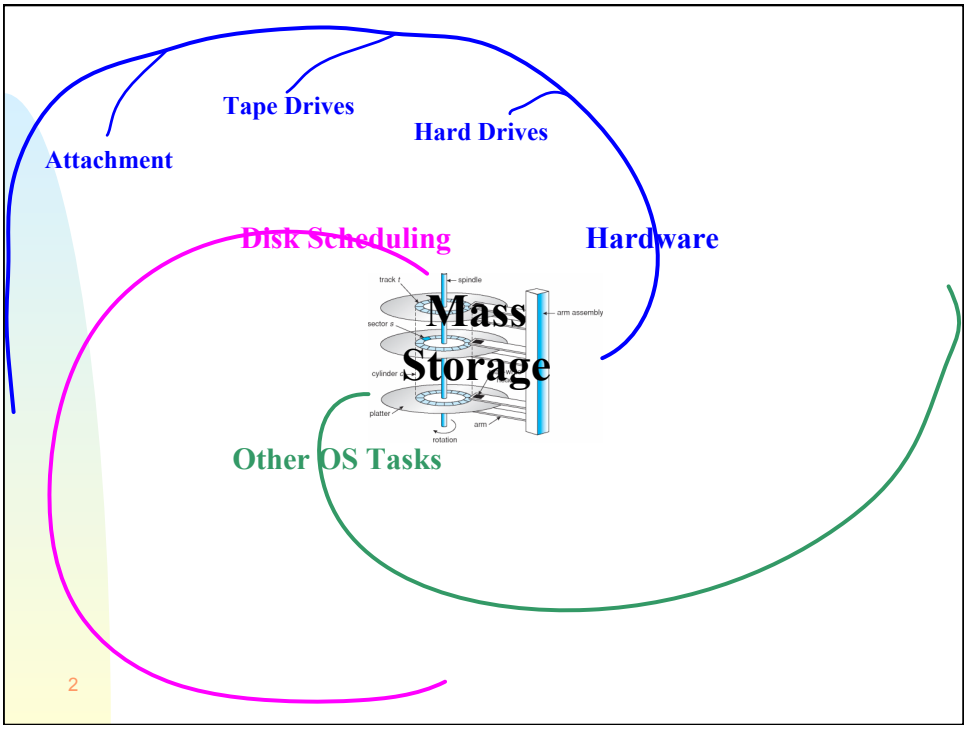
Module 10: Mass-Storage Structure

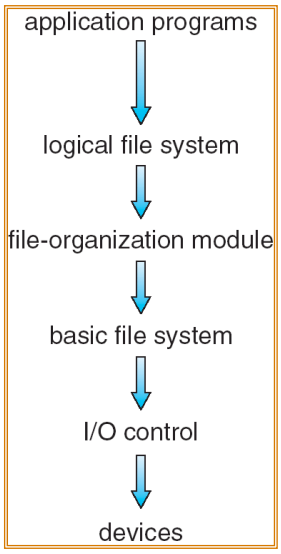
- **Reading: Chapter 10**
- **Objectives**
  - Describe the physical structure of secondary storage devices and the resulting effects on the uses of the devices
  - Explain the performance characteristics of mass-storage devices
  - Discuss operating-system services provided for mass storage

