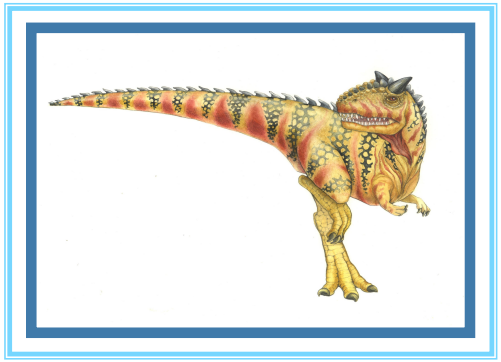
Chapter 10: Mass-Storage Systems



Chapter 10: Mass-Storage Systems

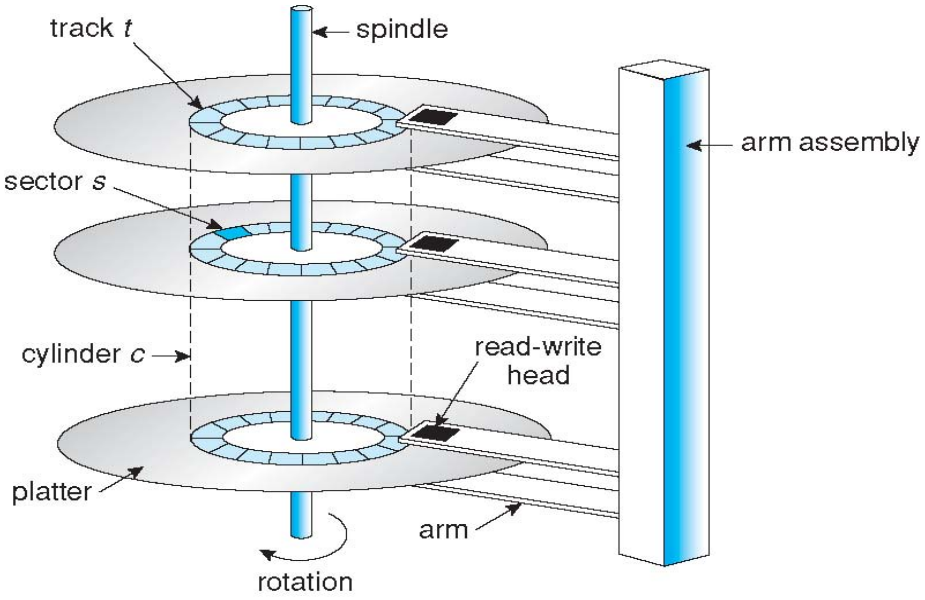
- Overview of Mass Storage Structure
- Disk Structure
- Disk Attachment
- Disk Scheduling
- Disk Management
- Swap-Space Management
- RAID Structure
- Stable-Storage Implementation
- Tertiary Storage Devices

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 disks** provide bulk of secondary storage of modern computers
  - Drives rotate at 60 to 250 times per second (5,400 to 15,000 RPM)
  - **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**) plus 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 news)
- Some disks are removable (e.g., CD, DVD, Blu-ray)
- Drive attached to computer via **I/O bus**
  - Busses vary, including **EIDE**, **ATA**, **SATA**, **USB**, **Fibre Channel**, **SCSI, SAS, Firewire**
  - **Host controller** in computer uses bus to talk to **disk controller** built into drive or storage array
    - typically communication over memory-mapped I/O ports
    - disk controller use its built-in cache to transfer



Magnetic Disks

- Platters range from .85” to 14” (historically)
  - Commonly 3.5”, 2.5”, and 1.8”
- Range from 30GB to 3TB per drive
- Performance
  - Transfer Rate – theoretical – 6 Gb/sec
    - Effective Transfer Rate – real – 1Gb/sec
  - Seek time from 3ms to 12ms – 9ms common for desktop drives
    - Average seek time measured or calculated based on 1/3 of the number of tracks
  - Latency based on spindle speed
    - 1/(RPM * 60)
    - Average latency = ½ latency

Spindle [rpm]	Average latency [ms]
4200	7.14
5400	5.56
7200	4.17
10000	3
15000	2

(from Wikipedia)

Magnetic Disk Performance

- **Access Latency** = **Average access time** = average seek time + average latency
  - For fastest disk 3ms + 2ms = 5ms
  - For slow disk 9ms + 5.56ms = 14.56ms
- Average I/O time = average access time + (amount to transfer / transfer rate) + controller overhead
- For example to transfer a 4KB block on a 7200 RPM disk with a 5ms average seek time, 1Gb/sec transfer rate with a 0.1ms controller overhead =
  - 5ms + 4.17ms + 4KB / 1Gb/sec + 0.1ms =
  - 9.27ms + 4 / 131,072 sec =
  - 9.27ms + 0.12ms = 9.39ms

