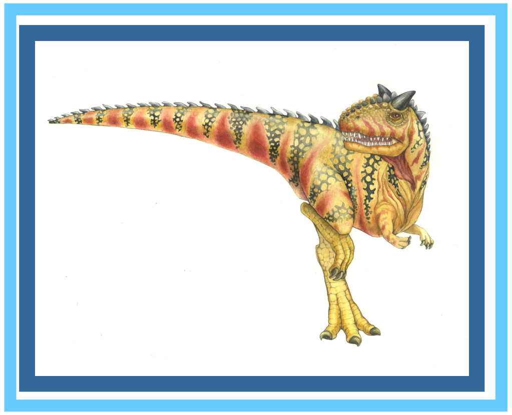
Chapter 12: Mass-Storage Systems



Chapter 12: Mass-Storage Systems

- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure
- Disk Attachment
- Stable-Storage Implementation
- Tertiary Storage Devices
- Operating System Issues
- Performance Issues

Objectives

- Describe the physical structure of secondary and tertiary 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, including RAID and HSM

Overview of Mass Storage Structure

- **Magnetic tape**
  - Was early secondary-storage medium
  - Relatively permanent and holds large quantities of data
  - Access time slow
    - And 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
- Today – primary mass storage structure is **magnetic disks**
  - Tape used for backup
  - Depending on capacity required, CD/DVD also sometimes used for backup

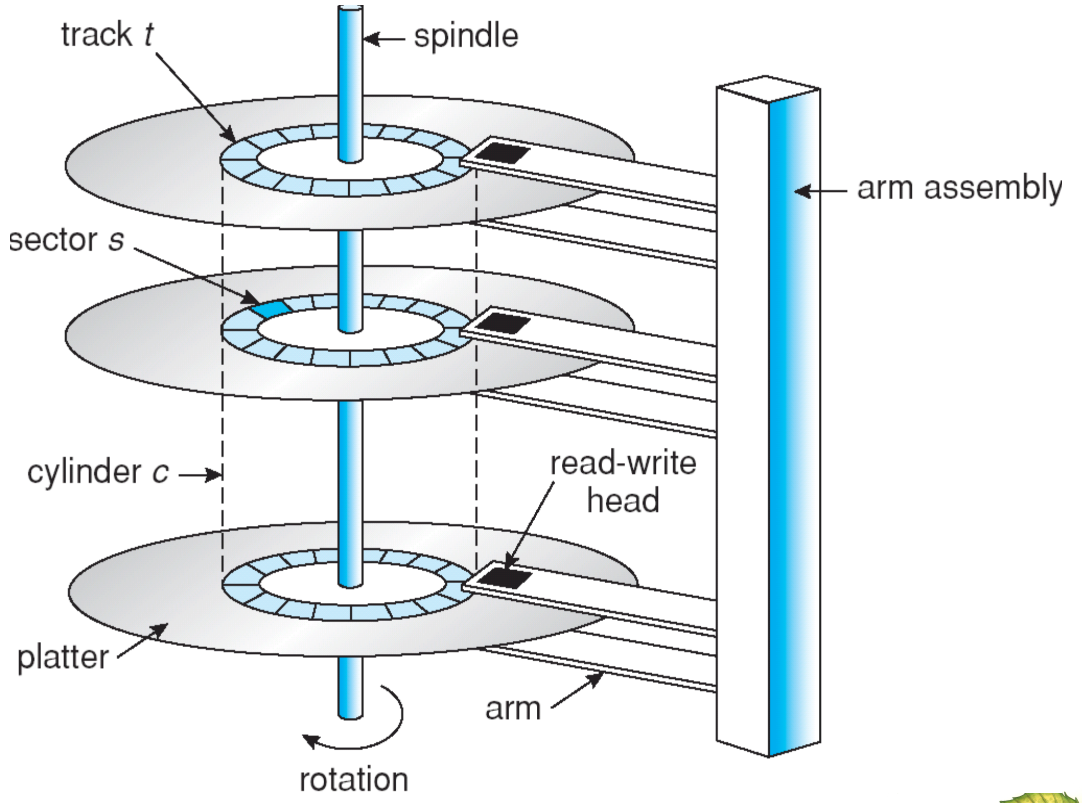
Overview of Mass Storage Structure (Cont.)

- **Magnetic disks** provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 200 times per second
  - **Transfer rate** is rate at which data flow between drive and computer
  - **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
  - **Head crash** results from disk head making contact with the disk surface
    - That's bad
- Disks can be **removable**
- Drive **attached** to computer via **I/O bus**
  - 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

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 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.

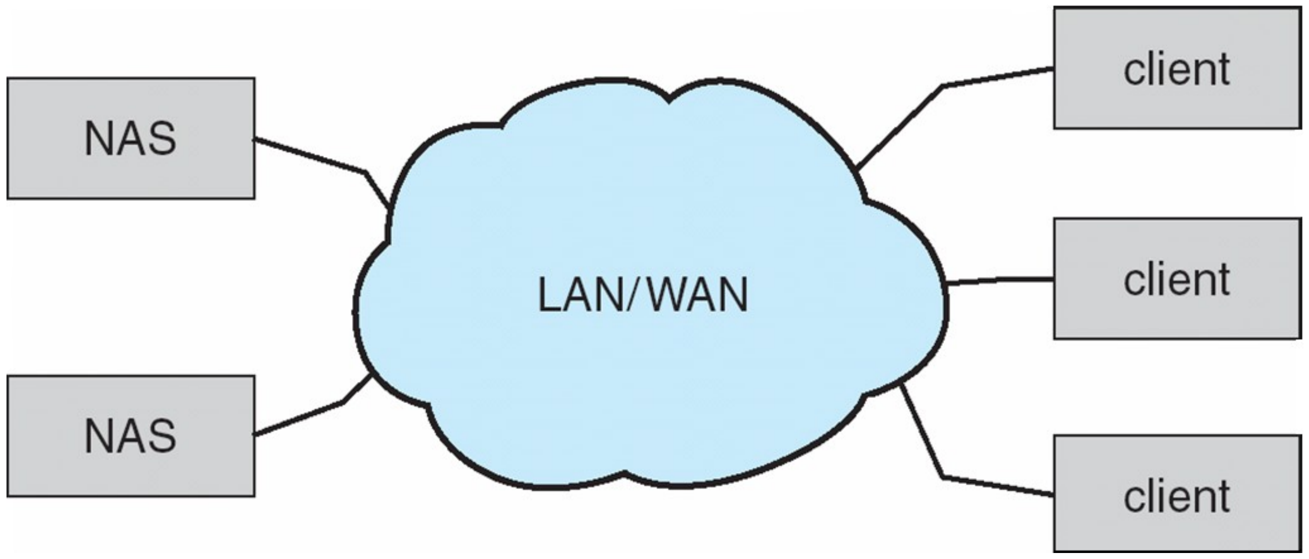


Disk Attachment

- Host-attached storage **accessed through I/O ports** talking to I/O busses
- **SCSI** itself is a type of bus, up to 16 devices on one cable, **SCSI initiator** requests operation and **SCSI targets** perform tasks
  - Each target can have up to 8 **logical units** (disks attached to device controller
- **Fibre Channel** (FC) is high-speed serial architecture
  - Can be switched fabric with 24-bit address space – the basis of **storage area networks** (**SAN**s) in which many hosts attach to many storage units
  - Can be **arbitrated loop** (**FC-AL**) of 126 devices