1



2

Layered File System Implementation



Mod 9: File System Interface

Mod 9: FS Implementation

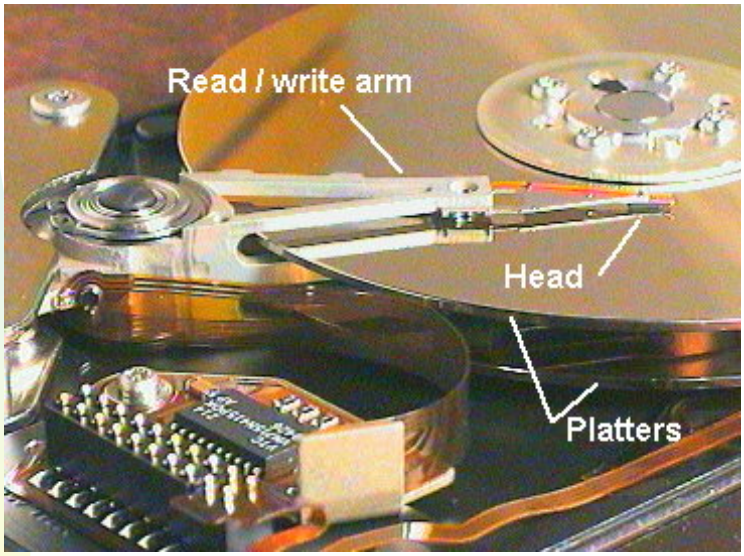
Mod 10: Mass storage & low-level FS

Mod 11: I/O Systems

Mod 10: Mass storage HW

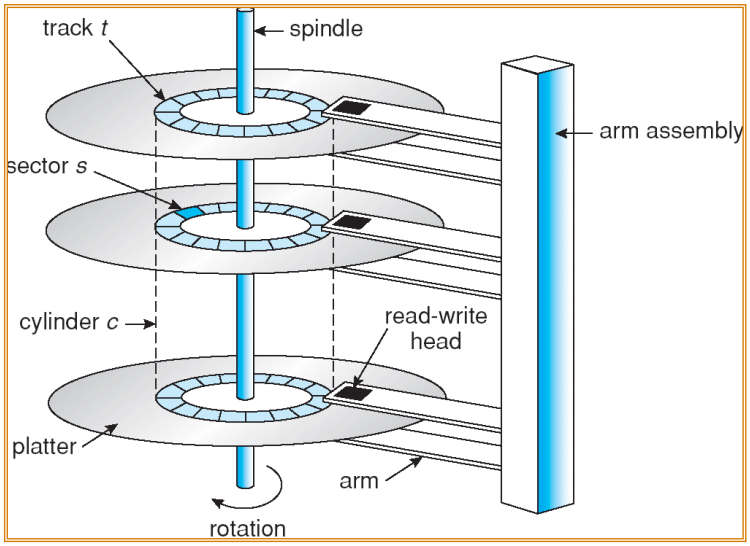
3

So, how does a disk look inside?



4

The insides of a hard drive!



5

Disk organization and properties

Disk organization:

- A set of platters covered with magnetic recording material.
- Sectors, tracks, cylinders, platters – 3D addressing space
  - Platter, track, sector
- The HD controller presents linear address space of sectors

Disk parameters:

- Latency – how long to get to the data
- Bandwidth – transfer rate
- Capacity, cost, reliability, noise

6

Disc Latency

What should happen in order to be able to read data from a given sector/track?

- **The head must move to the given track**
- **The platter must rotate to place the sector below the head**
- **There is a bit of command execution overhead**

Ok, so how long does it take?

- **7200RPM disc = 120 revolutions/s**
- **On average, needs half revolution to reach the sector = 1/240s ≈ 0.004s =4ms for** rotational latency
- **The** seek time **(head movement) is comparable to the rotational latency**

Expect approximately 8ms latency – in reality a bit higher

http://www.storagereview.com/articles/200505/20050518WD3200JD_1.html

7

Disc Transfer Rate

Ideal transfer rate

- **Contents of one cylinder/time for one revolution**
- **Can only be achieved for contiguously allocated large files**

But the tracks are shorter near the centre!

- **Have the inner tracks have less sectors, and spin up faster to maintain** constant linear velocity **(CD,DVD)**
  - **Harsher on HW (acceleration/deceleration)**
- **Pack the data denser in the inner tracks to maintain** constant angular velocity **(HD)**
  - **Wasted space on the outer tracks**

The connection of the HD to the computer (via I/O bus) should not limit the bandwidth

- **Busses vary, including** EIDE, ATA, SATA, USB, Fibre Channel, SCSI
- Host controller **in computer uses bus to talk to** disk controller **built into drive or storage array**

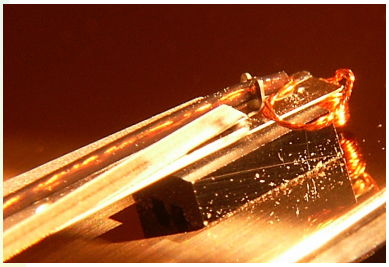
8

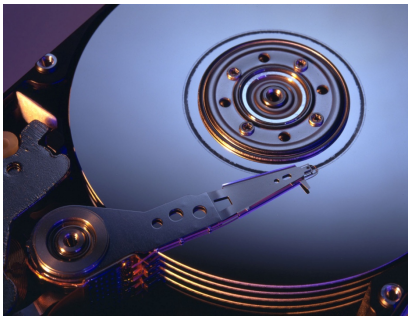
Head Crash

The read/write head floats on cca 0.0004mm of air above the platter spinning at 7200+ RPM

What happens if it touches it?