The First Commercial Disk Drive

1956
IBM RAMDAC computer included the IBM Model 350 disk storage system

5M (7 bit) characters
50 x 24” platters
Access time = < 1 second

Solid-State Disks

- Non-volatile memory
  - DRAM with battery, flash-memory single-level/multilevel cell (SLC/MLC)
- More reliable than disks (no moving parts), faster (no seek/latency time), use less power
- More expensive per MB, smaller capacity than disks, shorter life spans
  - good candidate for smaller/lighter/less power hungry disk replacement in laptops
- Sometimes used to hold file system metadata
- Can be used as a form of cache between disks and memory
- Can be connected directly to system bus

Magnetic Tape

- Was early secondary-storage medium
  - Evolved from open spools to cartridges
- Relatively permanent and holds large quantities of data
- Access time slow
- Random access about 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
  - 140MB/sec and greater
- 200GB to 1.5TB typical storage
- Common technologies are LTO-{3,4,5} and T10000

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of **logical blocks**, where the logical block is the smallest unit of transfer (typically 512 bytes)
- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially
  - Sector 0 is the first sector of the first track on the outermost cylinder
  - Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost
  - Logical to physical address should be easy
    - Except for bad sectors
    - Uniform bit density
      - 40% more sectors on outermost tracks
      - increase rotation speed from outermost to inner tracks for constant head reading speed
      - CD/DVD drives
    - Non-uniform bit density
      - constant rotation speed
      - outermost tracks have lower bit density

Disk Attachment

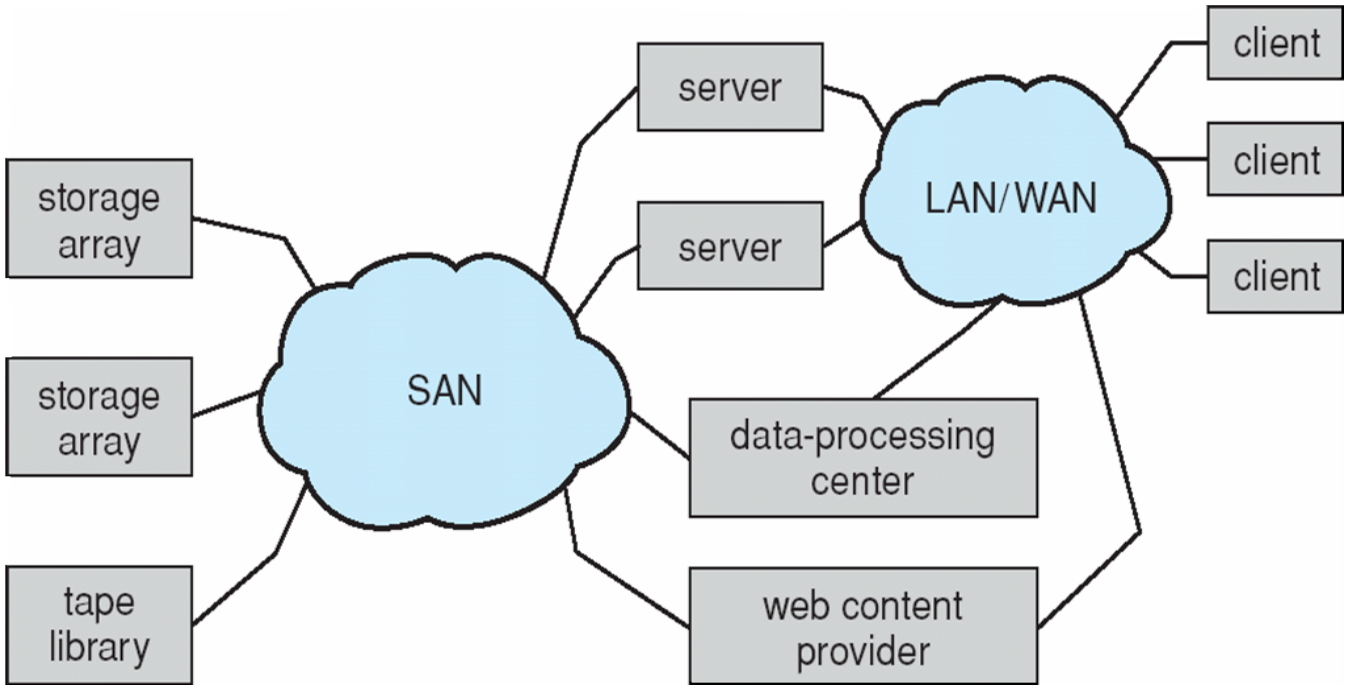
- Host-attached storage accessed through I/O ports talking to I/O busses
- SCSI itself is a bus, up to 16 devices on one cable, **SCSI initiator** requests operation and **SCSI targets** perform tasks
  - Each target can have up to 8 **logical units** (disks attached to device controller)
- FC is high-speed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

Storage Array

- Can just attach disks, or arrays of disks
- Storage Array has controller(s), provides features to attached host(s)
  - Ports to connect hosts to array
  - Memory, controlling software (sometimes NVRAM, etc.)
  - A few to thousands of disks
  - RAID, hot spares, hot swap (discussed later)
  - Shared storage -> more efficiency
  - Features found in some file systems
    - Snaphots, clones, thin provisioning, replication, deduplication, etc.