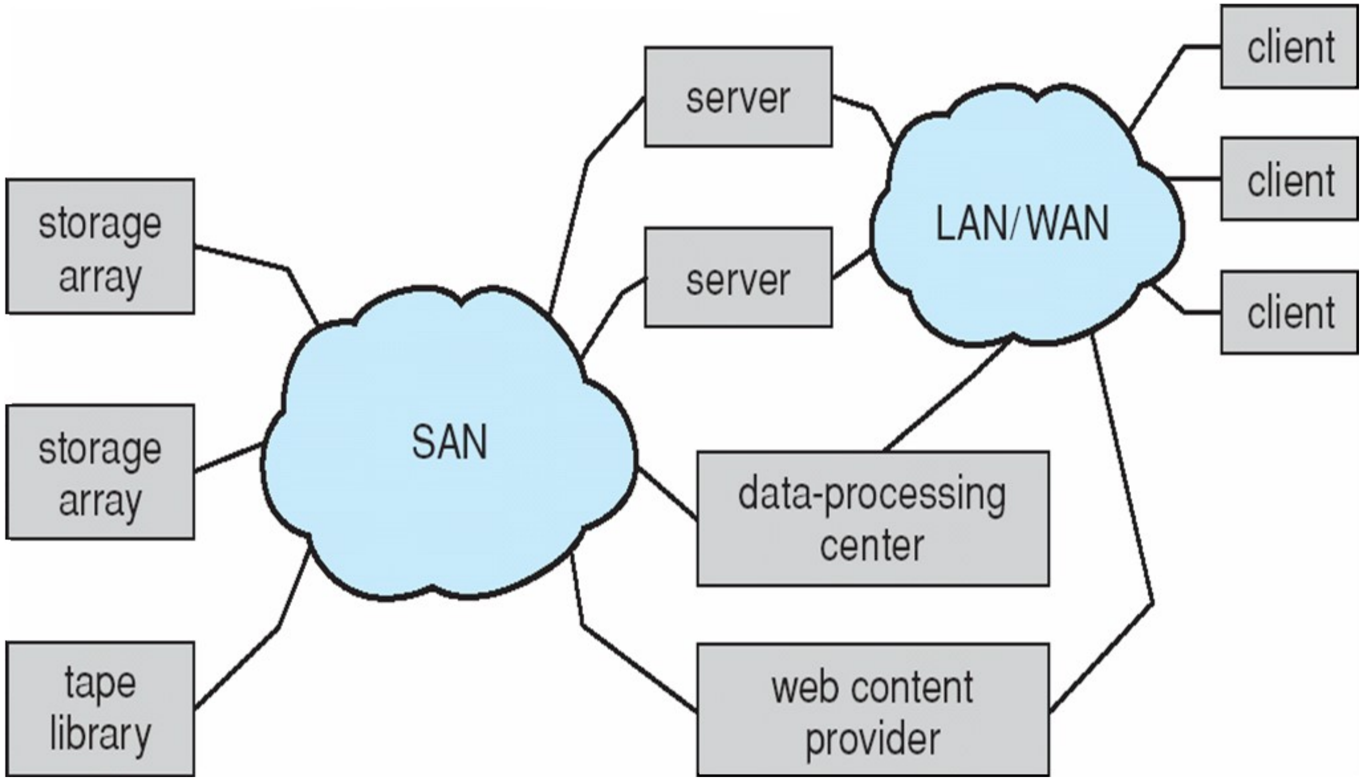
Network-Attached Storage

- **Network-attached storage** (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
- Although often “mounted” as a filesystem, is **really a remote server**
  - **NFS and CIFS are common** protocols (with NFS’s shortcomings)
  - Implemented via **remote procedure calls** (RPCs) between host and storage
  - New **iSCSI** protocol uses IP network to carry the SCSI protocol



Storage Area Network

- "Super" version of NAS servers – SAN masks complexity – redundancy
  - Large system / set of systems attached to nothing but storage (arrays)
- Common in large storage environments (and becoming more common)



Disk Scheduling

- Goal of OS / controller is to **minimize service time** (provide quickest response time) – by using hardware most efficiently
- Time factors in reading (or writing) a disk sector are:
  - ***Seek time*** is the time for the disk are to move the heads to the cylinder containing the desired sector.
  - ***Rotational latency time*** is the additional time waiting for the disk to rotate the desired sector to the disk head.
  - ***Transfer time*** is the time to read data and move it to the system
- **Cannot** do much about **transfer** and **latency** time (HW)
- **CAN** try to schedule reads/writes to minimize **seek time**
  - Do this by **minimizing seek distance**
  - If system / disk is reasonably busy, can have significant impact
  - Scheduling **can be done by system, disk, or in between**

Disk Scheduling (Cont.)

- 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**

(this is where the head is currently positioned)

- **Evaluate** algorithms on how well they **minimize seek time**, on **fairness**, and on **complexity**

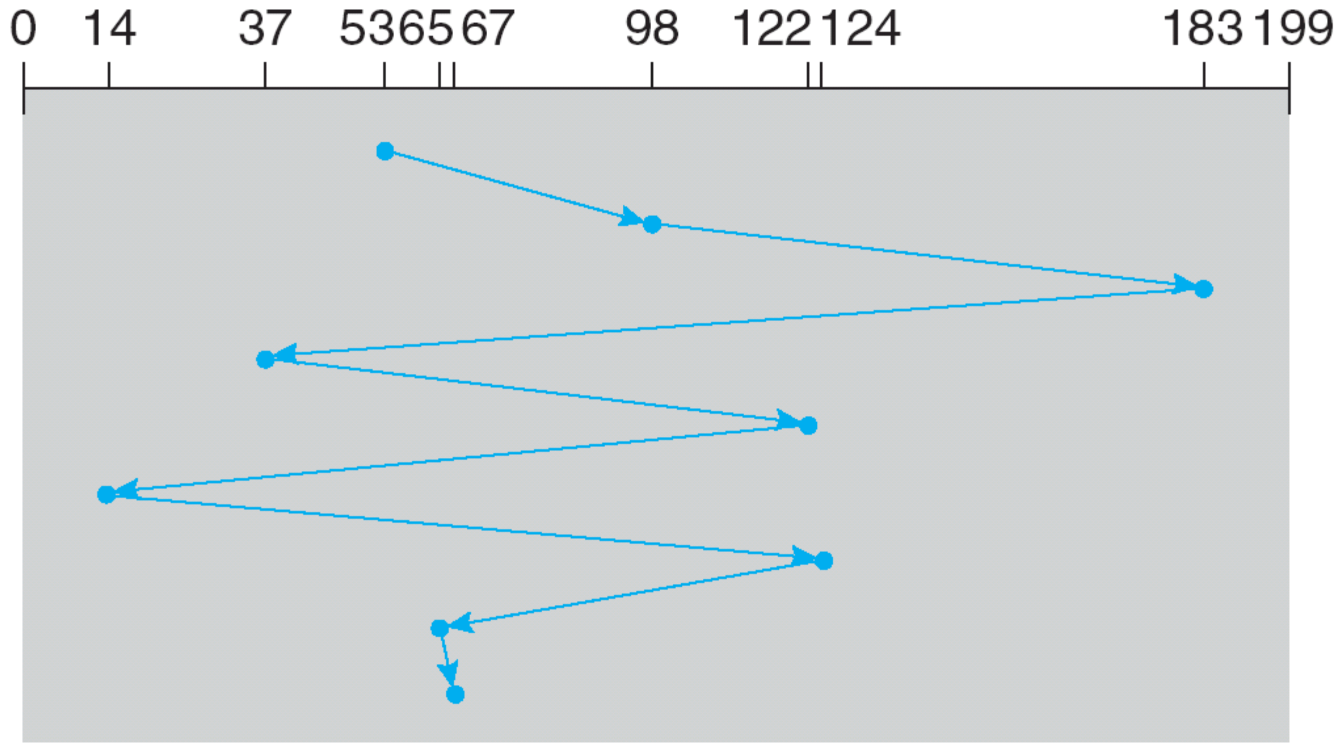
FCFS Scheduling

- **First-come-first-served scheduling** – FCFS
  - Serve disk read / write requests **in order in which they arrive**
- **Simple, fair, OK if utilization is low** (i.e., disk is not very busy)
- BUT – it **often results in excessive arm movement**

FCFS

Illustration shows total head movement of 640 cylinders/tracks.

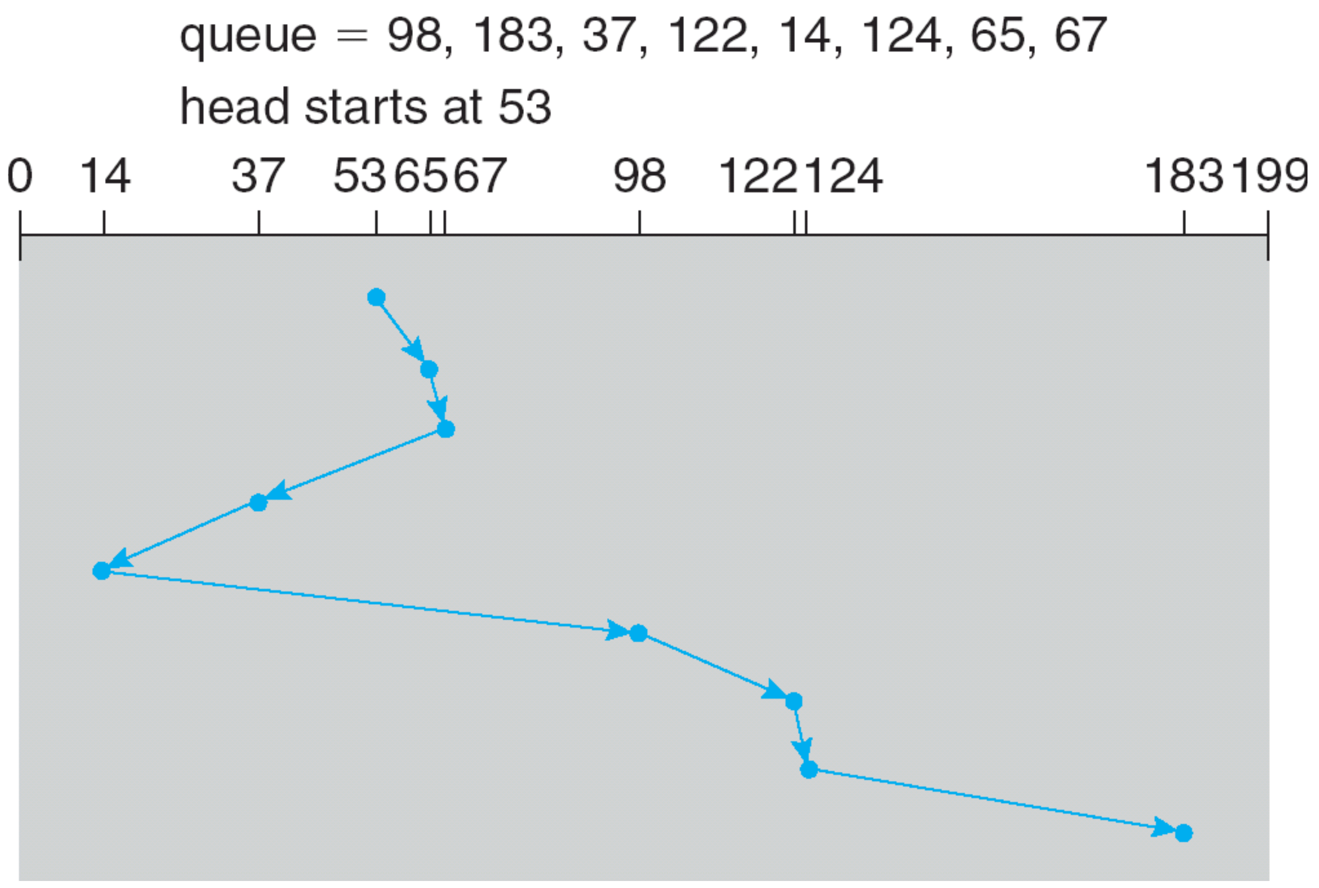
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



SSTF Scheduling

- **Shortest-seek-time-first** scheduling
  - The request chosen to be serviced next is the **one closest to the current head position**
  - **Similar in concept to SJF** process scheduling
- **PROBLEM** – may result in ***starvation***
- Can be **considerable improvement** over FCFS, but **not optimal** (because of possible starvation)
- Illustration shows total head movement of 236 cylinders.