- Head crash, say good-bye to your data
- A bit similar to an airplane landing on its belly





9

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of *logical blocks*, where the logical block is the smallest unit of transfer.
- The 1-dimensional array of logical blocks is mapped into the 3D structure (sector, track, platter) of the disk sequentially
- How to do the mapping?
  - Start with sector 0 of the outermost track of the top platter
  - Proceed with the sectors of that track
  - Which track to consider next?
    - The next inner track on the same platter
    - The track on the same cylinder, but on the next platter
  - Why the second option?
    - Reading a contiguously allocated file involves much less head movement

10

Electronic Disks

- Today, more and more other types of memory are treated like hard disks, but are completely electronic
  - For example, flash memory (memory sticks)
  - These devices do not contain seek time, rotational latency, etc.

11

Solid-State Disks

- Nonvolatile memory used like a hard drive
  - Many technology variations
- Can be more reliable than HDDs
- More expensive per MB
- Maybe have shorter life span
- Less capacity
- But much faster
- Busses can be too slow and limit throughput -> connect directly to the system bus
- No moving parts, so no seek time or rotational latency

Magnetic Tapes

- Was early secondary-storage medium
- Relatively permanent and holds large quantities of data
- Slow access time
  - Random access ~1000 times slower than disk
- Mainly used for backup, storage of infrequently-used data, transfer medium between systems
- Kept in spool and wound or rewound past read-write head
  - Once data under head, transfer rates comparable to disk
- 20-200GB typical storage
- Common technologies are 4mm, 8mm, 19mm, LTO-2 and SDLT

13

Disk Attachment

Attached locally to the host computer

- accessed through I/O ports of the controller talking to I/O busses
  - EIDE, ATA, SATA, USB, Fibre Channel, SCSI

Network-Attached Storage

- Storage (disks, tapes) attached to and accessed through a general-purpose network

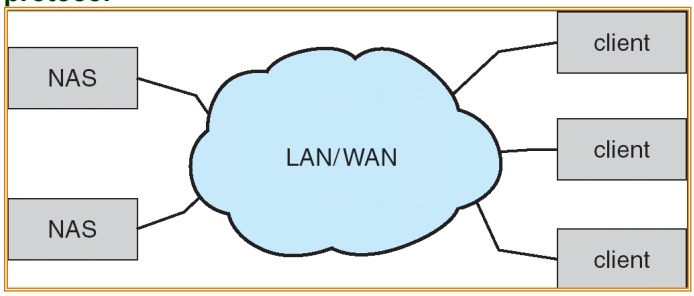
Storage Area Network

- Specialized network dedicated to storage

14

Network-Attached Storage

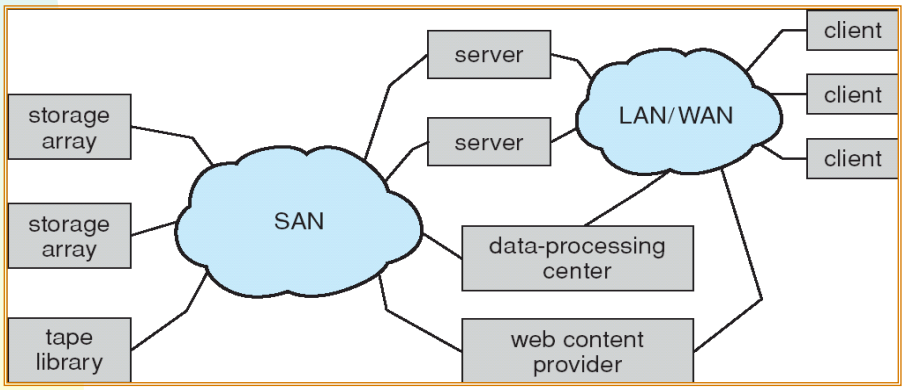
- Network-attached storage (NAS) is a device that makes storage available over a network rather than over a local connection (such as a bus)
- NFS and CIFS are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage
- New iSCSI protocol uses IP network to carry the SCSI protocol