Storage Area Network

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

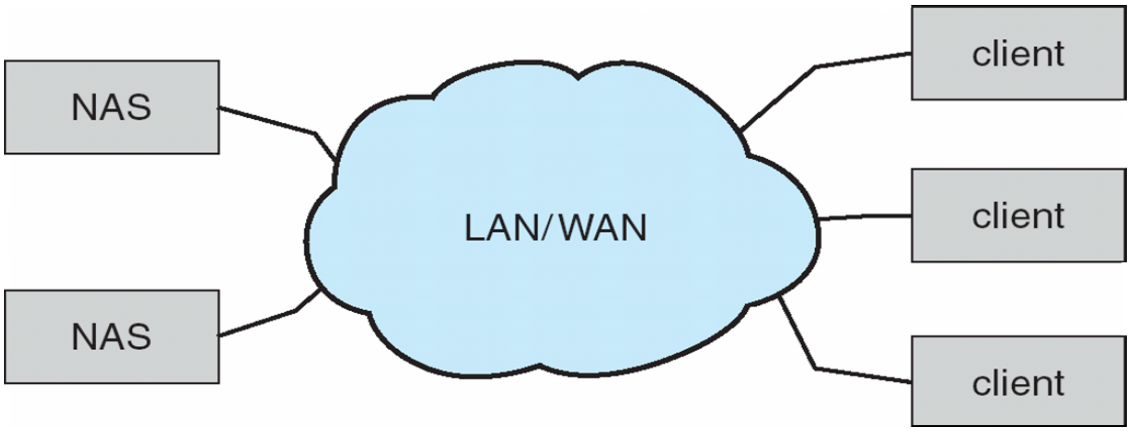


Storage Area Network (cont.)

- SAN is one or more storage arrays
  - Connected to one or more Fibre Channel switches
- Hosts also attach to the switches
- Storage made available via **LUN Masking** from specific arrays to specific servers
- Easy to add or remove storage, add new host and allocate it storage
  - Over low-latency Fibre Channel fabric
- Why have separate storage networks and communications networks?
  - Consider iSCSI, FCOE

Network-Attached Storage

- Network-attached storage (**NAS**) is storage made available over a network rather than over a local connection (such as a bus)
  - Remotely attaching to file systems
- NFS and CIFS are common protocols
- Implemented via remote procedure calls (RPCs) between host and storage over typically TCP or UDP on IP network
- **iSCSI** protocol uses IP network to carry the SCSI protocol
  - Remotely attaching to devices (blocks)



Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth
- Minimize seek time
- Seek time is related to seek distance for disks
- Disk **bandwidth** is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer

Disk Scheduling (cont.)

- There are many sources of disk I/O request
  - OS
  - System processes
  - Users processes
- I/O request includes input or output mode, disk address, memory address, number of sectors to transfer
- OS maintains queue of requests, per disk or device
- Idle disk can immediately work on I/O request, busy disk means work must queue
  - Optimization algorithms only make sense when a queue exists
- Note that drive controllers have small buffers and can manage a queue of I/O requests (of varying “depth”)

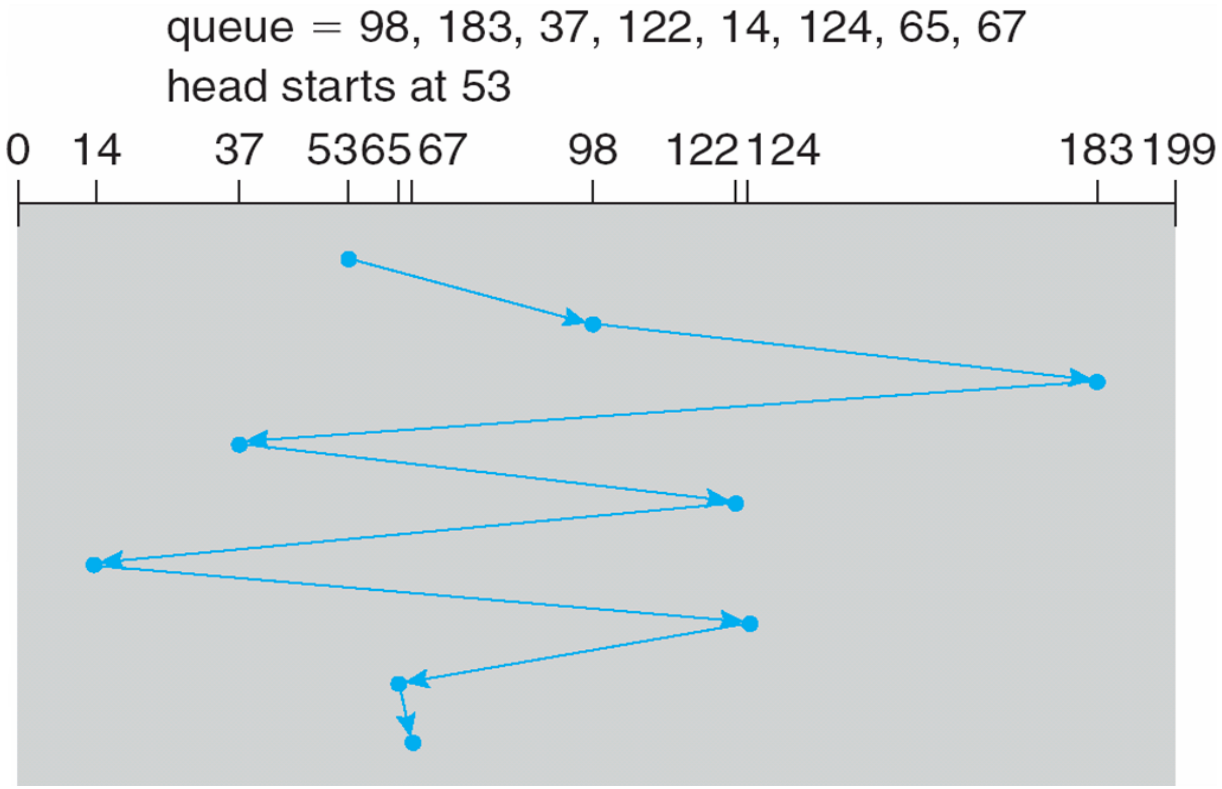
- Several algorithms exist to schedule the servicing of disk I/O requests
- The analysis is true for one or many platters
- We illustrate scheduling algorithms with a request queue (0-199) for track/cylinder number

