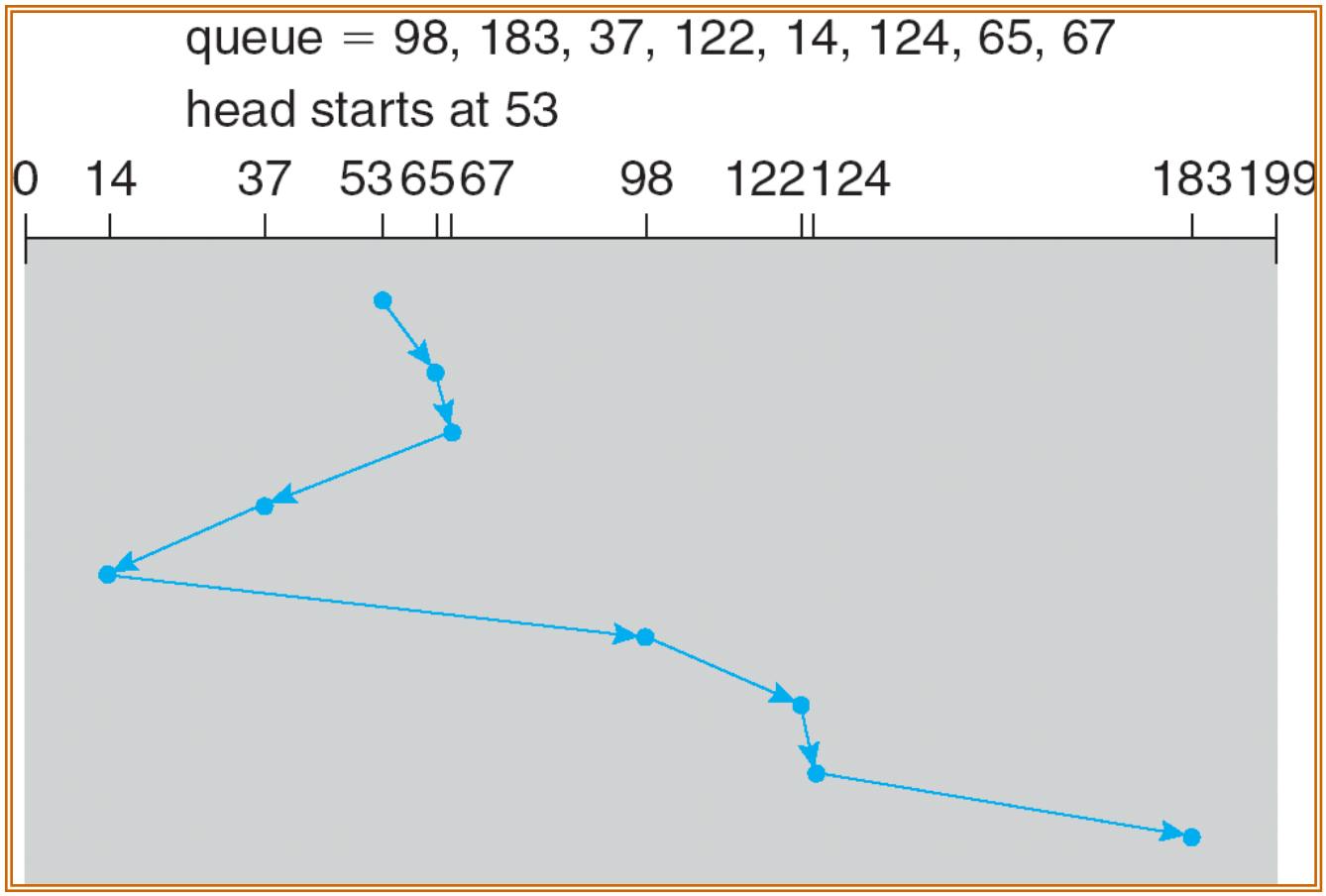
= 45+85+146+85+108+
110+59+2

=640

SSTF

- Shortest Seek Time First -- selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of SJF scheduling; may cause starvation of some requests
- Disadvantages:
  - Starvation for some request
  - Switching direction on the frequent basis slows working of algorithm
  - Not most optimal
- Illustration shows total head movement of 236 cylinders