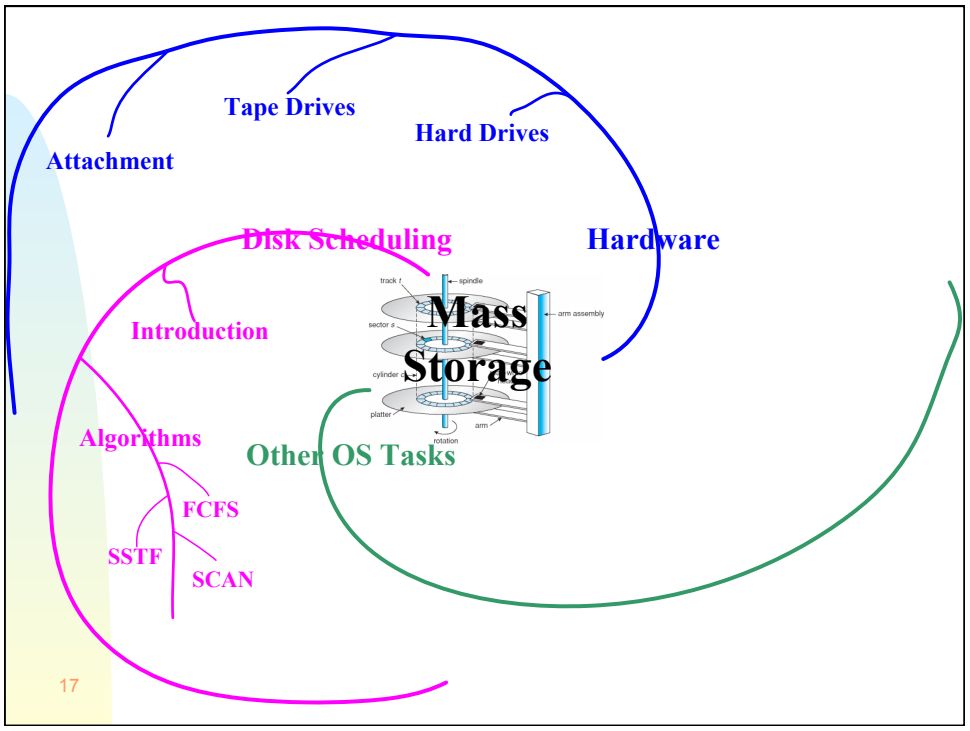


15

Storage Area Network

- Common in large storage environments (and becoming more common)
- Multiple hosts attached to multiple storage arrays - flexible





Disk Scheduling - Introduction

- A read request has arrived to the File System …
- From the CPU point of view, it takes forever to execute
- So, tell the HD to read the data and schedule a different process
- Can that process request a disc I/O?
  - Easily, why not?
- Therefore the disc I/Os can pile up even though they appear synchronous to each process
- Who should queue and process them?
  - The disk controller (HW on the disk)
  - The OS
- In reality, both do
  - The disk HW knows better about physical layer
  - The OS knows better about priorities

18

Disk Scheduling

We have a queue of disk I/O requests

Each request references some cylinder

- Changing cylinders involves head movement

It makes a lot of sense to examine the queue and choose the order of executing the operations in a way that minimizes head movement

Several algorithms exist to schedule the servicing of disk I/O requests.