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

  Head pointer 53

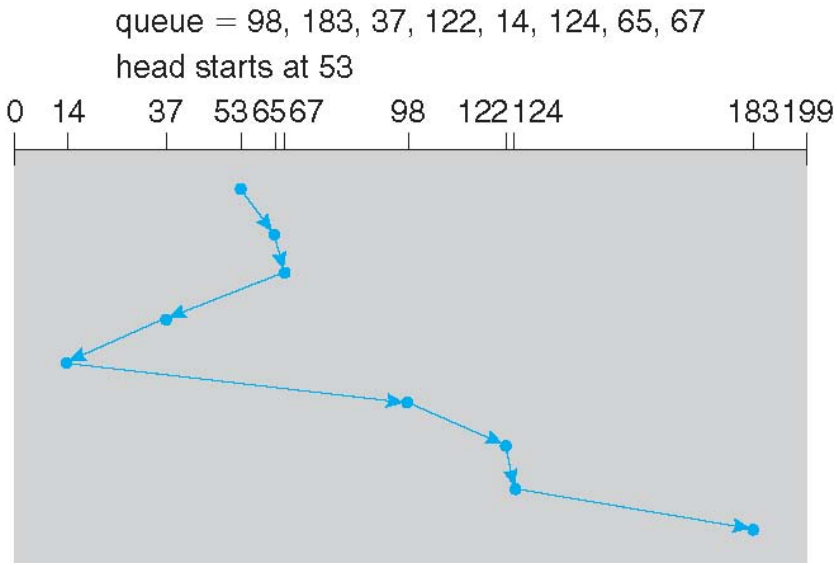
First-come first-served (FCFS)

Illustration shows total head movement of 640 cylinders



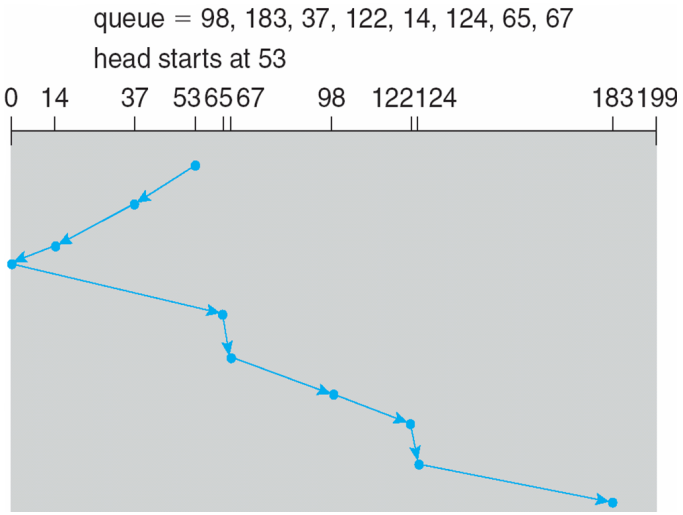
Shortest Seek Time First (SSTF)

- Shortest Seek Time First selects the request with the minimum seek time from the current head position
- SSTF scheduling is a form of Shortest-job-first (SJF) scheduling; may cause starvation of some requests
- Improvement over FCFS, but not optimal
- Illustration shows total head movement of 236 cylinders



