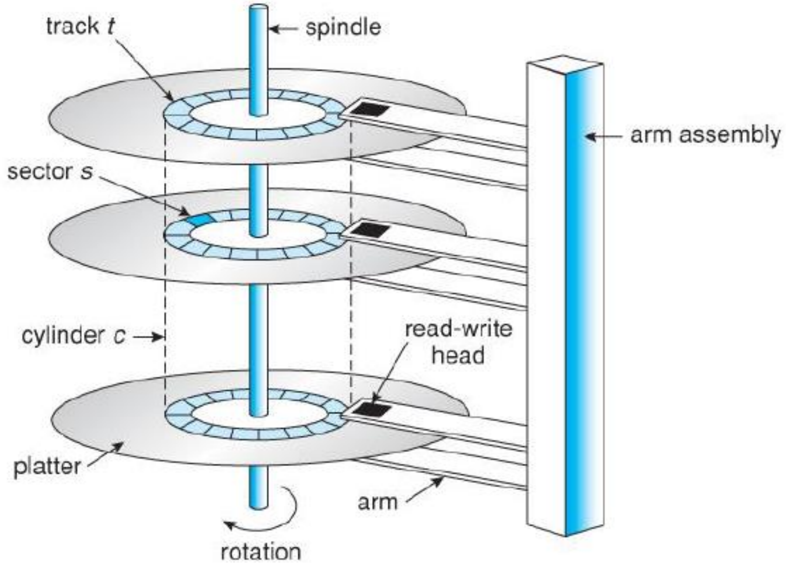
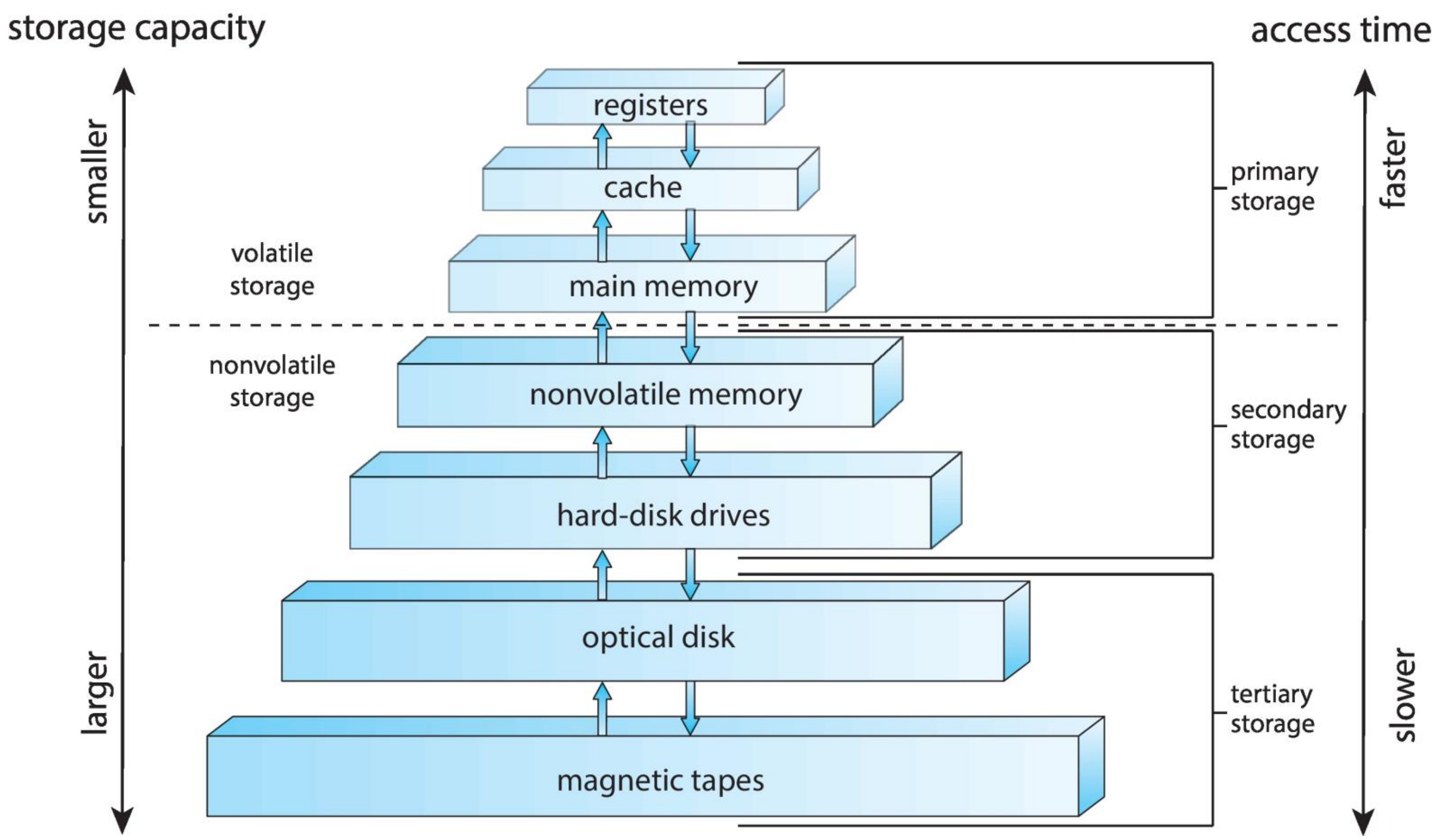
CSCI 509

OPERATING SYSTEMS

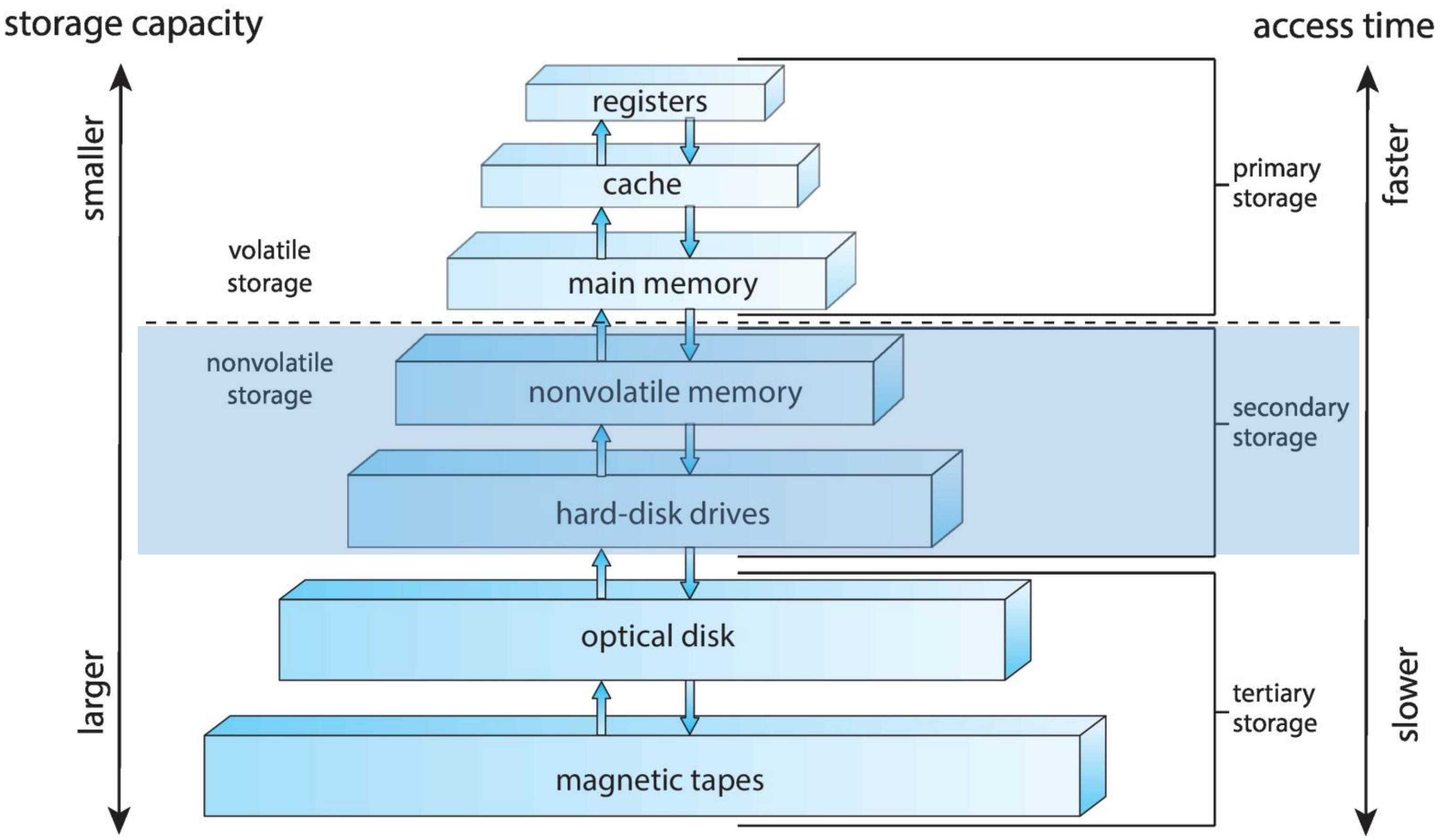
CHAPTER 11: MASS STORAGE



MASS STORAGE



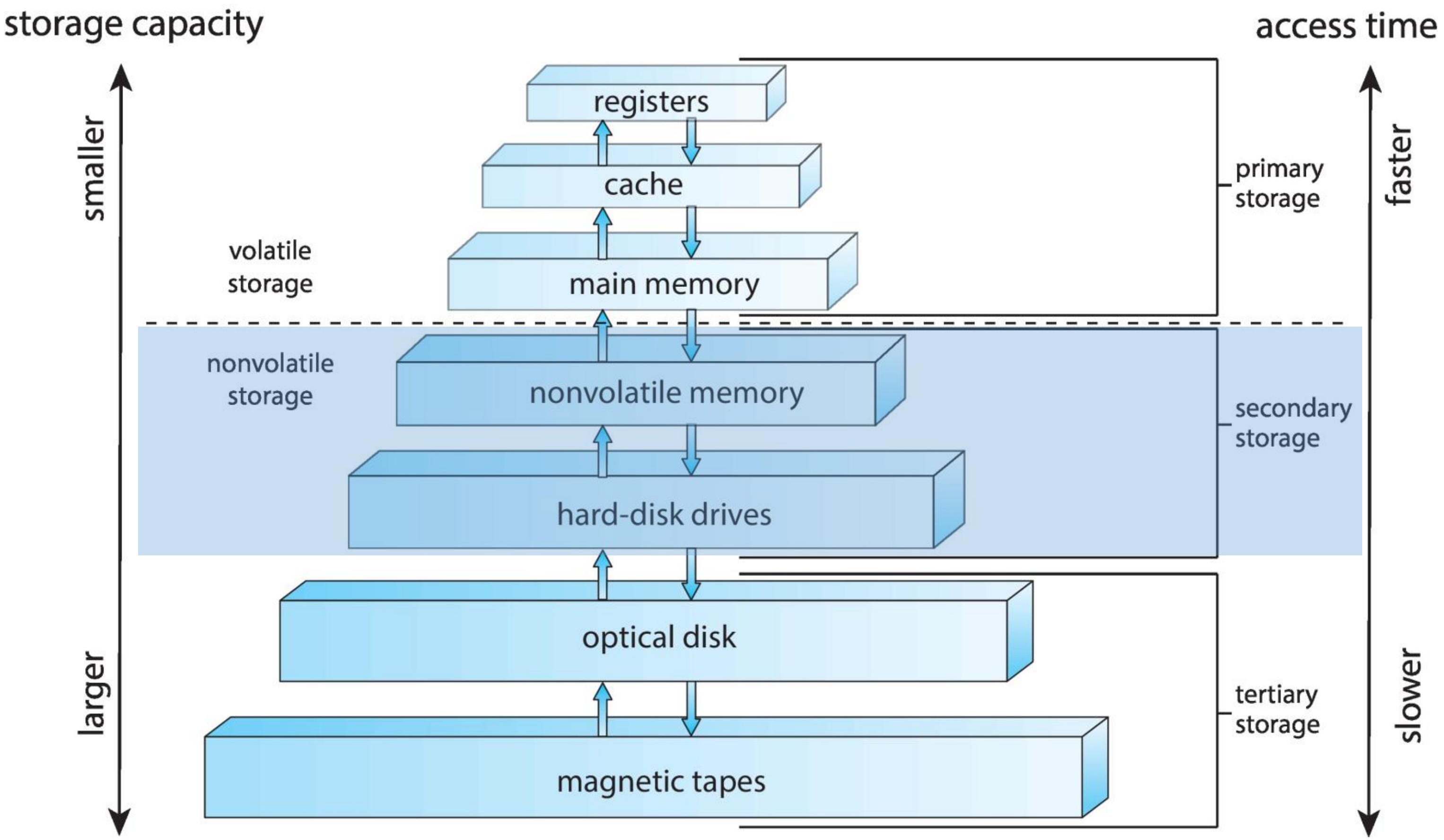
MASS STORAGE



MASS STORAGE

Mass storage is where:

- The operating system is stored
- Swap Space and Virtual Memory
- Files System Implemented/Stored



OVERVIEW OF MASS STORAGE

Magnetic Disks

OVERVIEW OF MASS STORAGE

Magnetic Disks

Solid State Drives

OVERVIEW OF MASS STORAGE

Magnetic Disks

Solid State Drives

RAM Storage

OVERVIEW OF MASS STORAGE

Magnetic Disks

Slow

Cost Efficient

Solid State Drives

Fast

Less Cost Efficient

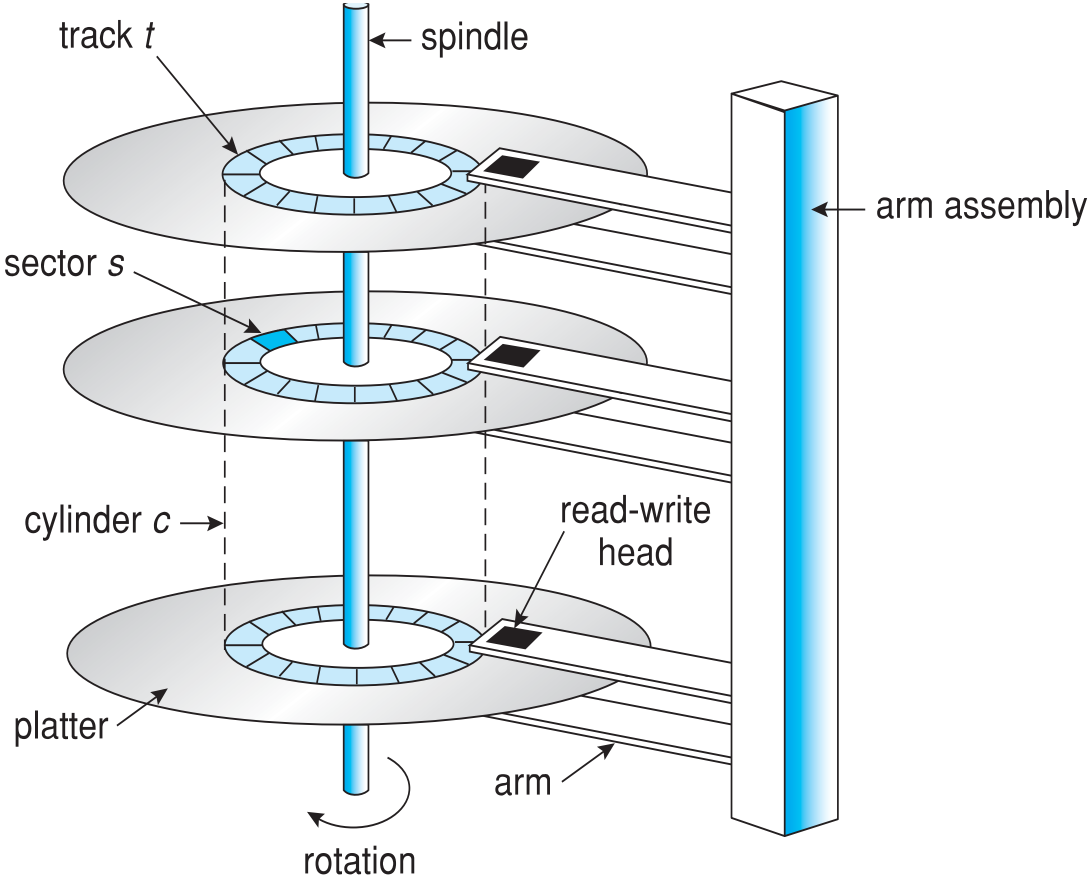
RAM Storage

Fastest

Very Expensive

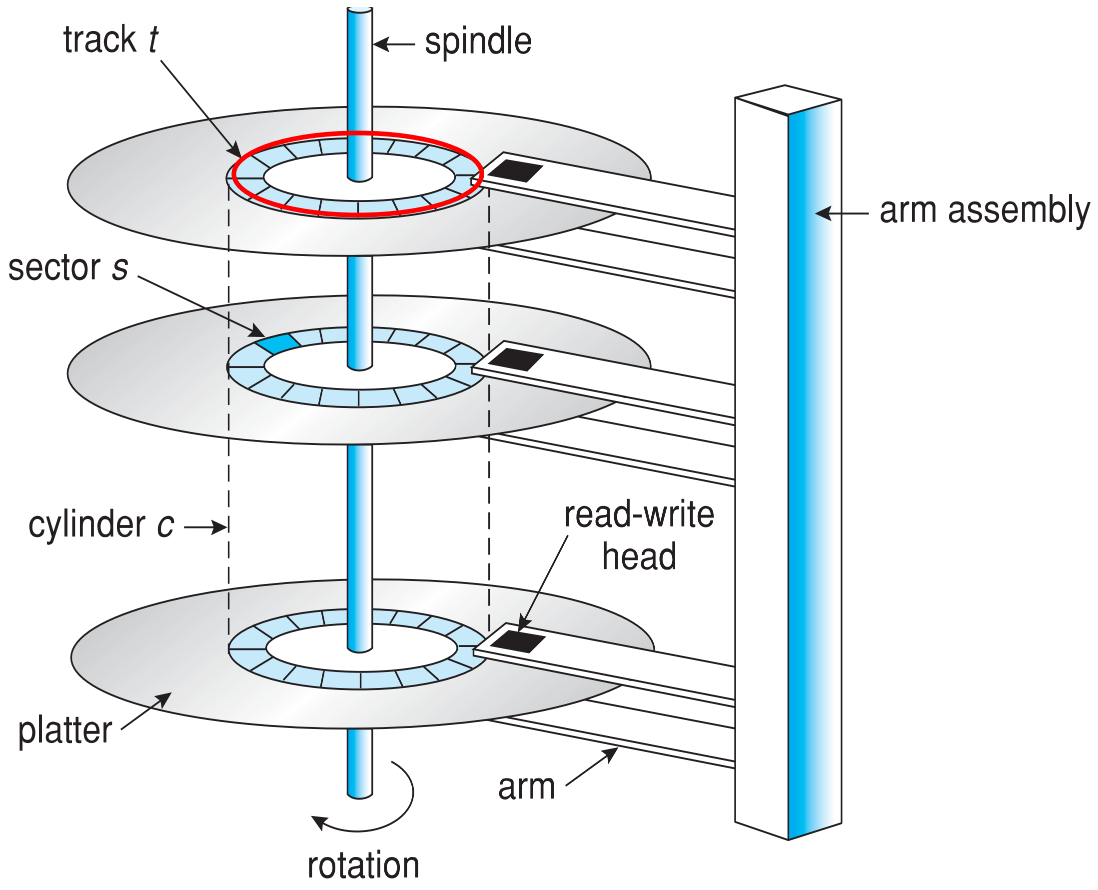
Needs back up battery since it’s volatile.

MAGNETIC DISKS



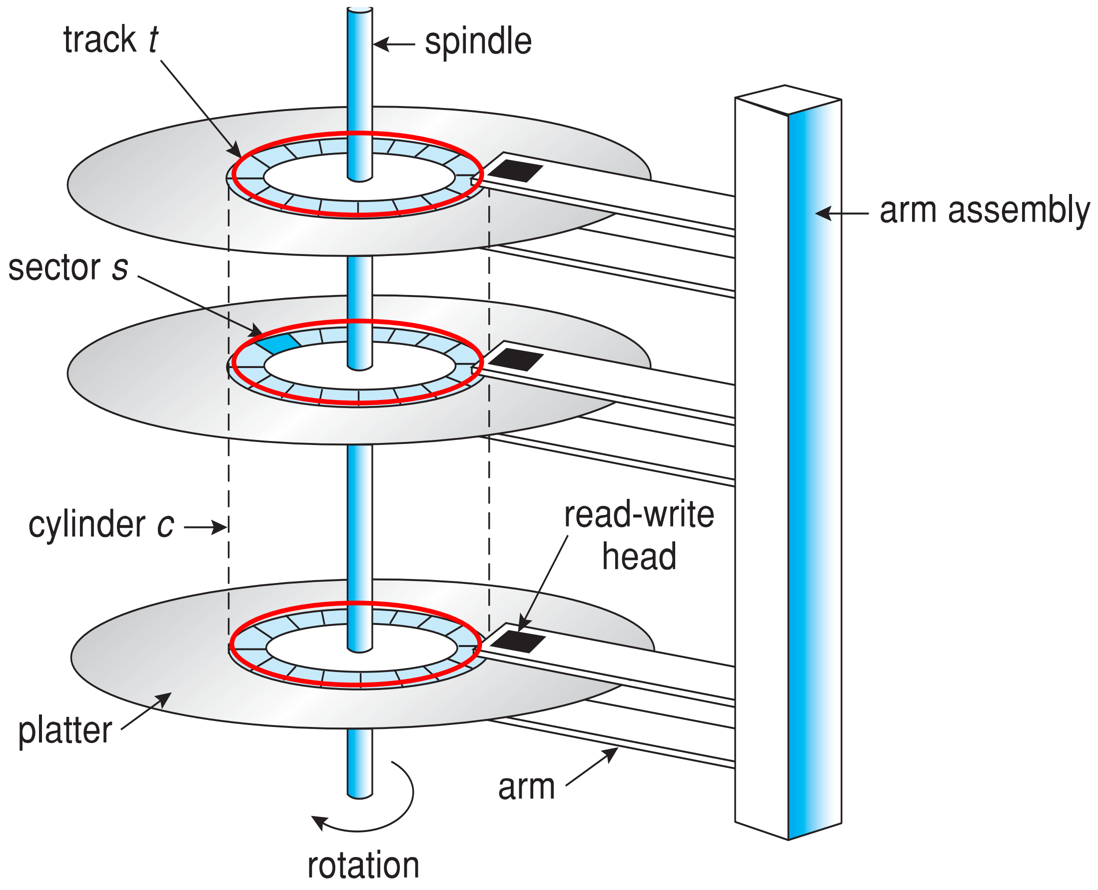
MAGNETIC DISKS

- Track a ‘1-bit’ width ring.



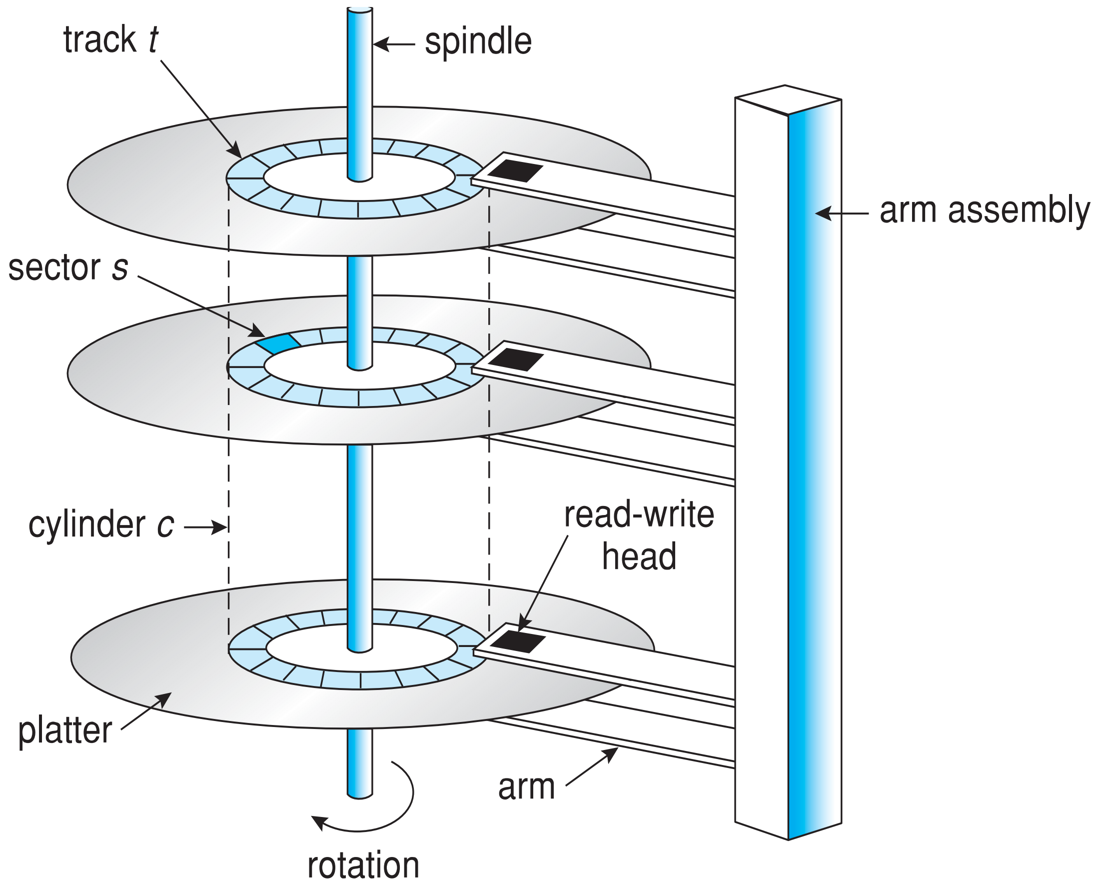
MAGNETIC DISKS

- Track a '1-bit' width ring.
- Cylinder: the set of tracks among all platters for a single seek position



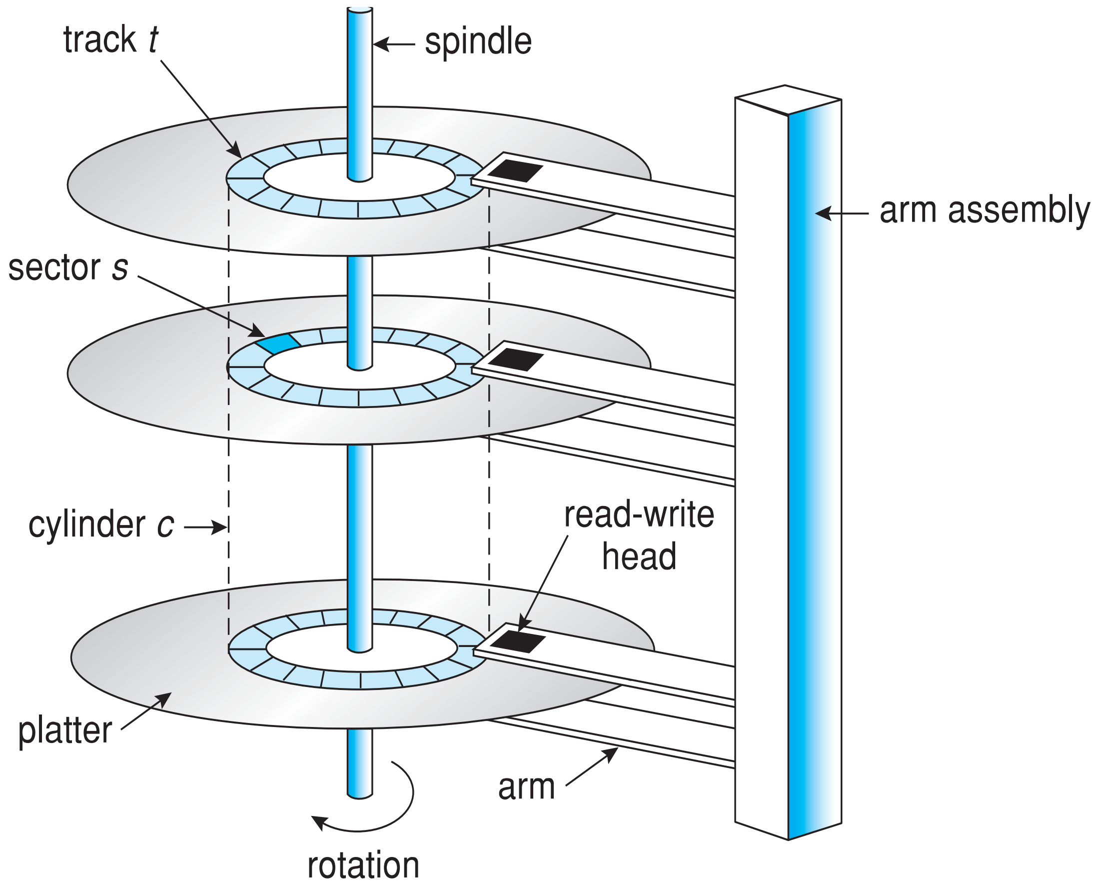
MAGNETIC DISKS

- Track a '1-bit' width ring.
- Cylinder: the set of tracks among all platters for a single seek position
- Sector: a portion of the track.