SSTF (Cont.)



Total tracks moved: 236

SCAN

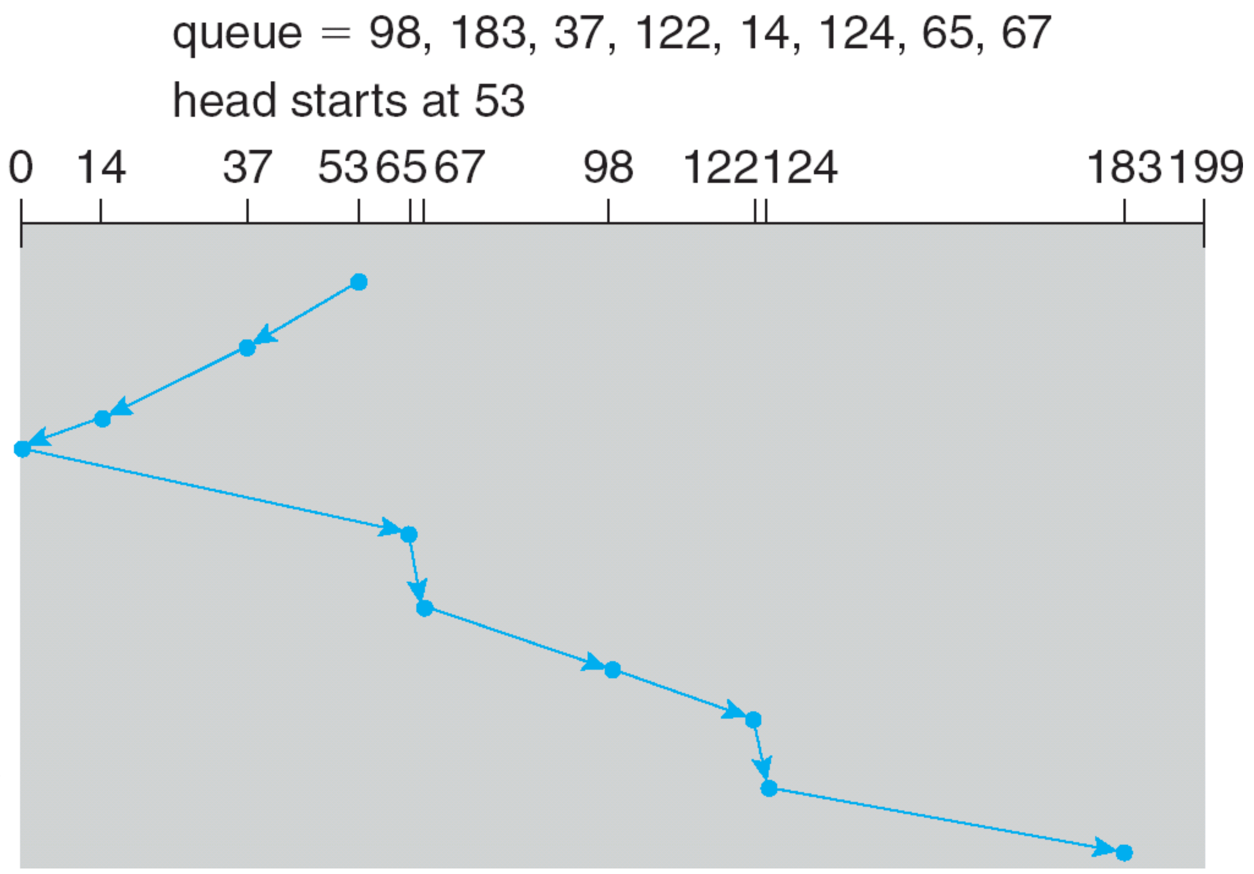
- The disk arm starts at one end of the disk, and moves toward the other end, servicing all possible requests along the way, then moves back toward the other end of the disk, doing the same
  - Note: it has to go all the way to the end of the disk (at each end) before it can turn around and start back
- Sometimes called the ***elevator algorithm*.**
  - Think of an elevator that must go to very top and very bottom floors before reversing directions

Your text here 1

- Can result in **uneven service time**, but **no starvation**
- Illustration shows total head movement of ~~208~~ cylinders.

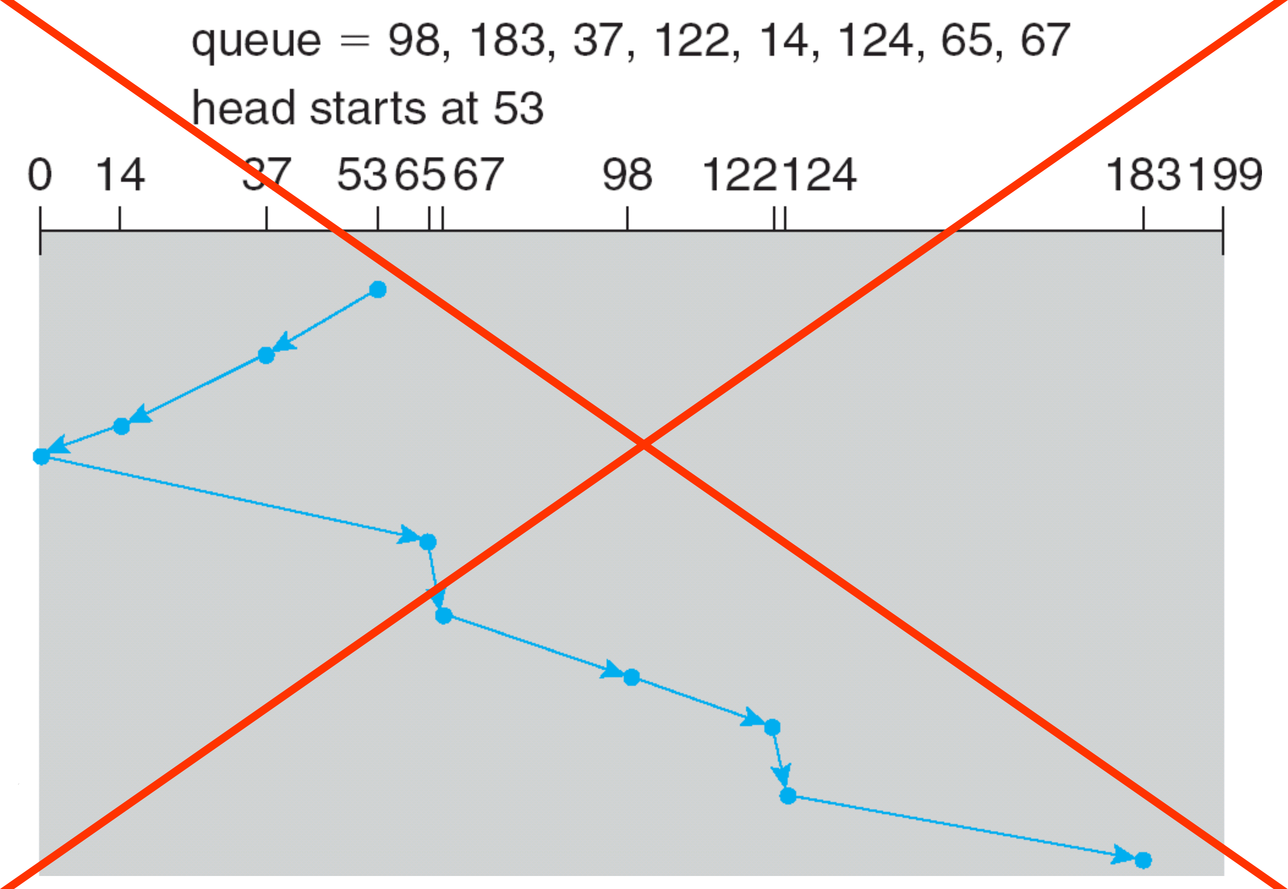
236

SCAN (Cont.)