- FirstComeFirstServe – keep it simple
- ShortestSeekTimeFirst – this is the age of greed, no?
- SCAN/elevator
  - C-SCAN
  - (C-)LOOK

19

Disk I/O Queue

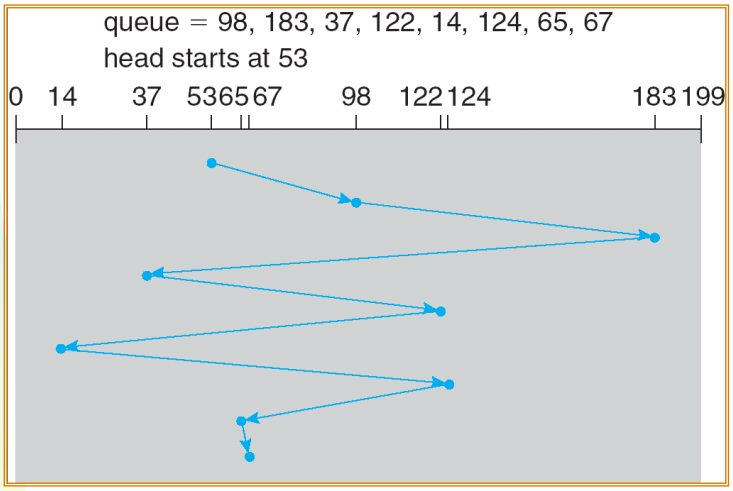
- Shall study different methods using an I/O waiting queue containing requests:

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

- Each number represents a cylinder number.
- Must also known the starting cylinder: 53
- What order of requests will minimize the total seek time (movement of head to cylinders).
- Simple hypothesis: moving 1 cylinder takes 1 time unit.

20

FCFS



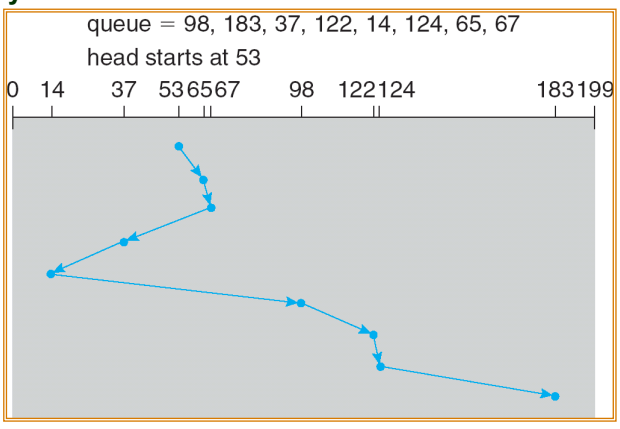
In this example: 640 cylinders of head movement

In general: Can be quite inefficient

21

SSTF

- **Selects the request with the minimum seek time from the current head position**
- **Can be seen as a form of SJF scheduling**
- **Illustration shows total head movement of 236 cylinders.**



22

SSTF

Looks much better then FCFS!

Any problems?

- **What if there is steady stream of requests for cylinders 100-150, and a lone request for cylinder 20?**
  - **The head will never leave range 100-150**
- **May cause starvation of some requests.**

Is it optimal?

- **SJB was…**
- **But SSTF is not:**
  - **Head: 10, Queue: 12,5,20**
  - **SSTF: 10 -> 12 -> 5 ->20, total movements: 2+7+15=22**
  - **Optimal: 10->5->12->20, total movements: 5+7+8 = 20**

23

SCAN

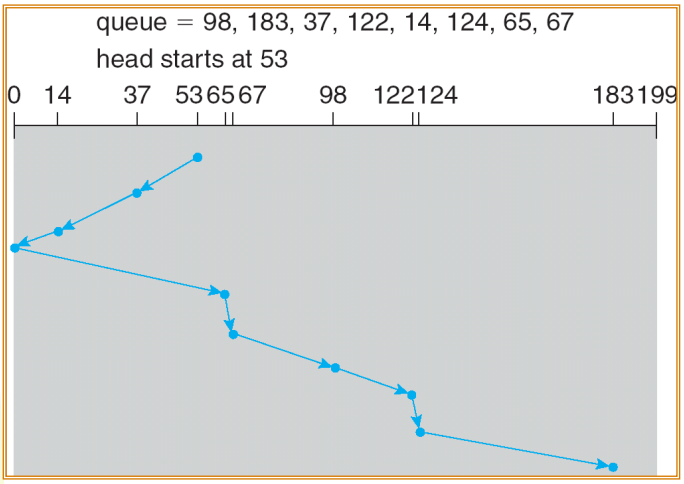
Goal: **Avoid starvation, while still being efficient**