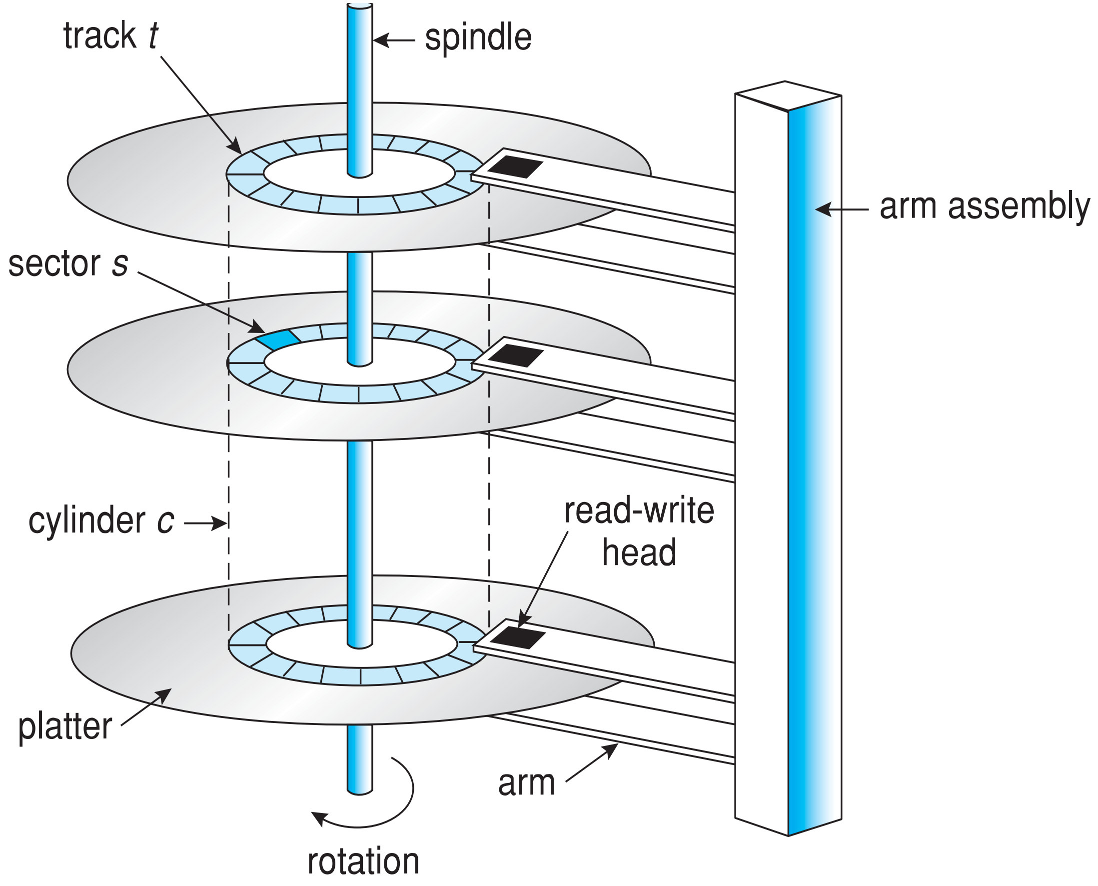
MAGNETIC DISKS

- Track a '1-bit' width ring.
- Cylinder: the set of tracks among all platters for a single seek position
- Sector: a portion of the track.
- All read/write operation are done as 'sector' read/writes.
- To read one byte, you need to read the whole sector.



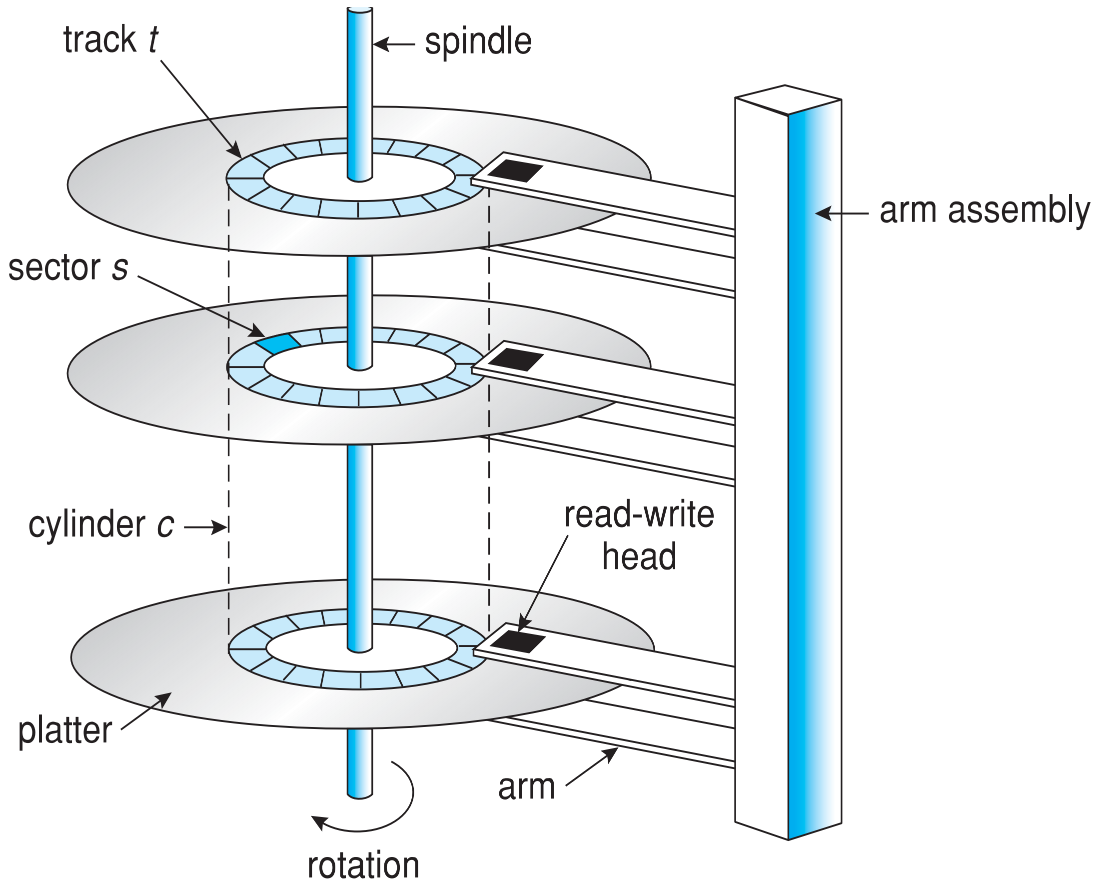
MAGNETIC DISKS

- Track a '1-bit' width ring.
- Cylinder: the set of tracks among all platters for a single seek position
- Sector: a portion of the track.
- All read/write operation are done as 'sector' read/writes.
- To read one byte, you need to read the whole sector.
- Sector size is almost always 512 bytes.



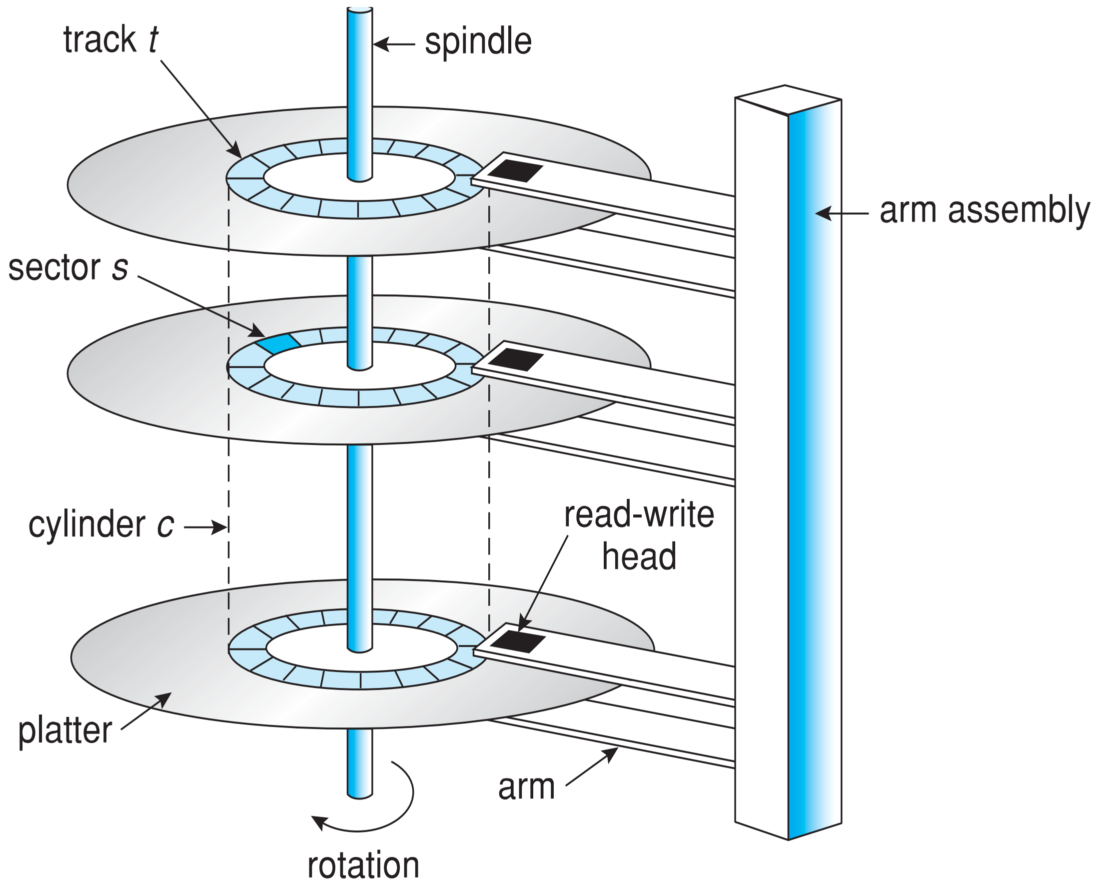
MAGNETIC DISKS

- The operating System, does NOT address the sectors of an HDD.
- Instead it uses 'Block', made up of several sectors.



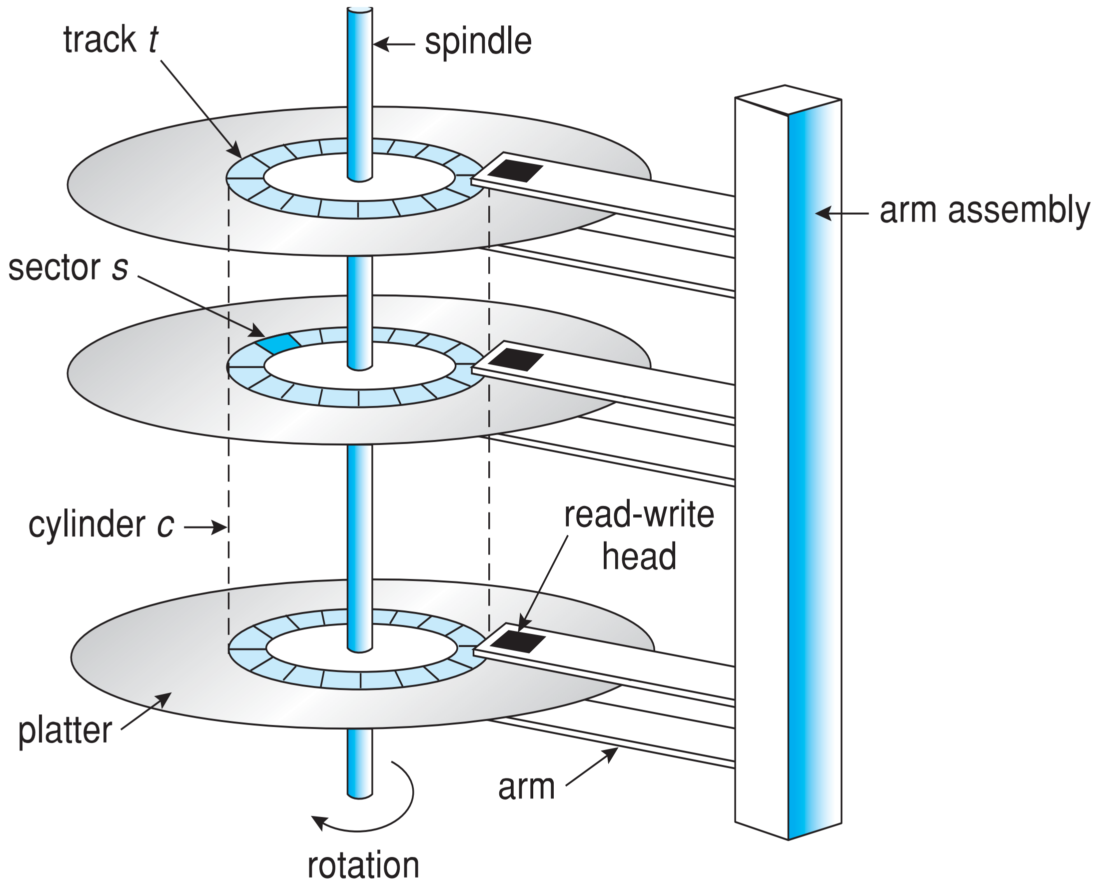
MAGNETIC DISKS

- The operating System, does NOT address the sectors of an HDD.
- Instead it uses 'Block', made up of several sectors.
- OS always reads/write at least one block from the HDD.
- This is done to simplify addressing and usually data is fetched in pages rather than sectors.



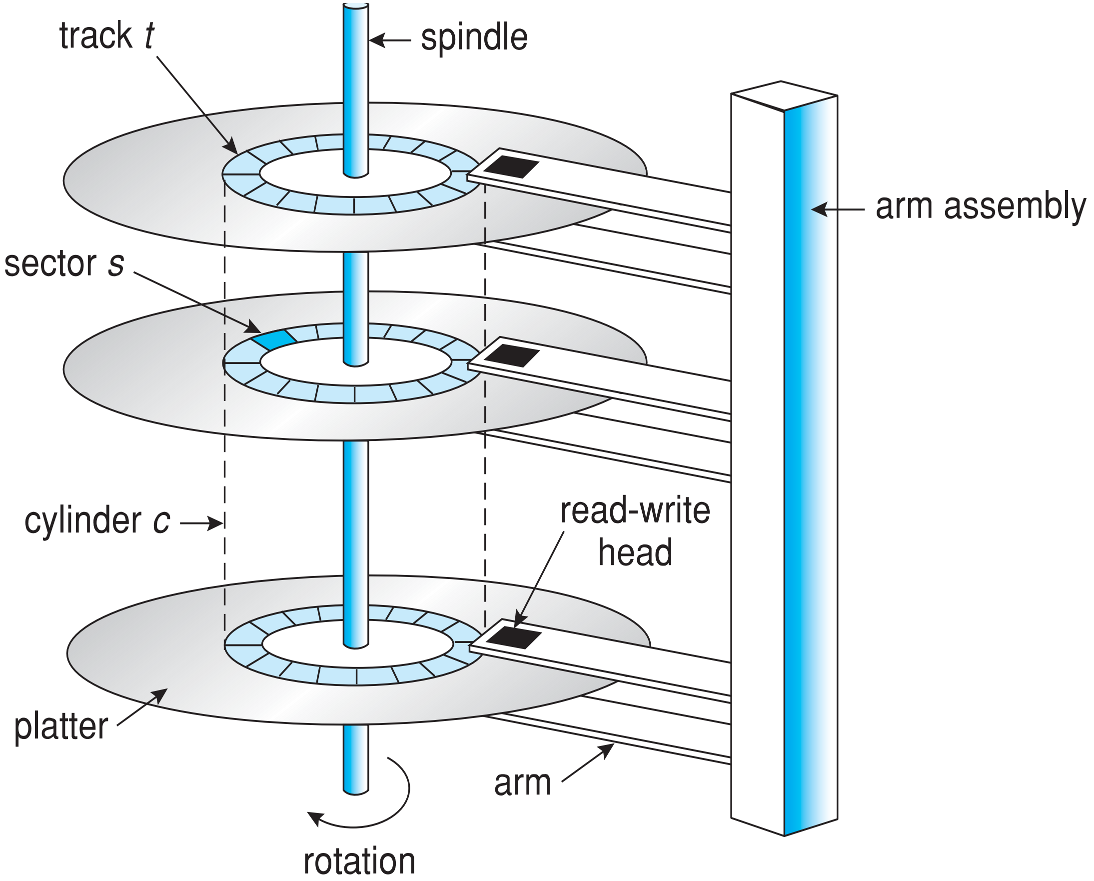
MAGNETIC DISKS

- Most hard drives have 2- 3 platters.
- Both sides of a platter are used for storage.
- When in use, platter spin (6-250 revolutions/second; up to 15,000 RPM).



MAGNETIC DISKS

- Most hard drives have 2- 3 platters.
- Both sides of a platter are used for storage.
- When in use, platter spin (6-250 revolutions/second; up to 15,000 RPM).
- The HDD head should never touch the platter.
- If it does, the event is called a "head crash" and the HDD will be dead.



MAGNETIC DISKS

- Most hard drives have 2- 3 platters.
- Both sides of a platter are used for storage.
- When in use, platter spin (6-250 revolutions/second; up to 15,000 RPM).
- The HDD head should never touch the platter.
- If it does, the event is called a "head crash" and the HDD will be dead.

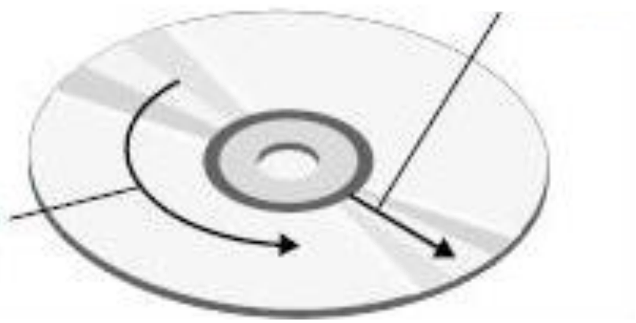
DISK SCHEDULING

- Objective of 'Disk' Scheduling it to minimize mean access time.
- Scheduling: What order to service pending Disk I/O requests?

DISK SCHEDULING

Disk access required mechanical operations. Disk latency is dominated by by:

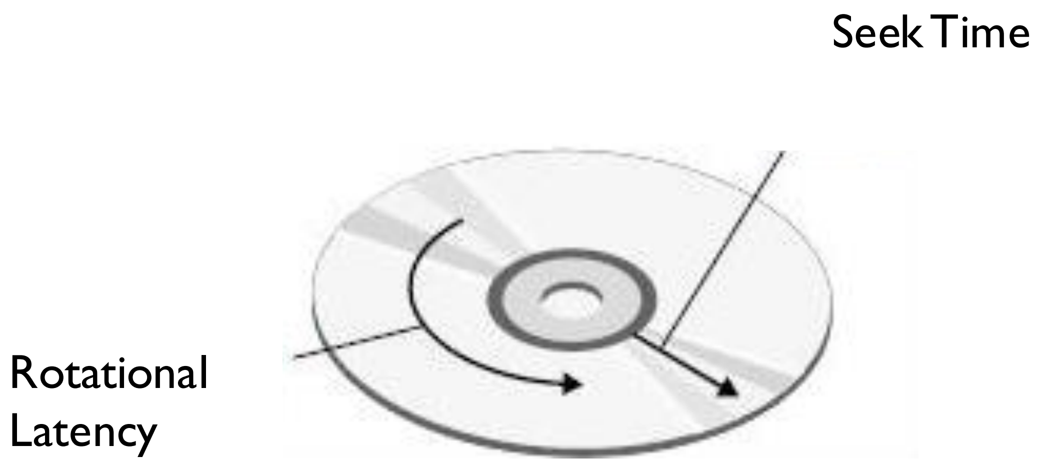
- Seek latency: time required for the head to move to the track.
- Rotational latency: time required for the disk to rotate to the sector.



DISK SCHEDULING

Disk access required mechanical operations. Disk latency is dominated by by:

- Seek latency: time required for the head to move to the track.
- Rotational latency: time required for the disk to rotate to the sector.

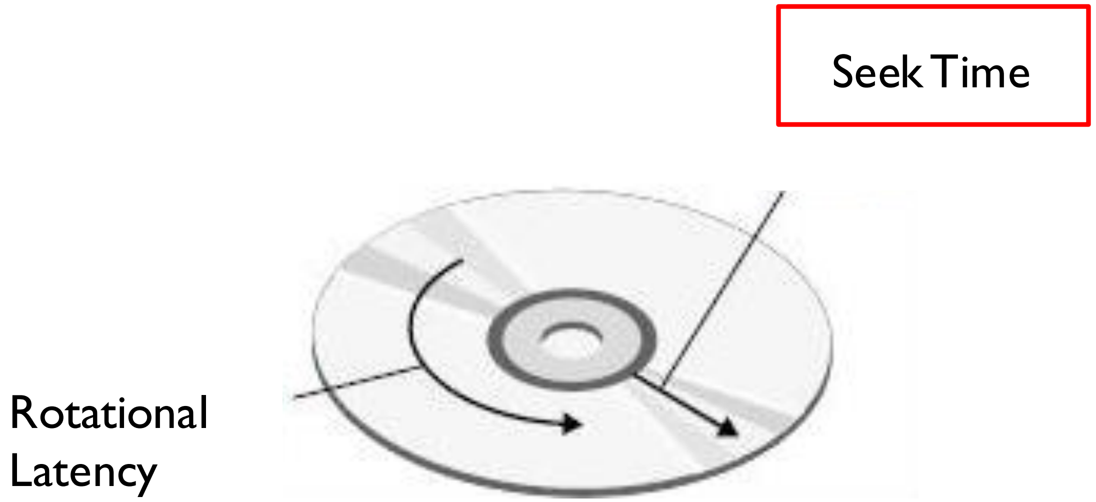


The objective of disk scheduling algorithms is to minimize the *average* latency.

DISK SCHEDULING

Disk access required mechanical operations. Disk latency is dominated by by:

- Seek latency: time required for the head to move to the track.
- Rotational latency: time required for the disk to rotate to the sector.



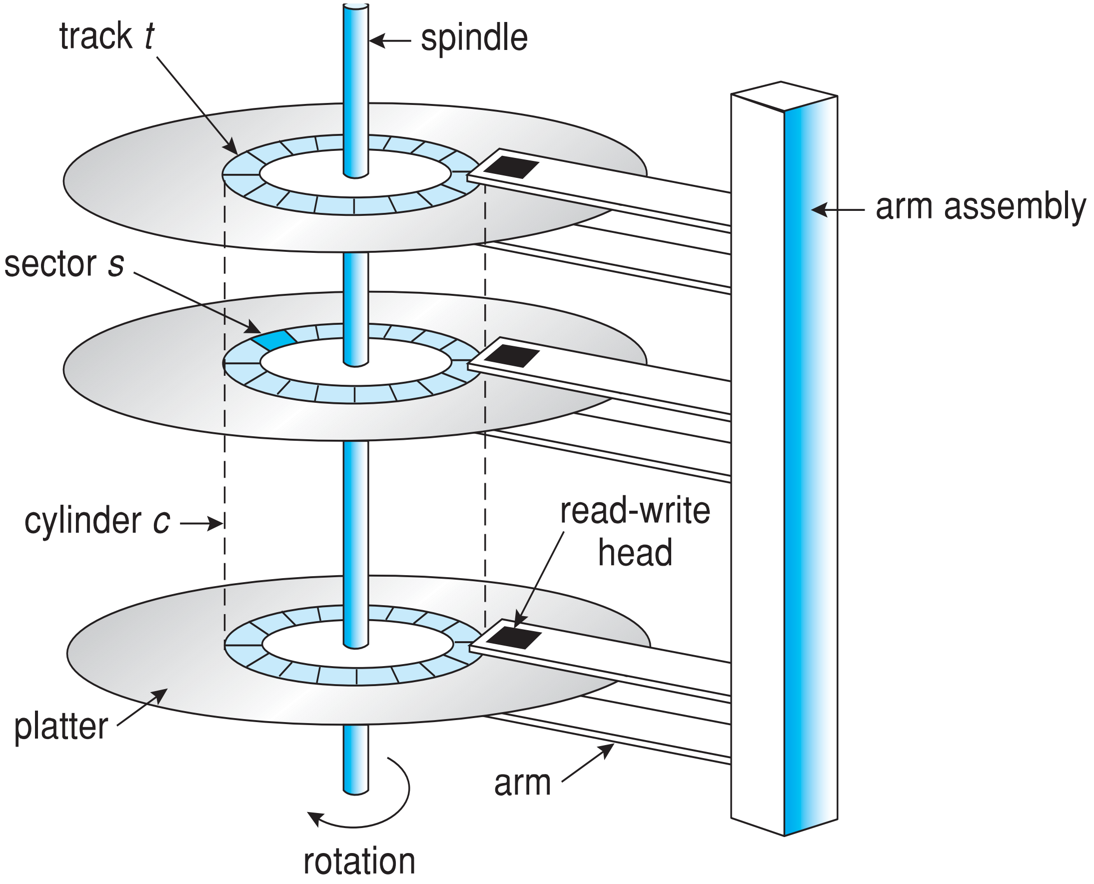
The objective of disk scheduling algorithms is to minimize the <u>*average*</u> latency.

Specifically, scheduling algorithm reduce the average seek latency as there is not much to do regarding rotational latency.

DISK SCHEDULING

Objective: Minimize Seek Time

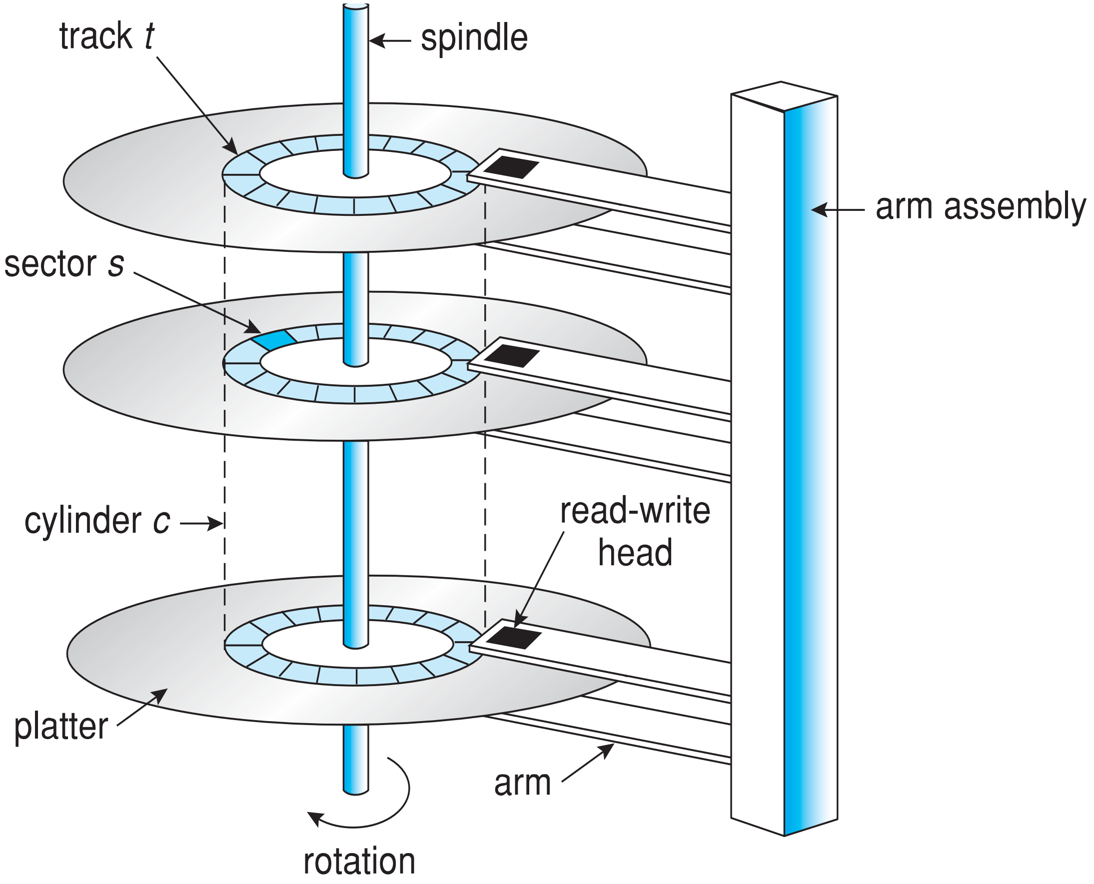
- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.



DISK SCHEDULING

Objective: Minimize Seek Time

- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.
- Each request corresponds to an address that contains:
  - Platter Number
  - Track Number
  - Sector Number

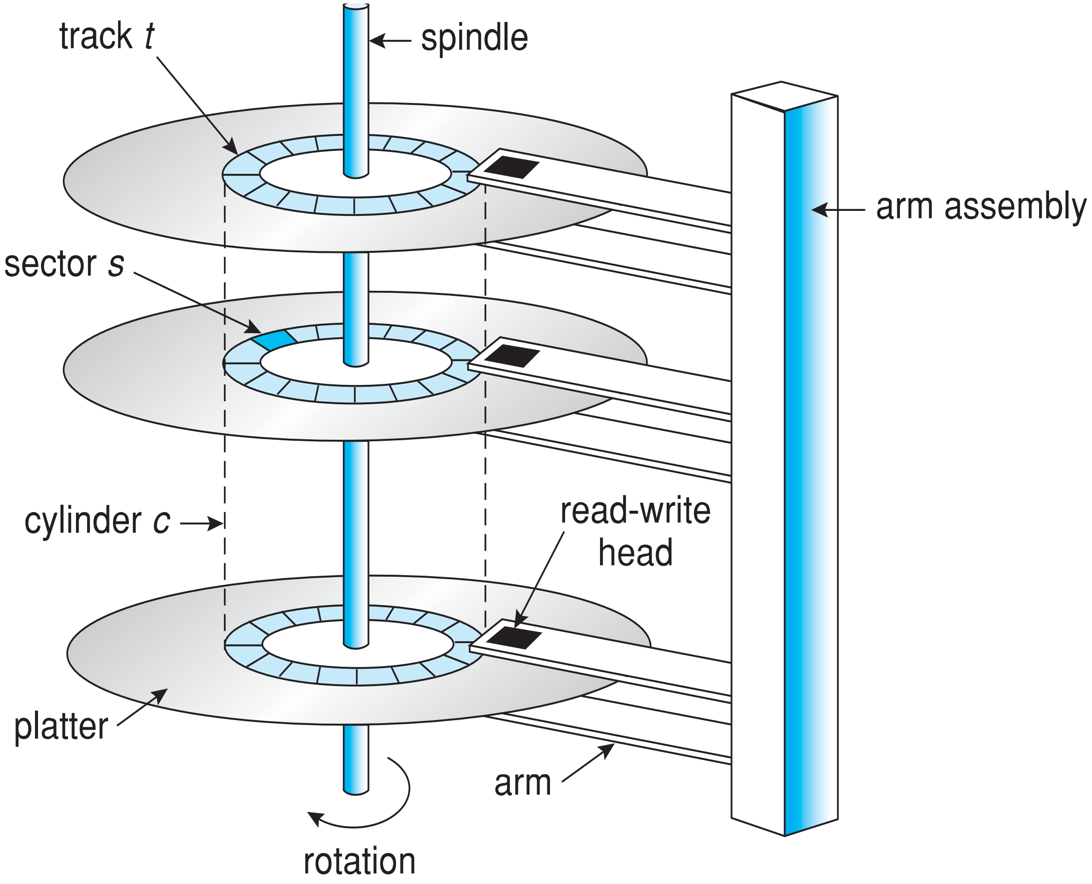


DISK SCHEDULING

Objective: Minimize Seek Time

- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.
- Each request corresponds to an address that contains:
  - Platter Number
  - Track Number
  - Sector Number

Disk Scheduling: What order to service these requests?