Total tracks moved: ~~208~~ 236

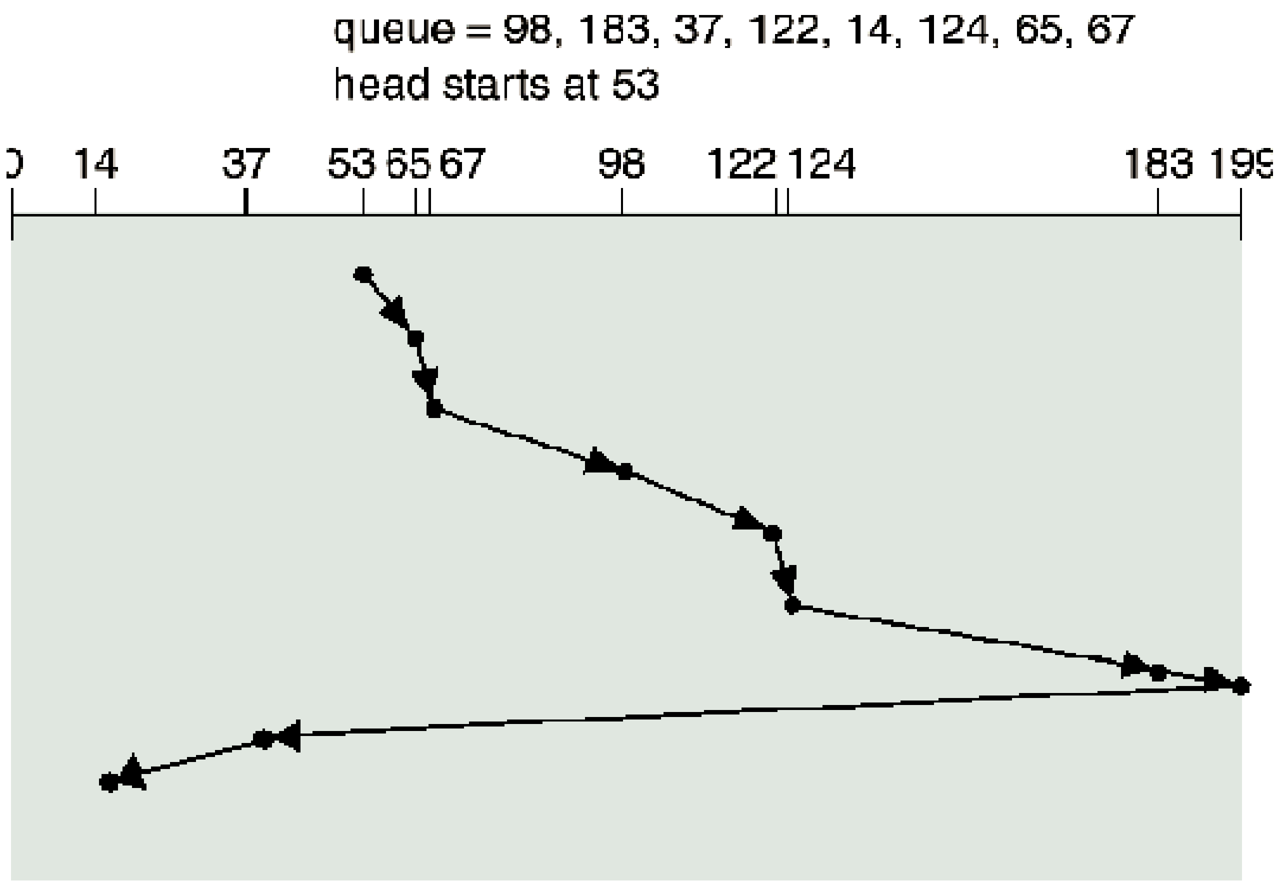
SCAN (Cont.)



Total tracks moved: ~~208~~ 236

NOTE – this example starts in the opposite direction from all the others, so is not a good comparison

SCAN (Cont.)



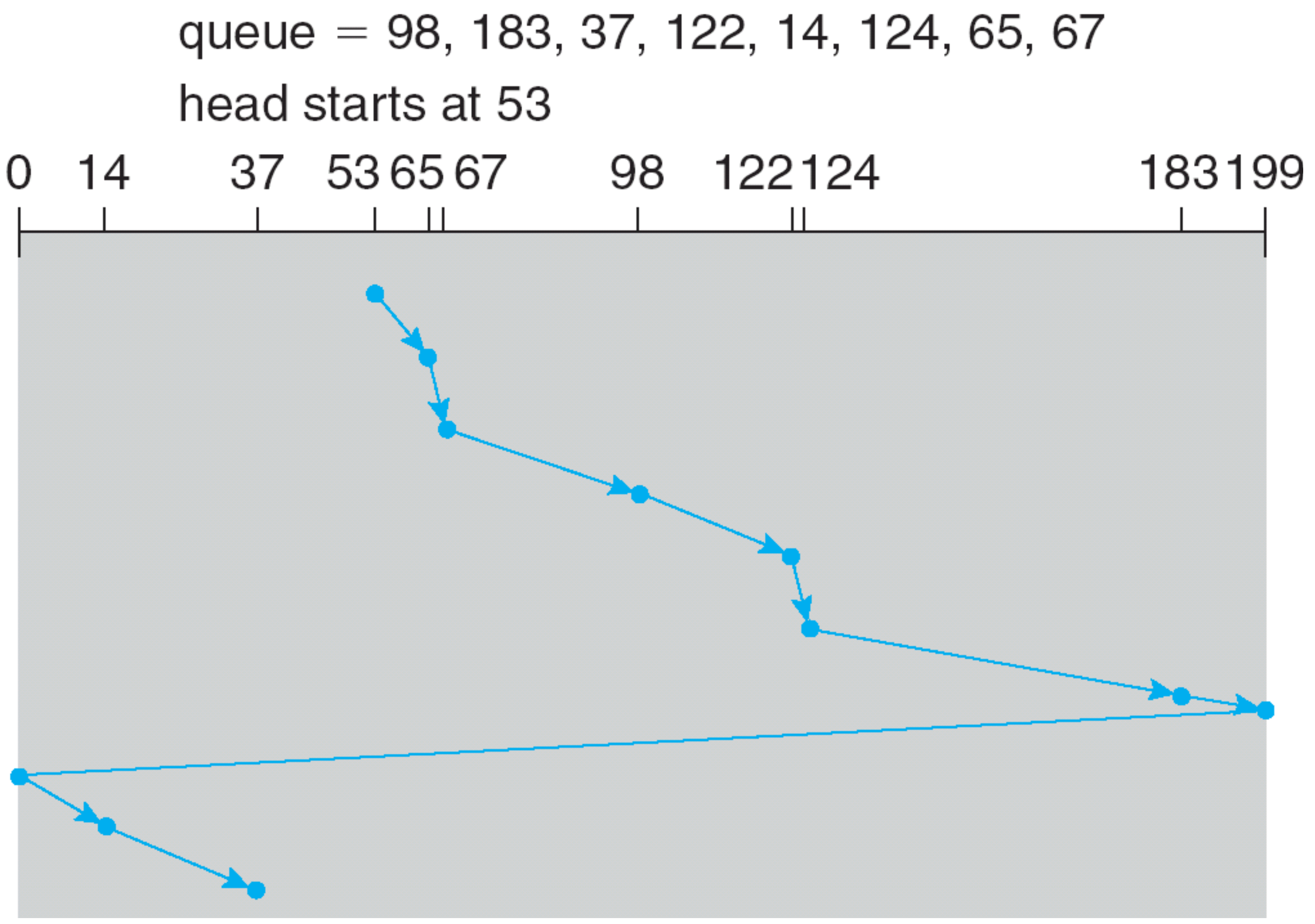
Total tracks moved: 331

Note – this example goes in SAME DIRECTION as all the others

C-SCAN

- **Circular 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.
  - **Only reads/writes when moving one direction**
  - Like SCAN, **goes from first to last sector of disk**, even though there may be no requests that far
- Provides a **more uniform wait time** than SCAN.
- **Treats the cylinders as a circular list** that wraps around from the last cylinder to the first one.

C-SCAN (Cont.)



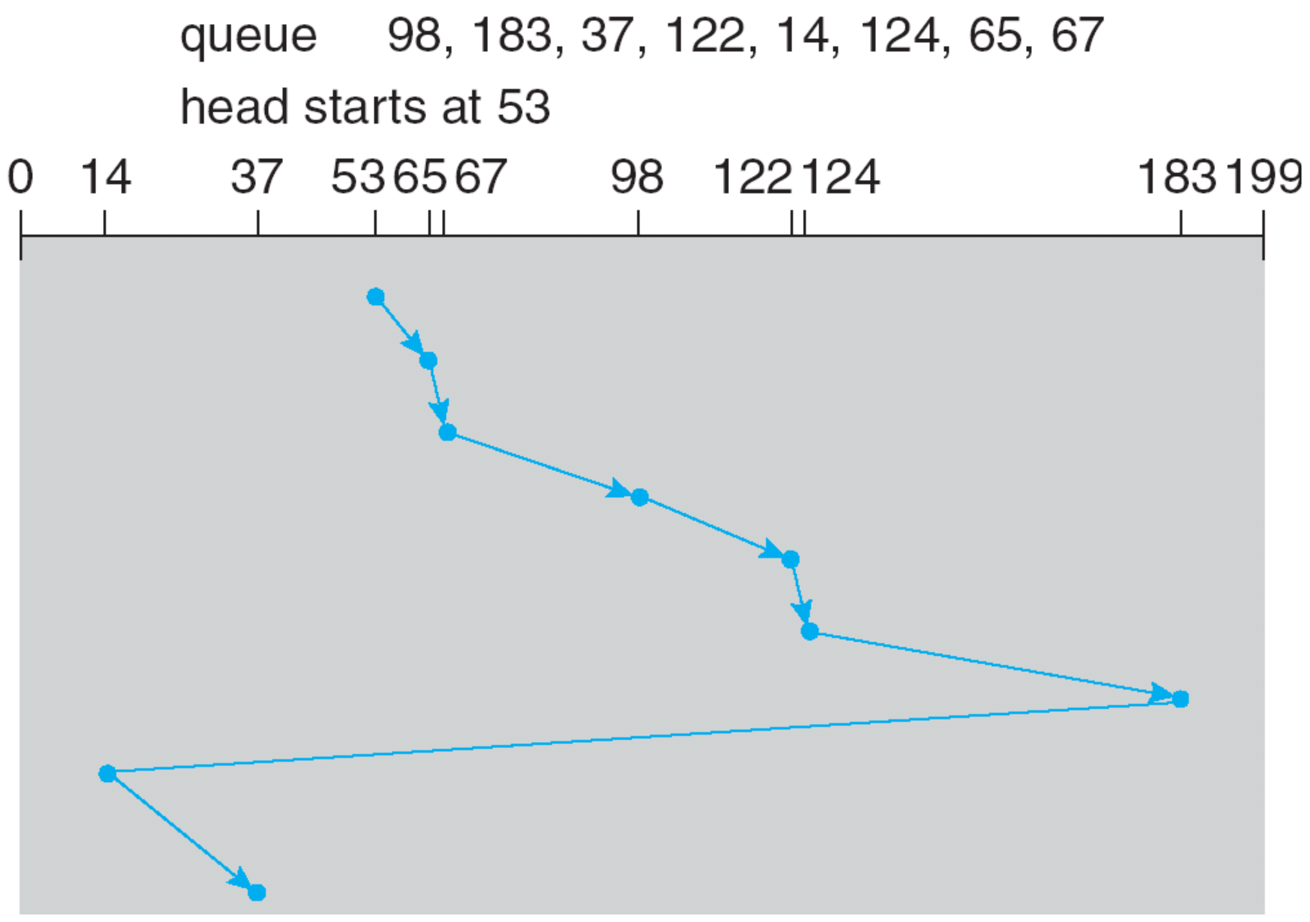
Total tracks moved: 382

NOTE: when counting tracks moved, must also count while moving from one end to other, even if doing no I/O