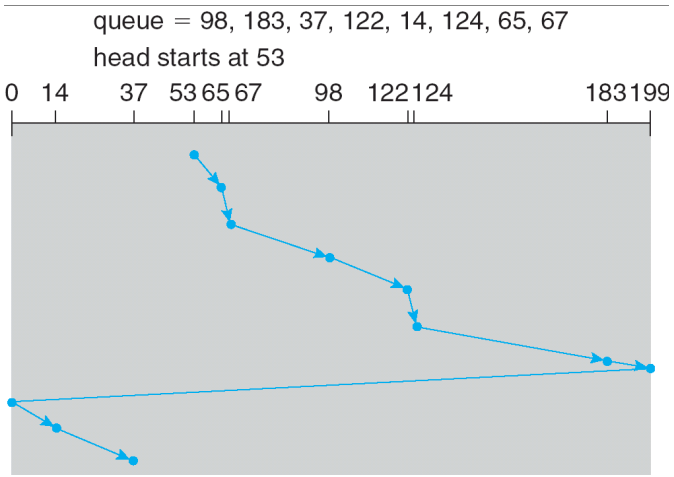
SCAN

- 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
- **SCAN algorithm** sometimes called the **elevator algorithm**
- Illustration shows total head movement of 208 cylinders
- But note that if requests are uniformly dense, largest density at other end of disk and those wait the longest



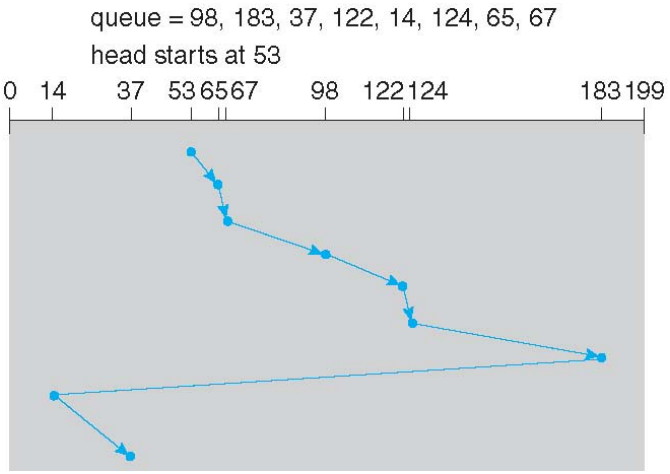
Circular SCAN (C-SCAN)

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



C-LOOK

- LOOK a version of SCAN, C-LOOK a version of C-SCAN
- Arm only goes as far as the last request in each direction, then reverses direction immediately, without first going all the way to the end of the disk
- Total number of cylinders?



Selecting a Disk-Scheduling Algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation
- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method
  - And metadata layout
- 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
- What about rotational latency?
  - Difficult for OS to calculate
- Disk controller might have its own scheduling algorithm, but relying on it would defer all knowledge the OS has about priorities (crash, write vs read, small remaining space for memory paging out, etc.)
- How does disk-based queuing effect OS queue ordering efforts?

Disk Management

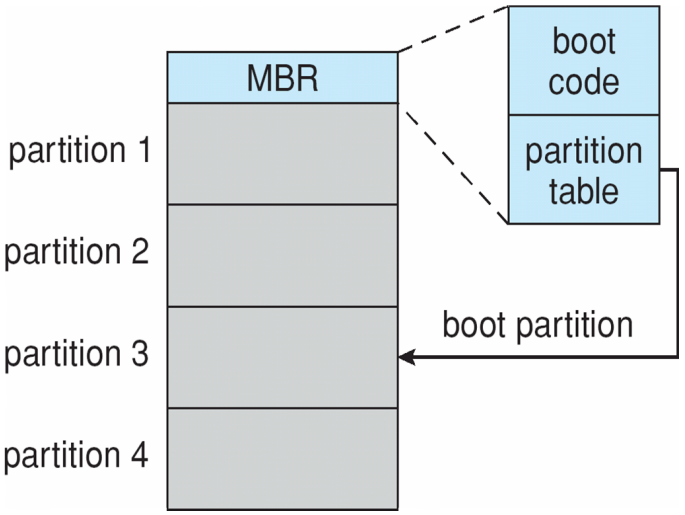
- **Low-level formatting**, or **physical formatting** — Dividing a disk into sectors that the disk controller can read and write
  - Each sector can hold header information, plus data, plus error correction code (**ECC**)
  - Usually 512 bytes of data but can be selectable
  - Controller tests the ECC after reading, and indicates if it can correct any error (a few bits)

- To use a disk to hold files, the operating system needs to record its own data structures on the disk
  - **Partition** the disk into one or more groups of cylinders, each treated as a logical disk
    - separate partitions for OS, user files, exported file system
  - **Logical formatting** or "making a file system"
  - To increase efficiency most file systems group blocks into **clusters**
    - Disk I/O done in blocks
    - File I/O done in clusters
  - Partition could be **raw**, i.e., without file system data structure overhead (e.g., for databases)

Disk Management (cont.)