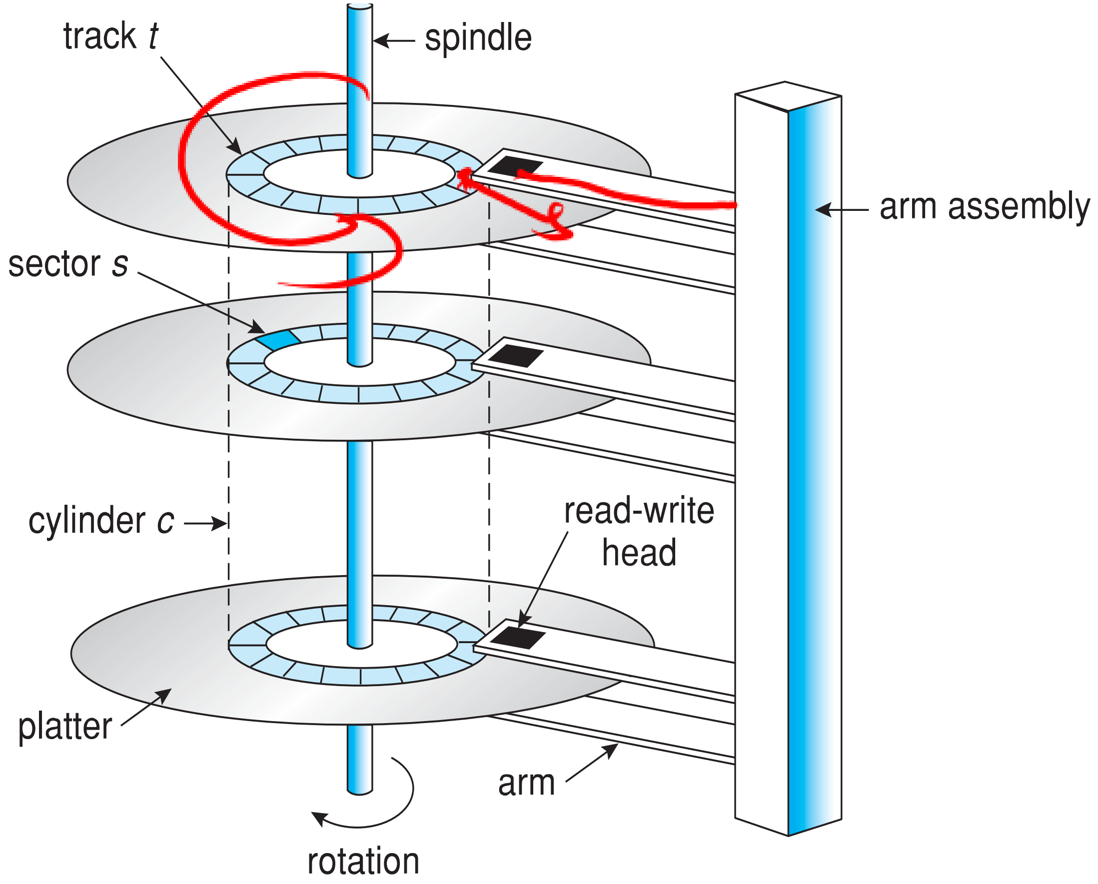


DISK SCHEDULING

Objective: Minimize Seek Time

- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.
- Each request corresponds to an address that contains:
  - Platter Number
  - Track Number
  - Sector Number

Disk Scheduling: What order to service these requests?



Q: What should we look at?

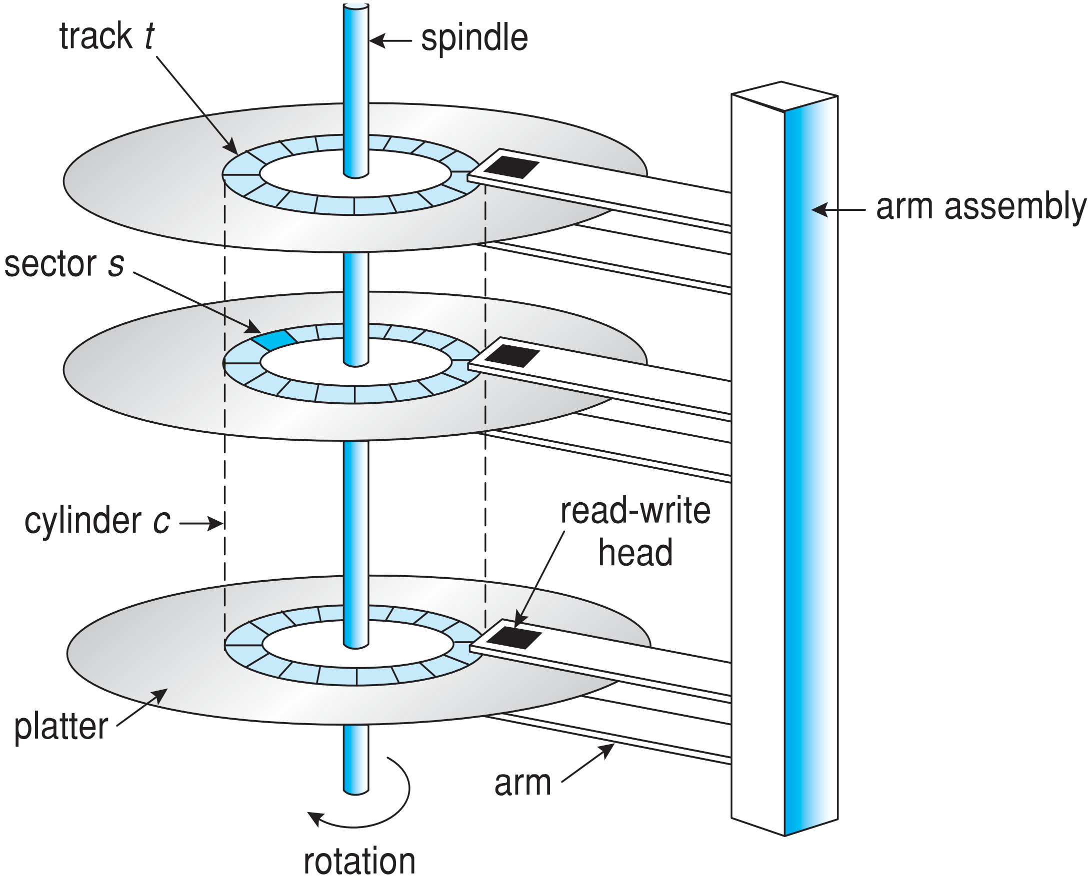
A B
C D

A: Track
B: Platter
C: Sector
D: All

DISK SCHEDULING

Objective: Minimize Seek Time

- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.
- Each request corresponds to an address that contains:
  - Platter Number
  - Track Number
  - Sector Number



Disk Scheduling: What order to service these requests?

Q: What should we look at?

A: Track
B: Platter
C: Sector
D: All

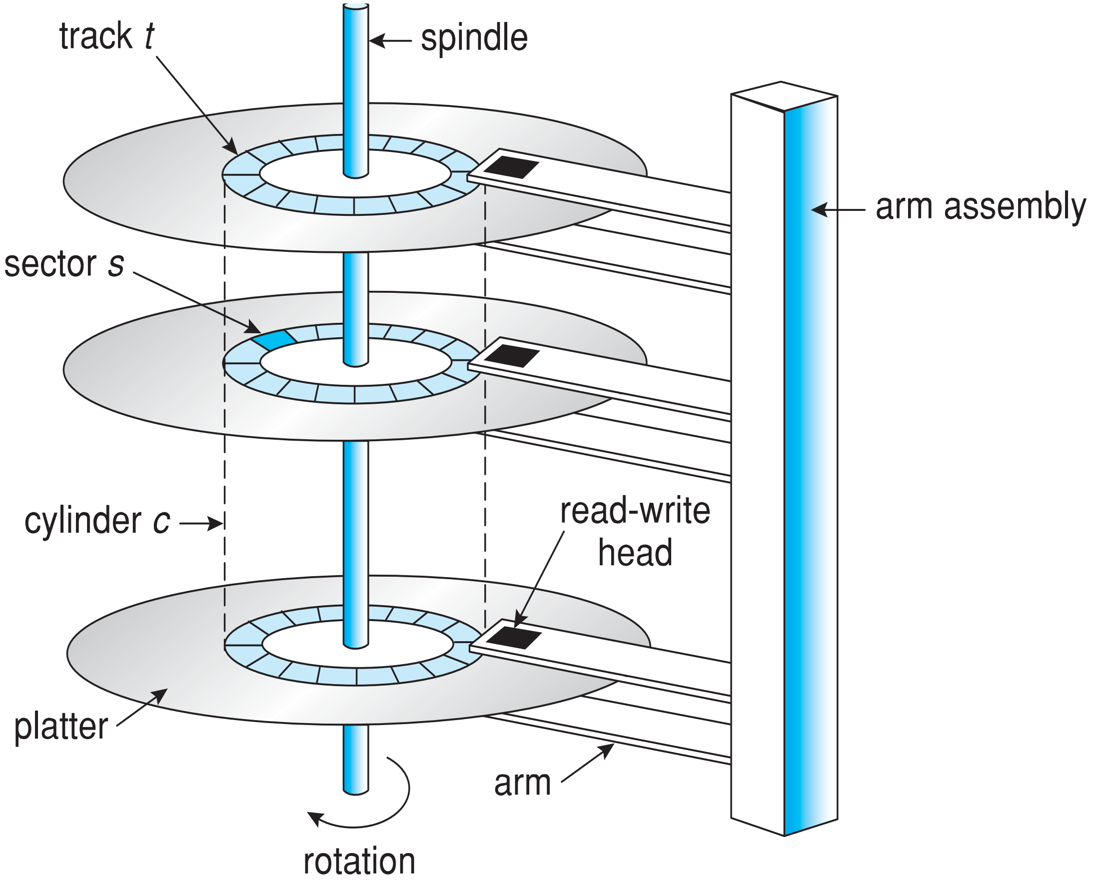
DISK SCHEDULING

Objective: Minimize Seek Time

- I/O requests have arrived.
- Multiple I/O requests have arrived and are pending.
- Each request corresponds to an address that contains:
  - Platter Number
  - Track Number
  - Sector Number

Disk Scheduling: What order to service these requests?

Only the track is relevant to seek time.

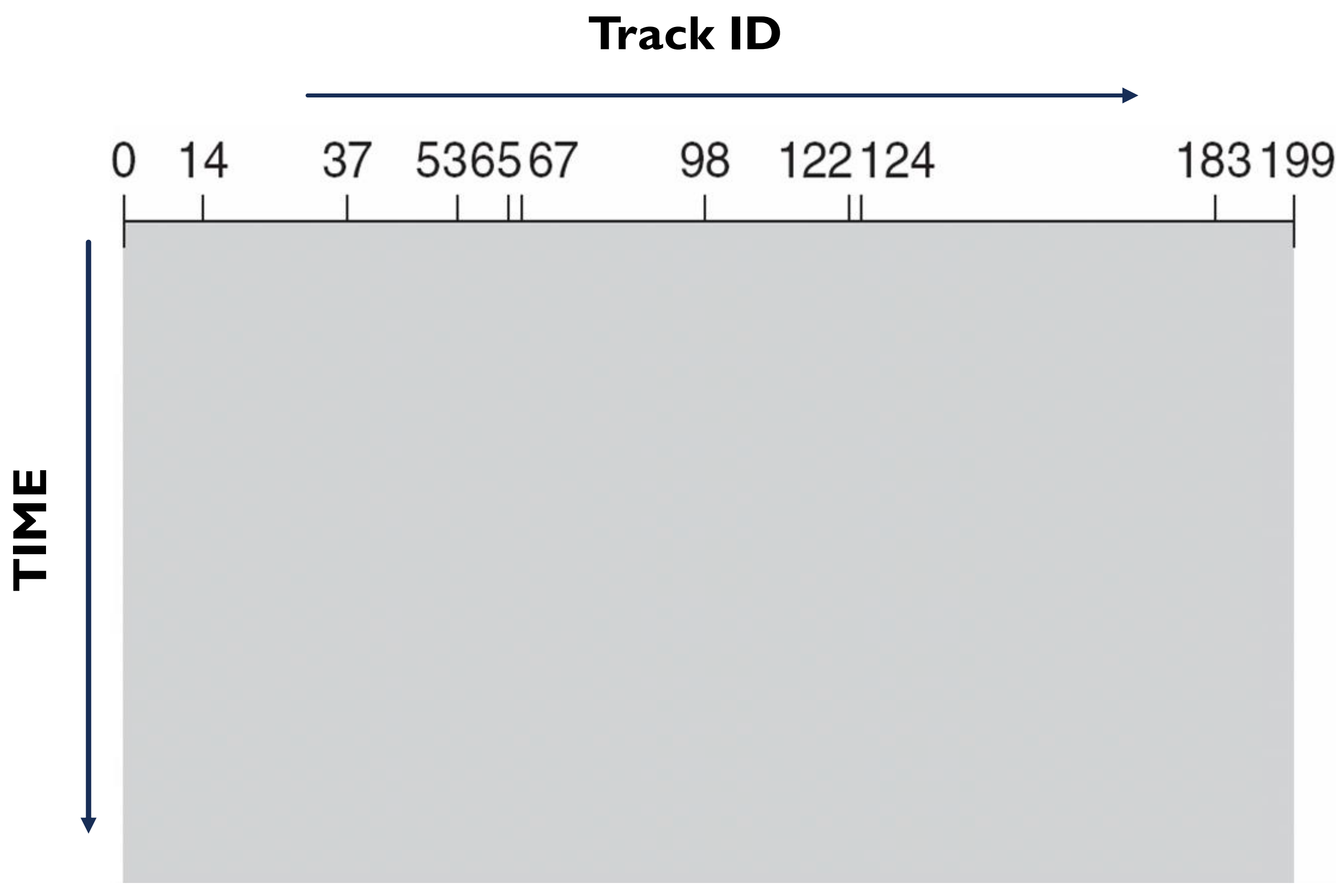


DISK SCHEDULING

- Assume I/O requests for tracks arrived in this order:
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Head is initially positioned at 53.

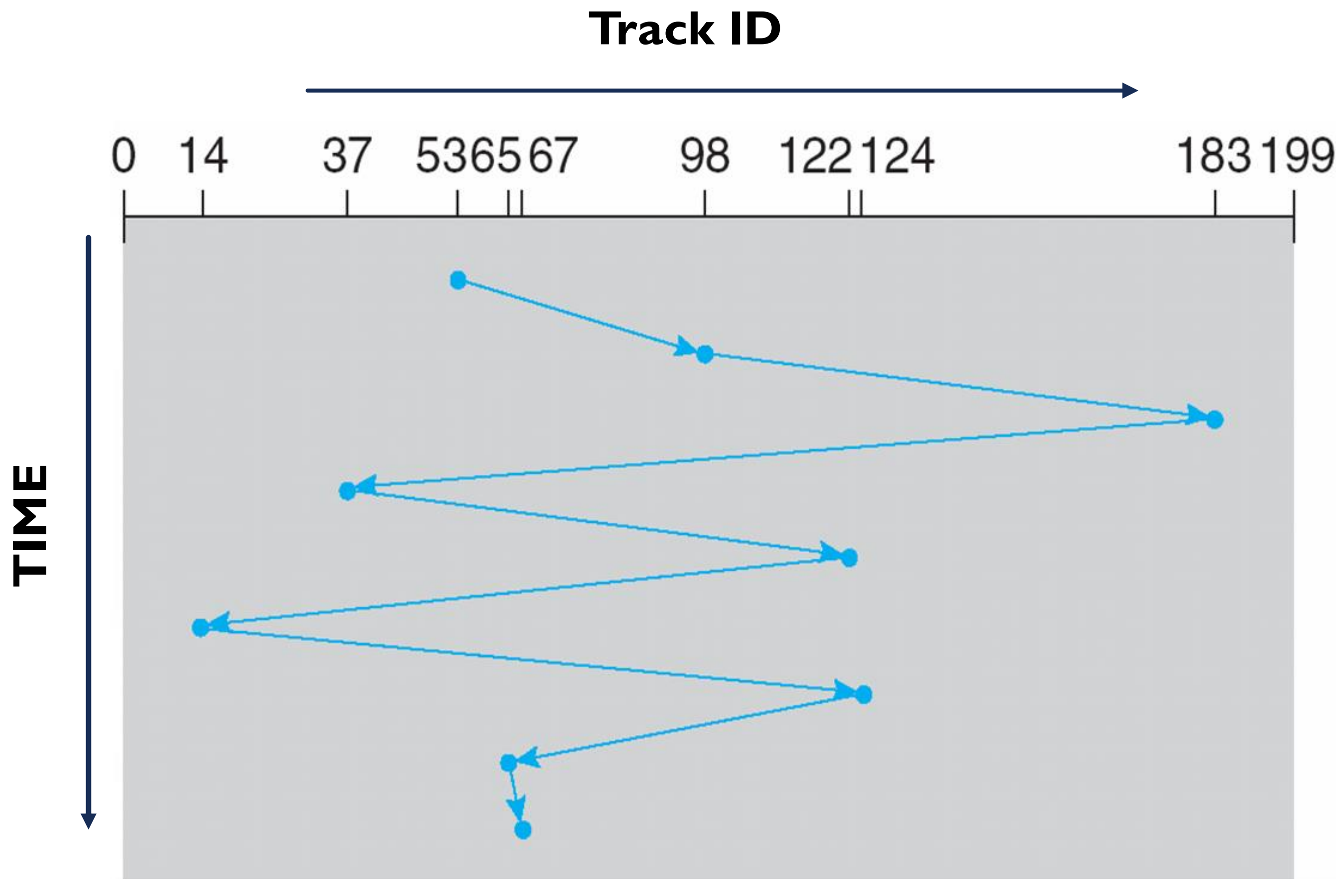
DISK SCHEDULING: FIFO

- Assume I/O requests for tracks arrived in this order:
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Head is initially positioned at 53.



DISK SCHEDULING: FIFO

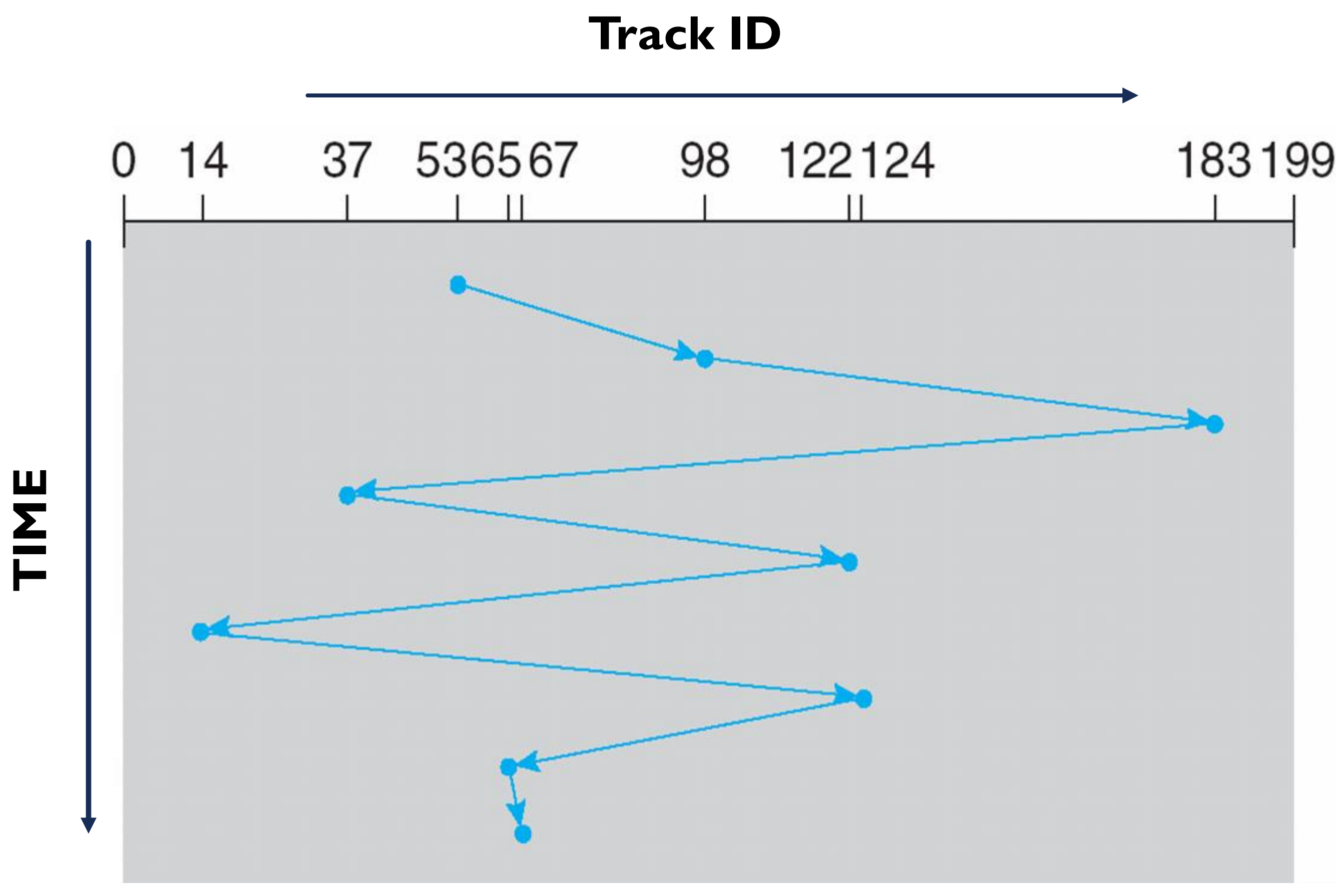
- Assume I/O requests for tracks arrived in this order:
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Head is initially positioned at 53.



DISK SCHEDULING: FIFO

- Assume I/O requests for tracks arrived in this order:
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Head is initially positioned at 53.

Q: Any issues with this algorithm?

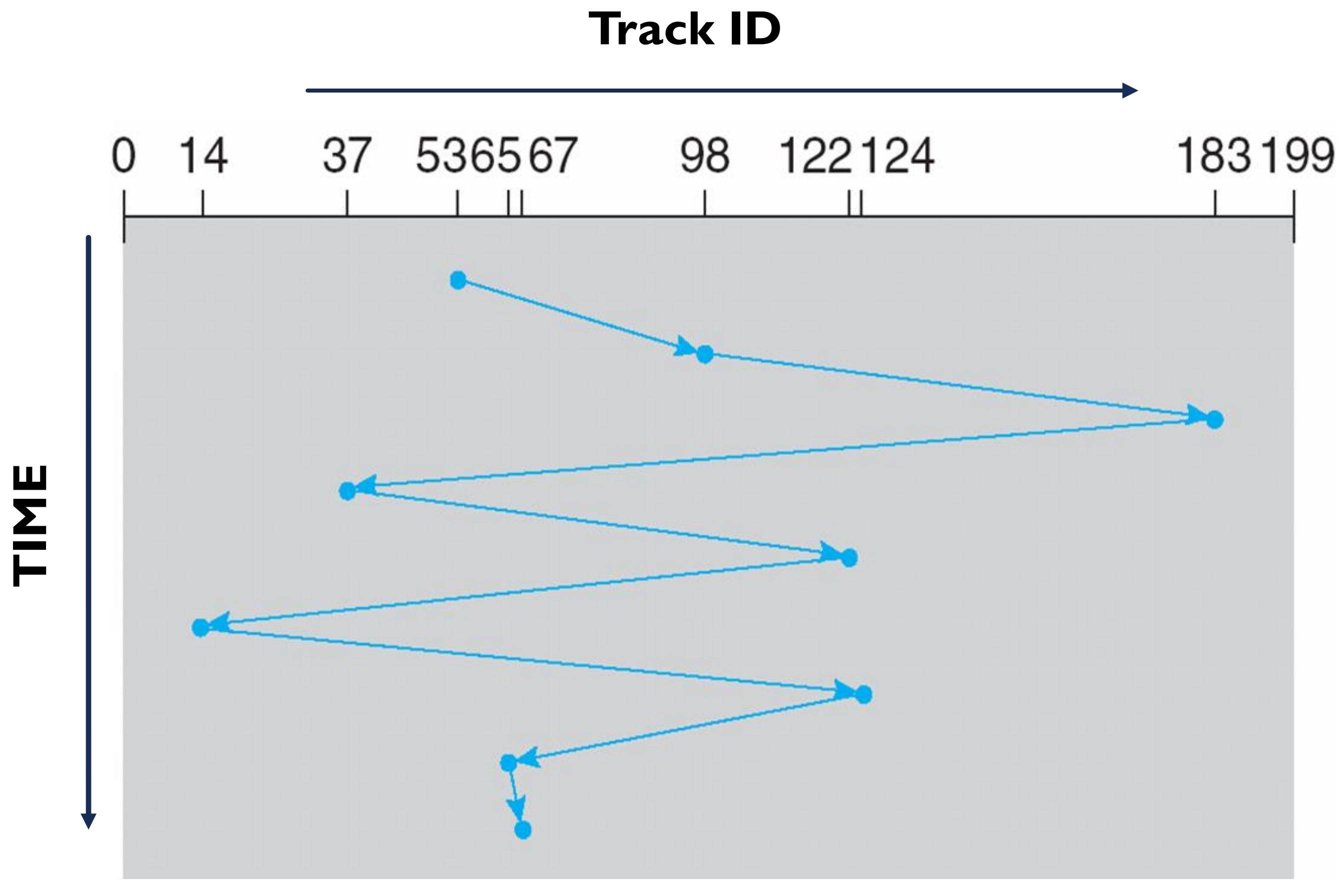


DISK SCHEDULING: FIFO

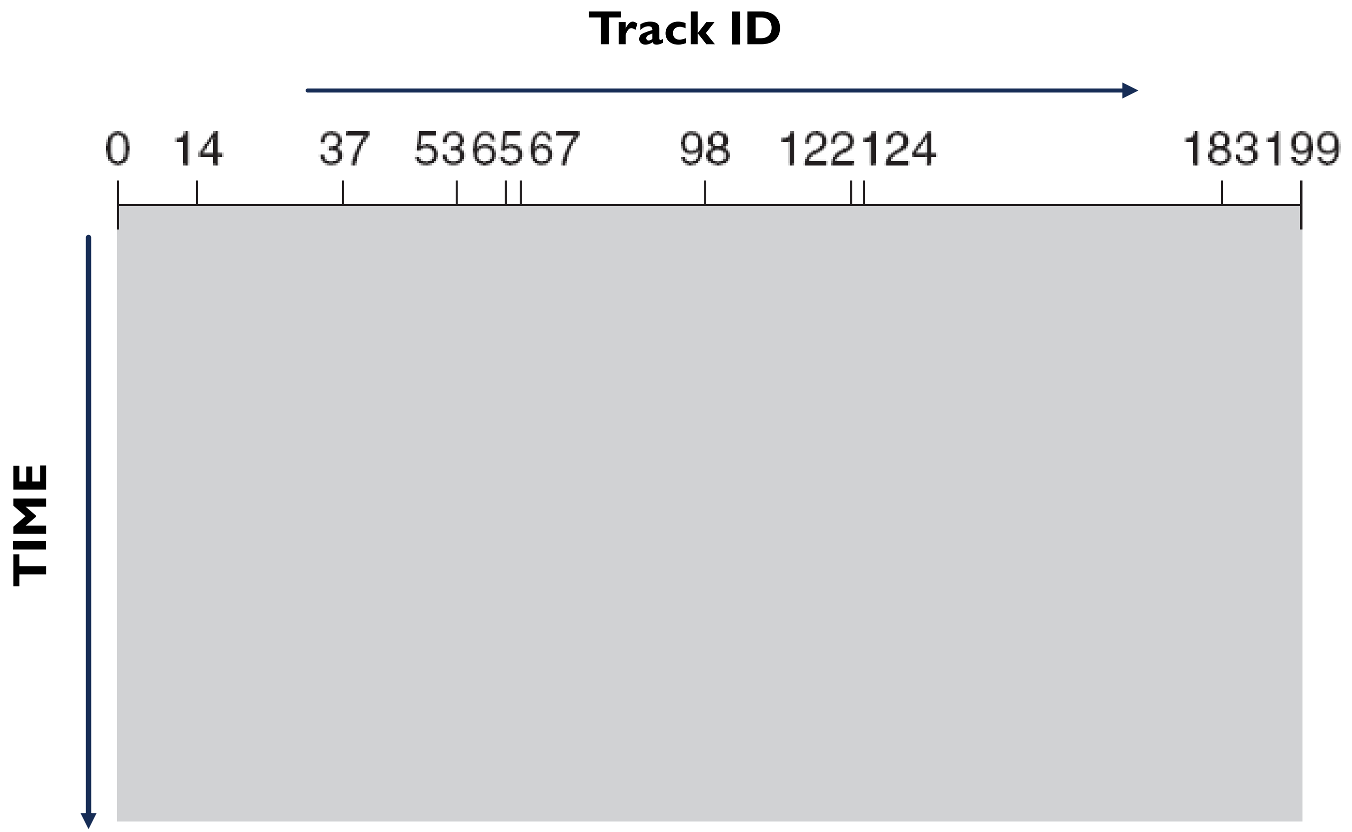
- Assume I/O requests for tracks arrived in this order:
  - 98, 183, 37, 122, 14, 124, 65, 67
  - Head is initially positioned at 53.