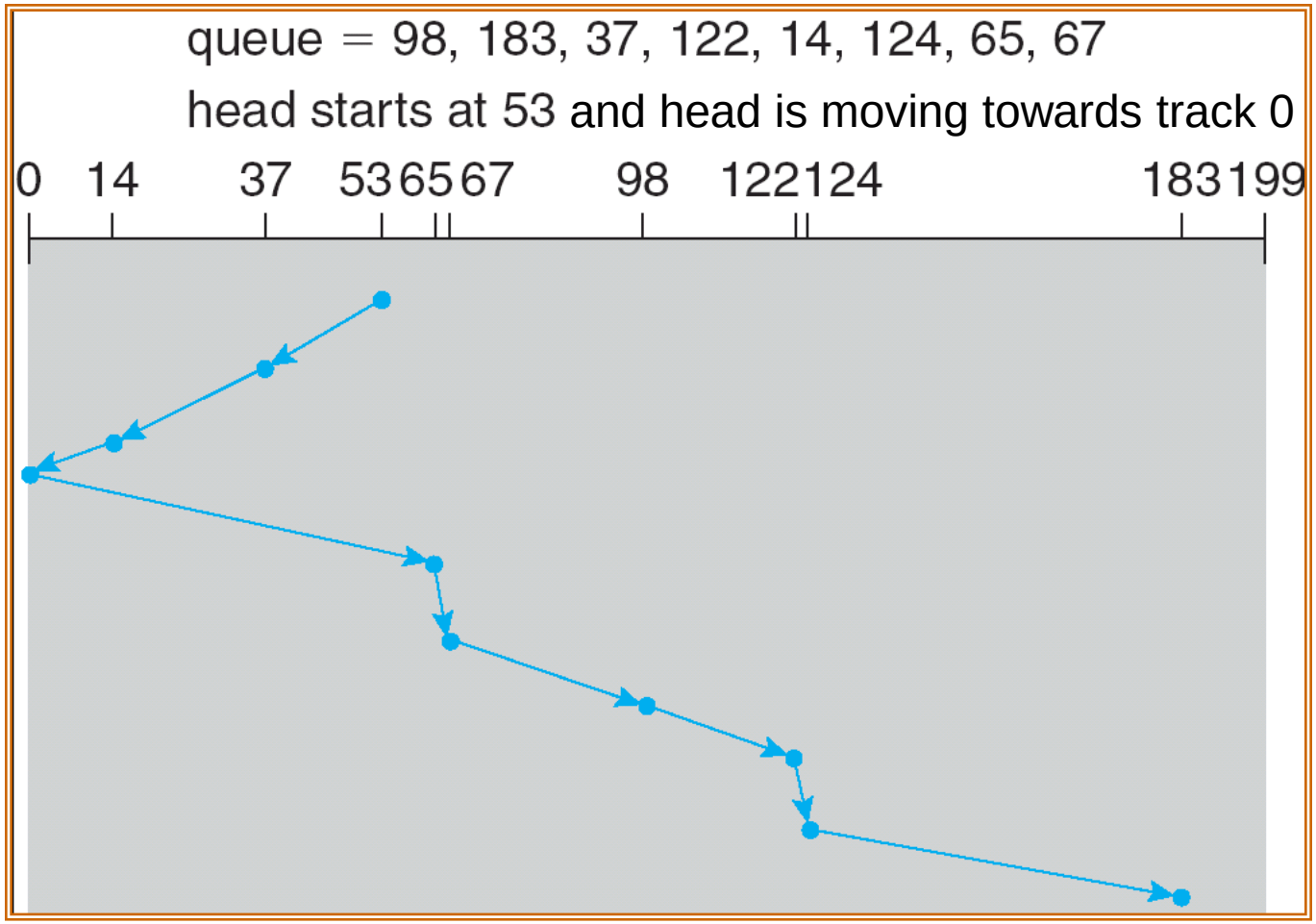
SSTF (236 moves)



SCAN

- **SCAN algorithm.** The disk arm starts at one end of the disk, and moves toward the other end, servicing requests until it gets to the other end of the disk, where the head movement is reversed and servicing continues.
- Sometimes it is called the **elevator algorithm**
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest

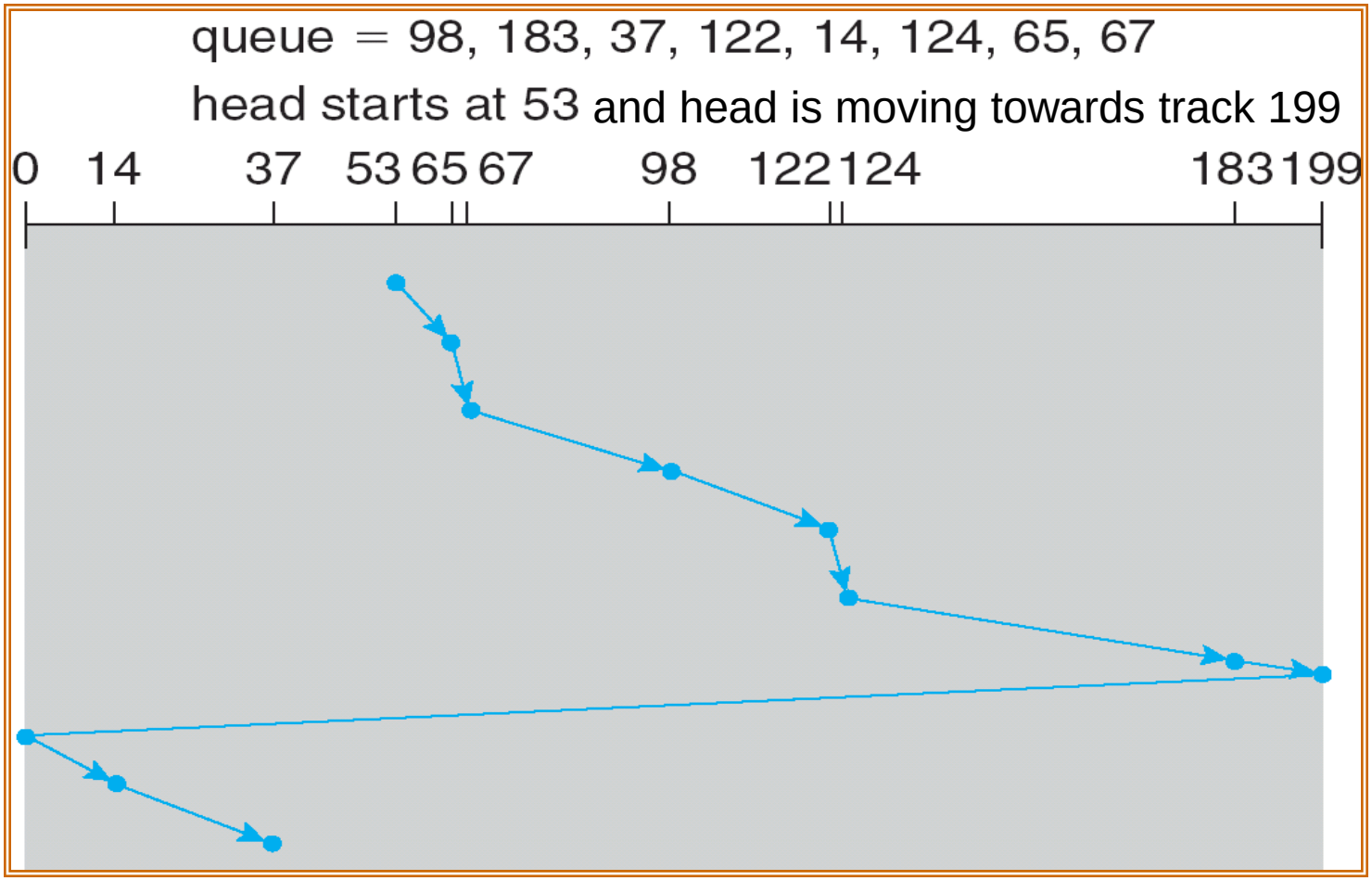
SCAN / Elevator



C-SCAN

- Provides a more uniform wait time than SCAN
- The head moves from one end of the disk to the other, servicing requests as it goes
  - When it reaches the other end, however, it immediately returns to the beginning of the disk, without servicing any requests on the return trip
- Treats the cylinders as a circular list that wraps around from the last cylinder to the first one