- Boot block initializes system
  - The bootstrap is stored in ROM
  - **Bootstrap loader** program stored in boot blocks of boot partition
- Bad disk blocks appears when hardware failures occur on it
  - can be detected and flagged as unusable (live)
  - Methods such as **sector sparing** used to handle bad blocks
    - reserve sectors and use them as replacements, as much as possible in the same cylinder to reduce scheduling difficulties
    - controller reassign the replacement to the bad block address
  - **Sector slipping** copies all sectors up to the spare sector in order to keep the sector adjacent to its bad sector

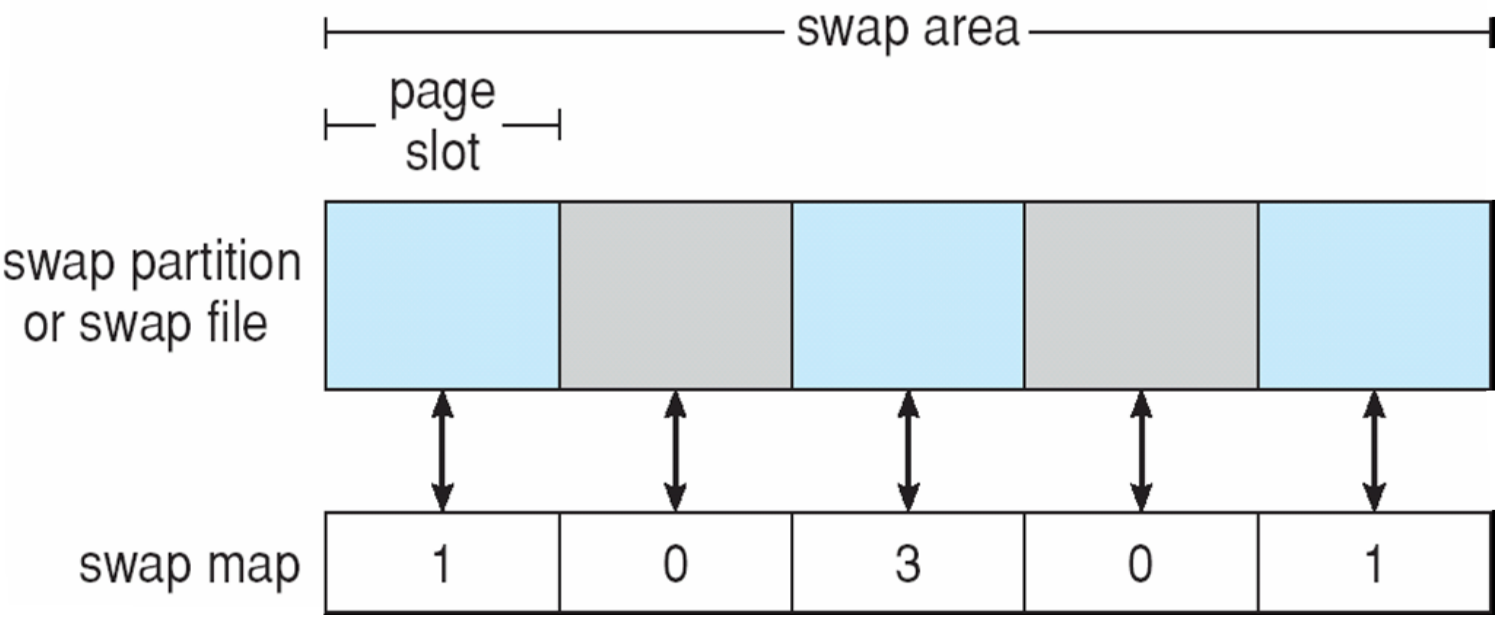
Booting from disk in Windows 2000



Swap-Space Management

- Swap-space — Virtual memory uses disk space as an extension of main memory
  - Less common now due to memory capacity increases
- Swap-space can be carved out of the normal file system, or, more commonly, it can be in a separate disk partition (raw)
- Swap-space management
  - 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
  - Solaris 2 allocates swap space only when a dirty page is forced out of physical memory, not when the virtual memory page is first created
    - File data written to swap space until write to file system requested
    - Other dirty pages go to swap space due to no other home
    - Text segment pages thrown out and reread from the file system as needed
- What if a system runs out of swap space?
- Some systems allow multiple swap spaces

Data Structures for Swapping on Linux Systems



number indicates the number of processes using the swapped page
0 means space available

RAID Structure

- RAID – multiple disk drives provides reliability via **redundancy**
- Increases the **mean time to failure**
  - 100,000 hours: mean failure for one disk
  - 100,000/100 = 1000 hours: mean failure for one disk out of a set of 100
  - With mirroring, failure of two specific disks 100,000x100,000 / 2 x 10 hours repair = 500×10^6
  - But not all failures can be considered independent events, quite the contrary...
- With mirroring, could also write one copy, wait for completion, then write the mirrored copy
- Frequently combined with **NVRAM** to improve write performance
- Reading two disks in parallel allows doubling the transfer of data
- RAID is arranged into six different levels