Idea:

- **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.**
- **Starvation is avoided, because we scan the whole disk**
- **Performance should also be reasonably good, as zig-zagging is avoided (i.e. queue: 1,200,2,199,3,198,4,197; requests will be serviced in one pass)**
- **Also called the *elevator algorithm.***
- **Problems**
  - **Not much work to do when reversing directions, since requests will have accumulated at the other end of the disk.**
  - **Waste time moving to the end of the disk when have serviced the last request.**

24

SCAN (Cont.)



- Illustration shows total head movement of 208 cylinders.

25

C(ircular)-SCAN

First problem with SCAN:

- **Not much work to do when reversing directions, since requests will have accumulated at the other end of the disk.**

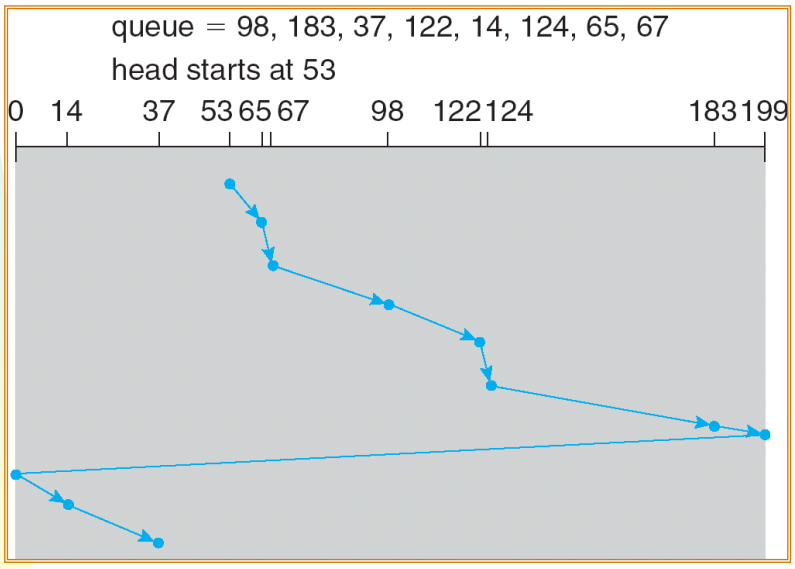
Idea:

- **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.**
- **The return trip is relatively fast, as it does not need to accelerate/decelerate for each request**

Treats the cylinders as a circular list that wraps around from the last cylinder to the first one.

26

C-SCAN (Cont.)

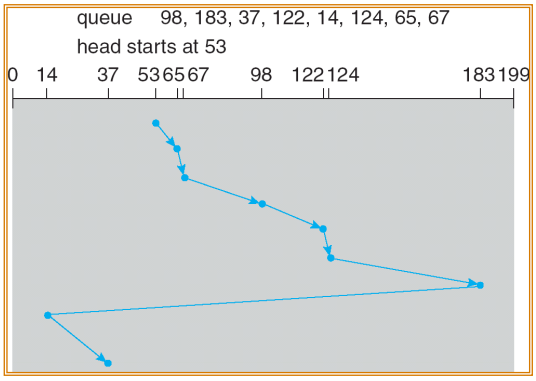


C-LOOK

SCAN Problem 2: Waste time moving to the end of the disk when have serviced the last request.

Can we optimize? Yes!

- **Move the head to the cylinder referred by a request that is closest to the ends of the disk.**