Q: Any issues with this algorithm?

Distance travelled by head is too long.
Time is wasted.



SSTF ALGORITHM

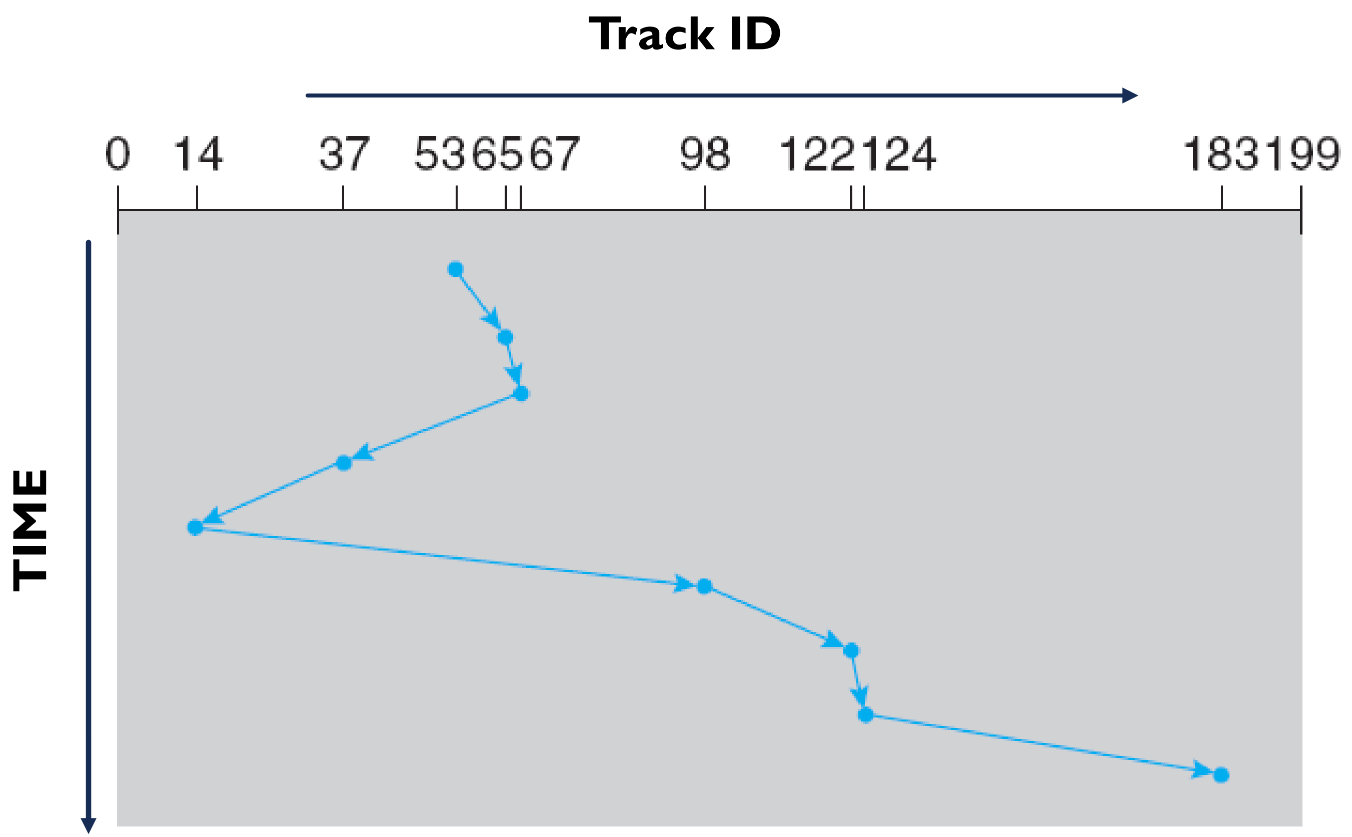


- Shortest Seek Time First
- Simple to implemented and decreases distance traveled by head dramatically compared to FIFO.

Worksheet Q1

SSTF ALGORITHM

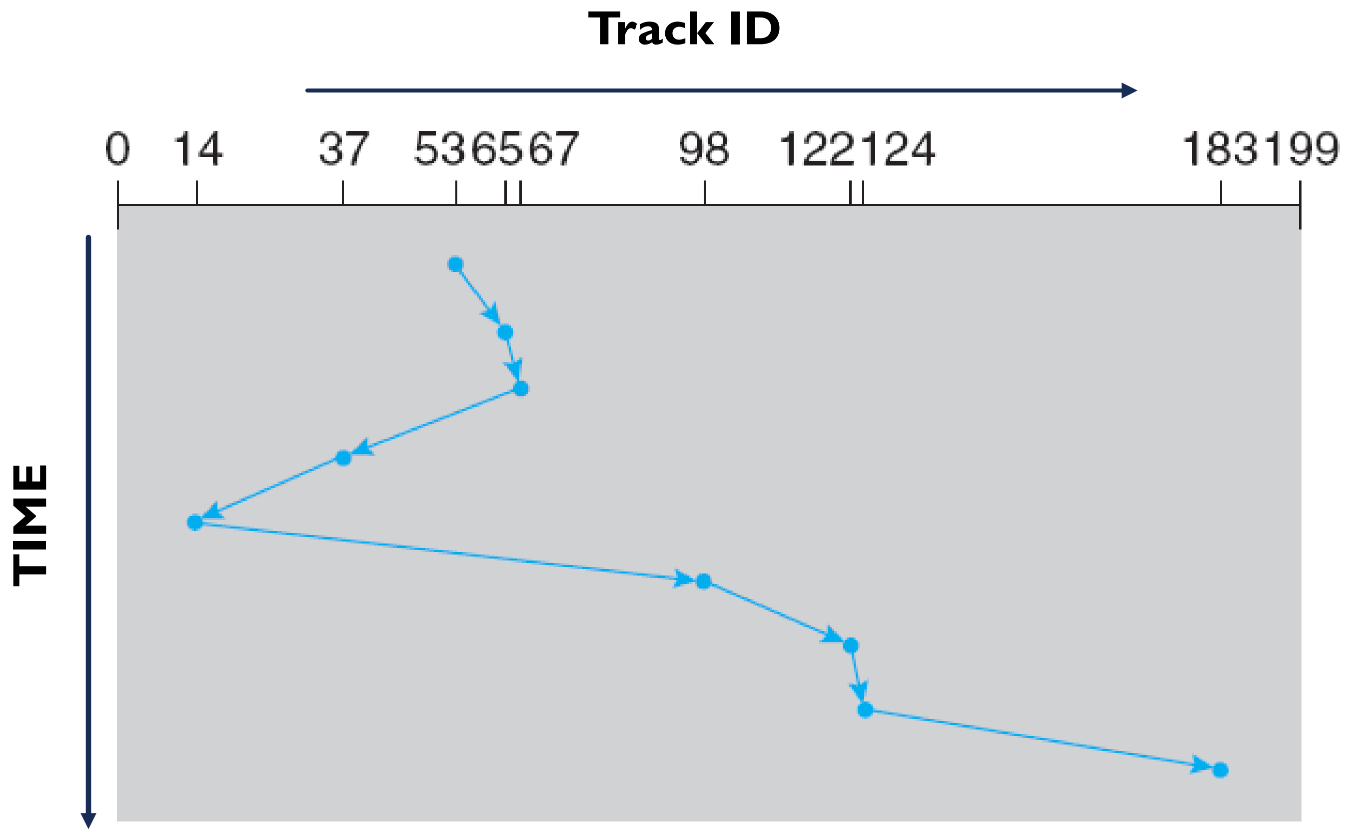
- Shortest Seek Time First
- Simple to implemented and decreases distance traveled by head dramatically compared to FIFO.



SSTF ALGORITHM

- Shortest Seek Time First
- Simple to implemented and decreases distance traveled by head dramatically compared to FIFO.

Worksheet Q2: Possible disadvantages?

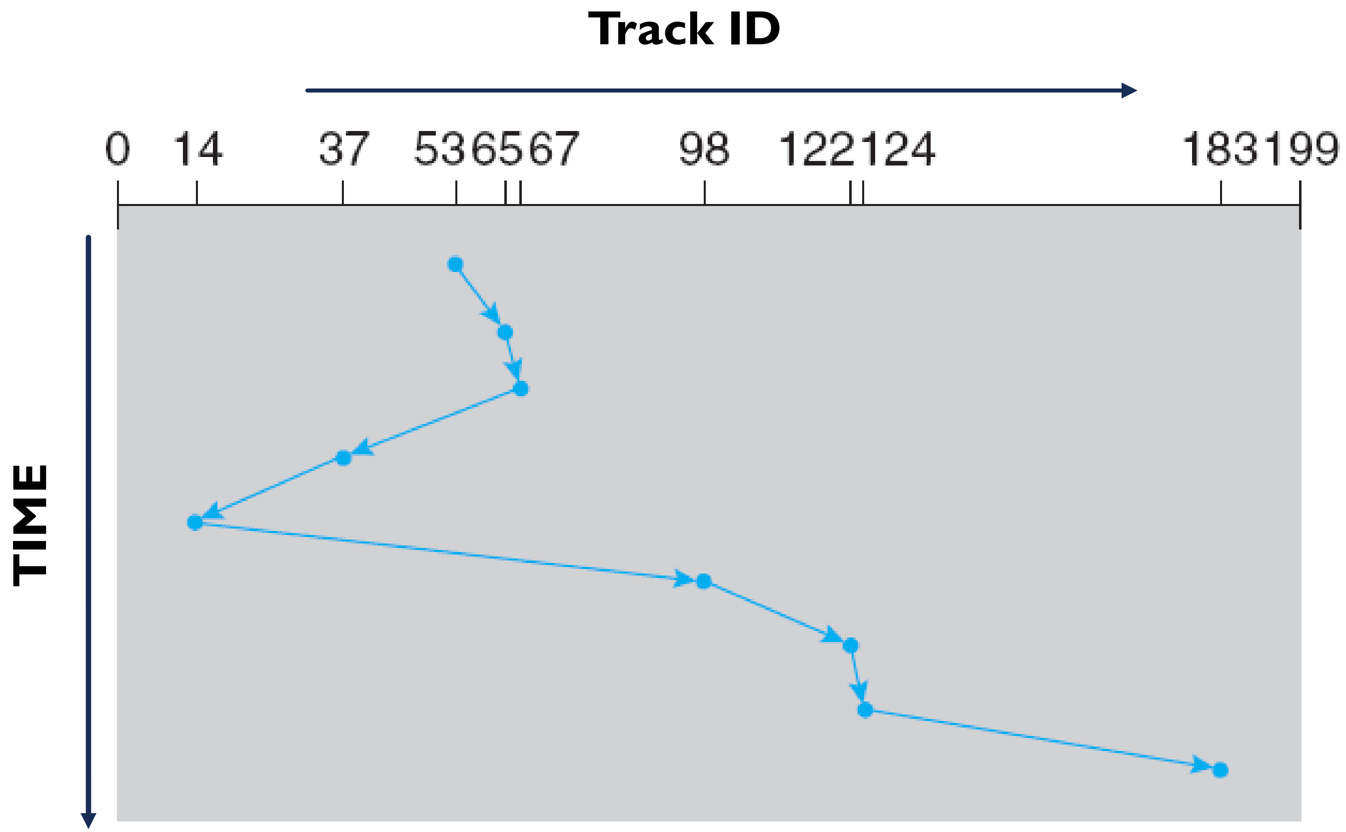


SSTF ALGORITHM

- Shortest Seek Time First
- Simple to implemented and decreases distance traveled by head dramatically compared to FIFO.

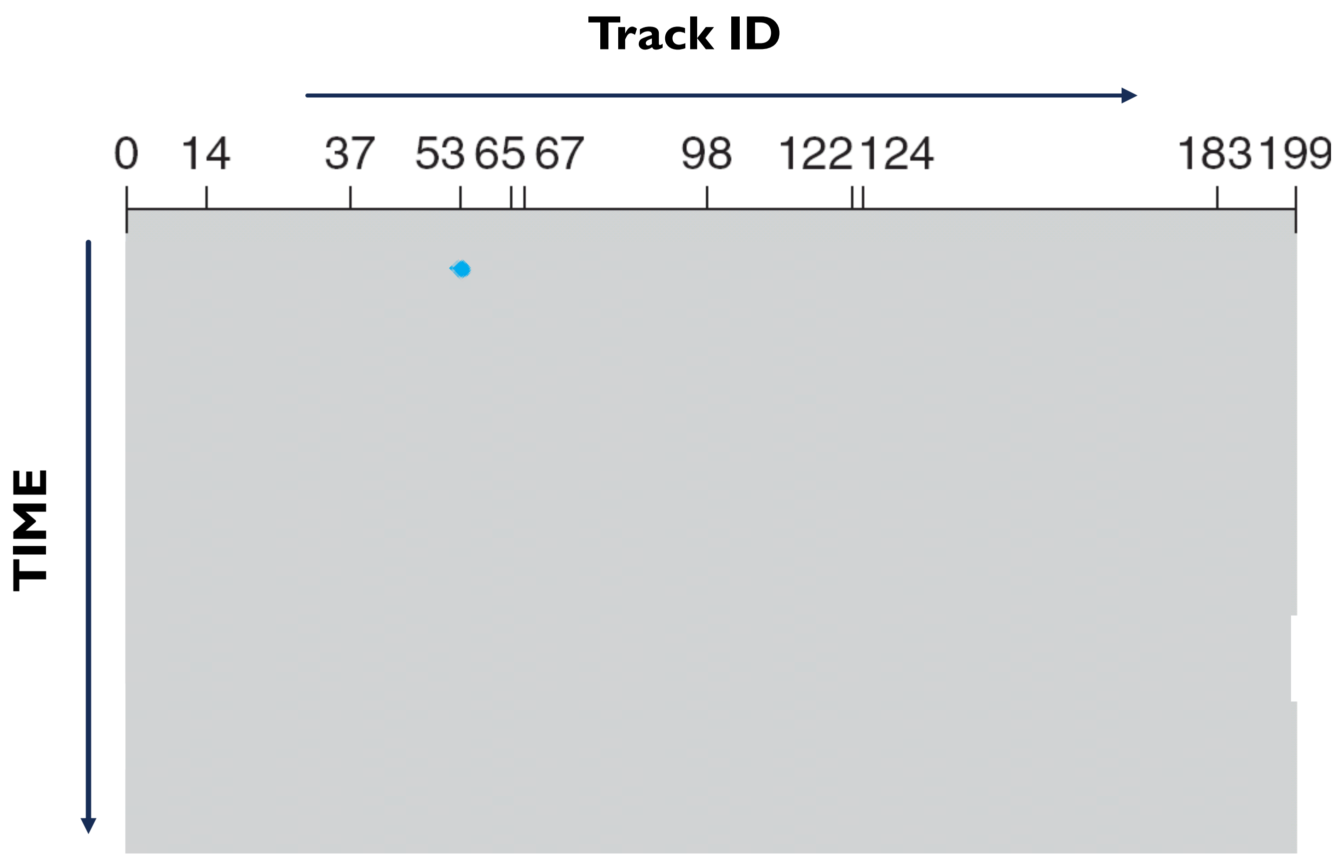
Worksheet Q2: Possible disadvantages?

- Need to calculate distance
- Starvation



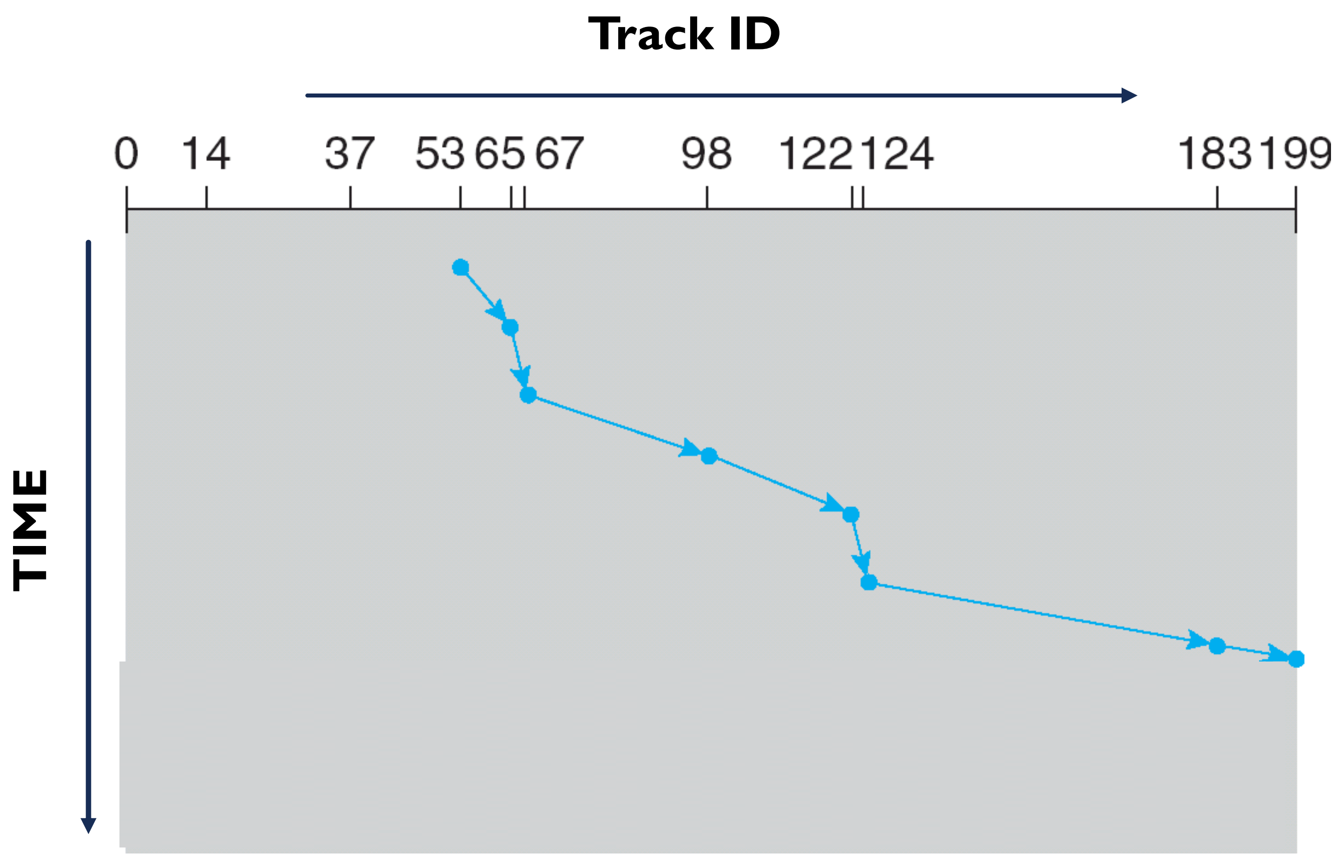
C-SCAN ALGORITHM

- Move one direction, service all requests in that direction then reset to '0'.



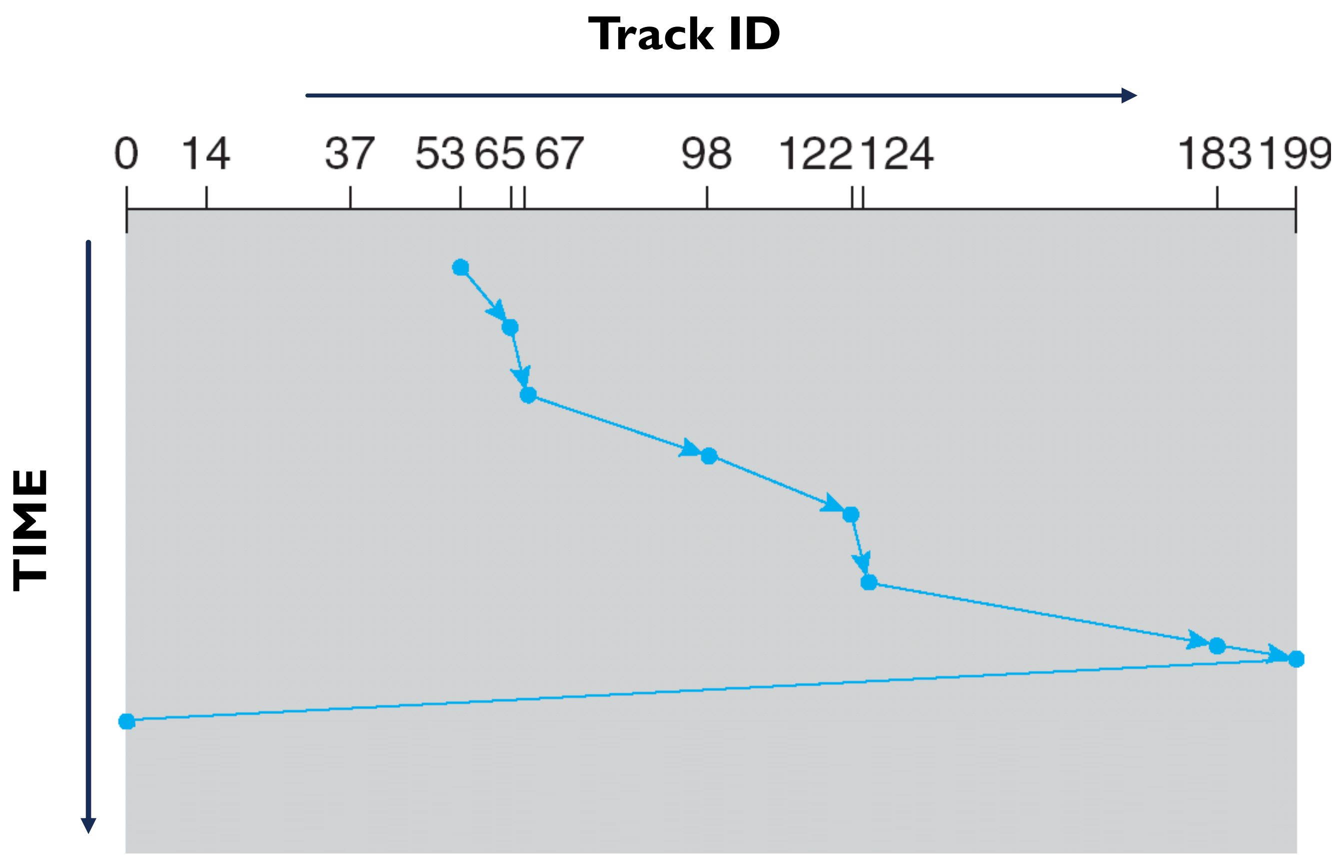
C-SCAN ALGORITHM

- Move one direction, service all requests in that direction then reset to ‘0’.

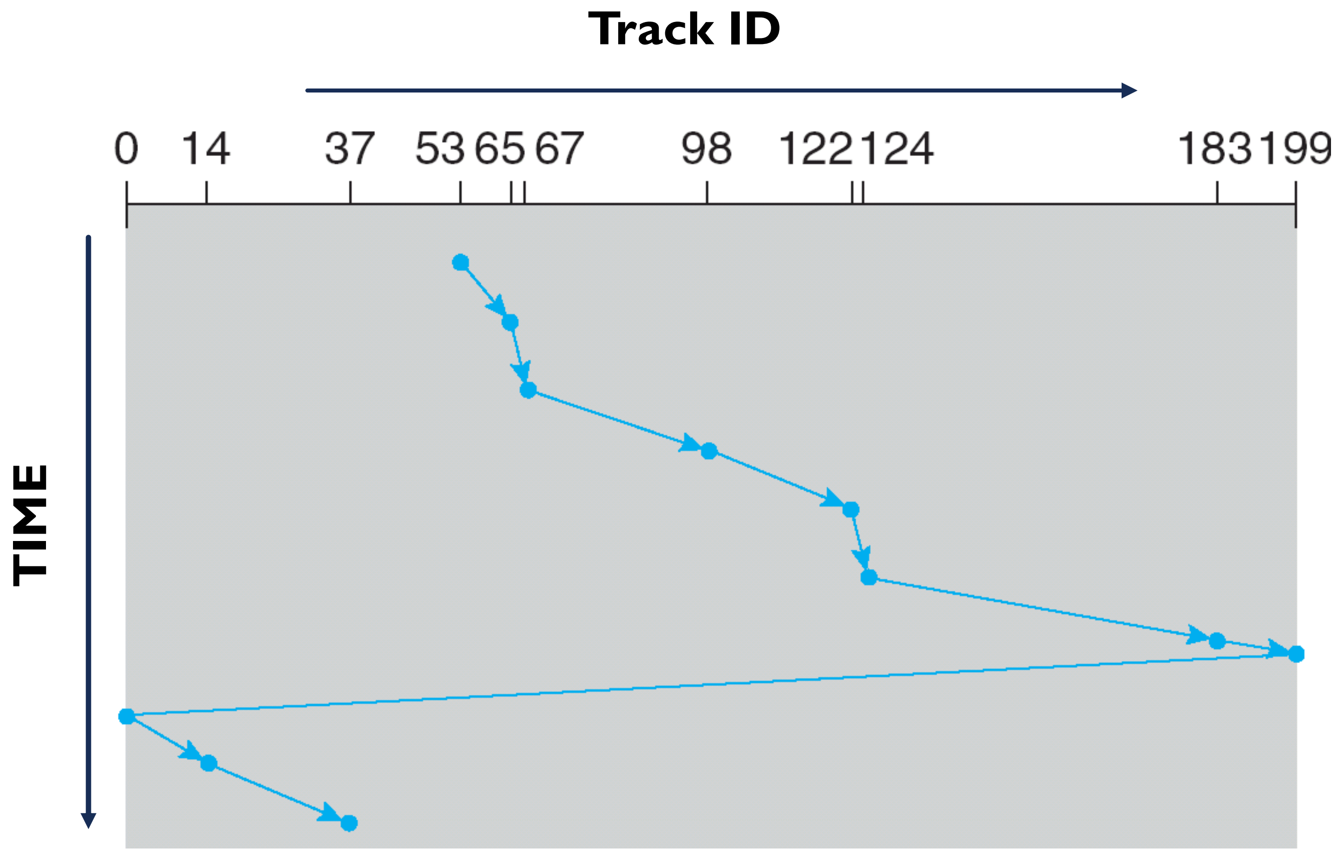


C-SCAN ALGORITHM

- Move one direction, service all requests in that direction then reset to ‘0’.



C-SCAN ALGORITHM

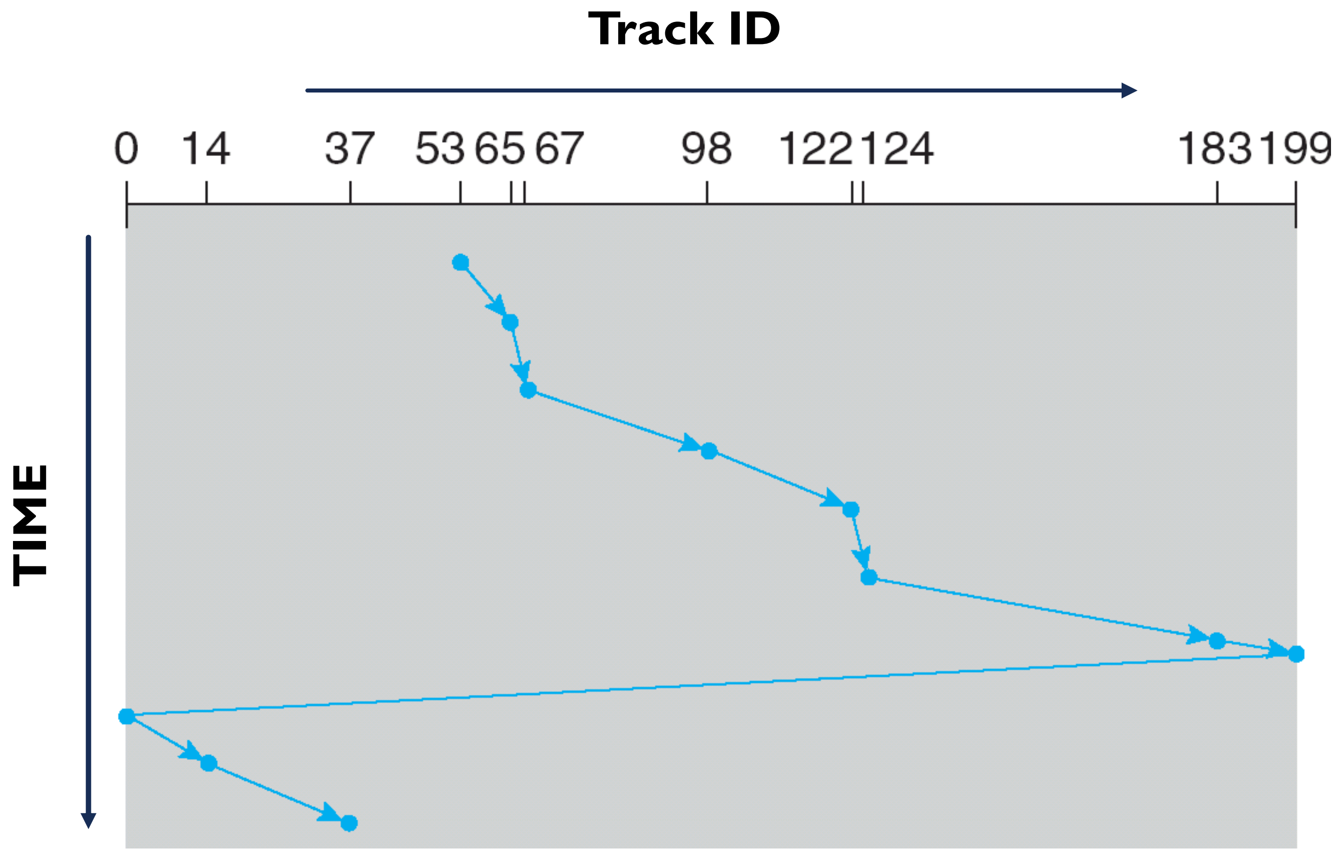


- Move one direction, service all requests in that direction then reset to '0'.