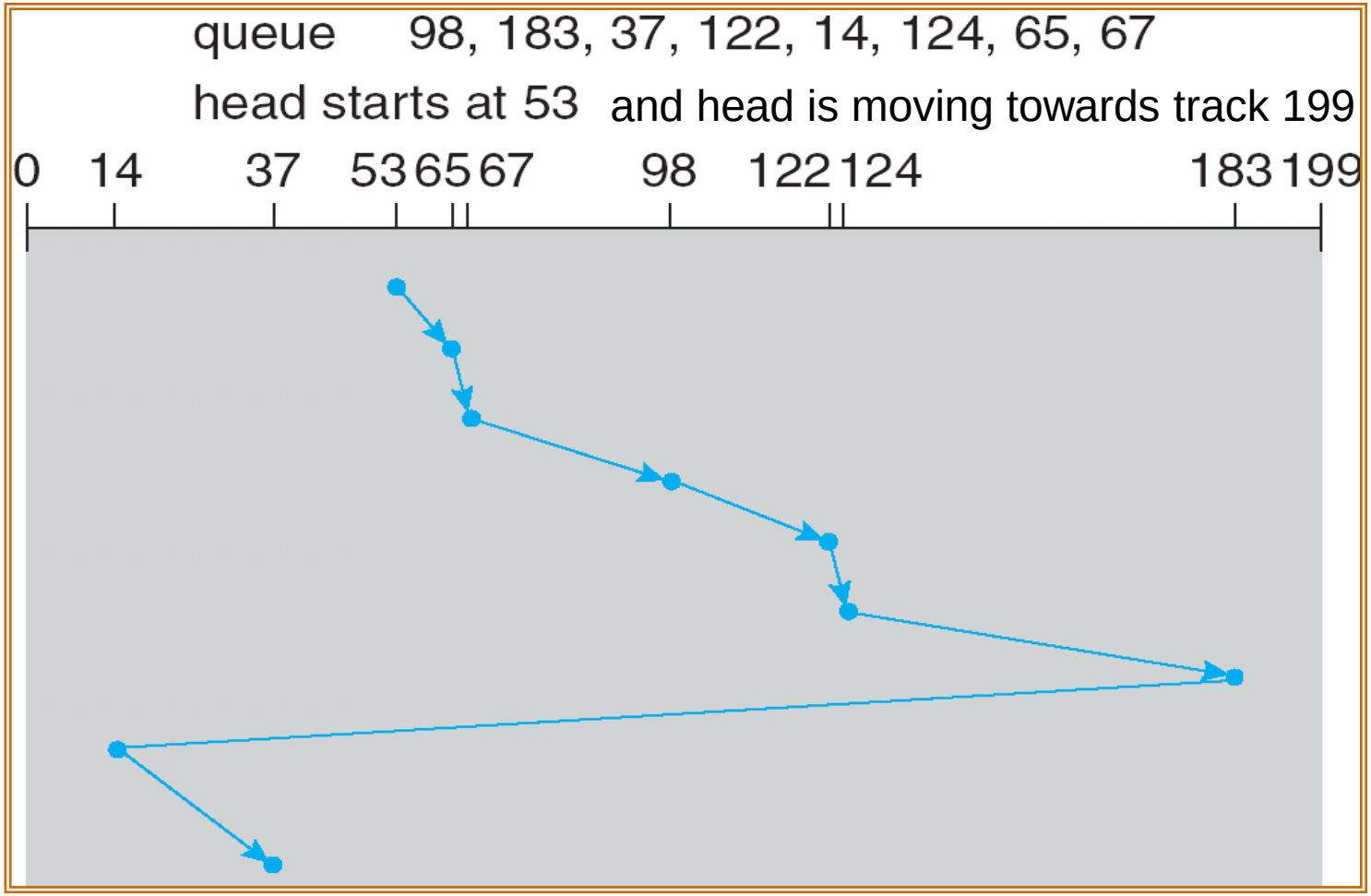
C-SCAN


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53 and head is moving towards track 199
0
14
37
53
65
67
98
122
124
183
199