Selecting a Disk-Scheduling Algorithm

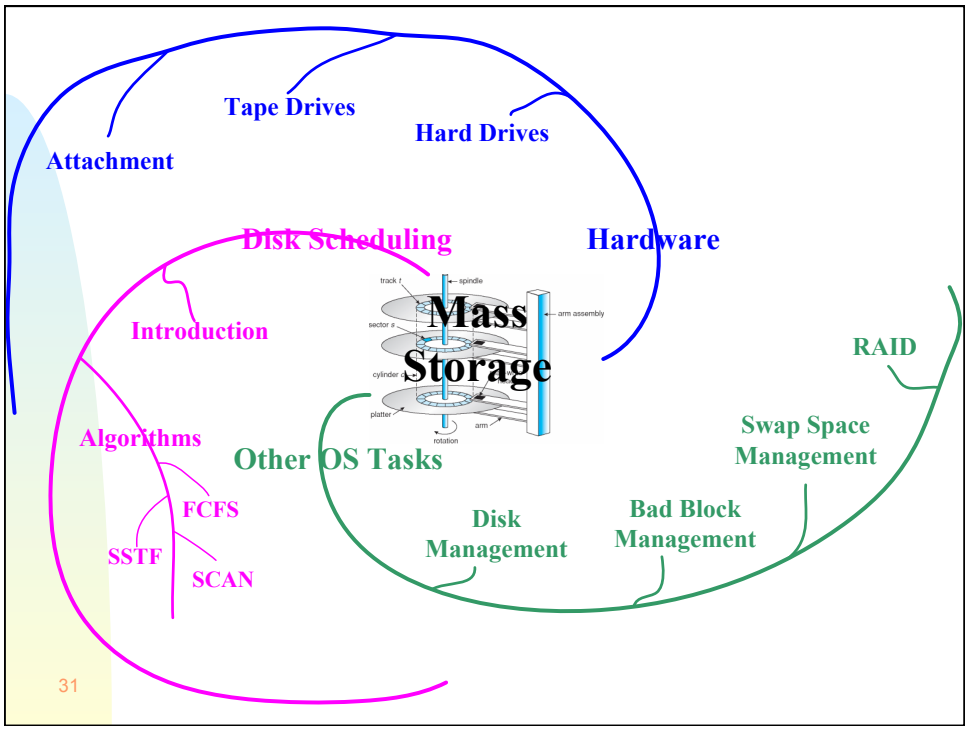
- Performance depends on the number and types of requests.
  - SSTF is common and has a natural appeal
    - If the load is light, there will be an idle time to serve the potentially starved requests
  - SCAN and C-SCAN perform better for systems that place a heavy load on the disk.
  - With very light loads, all algorithms perform equally well
- Requests for disk service can be influenced by the file-allocation method.
  - Files store contiguously limit head movement when accessed.
  - Locating directory entry on middle cylinder, and file contents close to index blocks, reduces movement
  - Caching directories and index blocks helps reduce disk access

29

Who can do scheduling?

- If I/O performance were the only consideration the OS could turn over the scheduling over to the disk controller.
  - Disk controller could increase performance by scheduling for rotational latency.
  - BUT, some I/O requests have higher priority
    - Demand paging has higher priority over application I/O
    - Order of disk writes may increase robustness in the face of system crashes: writing inodes and free list more important than writing the data
  - The OS can choose to do its own disk scheduling

30

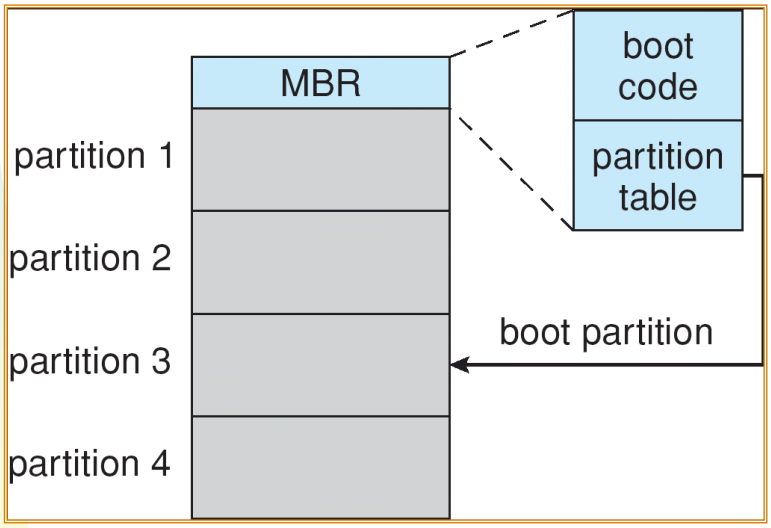


Disk Management

- *Low-level formatting*, or *physical formatting*
  - divide the disk into sectors the controller can read
  - initialize each sector (header, trailer)
  - Comes from factory
- *Partitioning*
  - logically *partition* the disk into one or more groups of cylinders, each of them having its own FS.
  - Set the boot partition
- *Logical formatting* or "making a file system"
  - Write the FS data on disc
  - i.e. write FAT, the root directory, inodes, …

32

Booting from a Disk in Windows 2000



33

MBR= Master Boot Record

Bad Block Management

There are tens and hundreds of millions of blocks on a HD

Bad blocks do happen

What to do about them?