C-SCAN ALGORITHM

- Move one direction, service all requests in that direction then reset to '0'.

Q: Can you spot a deficiency?



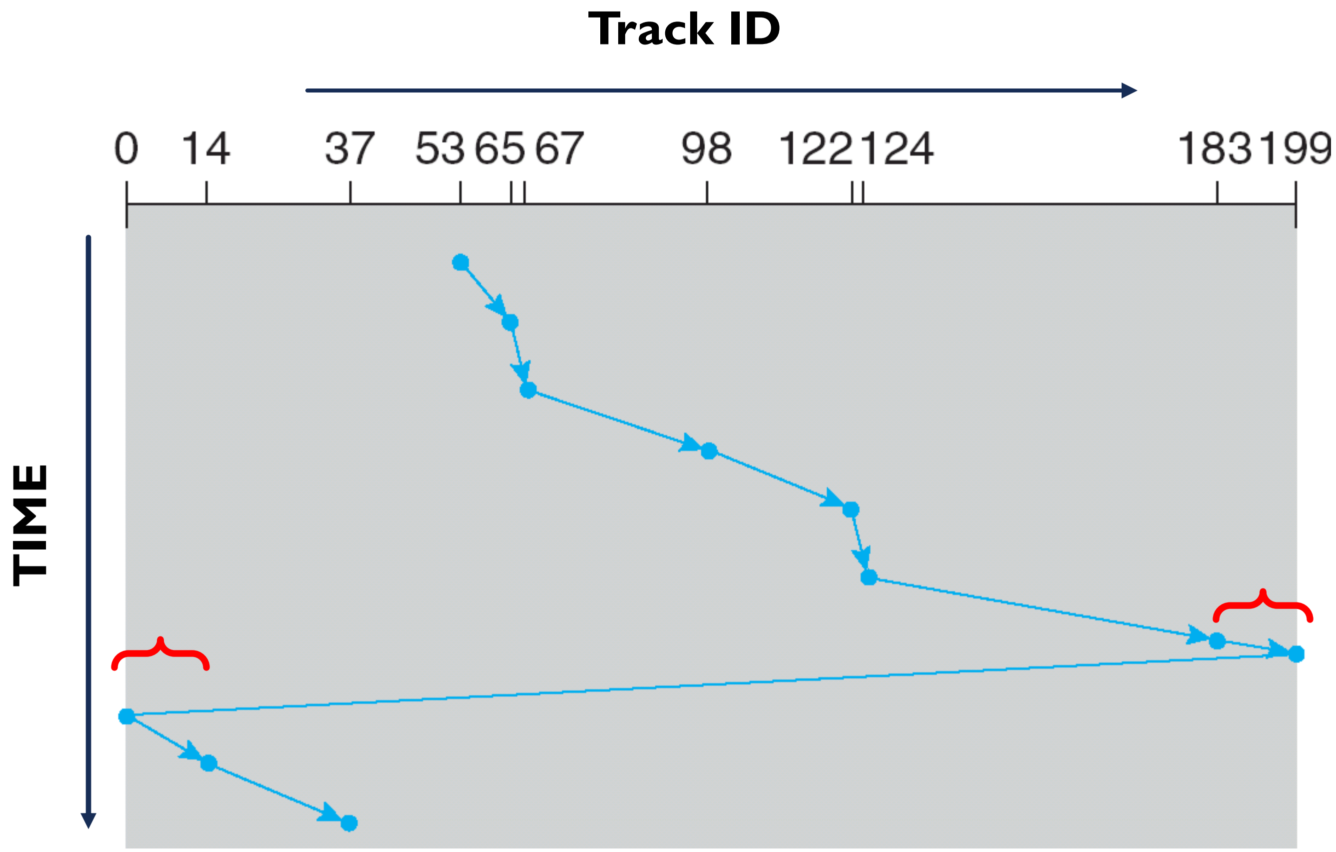
WESTERN
WASHINGTON UNIVERSITY

C-SCAN ALGORITHM

- Move one direction, service all requests in that direction then reset to ‘0’.

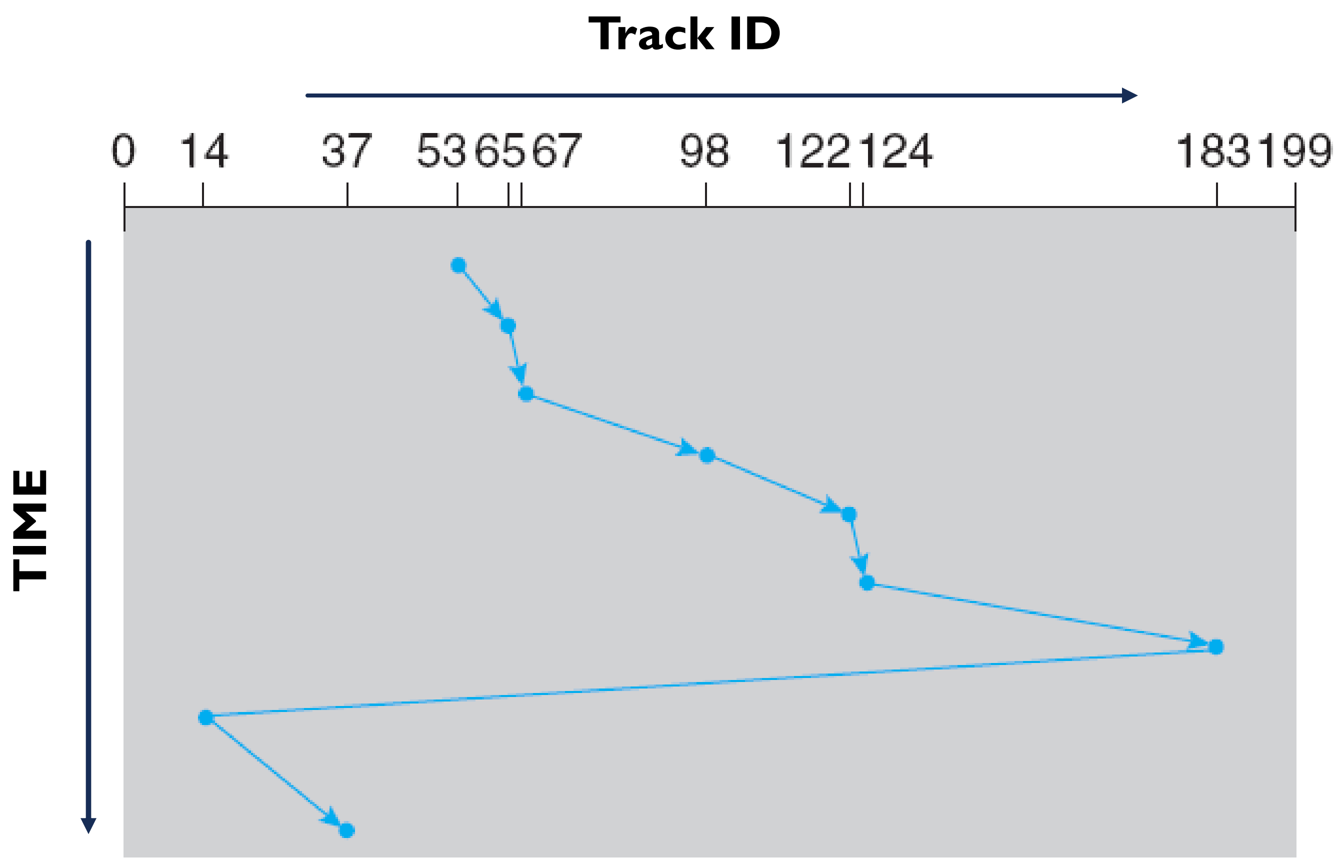
Q: Can you spot a deficiency?

There is really no need for the head to reach the last track or first track …



LOOK ALGORITHM

- Look algorithm: improvement on scan.
- Head only reaches the track of the last request before switching direction.



NON-VOLATILE MEMORY SCHEDULING

- NVM storage have uniform access as there is no mechanical movement.
- Read requests are serviced FCFS.

WESTERN
WASHINGTON UNIVERSITY

NON-VOLATILE MEMORY SCHEDULING

- NVM storage have uniform access as there is no mechanical movement.
- Read requests are serviced FCFS.
- Writes are not uniform and hence adjacent writes are scheduled together.
- Overall much simpler than disk scheduling.

NON-VOLATILE MEMORY SCHEDULING

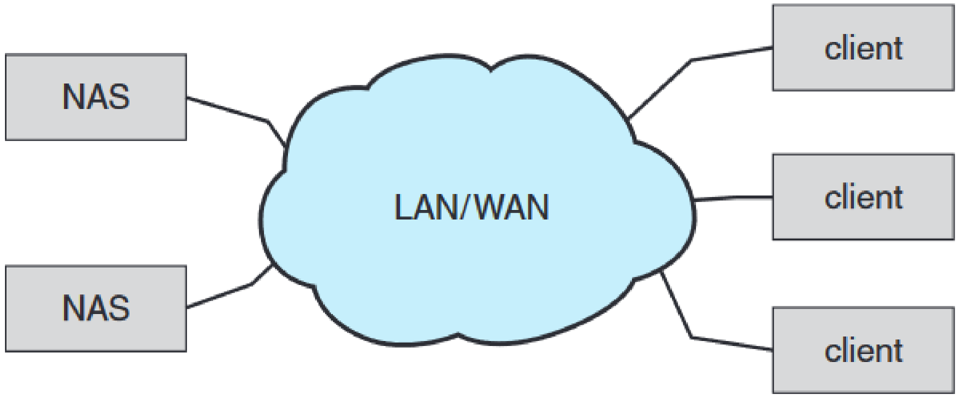
- NVM: susceptible to wear.
- Blocks can support limited number of writes.

NON-VOLATILE MEMORY SCHEDULING

- NVM: susceptible to wear.
- Blocks can support limited number of writes.
- Wear leveling: scheduling that aims to "level" the writes to prolong the lifetime of the device.

NETWORK ATTACHED STORAGE

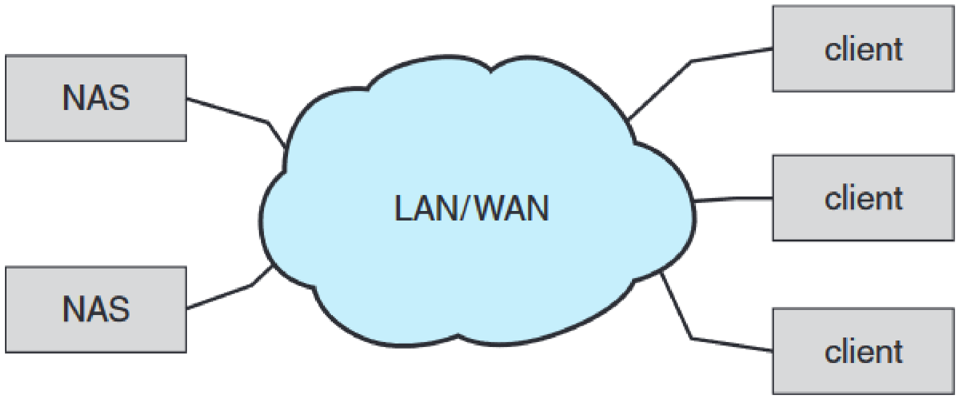
- Storage that is accessed through local LAN/WAN.



NETWORK ATTACHED STORAGE

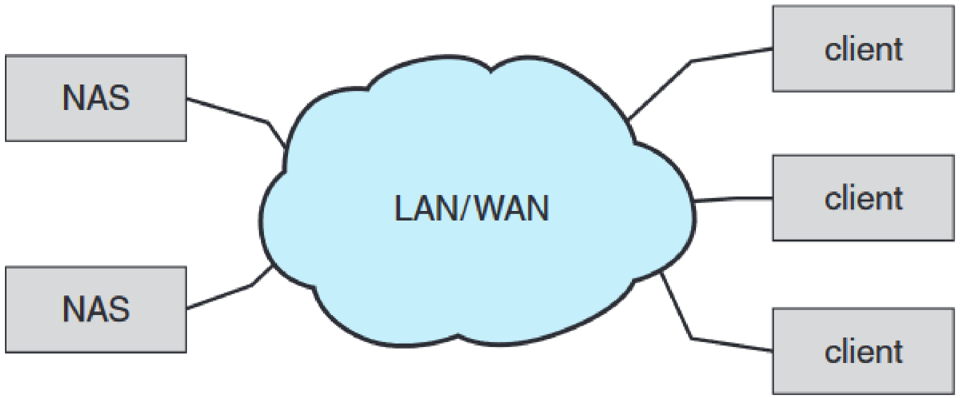
- Storage that is accessed through local LAN/WAN.

Q1 Worksheet: Think of the advantages and disadvantages of network attached storage.



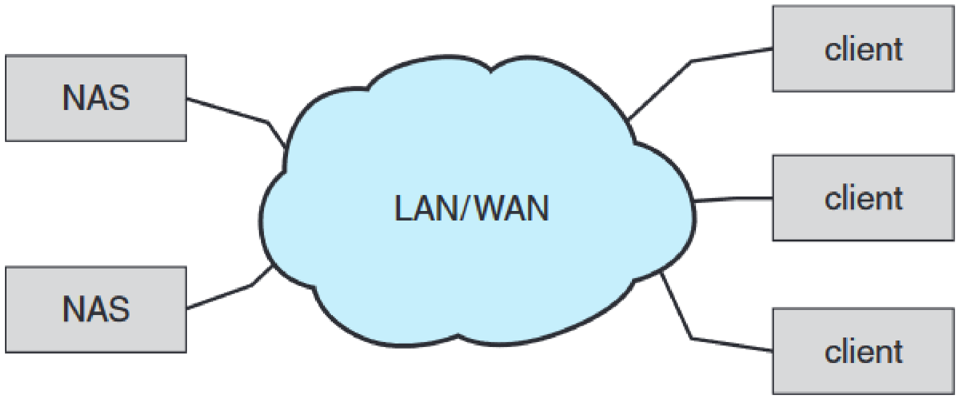
NETWORK ATTACHED STORAGE

- Storage that is accessed through local LAN/WAN.
- Advantages:
  - Storage can be easily shared among many clients.
  - Centralized management
  - Easier to secure.



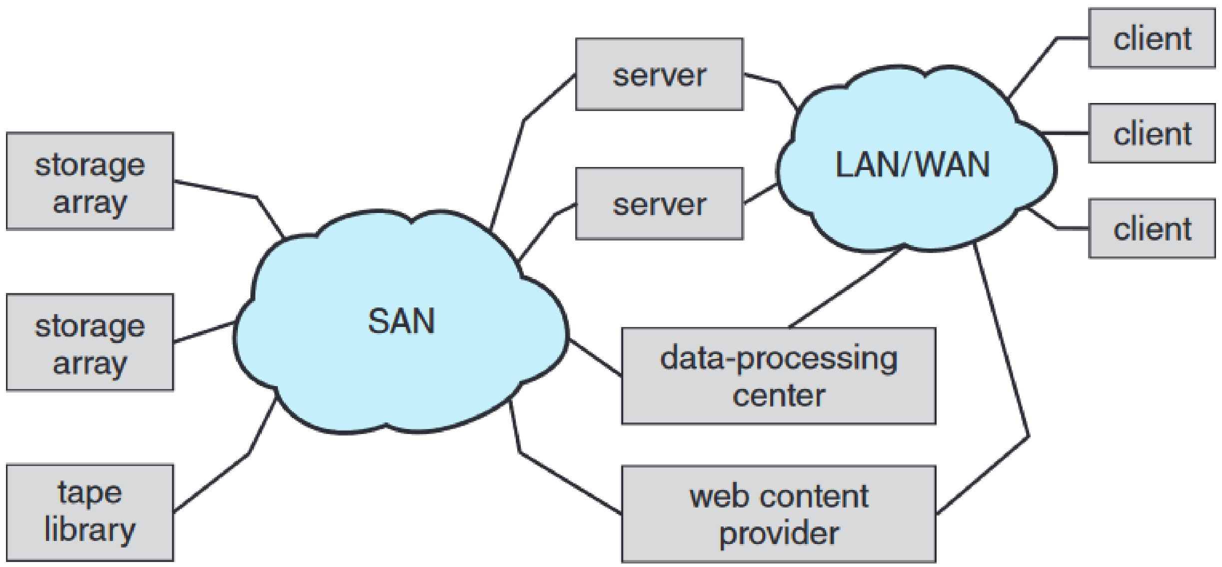
NETWORK ATTACHED STORAGE

- Storage that is accessed through local LAN/WAN.
- Advantages:
  - Storage can be easily shared among many clients.
  - Centralized management
  - Easier to secure.
- Disadvantages:
  - Overload the network with huge volume transfers.
  - Traditional LAN/WAN are not optimized for large volume of data transfers.



STORAGE AREA NETWORKS

- Uses storage protocols rather than networking protocols.
- Clients can’t access storage directly.
- Better bandwidth management but there will always be a bandwidth cost.



STORAGE ARRAYS

- JBOD:
- RAID:

STORAGE ARRAYS

- JBOD: Just a Bunch Of Disks
- RAID: Redundant Array of Inexpensive Disks

RAID

Disks are fairly reliable … the **mean time to failure** of a disk is ~ 100,000 hours (11 years)

Disk storage is cheap … so let's buy lots of them

RAID

Disks are fairly reliable … the **mean time to failure** of a disk is ~ 100,000 hours (11 years)

Disk storage is cheap … so let's buy lots of them

Assume an array of 100 disks

Q: What is the mean time of failure of some disk in the 100 disk array?

RAID

Disks are fairly reliable … the **mean time to failure** of a disk is ~ 100,000 hours (11 years)

Disk storage is cheap … so let's buy lots of them

Assume an array of 100 disks

Q: What is the mean time of failure of some disk in the 100 disk array?

100,000 / 100 = 1,000 hours = ~ 40 days

Every 40 days, you throw out a disk drive and lose its data … this is unacceptable.

RAID

Disks are fairly reliable … the **mean time to failure** of a disk is ~ 100,000 hours (11 years)

Disk storage is cheap … so let's buy lots of them

Assume an array of 100 disks

Q: What is the mean time of failure of some disk in the 100 disk array?

100,000 / 100 = 1,000 hours = ~ 40 days

Every 40 days, you throw out a disk drive and lose its data … this is unacceptable.

Q: What are possible solutions?

RAID

Redundancy

- Mirroring: write data to duplicate discs.

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years
MTTR: Mean Time To Repair: 10 hours

Assume that you have 2 independent disk …

RAID

Redundancy

- Mirroring: write data to duplicate discs.

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years

MTTR: Mean Time To Repair: 10 hours

Assume that you have 2 independent disk …

Q2: What is the Mean Time To Data Loss (MTTDL)?

RAID

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years
MTTR : Mean Time To Repair: 10 hours

Assume that you have 2 independent disk …

Q2: What is the Mean Time To Data Loss (MTTDL)?

Redundancy

- Mirroring: write data to duplicate discs.

$$MTTDL = \frac{(MTBF^2)}{N(N-1) \times MTTR}$$

Where N is the number of disks.

RAID

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years

MTTR : Mean Time To Repair: 10 hours

Redundancy

- Mirroring: write data to duplicate discs.

Assume that you have 2 independent disk …

Q2: What is the Mean Time To Data Loss (MTTDL)?

$$MTTDL = \frac{(MTBF^2)}{N(N-1) \times MTTR}$$

Where N is the number of disks.

Worksheet Q2

RAID

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years
MTTR : Mean Time To Repair: 10 hours

Redundancy

- Mirroring: write data to duplicate discs.

Assume that you have 2 independent disk …

2 is mirroring level

Q: What is the Mean Time To Data Loss (MTTDL)?

$$MTTDL = \frac{(MTBF^2)}{N(N-1) \times MTTR}$$

Where N is the number of disks.

Worksheet Q2

RAID

Redundancy

- Mirroring: write data to duplicate discs.

$$MTTDL = \frac{(MTBF^2)}{N(N-1) \times MTTR}$$

Where N is the number of disks.

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years

MTTR : Mean Time To Repair: 10 hours

Assume that you have 2 independent disk …

Q: What is the Mean Time To Data Loss (MTTDL)?

$$MTTDL = \frac{(100000^2)}{2(2-1) \times 10} = \frac{10^{10}}{20} = 5 \times 10^8 \sim 57\mathrm{K\ years}$$

RAID

Redundancy

- Mirroring: write data to duplicate discs.

$$MTTDL = \frac{(MTBF^2)}{N(N-1) \times MTTR}$$

Where N is the number of disks.

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years
MTTR : Mean Time To Repair: 10 hours

Assume that you have 2 independent disk …

Q: What is the Mean Time To Data Loss (MTTDL)?

$$MTTDL = \frac{(100000^2)}{2(2-1) \times 10} = \frac{10^{10}}{20} = 5 \times 10^8 \sim 57\text{K years}$$

MTTDL went from 11 years to 57K years using one level of mirroring.

RAID

For a single disk

MTBF: Mean Time Between Failures: 100,000 hours ~ 11 years

MTTR : Mean Time To Repair: 10 hours

Redundancy

- Mirroring: write data to duplicate discs.

Assume that you have 2 independent disk …

Q: What is the Mean Time To Data Loss (MTTDL)?

$$MTTDL = \frac{(MTBF^{L})}{N(N-1) \times MTTR}$$

Where N is the number of disks.

$$MTTDL = \frac{(100000^{2})}{2(2-1) \times 10} = \frac{10^{10}}{20} = 5 \times 10^{8} \sim 57\text{K years}$$

MTTDL went from 11 years to 57K years using one level of mirroring.

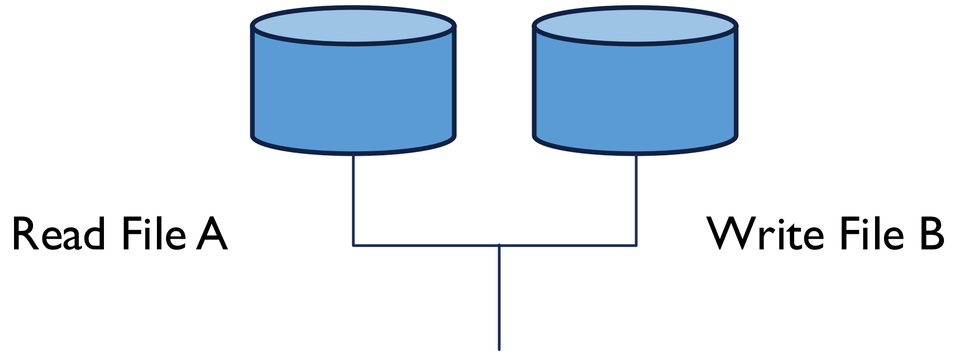
RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

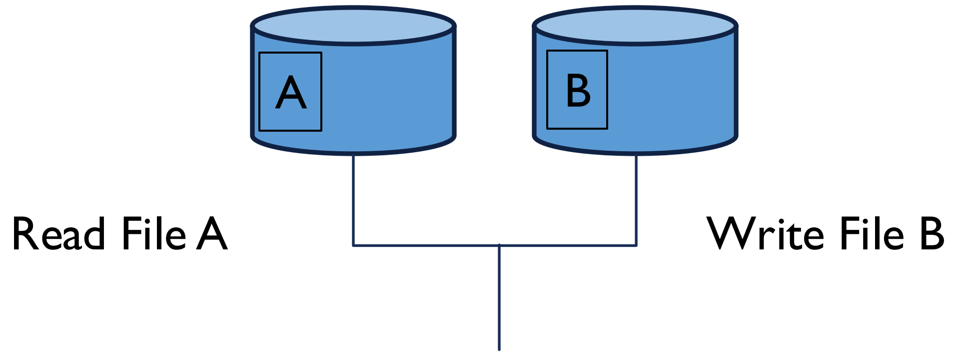
Consider two independent disks with no mirroring …
Each hold its own data.



RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.

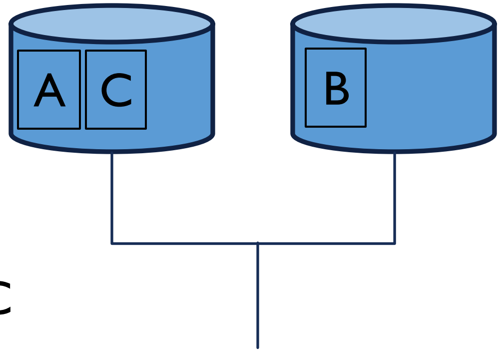


Increased performance when files accessed are on different disk.

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



Read File A

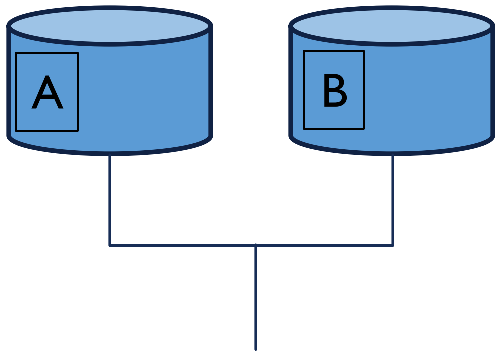
Write File C

What if the files being accessed are on the same disk?

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



Read File A

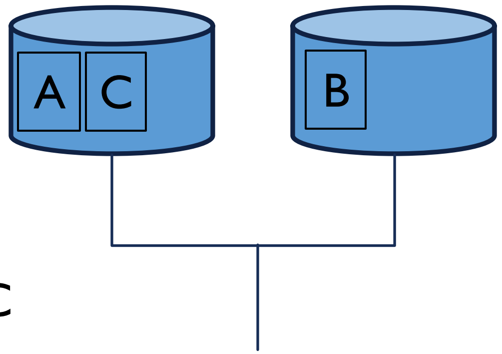
What if the files being accessed are on the same disk? No performance gain …

What if there is only one file I/O operation? No performance gain …

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



Read File A

Write File C

Q: What can we do to enhance performance for every I/O, even single file access?

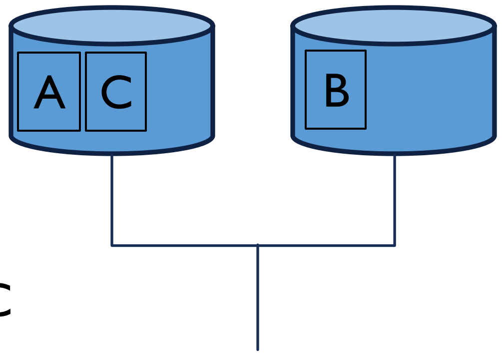
What if the files being accessed are on the same disk? No performance gain …

What if there is only one file I/O operation? No performance gain …

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



Read File A

Write File C

Q: What can we do to enhance performance for every I/O, even single file access?

Data striping: Split each file and all data among all Disks.

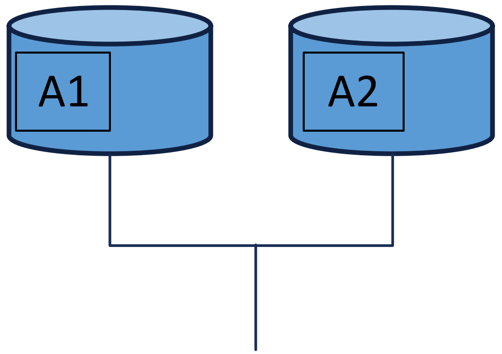
What if the files being accessed are on the same disk? No performance gain …

What if there is only one file I/O operation? No performance gain …

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



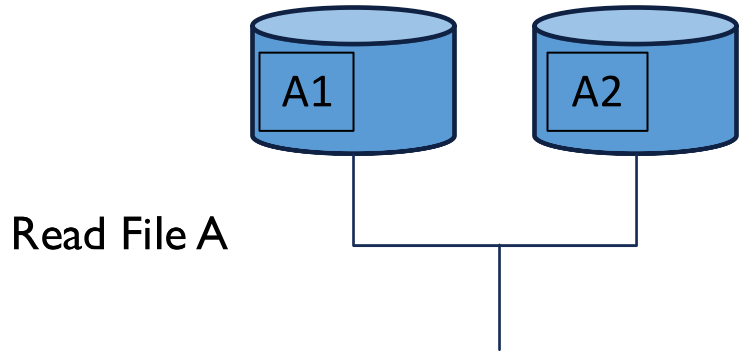
Q: What can we do to enhance performance for every I/O, even single file access?

Data striping: Split each file and all data among all Disks.

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.



Q: What can we do to enhance performance for every I/O, even single file access?