LOOK and C-LOOK

- **LOOK** is a version of SCAN. Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk
- **C-LOOK** a version of C-SCAN. Arm only goes as far as the last request in one direction, then reverses direction immediately, without first going all the way to the end of the disk

C-LOOK


queue 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53 and head is moving towards track 199
0 14 37 53 65 67 98 122 124 183 199

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

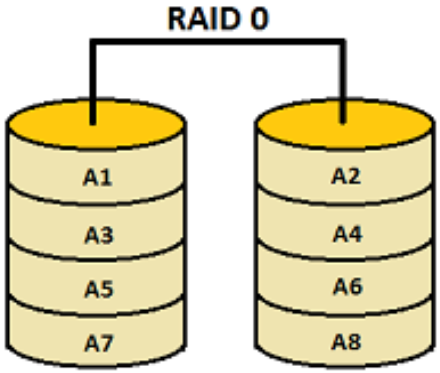
Question 6

On a disk with 1000 cylinders, numbers 0 to 999, compute the number of tracks the disk arm must move to satisfy all the requests in the disk queue. Assume the last request services was at track 345 and head is moving towards track 0. The queue in FIFO order contains the requests for the following tracks: 123, 874, 692, 475, 105, 376. Perform the computation for the following scheduling algorithms:

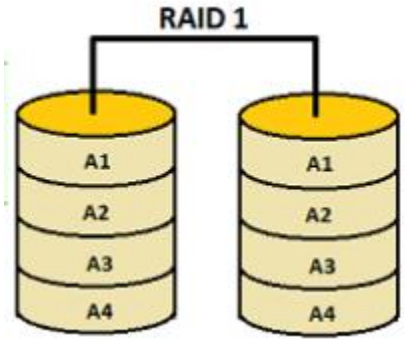
a) FIFO
b) SSTF
c) SCAN
d) LOOK
e) C-SCAN
f) C-LOOK

RAID Structure

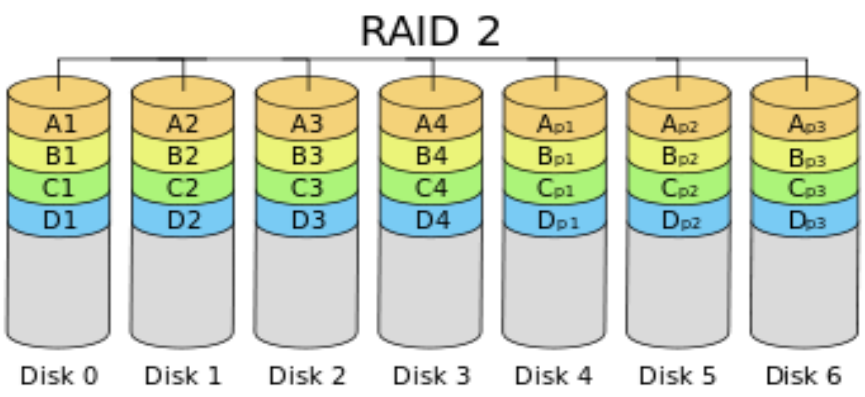
- RAID – redundant array of inexpensive disks
  - multiple disk drives provides reliability via **redundancy**
  - Now it is Redundant Array of Independent Disk
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively



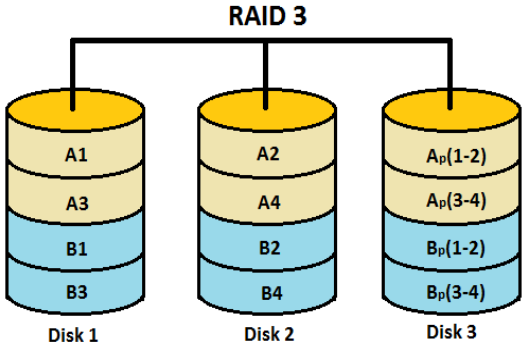
RAID 0: non-redundant striping



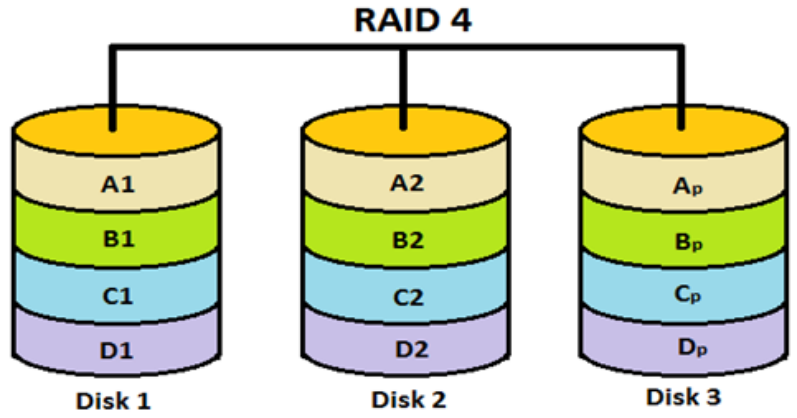
RAID 1: mirrored disks



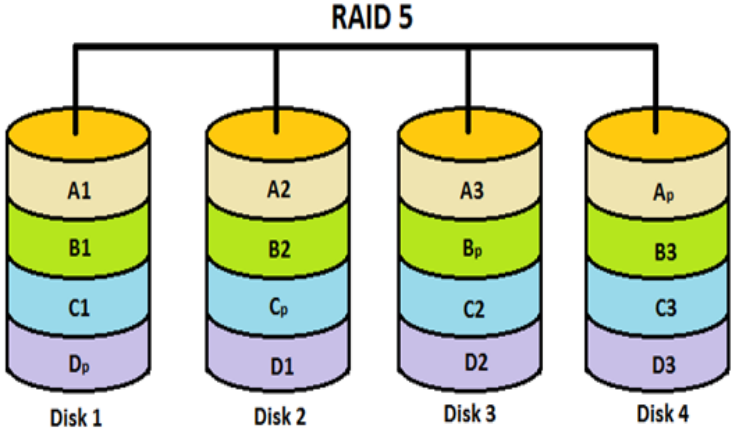
RAID 2: memory-style error-correcting codes



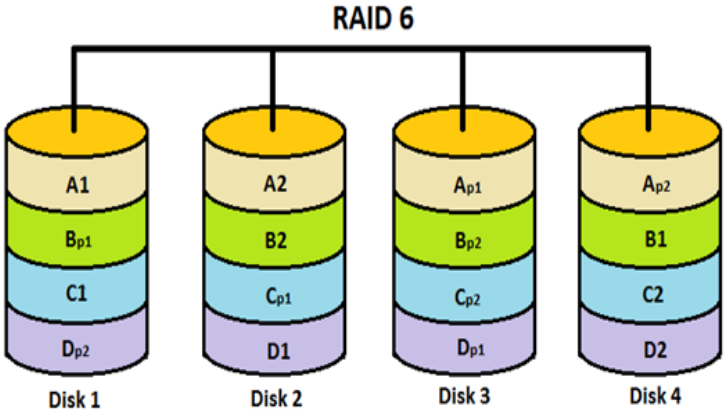
RAID 3: bit-interleaved parity



RAID 4: block-interleaved parity



RAID 5: block-interleaved distributed parity



RAID 6: P + Q redundancy

RAID 0

Pros	Cons
Data is stripped into multiple drives	No support for Data Redundancy
Disk space is fully utilized	No support for Fault Tolerance
Minimum 2 drives required	No error detection mechanism
High performance	Failure of either disk results in complete data loss in respective array

RAID-1

Pros	Cons
Performs mirroring of data i.e identical data from one drive is written to another drive for redundancy.	Expense is higher (1 extra drive required per drive for mirroring)
High read speed as either disk can be used if one disk is busy	Slow write performance as all drives has to be updated
Array will function even if any one of the drive fails	
Minimum 2 drives required	

RAID-2

Pros	Cons
BIT level stripping	It is used with drives with no built in error detection mechanism
Uses Hamming code for error detection	These days all SCSI drives have error detection
	Additional drives required for error detection