LOOK Scheduling

- Like SCAN, but **arm only goes as far as the last request in each direction** (i.e., does not go all the way to each end of the disk)
- Saves arm movement (and time)
- **C-LOOK** has a similar relationship to C-SCAN

C-LOOK (Cont.)



Total tracks moved: 322

Comparison of Scheduling Algorithms

TOTAL TRACKS MOVED

FCFS – 640 ✓

SSTF – 236 – smallest, although some requests may starve ✓

236

~~**SCAN** – 208 – *started opposite direction from others*~~

SCAN – 331 – going same direction as all the others ✓

C-SCAN – 382 ✓

LOOK – 299 ⟵ **Best overall in this example** ✓

C-LOOK – 322 ✓

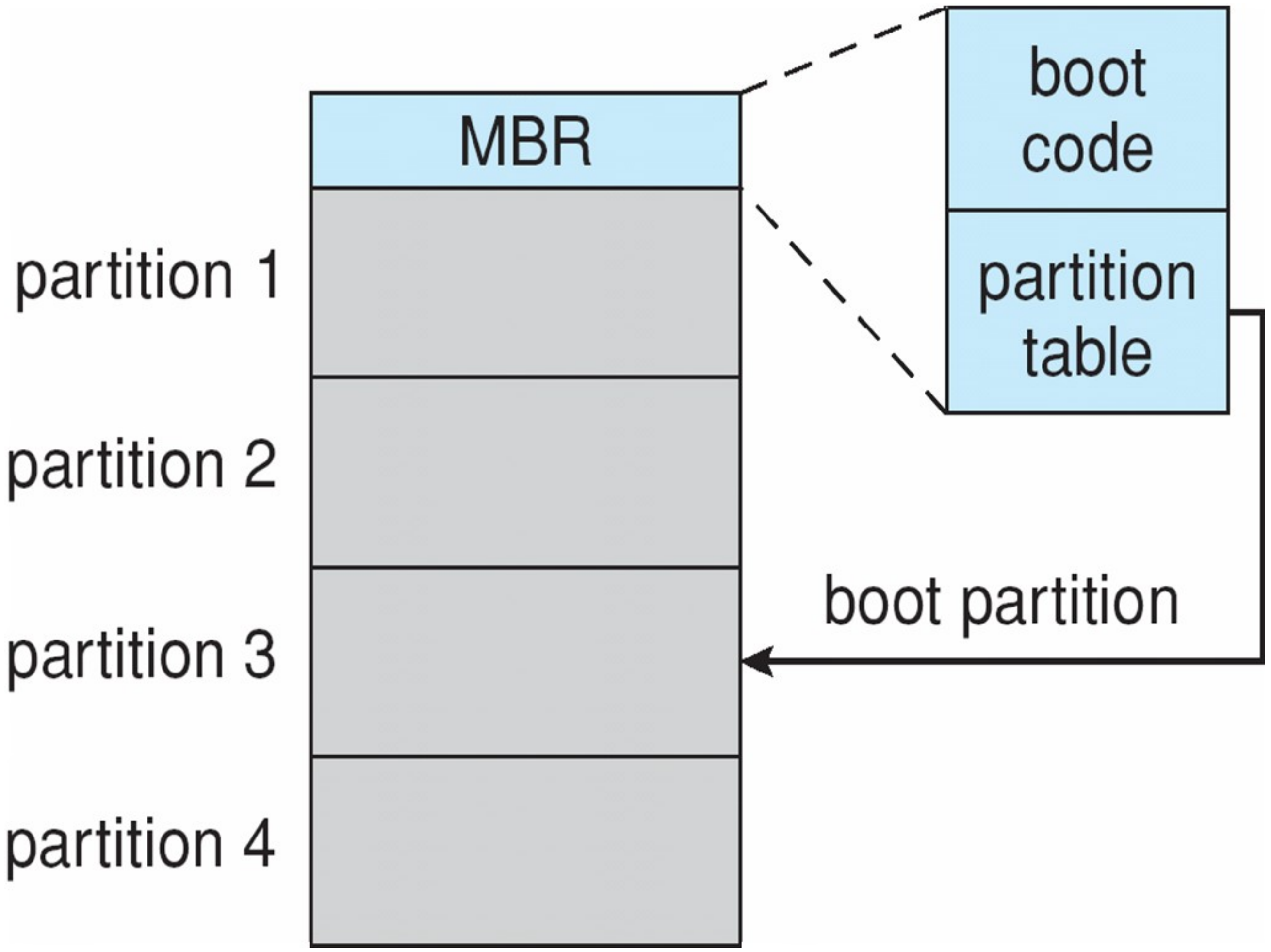
Selecting a Disk-Scheduling Algorithm

- **SSTF** is common and has a natural appeal
  - But **starvation** possible
- Text says that SCAN and C-SCAN perform better than SSTF for systems that place a heavy load on the disk
  - But only because they solve starvation problem – **they don’t really perform better**
- Performance depends on the number and types of requests.
- Requests for disk service can be influenced by the file-allocation method.
- 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 (if modify SSTF to avoid starvation).

Disk Management

- ***Low-level formatting*, or *physical formatting*** — Dividing a disk into sectors that the disk controller can read and write.
- To use a disk to hold files, the operating system still needs to record its own data structures on the disk.
  - ***Partition*** the disk into one or more groups of cylinders.
  - ***Logical formatting*** or "making a file system".
- **Boot block** initializes system.
  - The bootstrap is stored in ROM.
  - ***Bootstrap loader*** program.
- Methods such as ***sector sparing*** used to handle **bad blocks** (provide ***alternate sectors*** to replace bad ones)
  - Logically remove bad sectors or assign alternate sectors

Booting from a Disk in Windows 2000

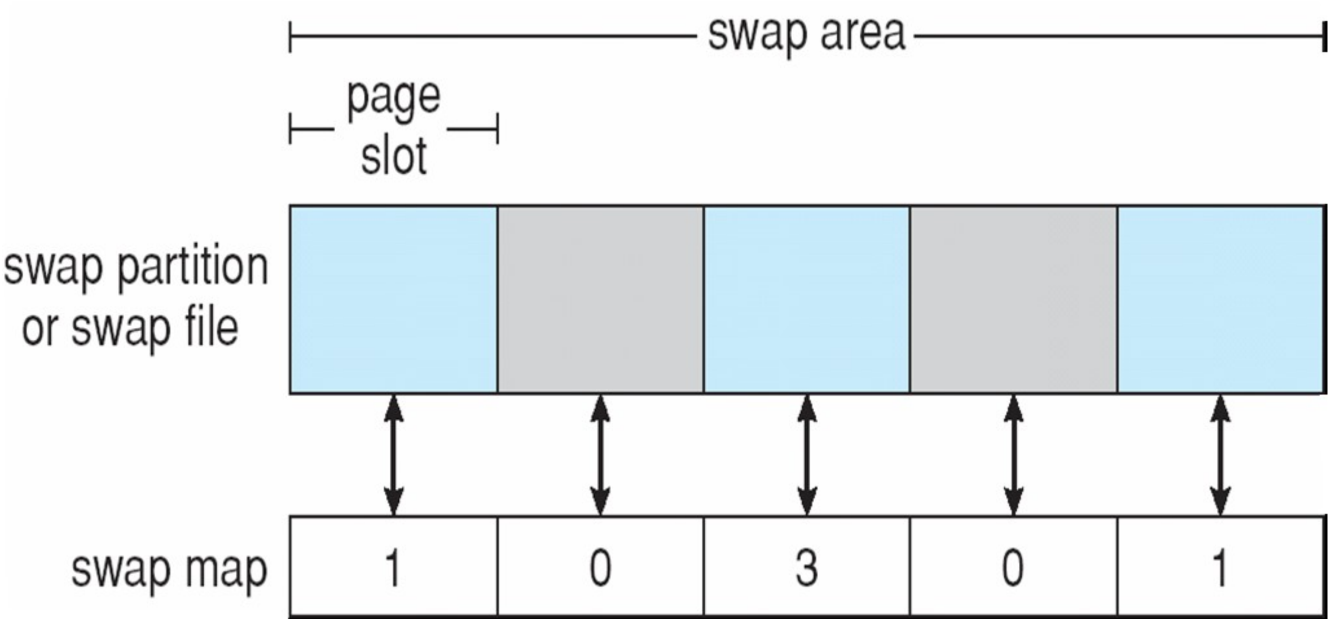


MBR = “Master Boot Record”

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.**
- Swap-space management – **various techniques**
  - 4.3BSD allocates swap space when process starts; holds *text segment* (the program) and *data segment.*
    - Kernel uses *swap maps* to track swap-space use.
  - Linux and late versions of Solaris only allocate swap space when a page is forced out
  - Linux supports multiple swap partitions, and suggests that they be at least 2X size of main memory (although there is some debate)
- **OS/400 does not use separate swap space** / swapping drive – all drives are paging devices in single-level store

Data Structures for Swapping on Linux Systems



- Swap space only allocated (for a process) when page first swapped out, not when process created
- Swap space is series of 4K page slots, used to hold swapped pages
- Swap map – integer counters, show number of processes using page
  - If = 0, then is available

RAID Structure

- **RAID** – **R**edundant **A**rray of **I**ndependent **D**isks
  - Multiple disk drives provides **reliability** via **redundancy**.
- Several improvements in disk-use techniques involve the use of multiple disks working cooperatively.
- **Disk striping** uses a group of disks as one storage unit.
- RAID schemes improve performance and / or improve the reliability of the storage system by **storing redundant data.**
  - ***Mirroring*** or ***shadowing*** keeps duplicate copy of each disk.
  - ***Block interleaved parity*** requires much less extra space for redundancy.