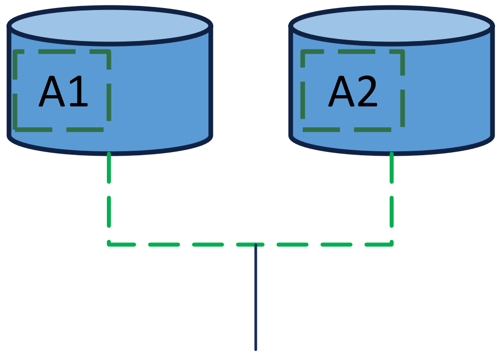
Data striping: Split each file and all data among all Disks.

RAID: PERFORMANCE

- Can we use redundant disks to increase I/O Performance?

Consider two independent disks with no mirroring …
Each hold its own data.

Read File A



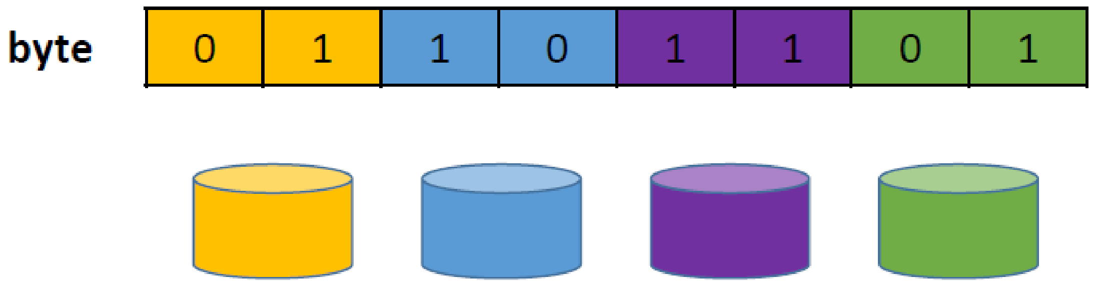
Even single file access now can be performed at twice the speed.

Q: What can we do to enhance performance for every I/O, even single file access?

Data striping: Split each file and all data among all Disks.

DATA STRIPING

- Data striping: divide all data across all disks.
- Data striping can be performed at Bit-level, Byte-level, Block level …

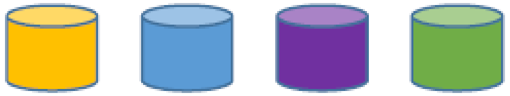


This is 'bit-level' splitting. The 'bits' of each byte are split across all disks.

RAID

Striping with no mirroring

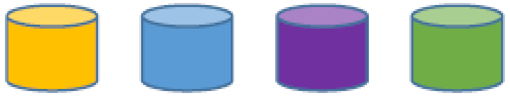
RAID 0



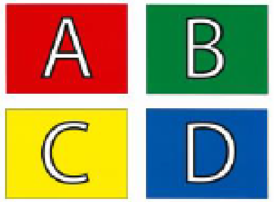
RAID

Striping with no mirroring

RAID 0



Q: RAID 0 …

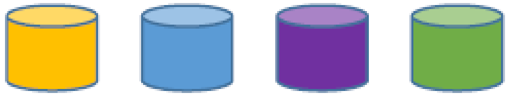


A : Improves Reliability Only
B : Improves Reliability and Performance
C : Improves Performance Only
D : Improves Neither

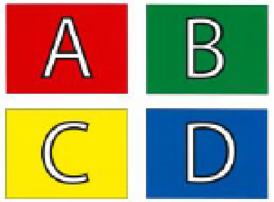
RAID

Striping with no mirroring

RAID 0



Q: RAID 0 …



A : Improves Reliability Only

B : Improves Reliability and Performance

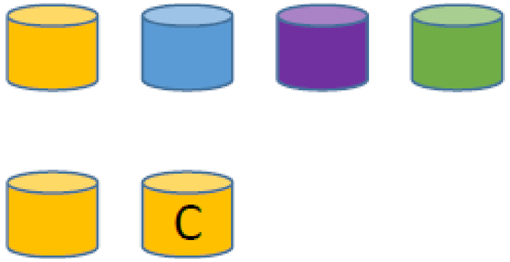
C : Improves Performance Only

D : Improves Neither

RAID

RAID 0

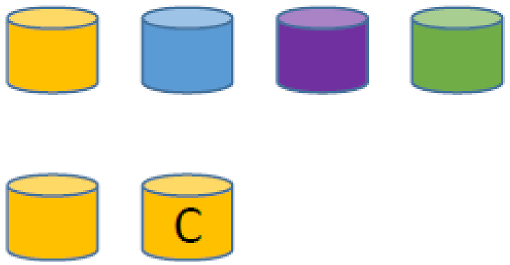
Mirroring Only RAID 1



RAID

RAID 0

Mirroring Only RAID 1

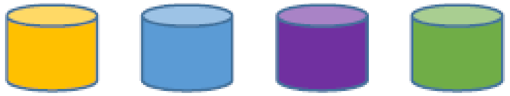


Q: RAID 1 …

A : Improves Reliability Only
B : Improves Reliability and Performance
C : Improves Performance Only
D : Improves Neither

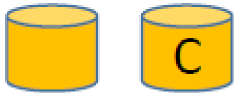
RAID

RAID 0



Mirroring Only

RAID 1



Q: RAID 1 ...

A : Improves Reliability Only

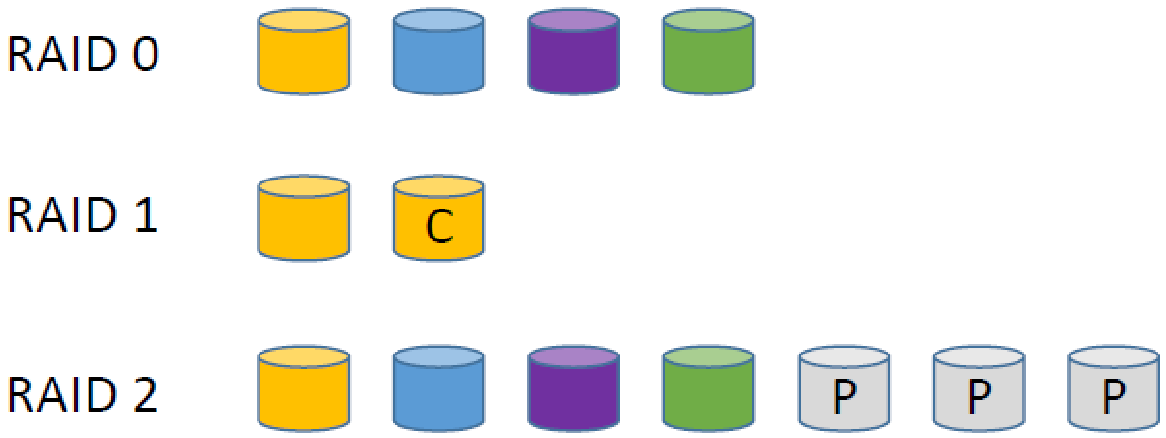
B : Improves Reliability and Performance

C : Improves Performance Only

D : Improves Neither

RAID

Striping w/ parity



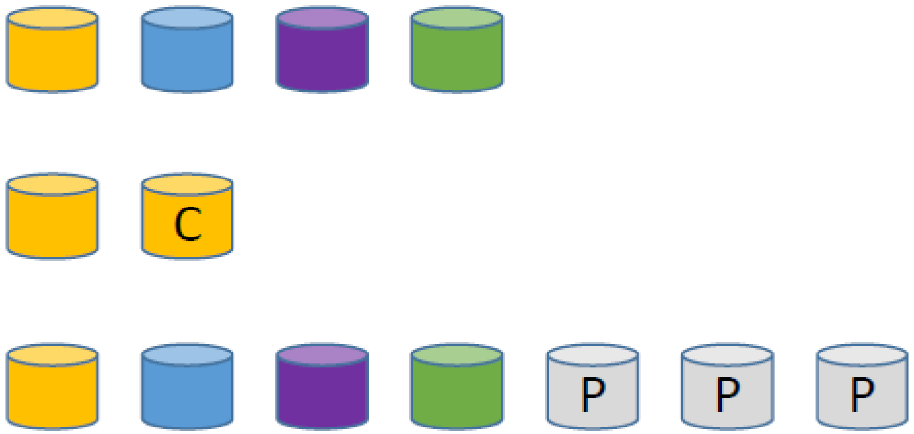
RAID

RAID 0

RAID 1

Striping w/
parity

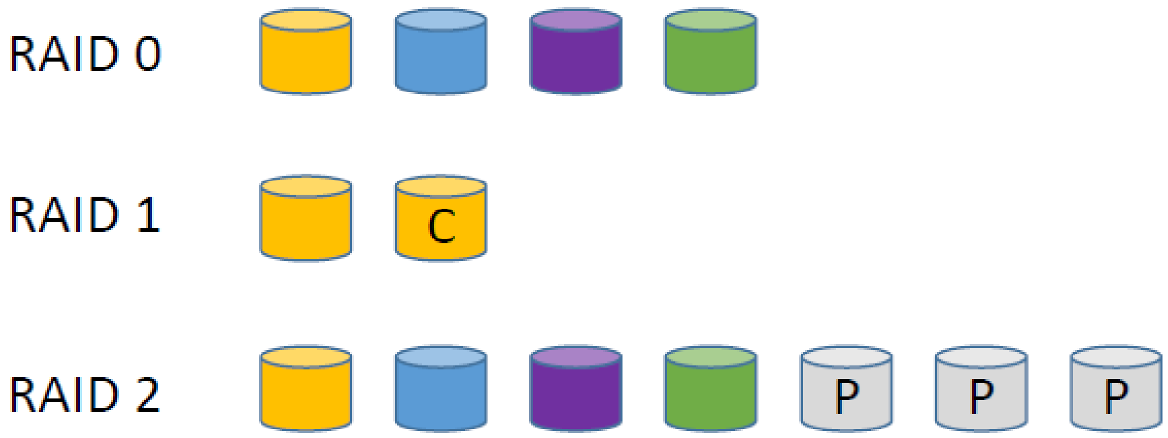
RAID 2



Q: What is the parity of a byte?

RAID

Striping w/
parity



Q: What is the parity of a byte?

If number of bits (1s) is even, then parity is 0
If number of bits (1s) is odd, then the parity is 1

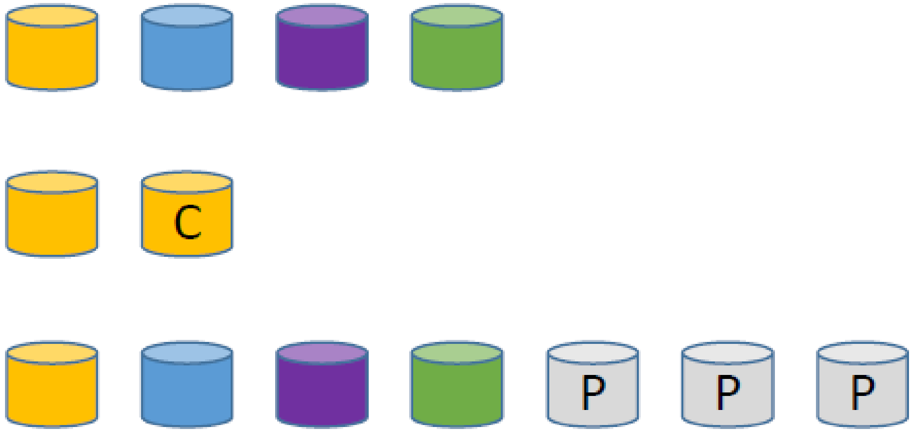
RAID

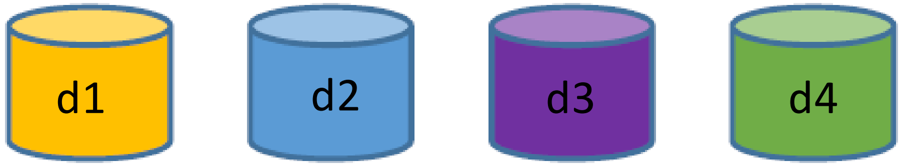
RAID 0

RAID 1

Striping w/ parity

RAID 2





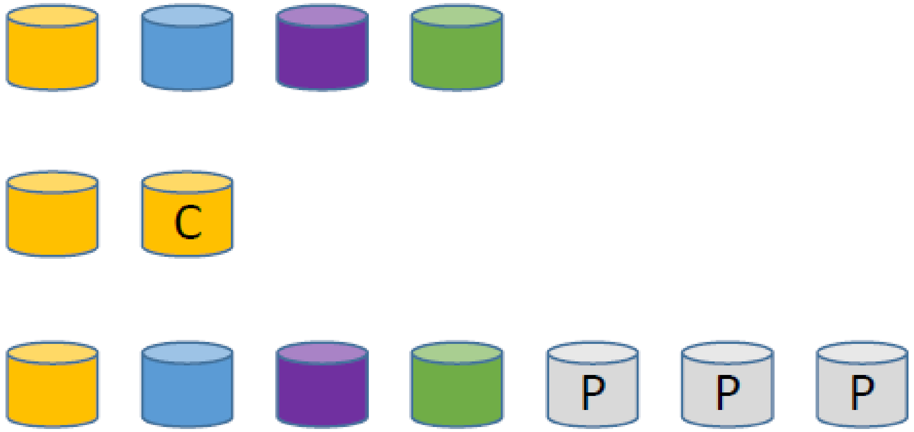
RAID

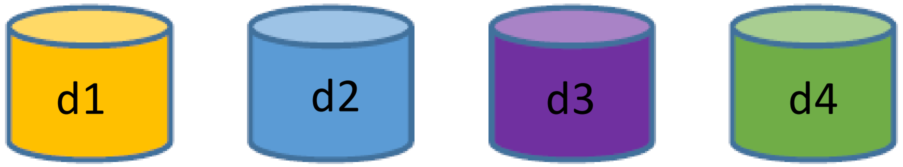
RAID 0

RAID 1

Striping w/ parity

RAID 2





d4

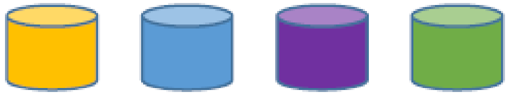
d2

d1

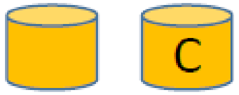
d3

RAID

RAID 0

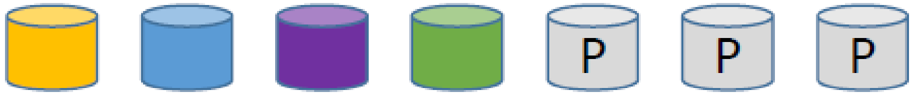


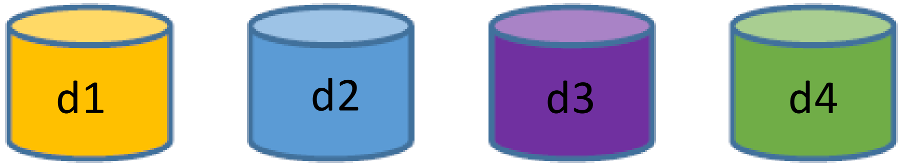
RAID 1

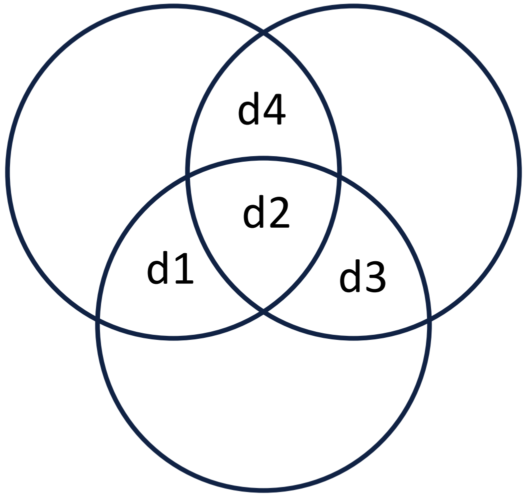


Striping w/
parity

RAID 2







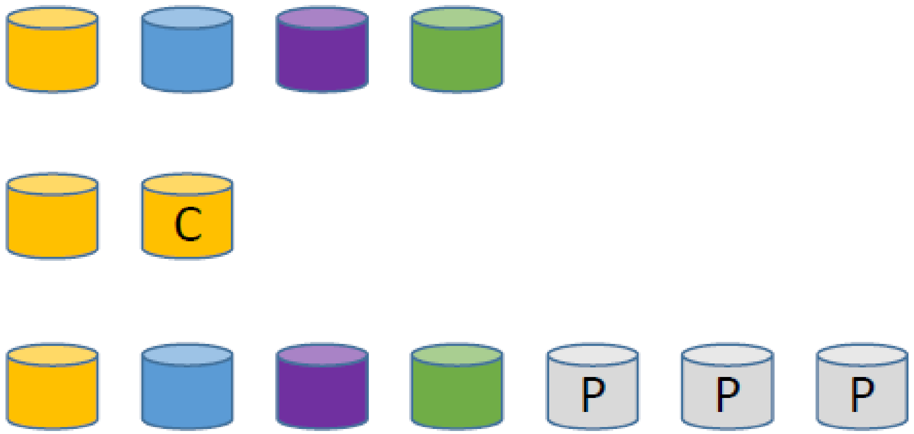
RAID

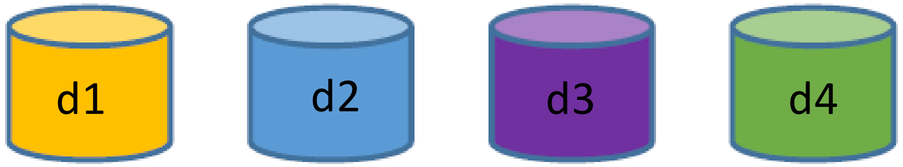
Striping w/ parity

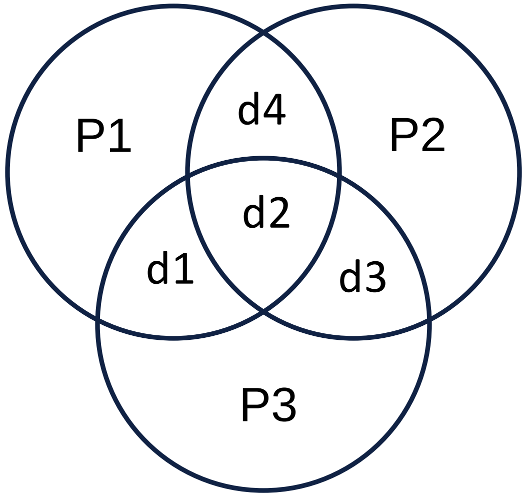
RAID 0

RAID 1

RAID 2







Calculate 3 Parity bits of 3 different combinations

This allows us to recover any 1-bit error.

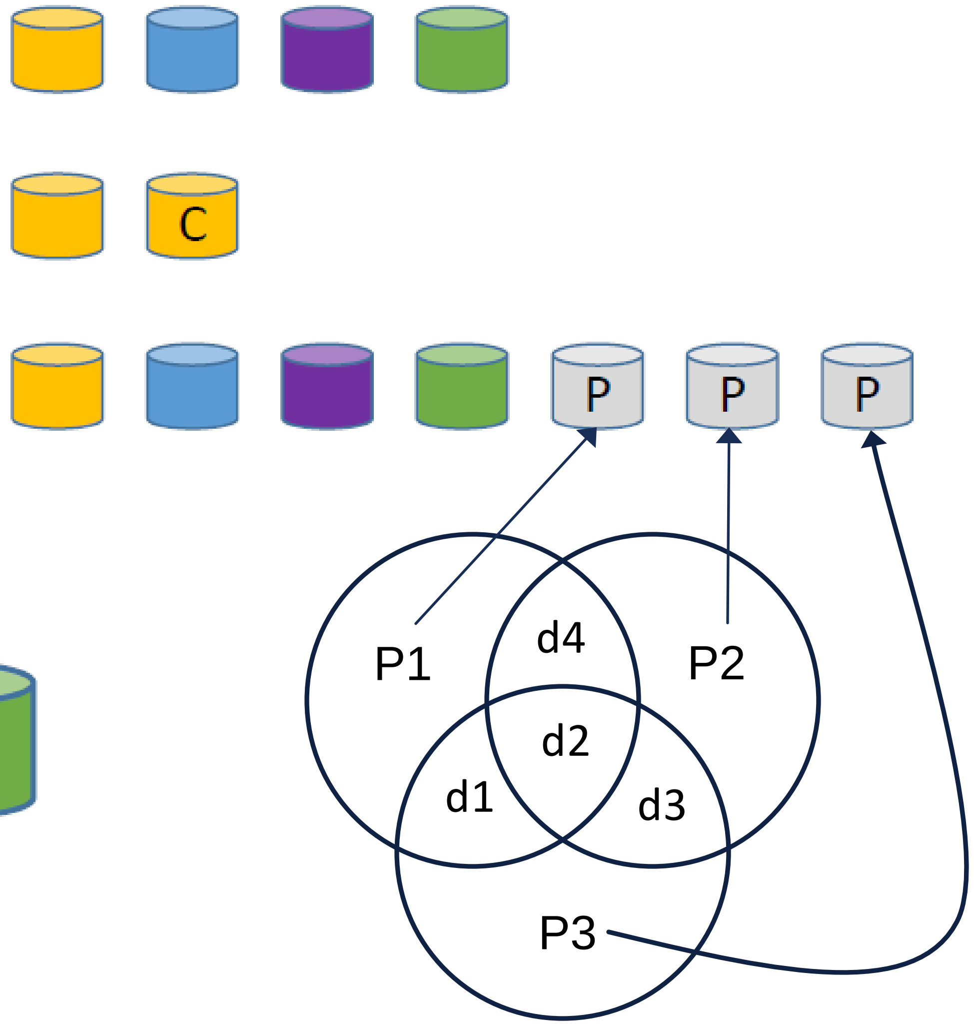
RAID

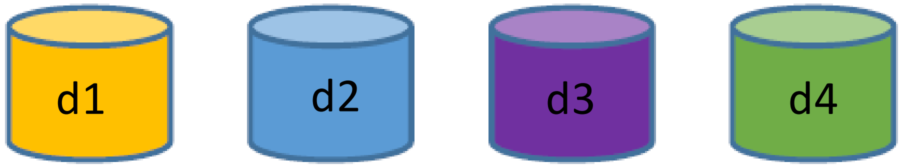
RAID 0

RAID 1

Striping w/
parity

RAID 2





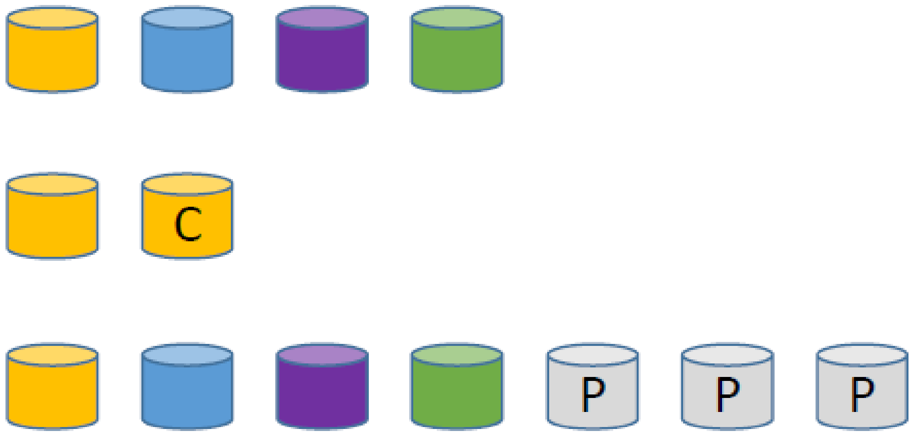
RAID

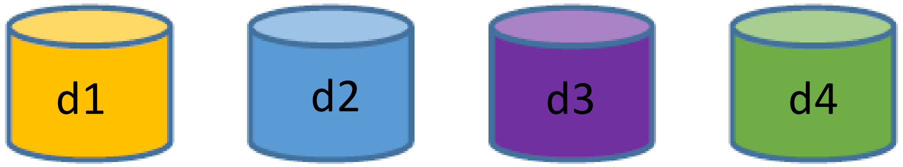
RAID 0

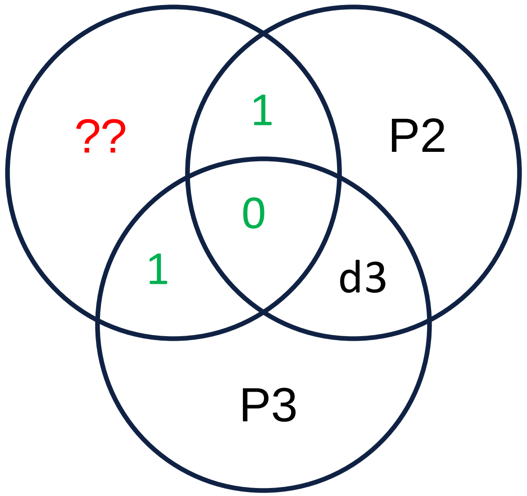
RAID 1

Striping w/ parity

RAID 2







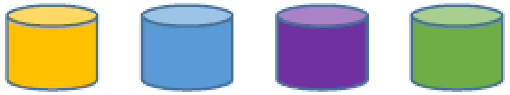
P1 = ?