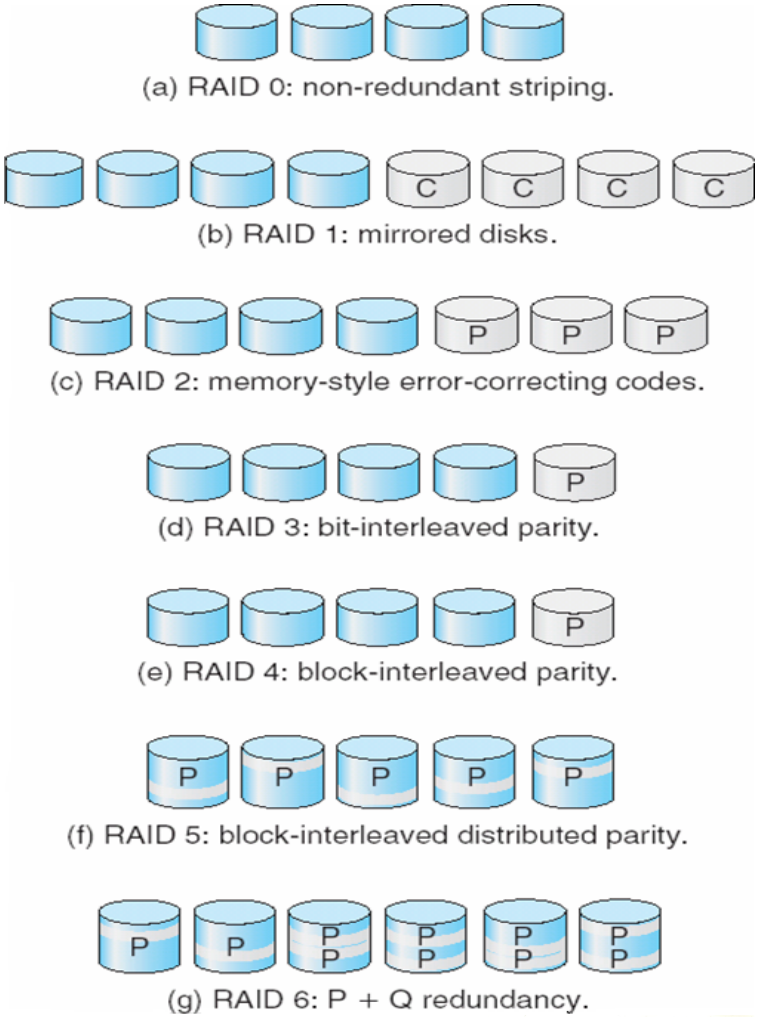
RAID (cont.)

- 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
  - E.g., with 8 bits of data, storing one bit per disk, 8 disks can transfer 8x more data
  - Can stripe by byte, sector, block (most common)
  - Striping increases throughput, reduces response time
  - Striping does not improve reliability

RAID (cont.)

Content for 4 disks of data
P: error-correcting bits
C: second copy

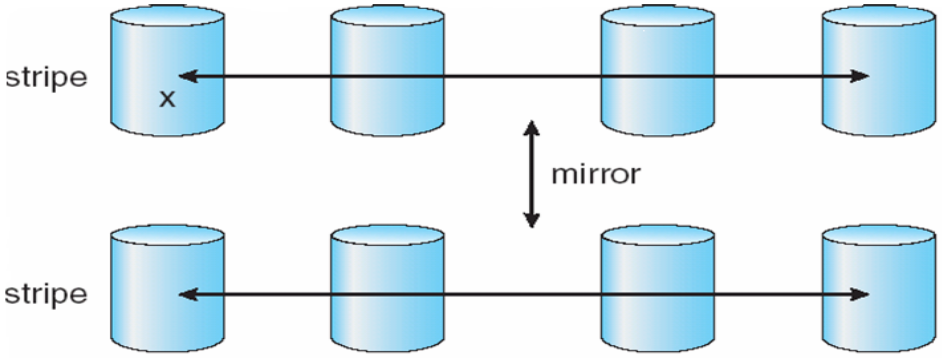
- RAID schemes improve performance and reliability of the storage system by storing redundant data
  - **Mirroring** or **shadowing** (**RAID 1**) keeps duplicate of each disk
  - **Block interleaved parity** (**RAID 4, 5, 6**) uses much less redundancy
  - **RAID 2** has error-correcting bits in other disks, so can correct also a failing disk; **RAID 3** uses disk controllers' detection mechanism to reduce to only 1 bit parity
  - **RAID 4** reads one block on one disk, but small accesses are expensive; **RAID 5** stores parity and blocks on any disk; **RAID 6** uses more bits for error correction (allow for 2 disk failures)
  - Striped mirrors (**RAID 1+0**) or mirrored stripes (**RAID 0+1**) provides high performance and high reliability
- RAID within a storage array can still fail if the array fails, so automatic **replication** of the data between arrays is common
- Frequently, a small number of **hot-spare** disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them