Redundancy for Improved Reliability

Why do we care?

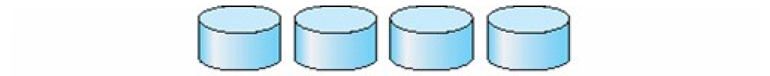
- **Systems becoming larger and larger** – now can have hundreds of disks attached, storing terabytes of data.
  - **Recovery time following a failure is much longer** for large systems than for smaller systems
  - Large systems are often **more mission critical** than small system
- Even with highly reliable hardware, as the number of instances of the hardware increases, the **probability of failure increases**
  - E.g., if mean time between failure for a disk drive is one failure every five years, if have 100 disks on the system, probability is that will have a disk failure every three weeks
  - This is not good for your bank (or hospital, or inventory control system, or e-commerce system, or . . .)
  - And many systems now have hundreds of drives attached
- **Use data redundancy to reduce impact** of hardware failures

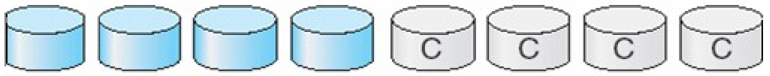
RAID Comes in Six Levels

- **Level 0** – non-redundant striping
  Data spread across all disks – no redundancy
- **Level 1** – mirroring
  Have duplicate copies of all disks
- Level 2 – memory-style ECC organization
  Store two or more extra bits to detect and correct errors
- Level 3 – bit interleaved parity organization
  Parity on one disk, no striping of data disks
- Level 4 – block interleaved parity organization
  Parity on one disk, data spread across n disks
- **Level 5** – block interleaved distributed parity
  Parity and data spread across all disks – quite common
- Level 6 – P+Q redundancy
  Stores extra parity to protect against multiple failures
- Level 0 + 1
  Spread data across all disks and mirror

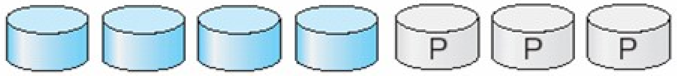
Levels 0, 1, & 5 most common, 6 starting to be used
Also work on 5 + 1

RAID Levels

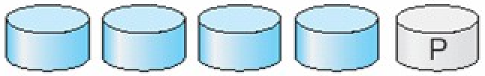

(a) RAID 0: non-redundant striping.



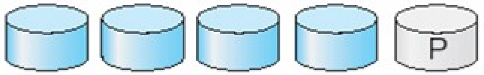
(b) RAID 1: mirrored disks.



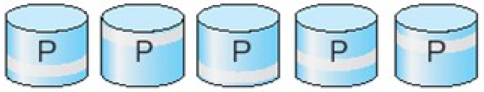
(c) RAID 2: memory-style error-correcting codes.



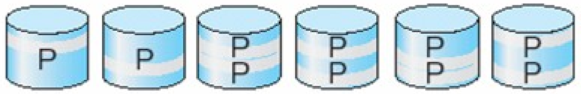
(d) RAID 3: bit-interleaved parity.



(e) RAID 4: block-interleaved parity.



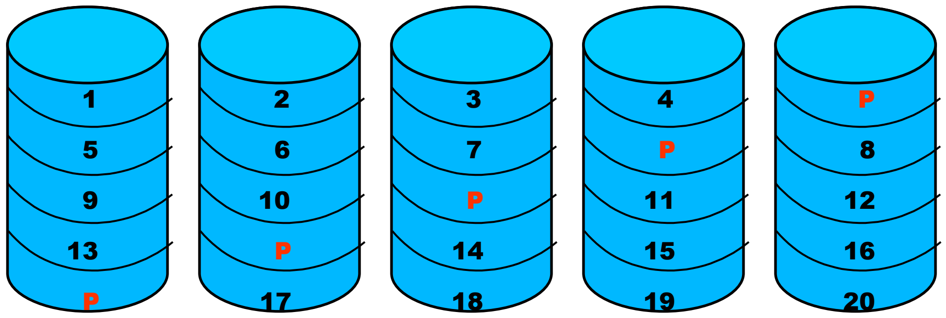
(f) RAID 5: block-interleaved distributed parity.



(g) RAID 6: P + Q redundancy.

RAID 5 Details

- Data and parity are **striped and interleaved** across a set of disks
  - **Requires one additional disk per parity set**
  - **Parity is spread across disks** in the set to avoid "hot spots"
  - Parity is the XOR of data in corresponding sectors on other disks of set

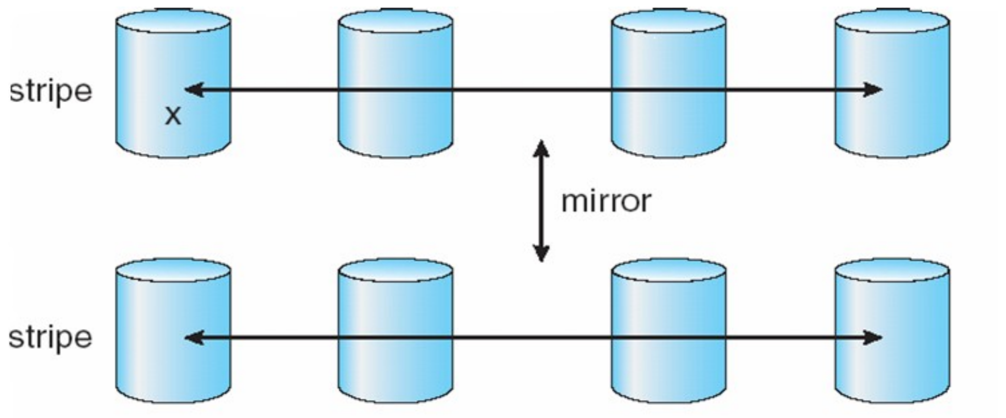


- If a disk in set fails, **can recreate data** by XORing together the corresponding sectors of remaining disks in the set
  - If two disks fail, lose all the data on the entire set
- **Reading data requires one I/O** (just read the data)
- **Writing data requires 4 I/Os plus system processing**
  - Read disk block that will be written, read corresponding parity block, XOR data block and parity block together, XOR result with data to be written to produce new parity value, write new data, write new parity

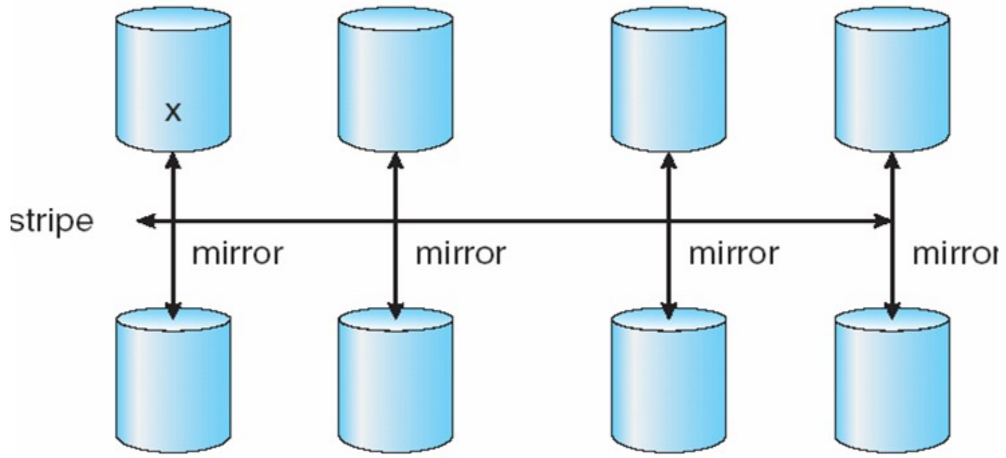
Selecting A Protection Level

- **RAID 0 (striping)** – popular for performance and ease of management
  no real protection from redundancy
- **RAID 1 (mirroring)** – provides best redundancy protection
  - But most expensive in terms of cost for extra disks
- **RAID 0 and RAID 1 combined** – best of both worlds
- Because the hardware required for **RAID 1 is expensive**, **RAID 5 is popular** as an alternative
  - However, **will not tolerate double failure in same RAID set** and often does not perform as well as RAID 1 (since a write requires two reads and two writes – data and parity – although caches help)
  - **Hot spares help avoid double failures**, but must have spare for each RAID set (and replace spare before a double failure occurs)
- **RAID 6 is an improvement**, since will tolerate a double failure without requiring 2x the disk space, but not widely available yet

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



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



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

Additional Data Protection Options

- Redundancy through **clustering**
- Maintain a **redundant copy of data** on another system in cluster
  - Use **remote journaling** or similar sort of replication
  - Or **remote mirroring**
- When primary copy of data fails, **switch to redundant copy** on different system in the cluster
- Can also use clustering to **protect against system failure** (not just disk failure)

Stable-Storage Implementation

- Information in ***Stable Storage*** is **never** lost
- Stable storage is **useful** or necessary for a number of scenarios
  - Write-ahead log scheme requires stable storage.
- **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 the stable data can be recovered after any failure during data transfer or recovery.
    - Successful completion
    - Partial failure
    - Total failure
    - Failures can occur anywhere in I/O hardware path

Tertiary Storage Devices

- **Extention** of secondary storage
  - Expand secondary storage – with tradeoffs
- **Low cost** is the defining characteristic of tertiary storage.
- Generally, tertiary storage is built using ***removable media***
- Tertiary storage is ***generally slower*** than secondary storage (disks)
- Common **examples** of removable media are floppy disks and CDs, DVDs, tape libraries, etc.; other types are available.