A: P1=1
B: P1=0
C: P1=2

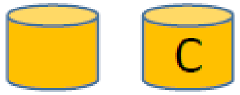
RAID

Worksheet Q3

RAID 0

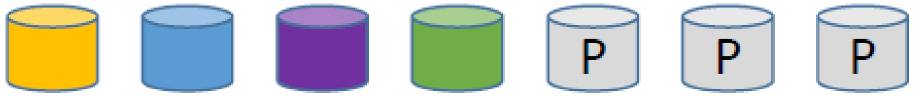


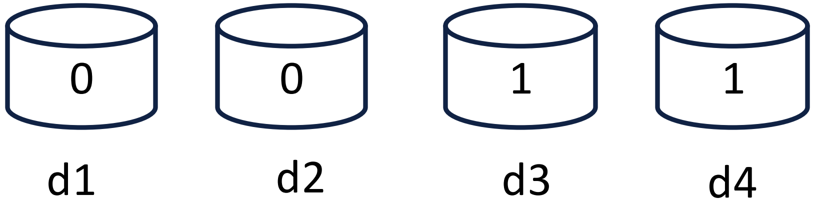
RAID 1

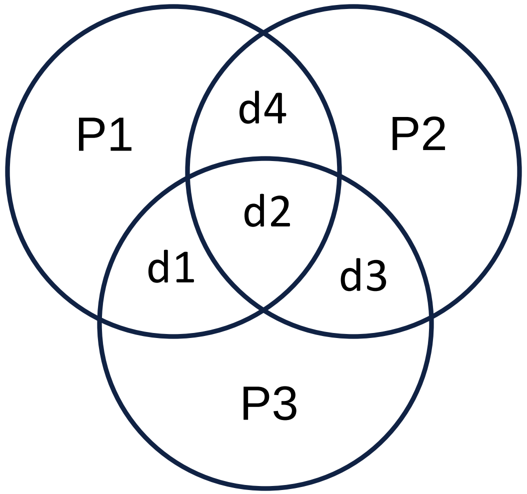


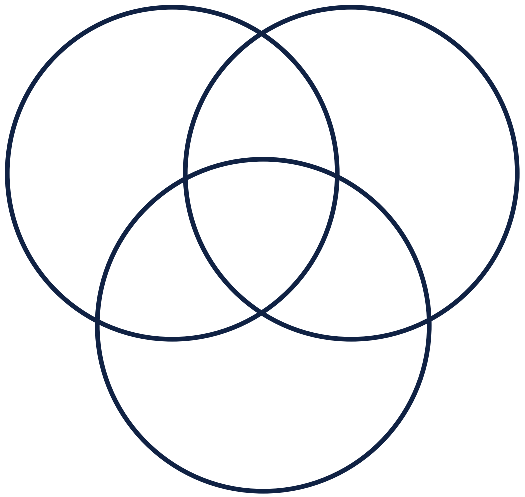
Striping w/ parity

RAID 2





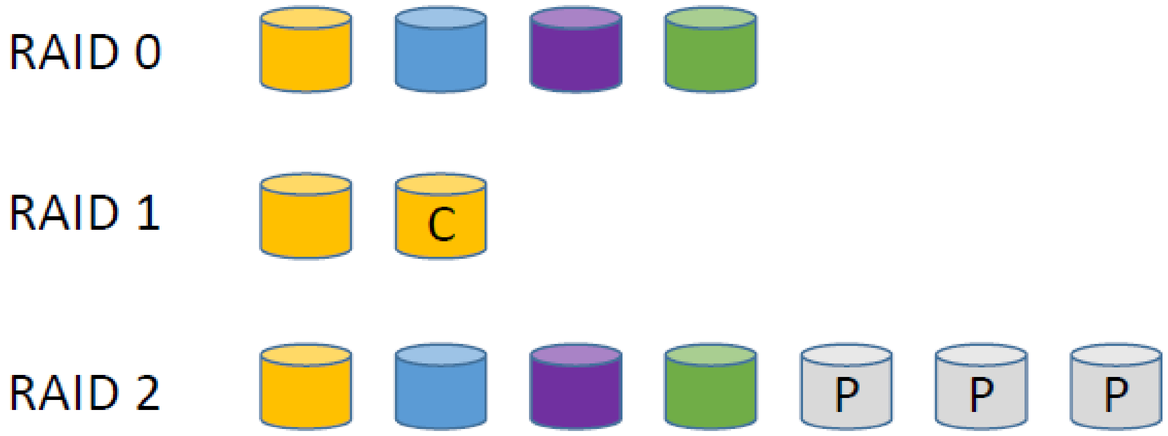


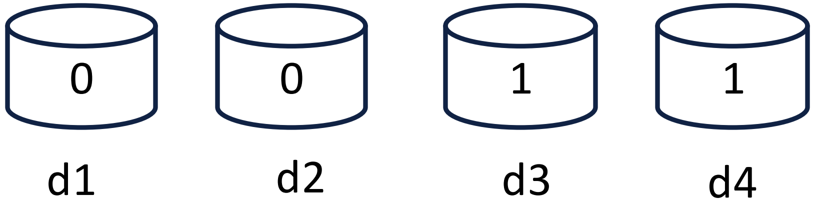


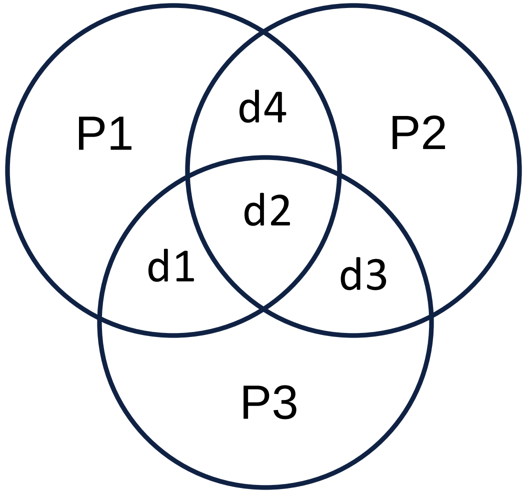
RAID

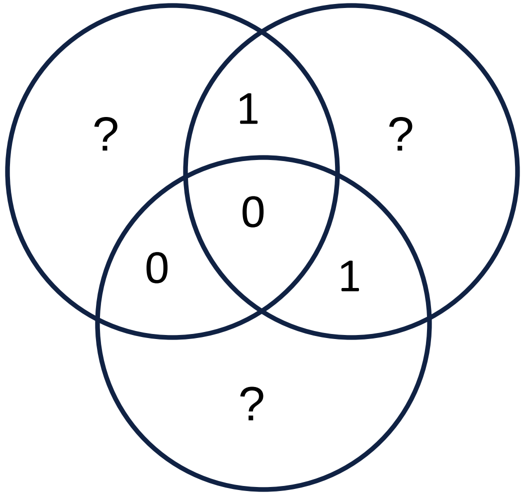
Worksheet Q3

Striping w/
parity



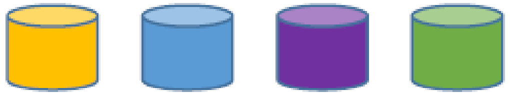




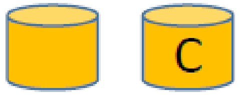


RAID

RAID 0

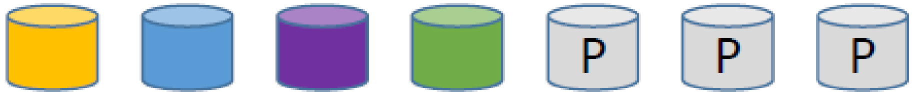


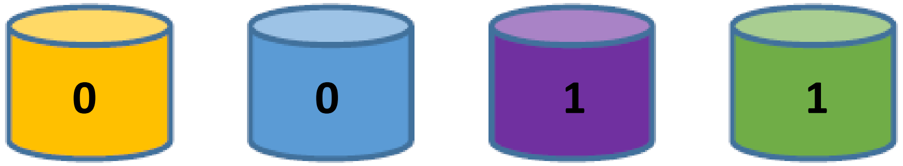
RAID 1

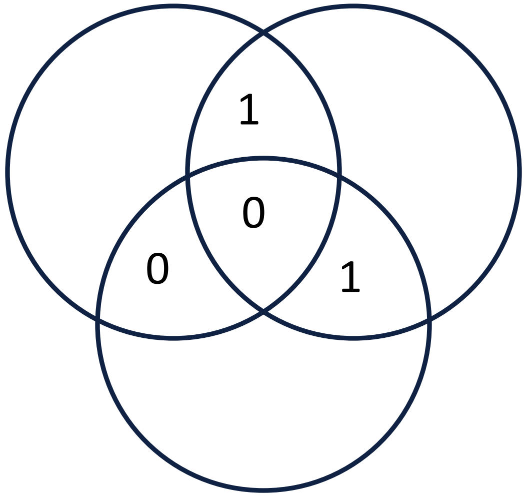


Striping w/ parity

RAID 2

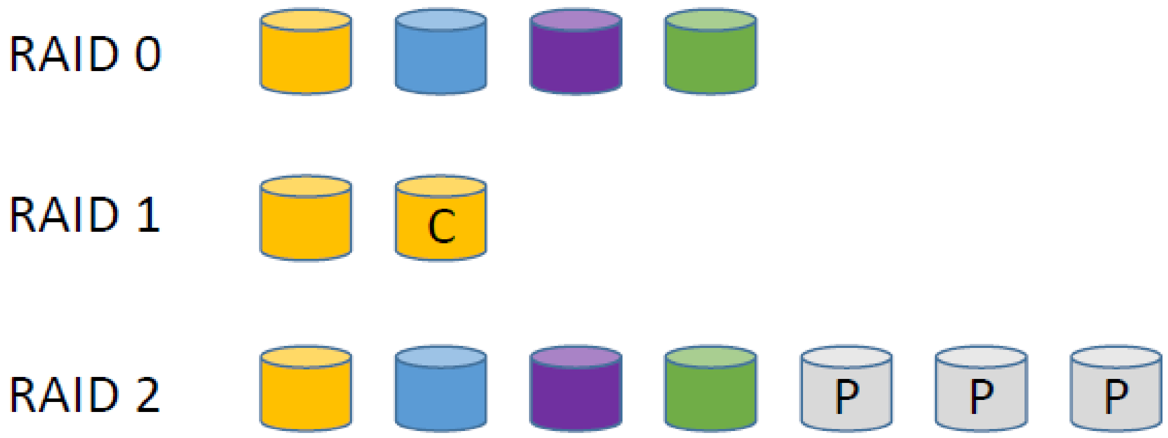


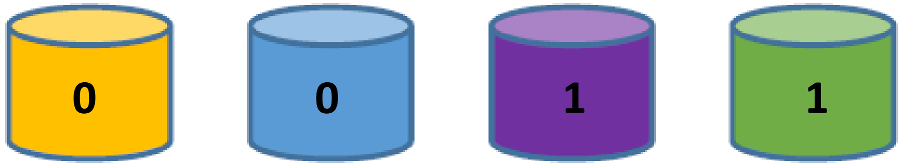


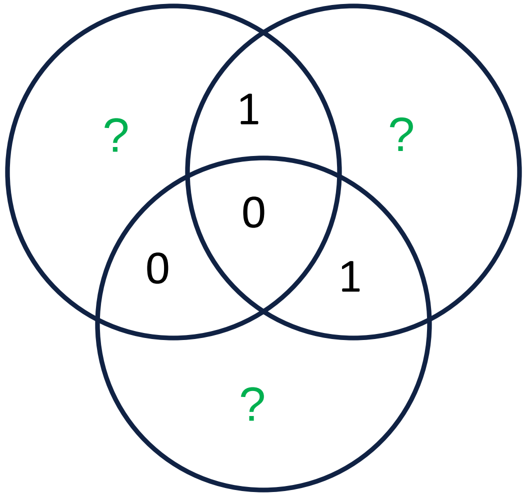


RAID

Striping w/ parity







Let’s calculate the three parity bits.

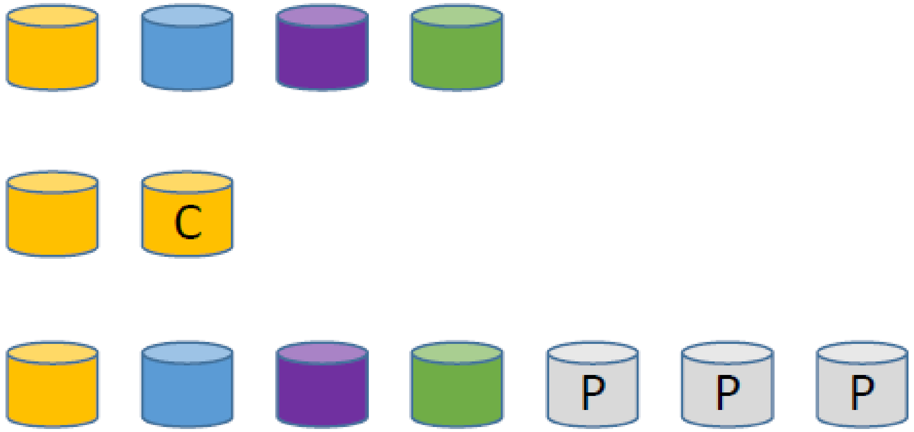
RAID

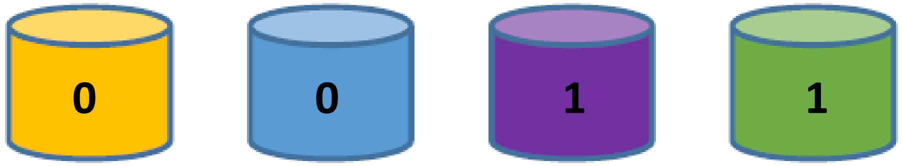
RAID 0

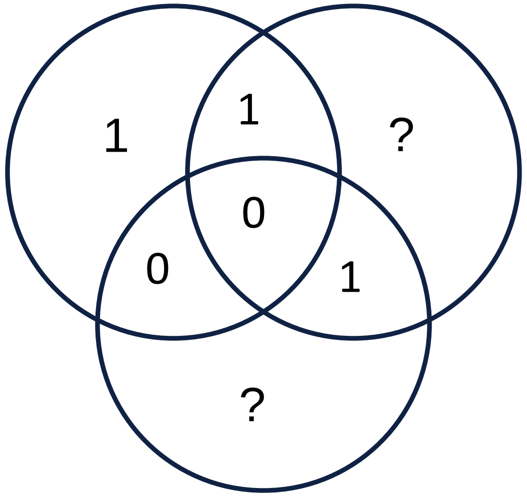
RAID 1

Striping w/ parity

RAID 2







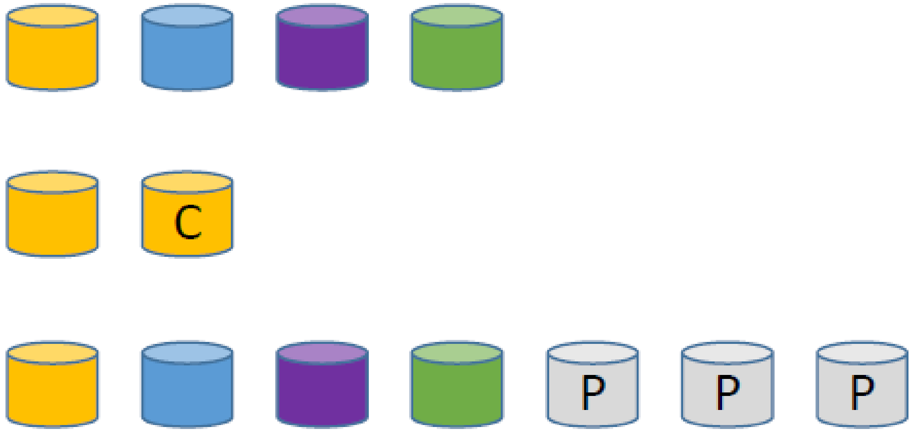
RAID

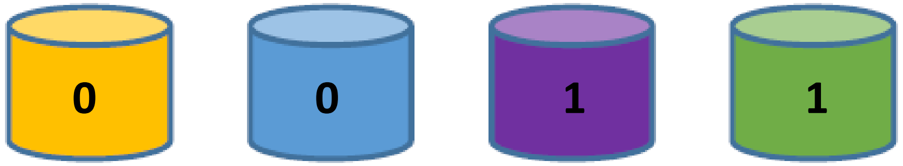
Striping w/ parity

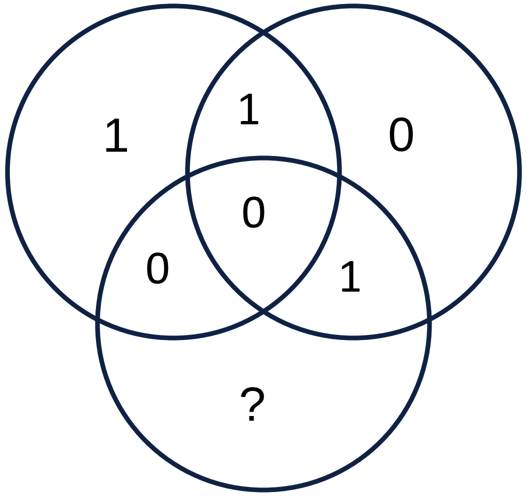
RAID 0

RAID 1

RAID 2







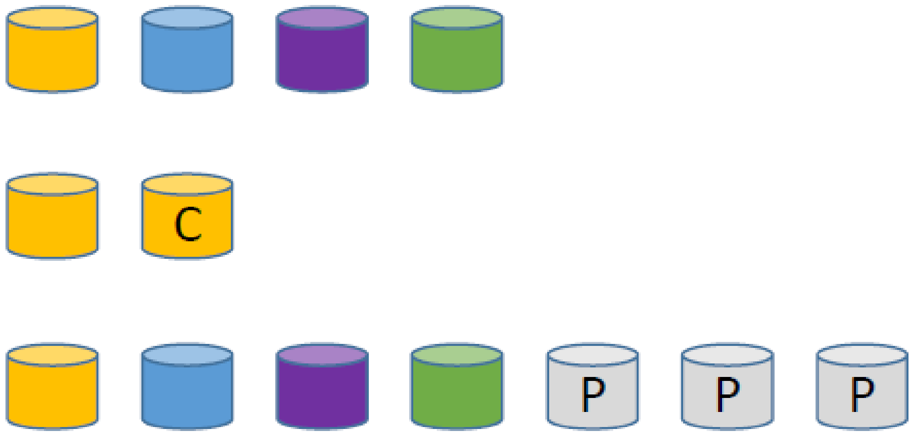
RAID

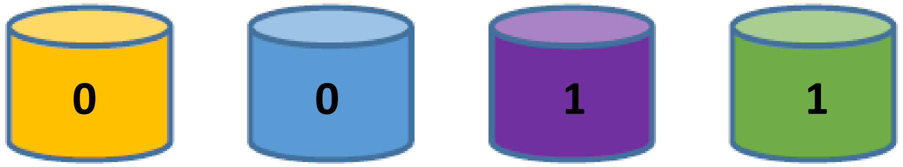
RAID 0

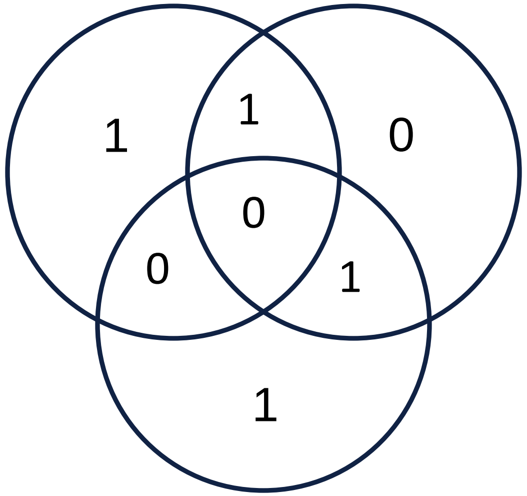
RAID 1

Striping w/ parity

RAID 2





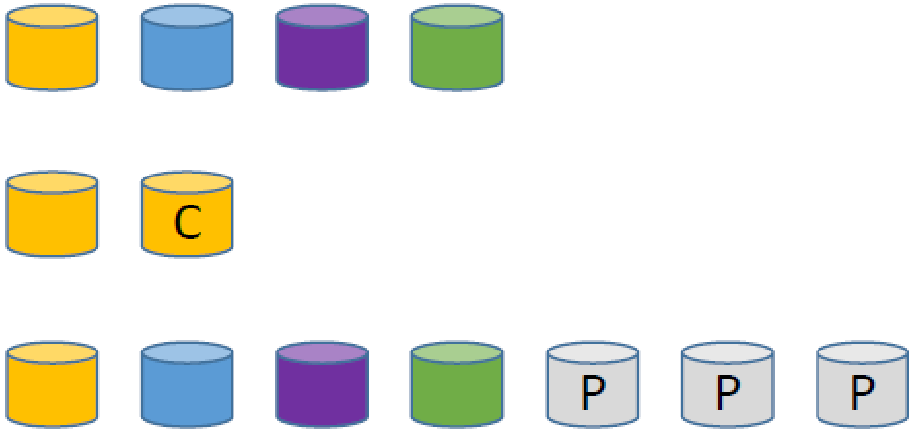


RAID

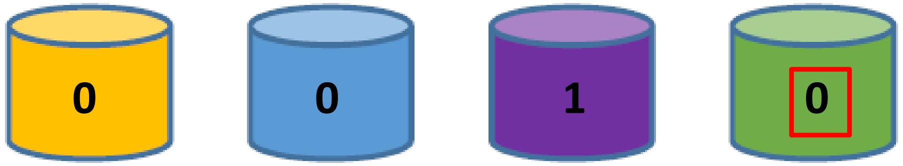
RAID 0

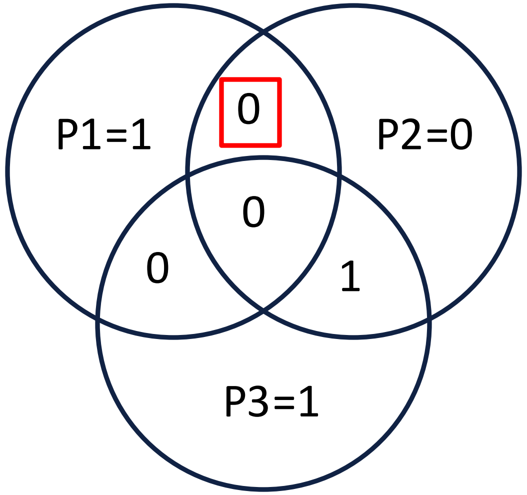
RAID 1

RAID 2



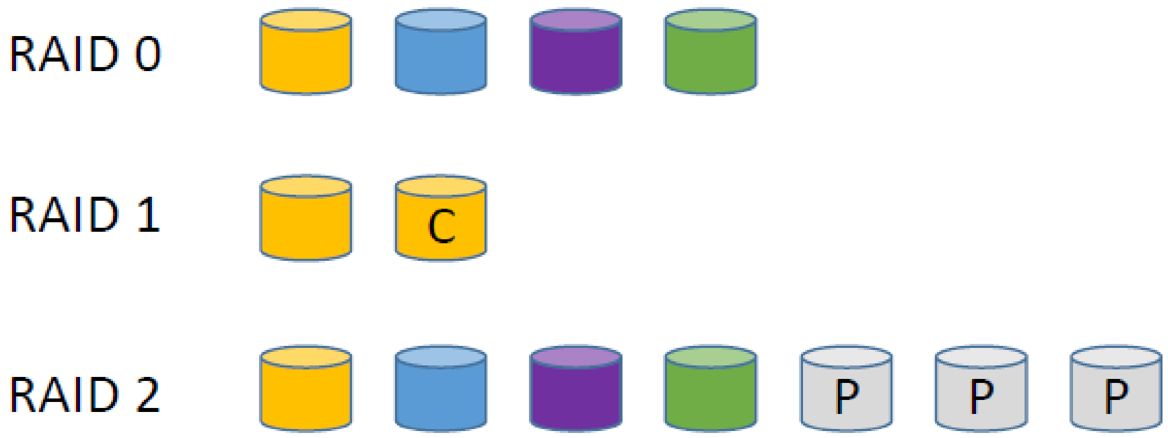
Stripping, no mirroring, but use of <u>parity</u> bits

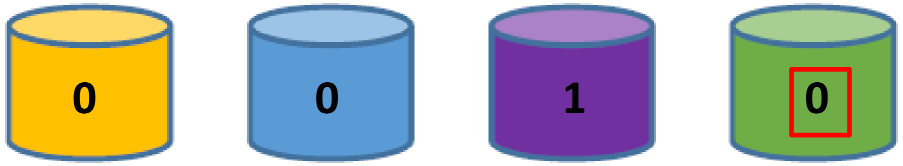




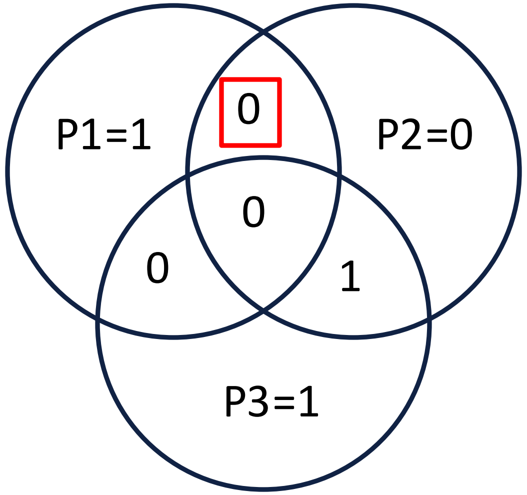
Suppose we had a single bit error at d4

RAID





Suppose we had a single bit error at d4



Which parity bits have conflicting values?

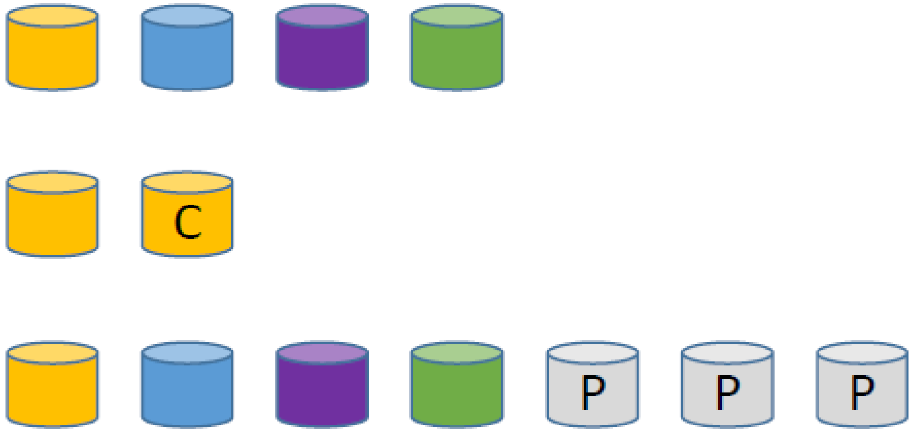
Worksheet Q5

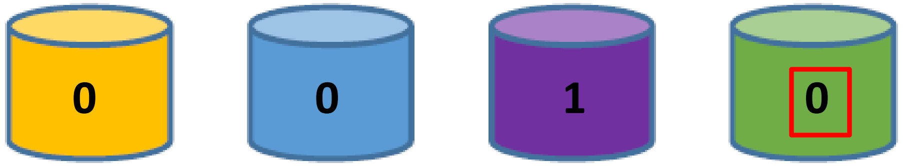
RAID

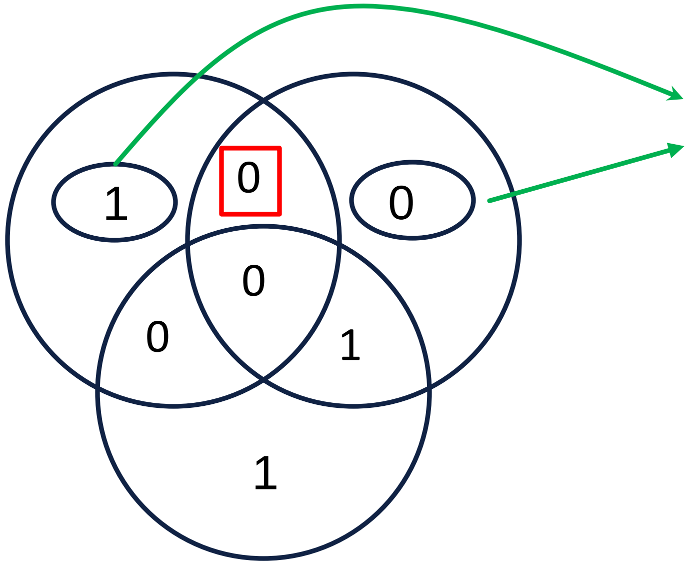
RAID 0

RAID 1

RAID 2



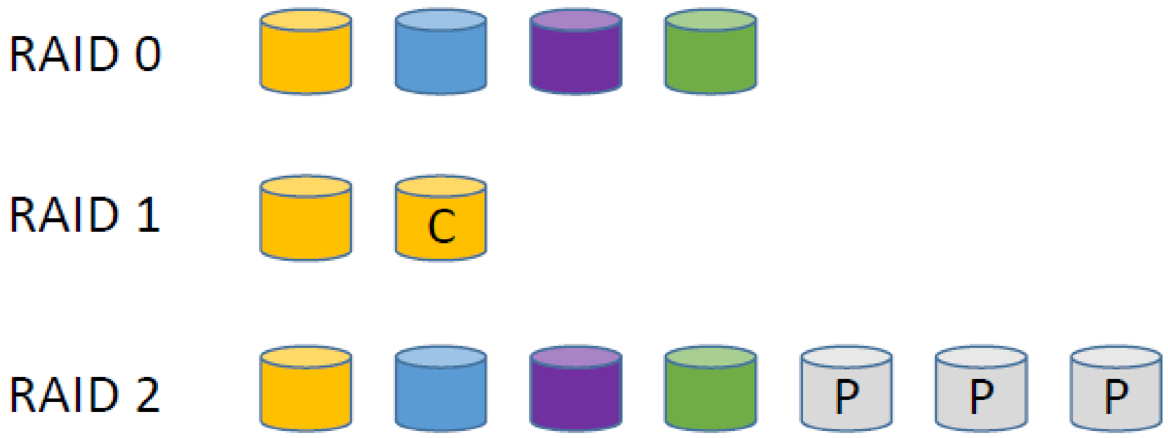


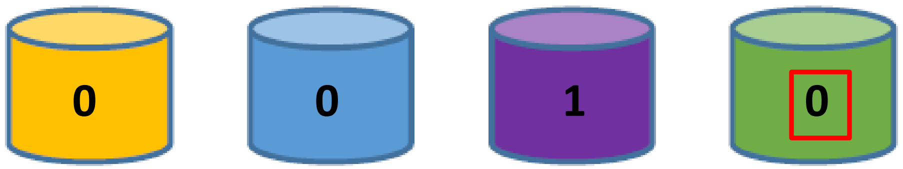


The new bit value of d4 conflicts with two Parity bits P1 and P2

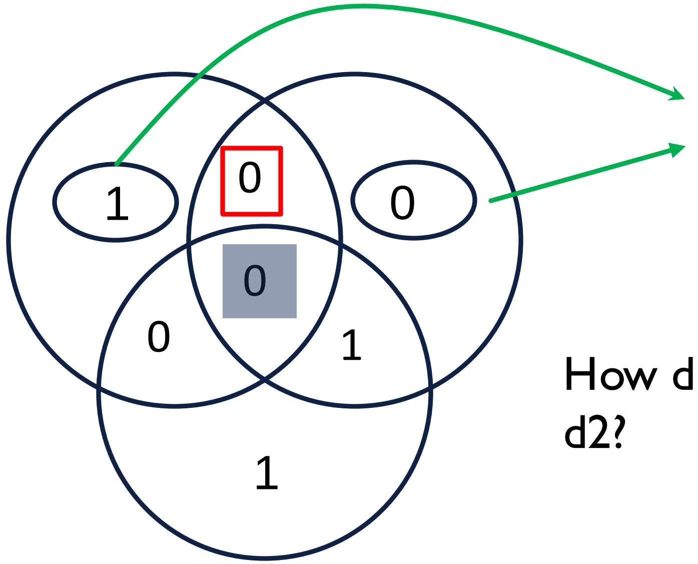
Suppose we had a single bit error at d4

RAID





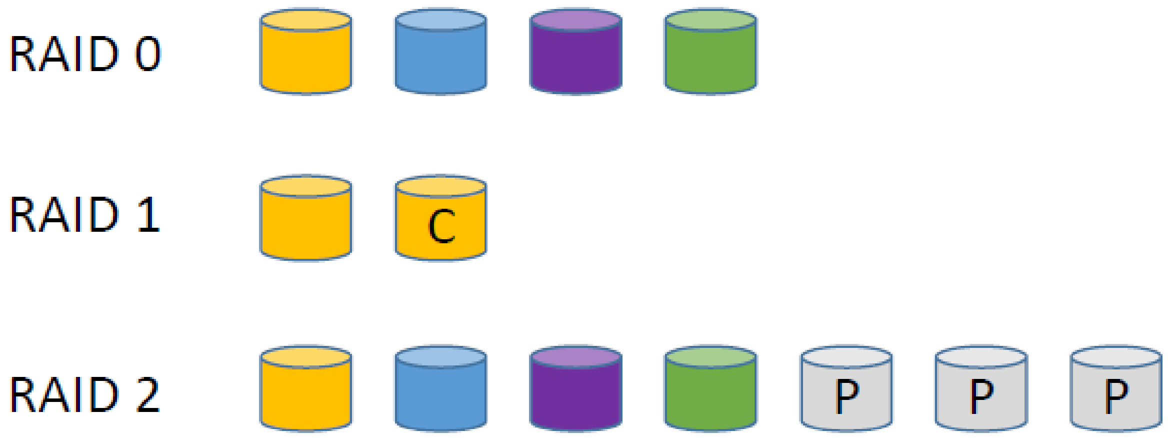
Suppose we had a
single bit error at d4

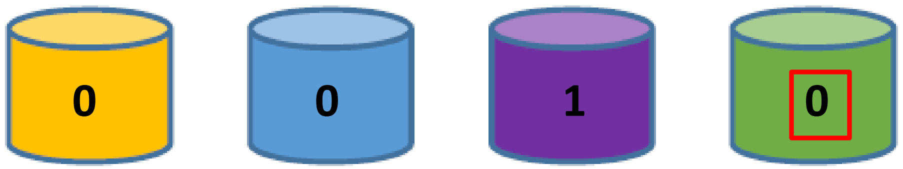


The new bit value of d4 conflicts with two Parity bits P1 and P2

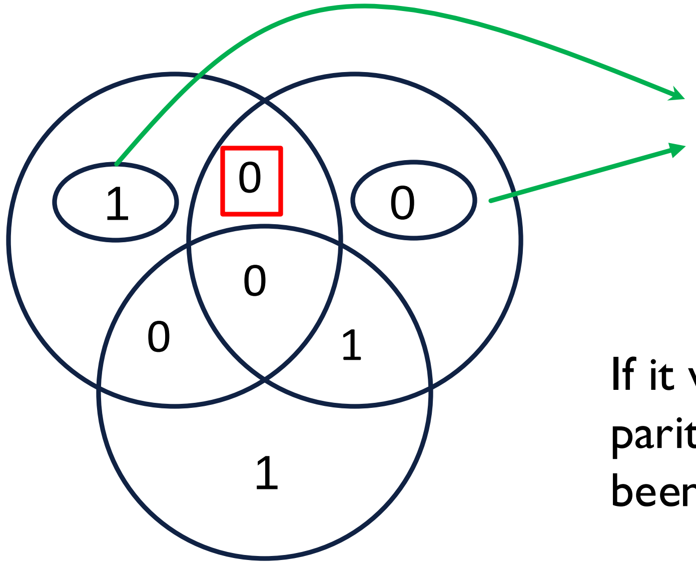
How do we know it's not d2?