RAID-3

Pros	Cons
BIT level stripping with parity	Additional drives required for parity
One designated drive is used to store parity	No redundancy in case parity drive crashes
Data is regenerated using parity drive	Slow performance for operating on small sized files
Data is accessed parallel	
High data transfer rates (for large sized files)	
Minimum 3 drives required	

RAID-4

Pros	Cons
BLOCK level stripping along with dedicated parity	Since only 1 block is accessed at a time so performance degrades
One designated drive is used to store parity	Additional drives required for parity
Data is accessed independently	Write operation becomes slow as every time a parity has to be entered
Minimum 3 drives required	
High read performance since data is accessed independently.	

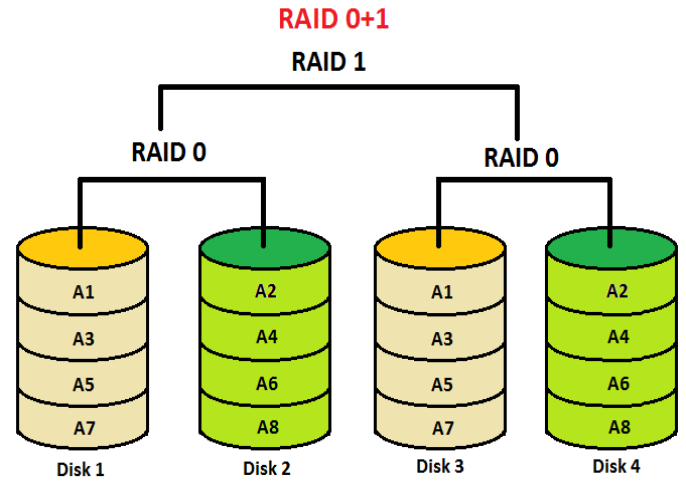
RAID-5

Pros	Cons
Block level stripping with DISTRIBUTED parity	In case of disk failure recovery may take longer time as parity has to be calculated from all available drives
Parity is distributed across the disks in an array	Cannot survive concurrent drive failures
High Performance	
Cost effective	
Minimum 3 drives required	

RAID-6

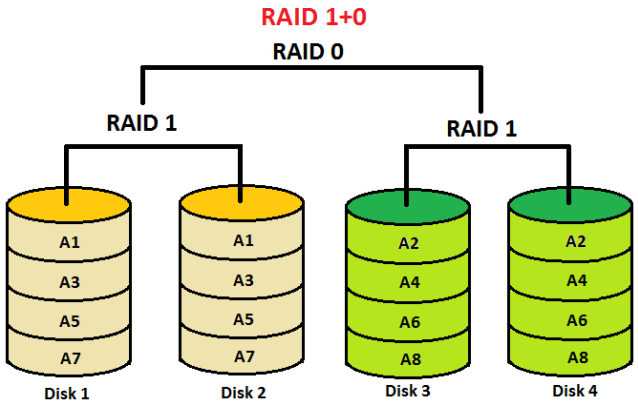
Pros	Cons
Block level stripping with DUAL distributed parity	Cost Expense can become a factor
2 parity blocks are created	Writing data takes longer time due to dual parity
Can survive concurrent 2 drive failures in an array	
Extra Fault Tolerance and Redundancy	

RAID (0+1)



Pros	Cons
No parity generation	Costly as extra drive is required for each drive
Performs RAID 0 to strip data and RAID 1 to mirror	100% disk capacity is not utilized as half is used for mirroring
Stripping is performed before Mirroring	Very limited scalability
Usable capacity is n/2 * size of disk (n = no. of disks)	
Drives required should be multiple of 2	
High Performance as data is stripped	

RAID (1+0)



Pros	Cons
No Parity generation	Very Expensive
Performs RAID 1 to mirror and RAID 0 to strip data	Limited scalability
Mirroring is performed before stripping	
Drives required should be multiple of 2	
Usable capacity is n/2 * size of disk (n = no. of disks)	
Better Fault Tolerance than RAID 0+1	
Better Redundancy and faster rebuild than 0+1	
Can sustain multiple drive failures	