Removable Disks

- **Floppy disk** — thin flexible disk coated with magnetic material, enclosed in a protective plastic case.
  - Most floppies hold about 1 MB
  - Similar technology is used for removable disks that hold more than 1 GB.
- Some removable magnetic disks can be nearly as fast as hard disks, but they are at a **greater risk of damage** from exposure.

Removable Disks (Cont.)

- A **magneto-optical disk** records data on a rigid platter coated with magnetic material.
  - **Laser heat** is used to amplify a large, weak magnetic field to **record** a bit.
  - **Laser light** is also used to **read** data (Kerr effect).
  - The magneto-optical head flies much farther from the disk surface than a magnetic disk head, and the magnetic material is covered with a protective layer of plastic or glass; resistant to head crashes.
- **Optical disks** do not use magnetism
  - They employ special materials that are altered by laser light
- **USB attached**, various forms
  - NVRAM "thumb drives" now commonly used instead of diskettes to save data for short periods of time or transfer between systems
  - Also regular "laptop" drives in case with USB adapter
  - They appear to the system as a removable disk

Optical Disks

- ***WORM*** ("Write Once, Read Many Times") disks can be written only once.
  - Thin aluminum film sandwiched between two glass or plastic platters.
  - To write a bit, the drive uses a laser light to burn a small hole through the aluminum; information can be destroyed, but not altered. *(i.e., "burn" a CD)*
  - Very durable and reliable (all these are very durable)
- ***Read Only*** disks, such ad *CD-ROM* and *DVD*, come from the factory with the data pre-recorded.
  - Pits are pressed, not burned, into media
- ***CD-R*** and ***DVD-R*** disks can be written only once
  - Similar to WORM disk but use organic polymer dye
- ***CD-RW*** and ***DVD-RW*** are ***phase change disks***
  - Disk coated with material that can be crystalline or amorphous – crystalline is more transparent
  - Laser uses different power levels to read, erase, write

Tapes

- Compared to a disk, a **tape is less expensive and holds more data, but random access is much slower.**
  - Tape is an economical medium for purposes that do not require fast random access, e.g., **backup** copies of disk data, holding huge volumes of data.

- Large tape installations typically use **robotic tape changers** that move tapes between tape drives and storage slots in a tape library.
  - ***stacker*** – library that holds a few tapes
  - ***silo*** – library that holds thousands of tapes

- A disk-resident file can be ***archived*** to tape for low cost storage; the computer can ***stage*** it back into disk storage for active use.

- Similar archival / library technology also used for CDs / DVDs

Operating System Issues

- **How are I/O devices managed by the OS** and presented to the user?
  - Major OS responsibilities are to **manage physical devices** and to present a **virtual machine abstraction** to applications
- For **hard disks**, the OS provides two kinds of abstractions:
  - **Raw device** – an array of data blocks.
  - **File system** – the OS queues and schedules the interleaved requests from several applications.

Application Interface

- Most OSs **handle removable disks almost exactly like fixed disks**
  - A new cartridge is formatted and an empty file system is generated on the disk.
- **Tapes are presented as a raw storage medium**, i.e., an application does not not open a file on the tape, it opens the whole tape drive as a raw device.
  - Usually the tape drive is reserved for the **exclusive use** of that application.
  - Since the OS **does not provide file system services**, the application must decide how to use the array of blocks.
  - Since every application makes up its own rules for how to organize a tape, a tape full of data **can sometimes be used only by the program that created it**
  - However, **often tapes are written in “standard” formats**, or as just a stream file.

Tape Drives

- The **basic operations** for a tape drive differ from those of a disk drive.
  - **locate** positions the tape to a specific logical block, not an entire track (corresponds to **seek**).
  - The **read position** operation returns the logical block number where the tape head is.
  - The **space** operation enables relative motion.
- Tape drives are **"append-only"** devices
  - Updating a block in the middle of the tape also effectively erases everything beyond that block.
- An **End of Tape** (EOT) mark is placed after the last block that is written.

File Naming

- The issue of naming files on removable media is especially **difficult** when want to write data on a removable cartridge on one computer, and then use the cartridge in another computer (with a different OS).
- Contemporary OSs generally **leave the name space problem unsolved for removable media**, and depend on applications and users to figure out how to access and interpret the data.
  - Can be **conventions from manufacturer** (e.g., Iomega)
  - And the removable media is usually considered a self-contained partition
- Some kinds of removable media (e.g., CDs) are so **well standardized** that all computers use them the same way.

Hierarchical Storage Management (HSM)

- A **hierarchical storage** system extends the storage hierarchy beyond primary memory and secondary storage to incorporate tertiary storage — usually implemented as a jukebox of tapes or removable disks.
- Usually **incorporate tertiary storage** by extending the file system.
  - Small and frequently used files remain on disk.
  - Large, old, inactive files are archived to the jukebox.
  - When try to access a file on tertiary storage, it just takes a little longer
- HSM is **usually found in supercomputing centers, large mainframes,** and other installations that have enormous volumes of data.
  - Becoming more common elsewhere
  - However, as online storage becomes cheaper, huge amounts of online storage also becoming more common

Evaluating Tertiary Storage

- Essential **considerations**:
  - **Speed**
  - **Reliability**
  - **Cost**

Speed

- Two aspects of speed in tertiary storage are **bandwidth and latency.**
- Bandwidth is measured in bytes per second.
  - **Sustained bandwidth** – average data rate during a large transfer; # of bytes/transfer time.
    - Data rate when the data stream is actually flowing.
  - **Effective bandwidth** – average over the entire I/O time, including **seek** or **locate**, and cartridge switching.
    - Drive's overall data rate.

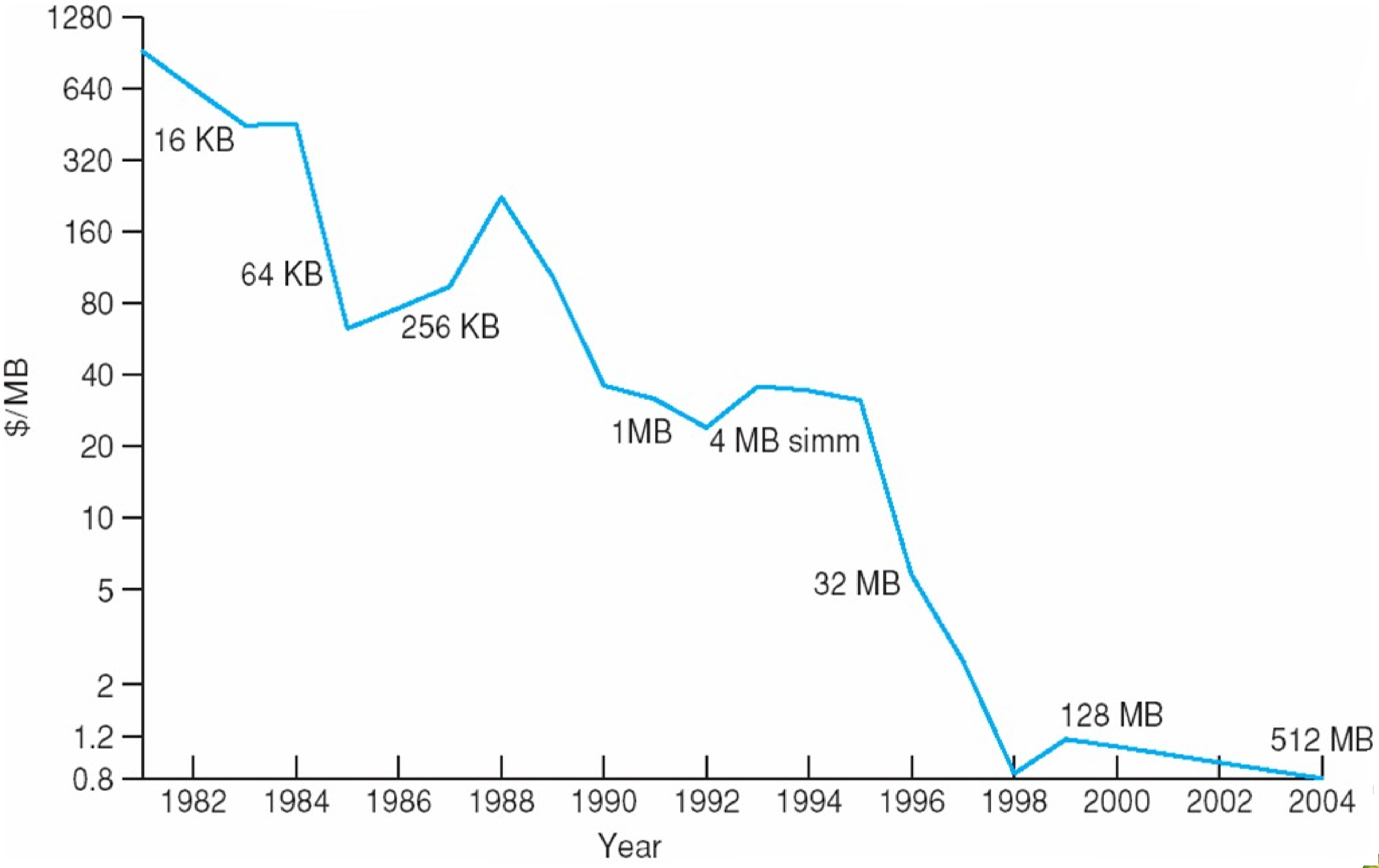
Reliability

- A **fixed disk drive** is likely to be more reliable than a removable disk or tape drive.
- An **optical cartridge** is likely to be more reliable than a magnetic disk or tape.
- A **head crash** in a fixed hard disk generally destroys the data, whereas the failure of a tape drive or optical disk drive often leaves the data cartridge unharmed.