RAID





Suppose we had a single bit error at d4



The new bit value of d4 conflicts with two Parity bits P1 and P2

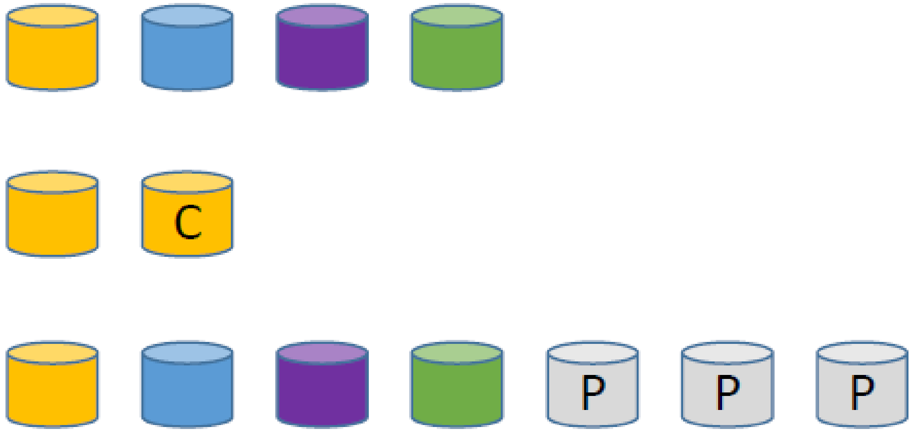
If it was d2, all three parity bits would have been invalid.

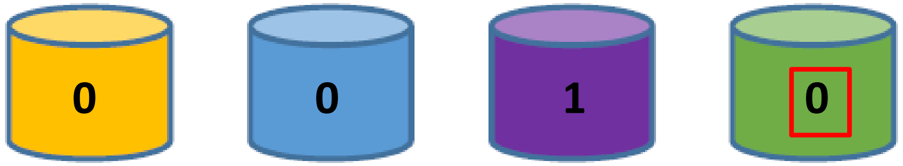
RAID

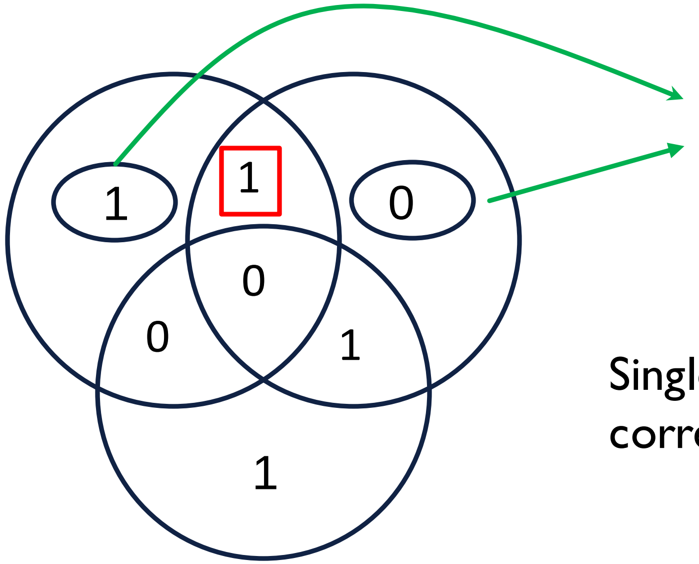
RAID 0

RAID 1

RAID 2







The new bit value of d4 conflicts with two Parity bits P1 and P2

Suppose we had a single bit error at d4

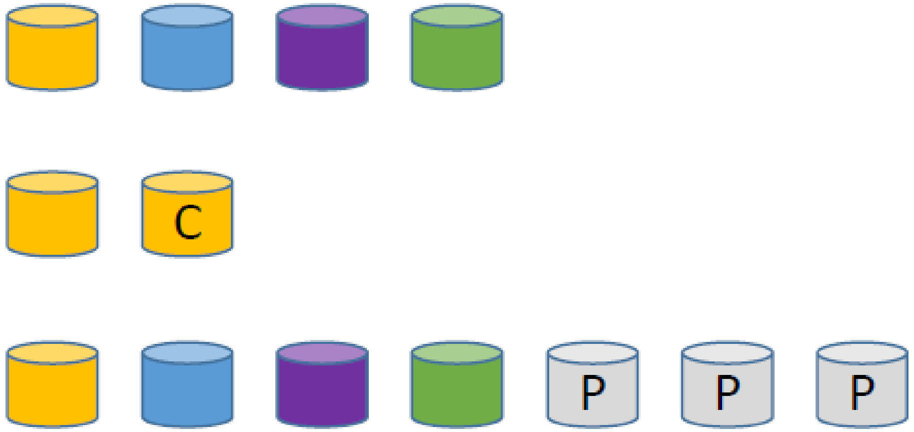
Single bit errors can be corrected!

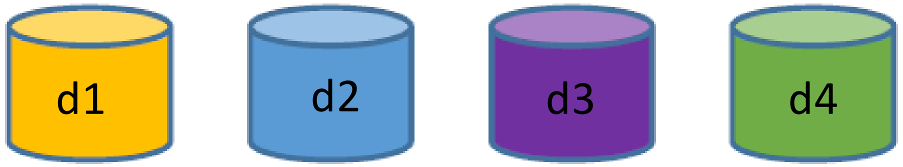
RAID

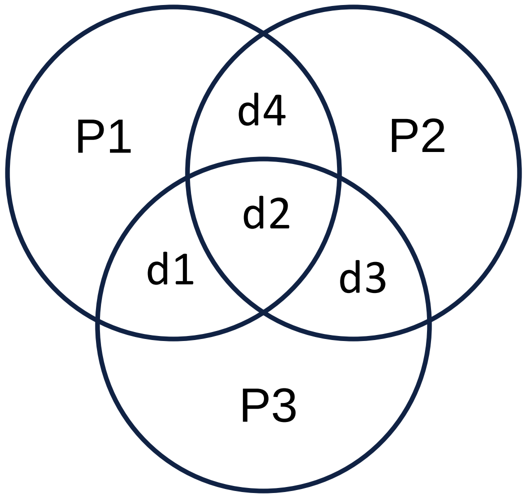
RAID 0

RAID 1

RAID 2



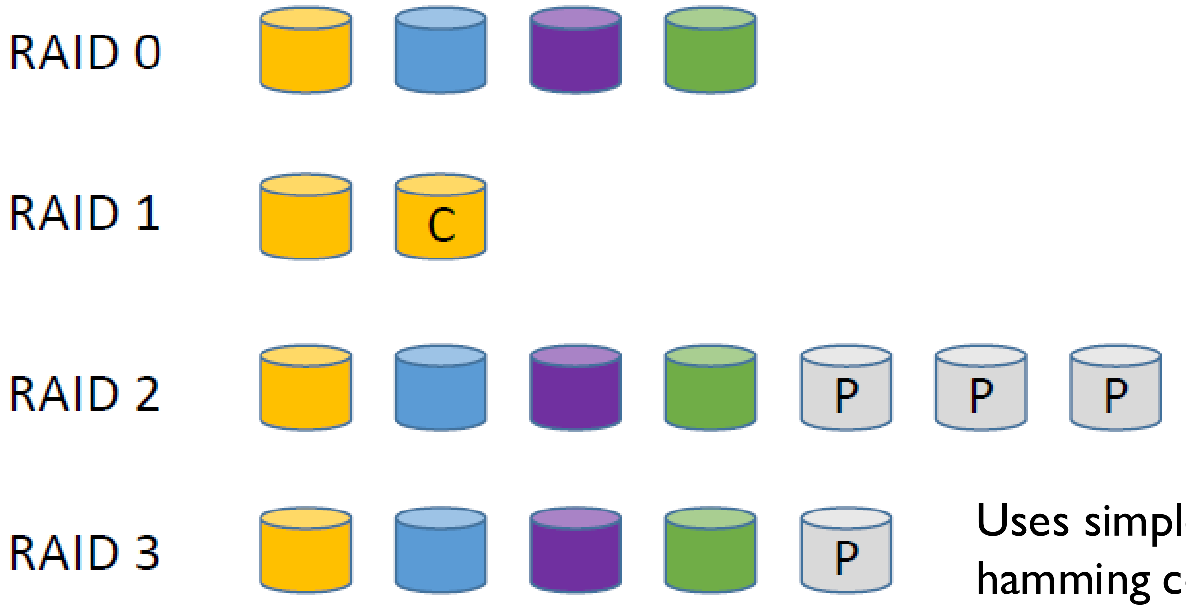




Hamming Code (7,4)

RAID

Striping w/
simplified parity



Uses simple parity and no hamming code.

RAID

RAID 0

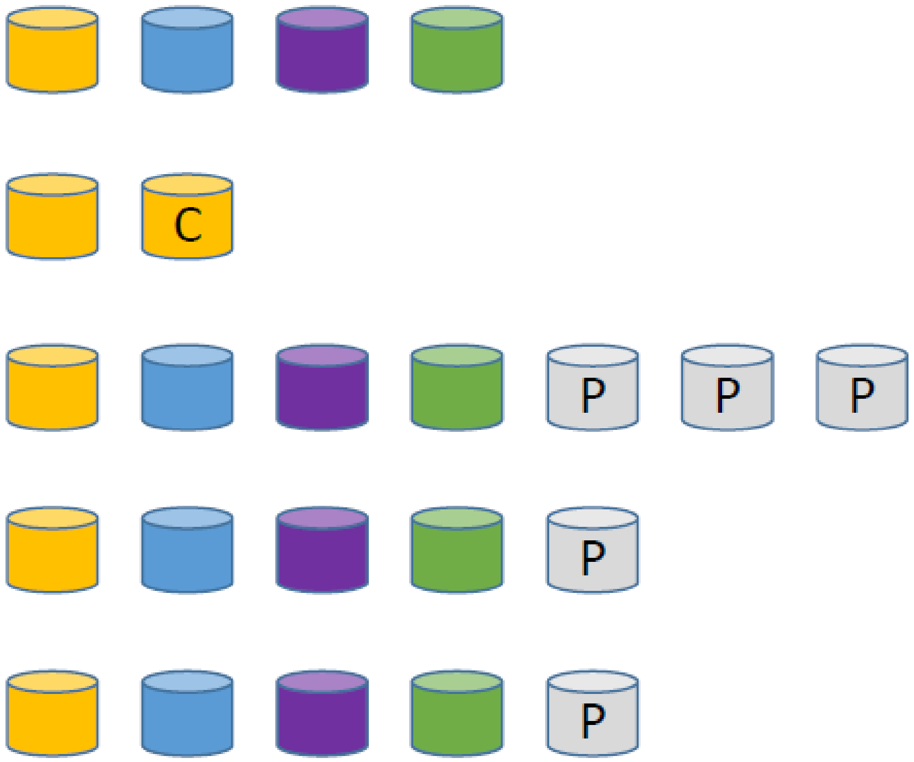
RAID 1

RAID 2

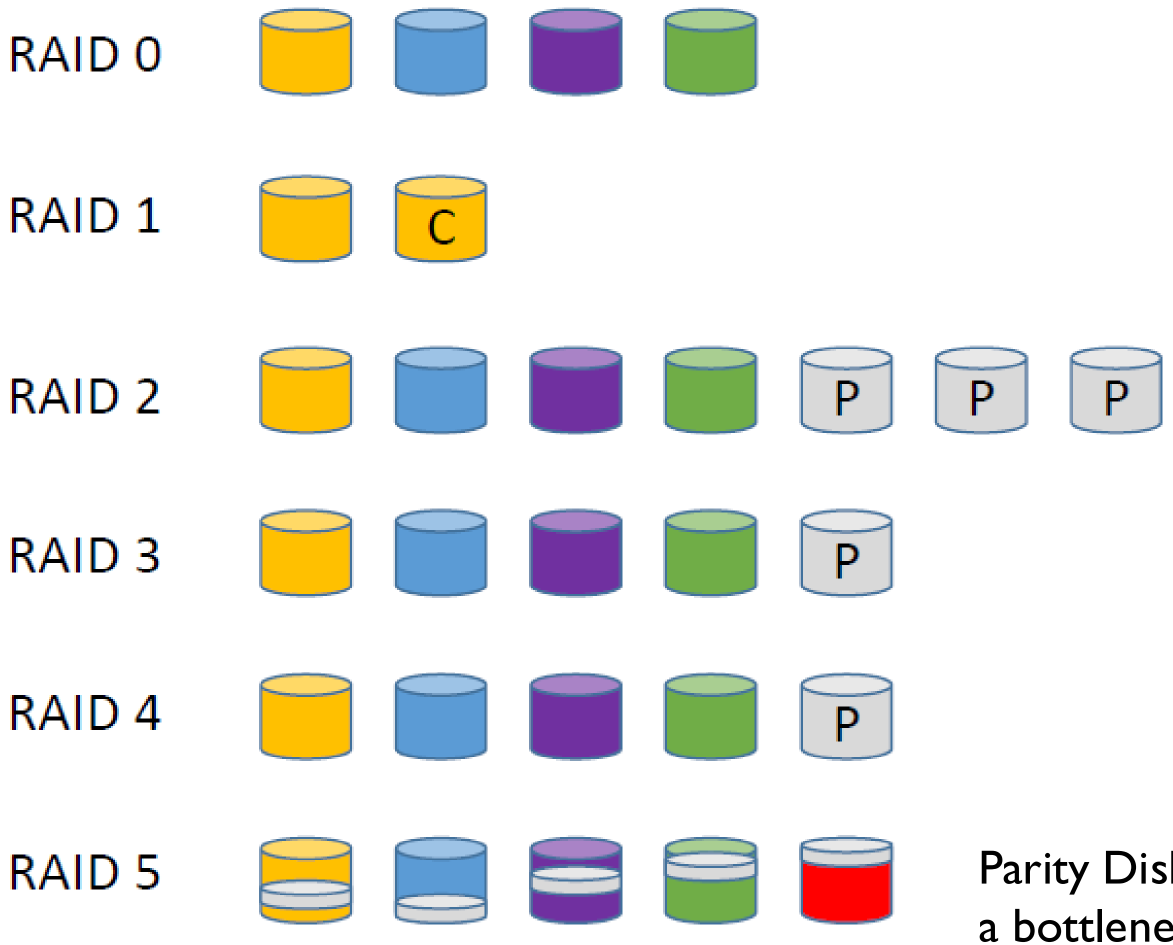
RAID 3

Block level
striping w/ parity

RAID 4



RAID

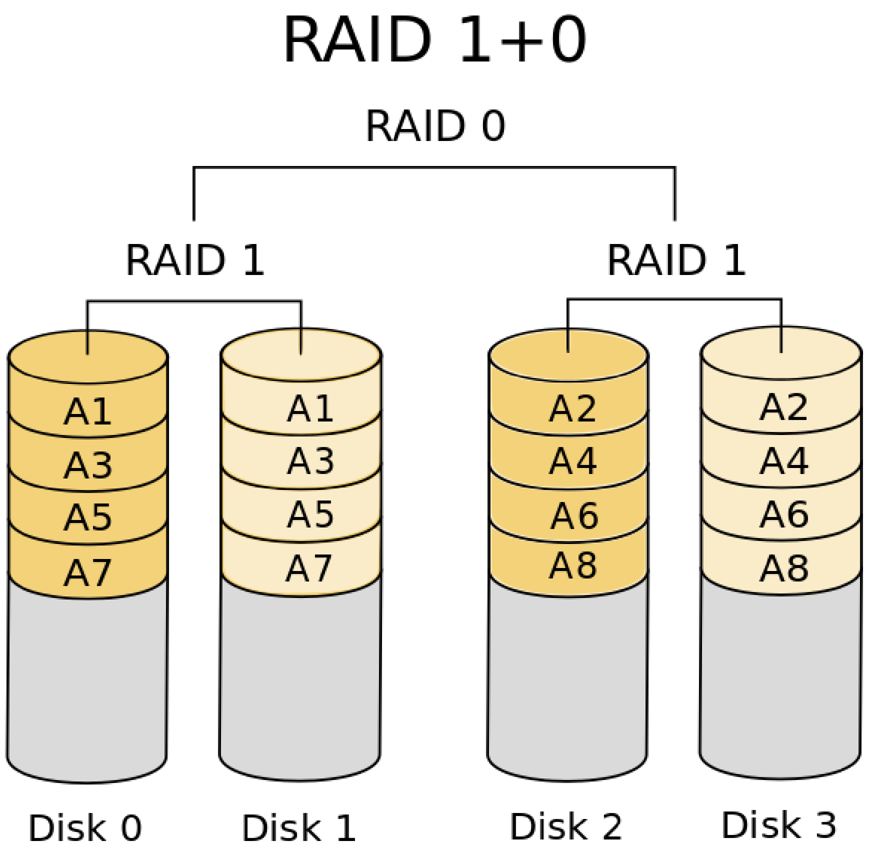


Block Level Striping w/ distributed Parity

Parity Disk in RAID 4 used to be a bottleneck and over "stressed" causing increased failure.

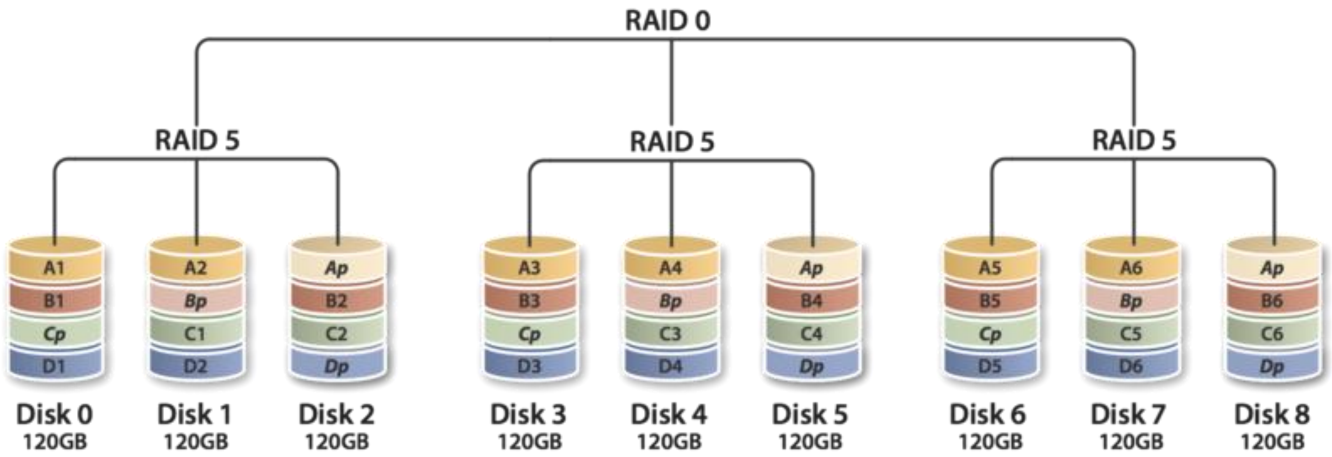
NESTED RAID SETUP

- RAID setups are usually nested.
- RAID 10 or RAID 1+0 is a very popular setup.
- Combines both mirroring and striping.
- No parity.
- Advantage:
  - No need to "pause" system during failure.
  - Replace disk and rebuild from mirror all while it's still running.



RAID 50

- Combine multiple Raid 5 systems using RAID 0 striping.
- Increased reliability.

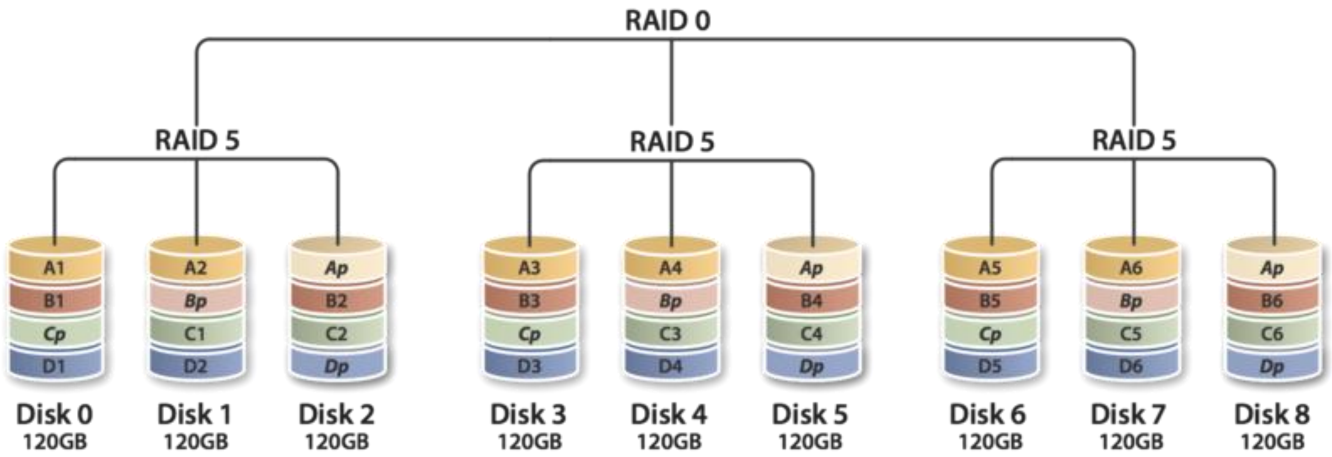


RAID 50

- Combine multiple Raid 5 systems using RAID 0 striping.
- Increased reliability.

How many disk failures can we tolerate in shown setup? Best case.

A: 1
B: 2
C: 3
D: 6

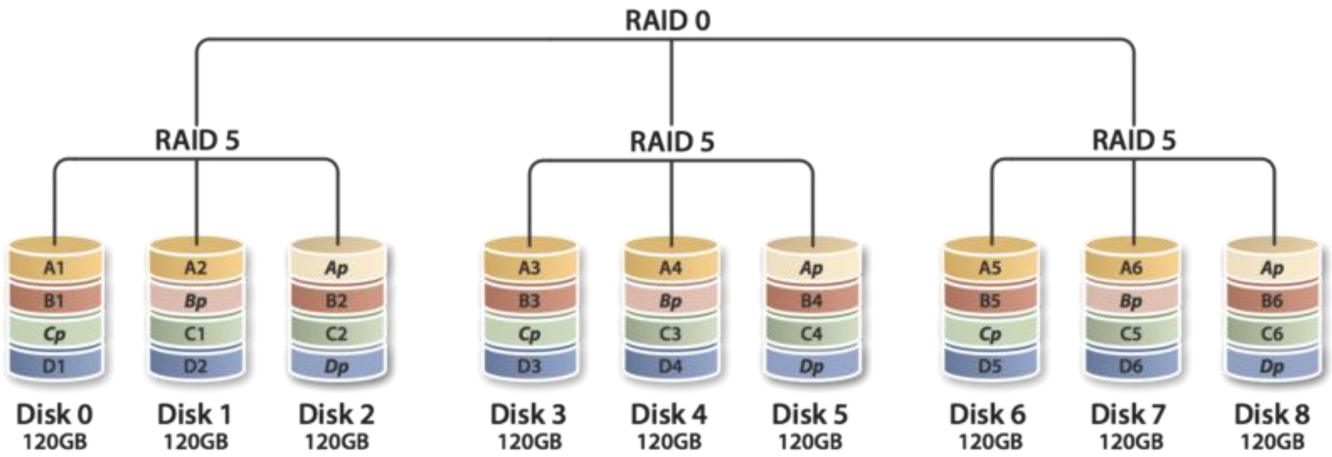


RAID 50

- Combine multiple Raid 5 systems using RAID 0 striping.
- Increased reliability.

How many disk failures can we tolerate in shown setup? Best case.

A: 1
B: 2
C: 3
D: 6

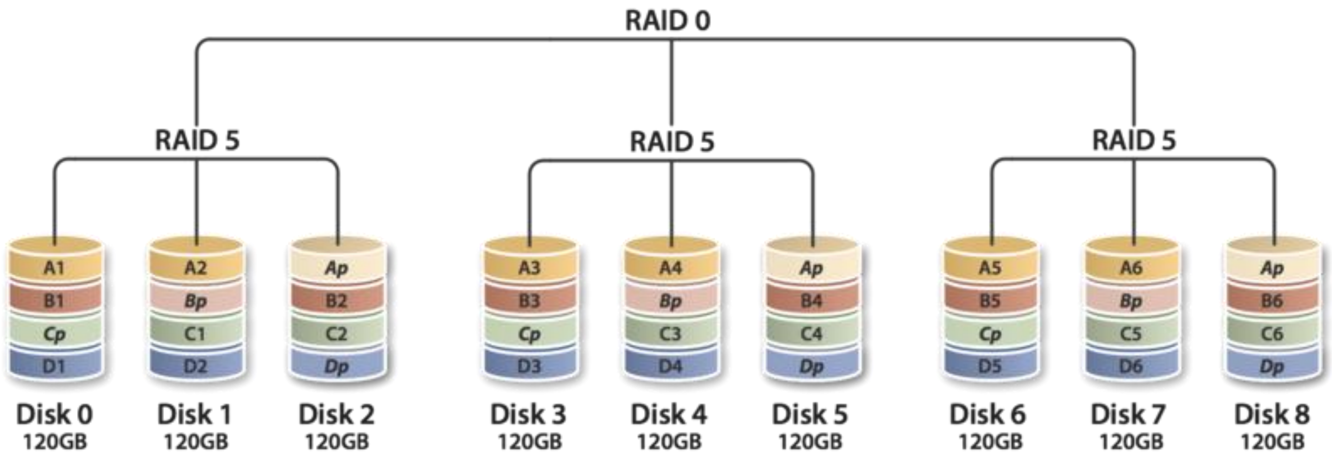


RAID 50

- Combine multiple Raid 5 systems using RAID 0 striping.
- Increased reliability.

How many disk failures are required to cause data loss? Worst case.

A: 1
B: 2
C: 3
D: 6



RAID 50

- Combine multiple Raid 5 systems using RAID 0 striping.
- Increased reliability.

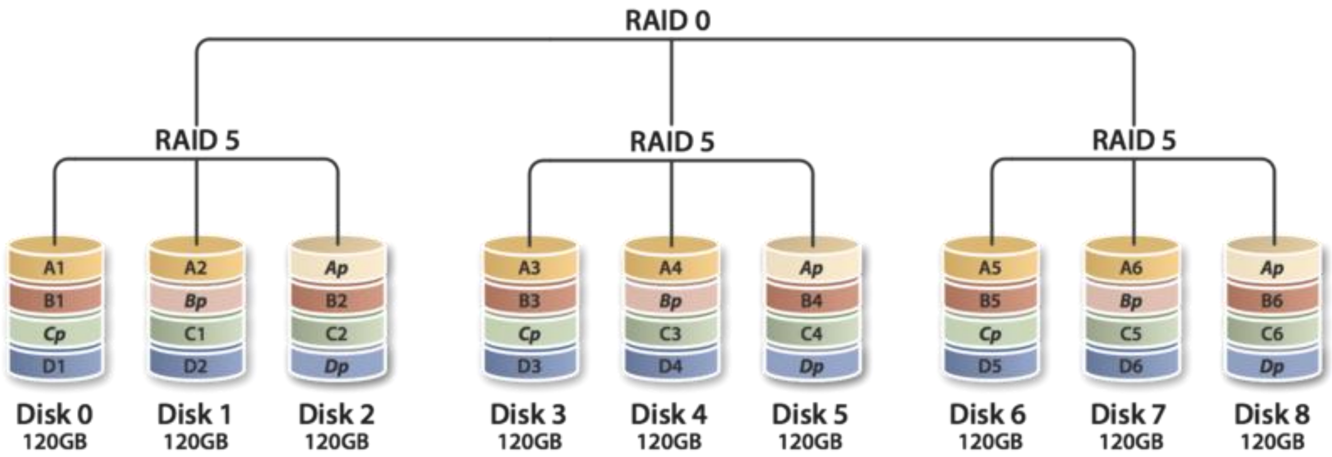
How many disk failures are required to cause data loss? Worst case.

A: 1

B: 2

C: 3

D: 6



RAID: FAILURES CAN STILL OCCUR

- It's assumed that disk failure is "random" but most often it's not. Result of power surge that hit all array …
- This is why it's advisable to have backup mirrors in a completely different location.

RAID: FAILURES CAN STILL OCCUR

- It's assumed that disk failure is "random" but most often it's not. Result of power surge that hit all array …
- This is why it's advisable to have backup mirrors in a completely different location.
- Some faults cannot be recovered through backup …
- Failure of a controller or overwrites to file system structure on disk can result in unrecoverable errors.