- **Have some spare blocks and when you detect a bad block, use the spare block instead**
  - **i.e. the HD has nominally 100 blocks, but was fabricated with 110, 0..99 are used, 100-109 are spare**
  - **If block 50 fails, the HD controller (after being told by the OS) will use block 100 to store the logical block 50**
- **But that messes up the efficiency of the disc scheduling algorithms!**
- **Don't panic! Just have spare blocks in each cylinder** (sector sparing) **or use** sector slipping

34

Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
- Where is the swap-space on the disc?
  - As a part of the normal file system
    - good: Simple, flexible
    - bad: SsssLOoooWwww
  - in a separate disk partition
    - Using its own, optimized algorithms and structures
    - i.e. fragmentation not such a big problem – on boot starts anew, but really want fast sequential access
    - Good: faster
    - Bad: might waste space

35

Swap-Space Management

- When to allocate space in the swap space?
  - When the process starts/the virtual page is created
  - When the page is replaced
- What to remember?
  - Kernel uses *swap maps* to track swap-space use.

36

RAID Structure

Redundant Array of Independent/Inexpensive Discs

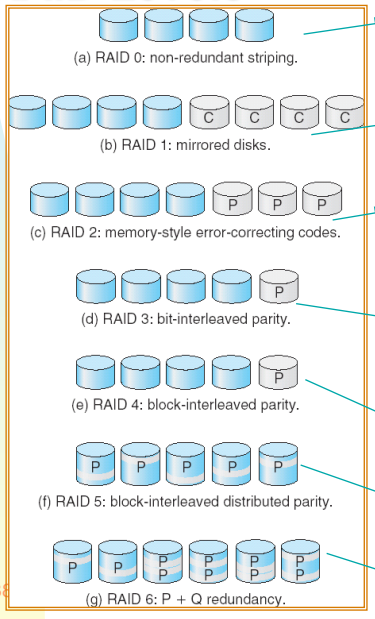
Goals: **performance and reliability**
Idea: **Combine and intelligently use several discs**
Techniques:

- striping: **(increases performance through parallelism)**
  - Bit interleaving
  - Bloc interleaving
- Redundancy for reliability
  - mirroring: **(reliability through replication)**
  - ECC/parity bits: **reliability**

There are six RAID levels/standards
The levels can also be combined

37

RAID Levels



Performance, low reliability

High reliability, lots of wasted space

Takes less disks than RAID 1

Little storage overhead, performance not as good as with other levels

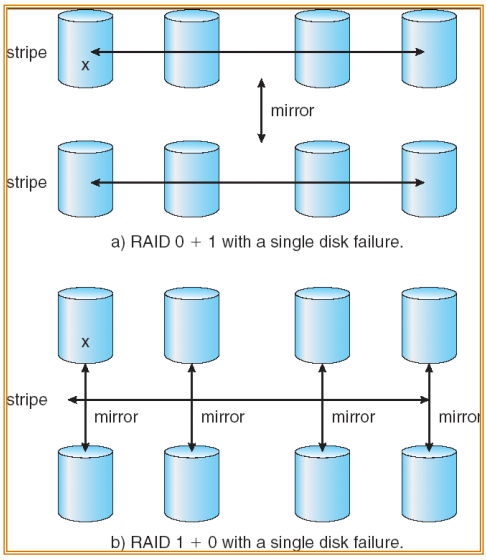
Good, but the parity disc bottleneck

Good, used in practice

Expensive, should be more reliable

38

RAID (0 + 1) and (1 + 0)



If a single disk fails, 0+1 one stripe fails, 1+0 works