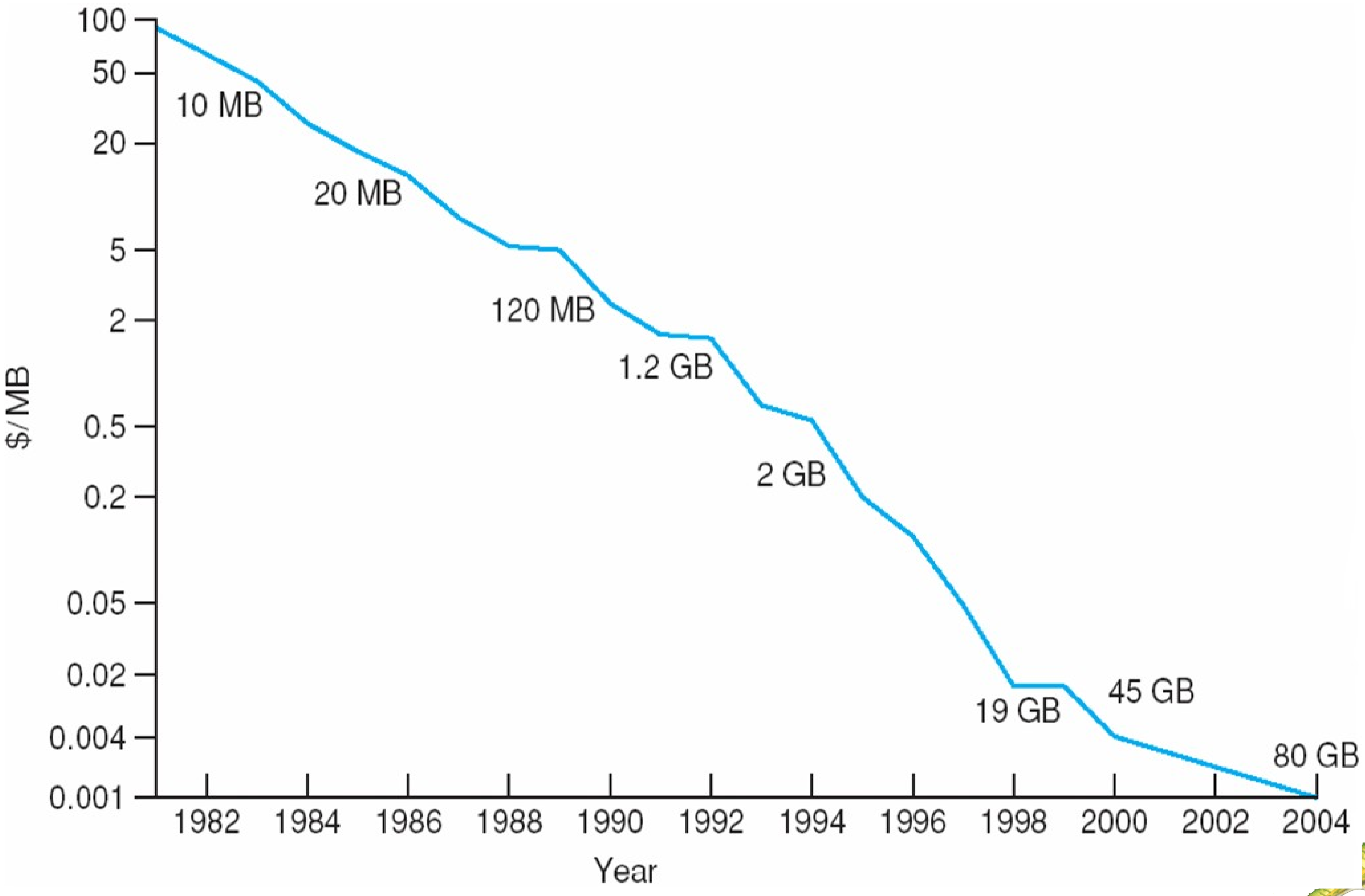
Cost

- Main memory is much more **expensive** than disk storage
- The cost per megabyte of hard disk storage is **competitive** with magnetic tape if only one tape is used per drive.
- The **cheapest** tape drives and the cheapest disk drives have had about the same storage capacity over the years.
- Tertiary storage gives a **cost savings** only when the number of cartridges is considerably larger than the number of drives.

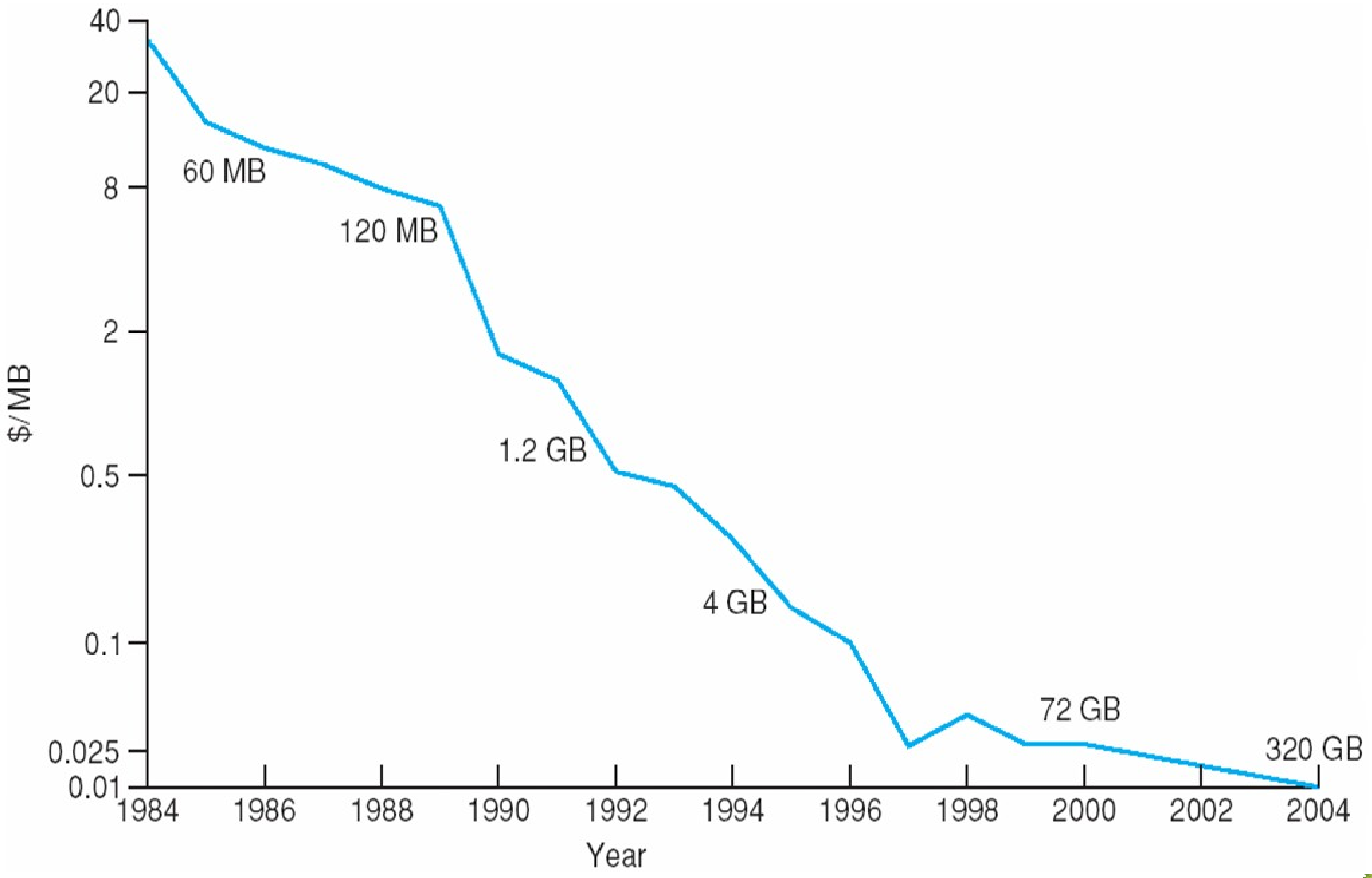
Price per Megabyte of DRAM, From 1981 to 2004



Price per Megabyte of Magnetic Hard Disk, From 1981 to 2004



Price per Megabyte of a Tape Drive, From 1984-2000



ADD INFO ON

- Cloud computing / storage
- Solid State Disks

End of Chapter 12