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

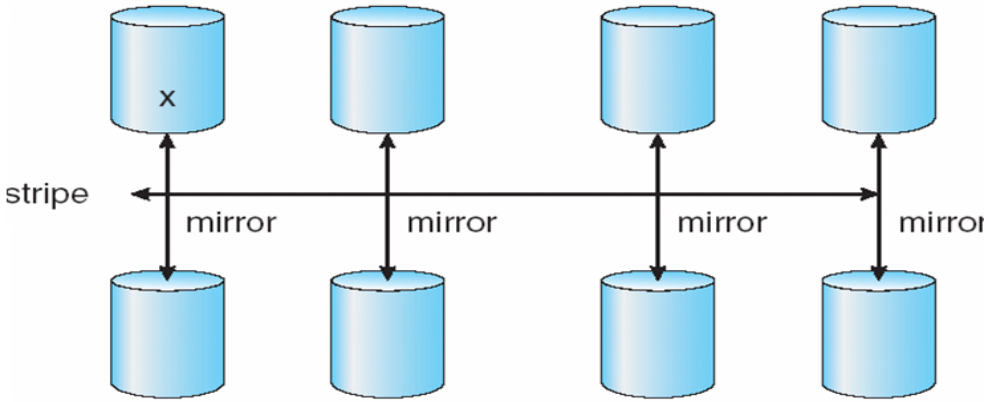
RAID 0+1:
disks are striped,
then stripe is mirrored

one disk failure cancels
an entire stripe



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

RAID 1+0:
disks are mirrored in pairs,
then mirrors are striped



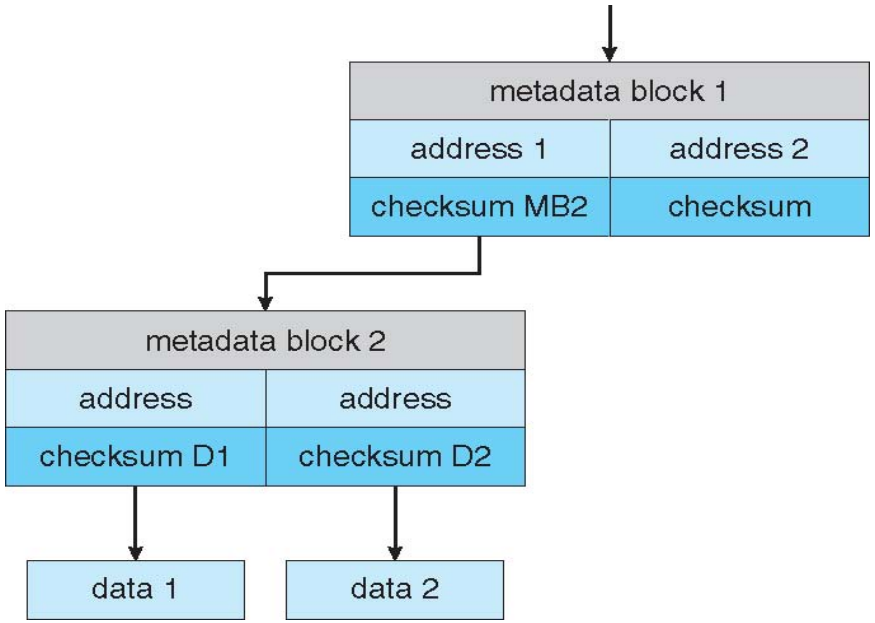
b) RAID 1 + 0 with a single disk failure.

Extensions

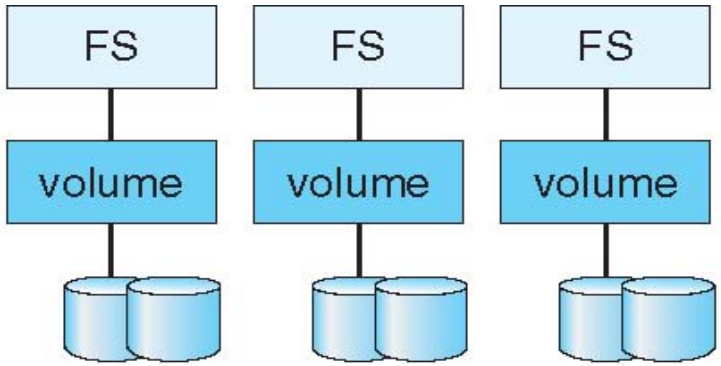
- Choosing a RAID level can also be dictated by rebuilding performance of a RAID, on top of throughput, bit correction, cost, etc.
- RAID is not very flexible, e.g., file systems can evolve, and they might need more than one disk, or be too small for one disk
- RAID technology can be applied to arrays of tapes, data broadcast over networks, etc.
- RAID is not the solution to all problems, RAID alone does not prevent or detect data corruption or other errors, just disk failures

Checksums

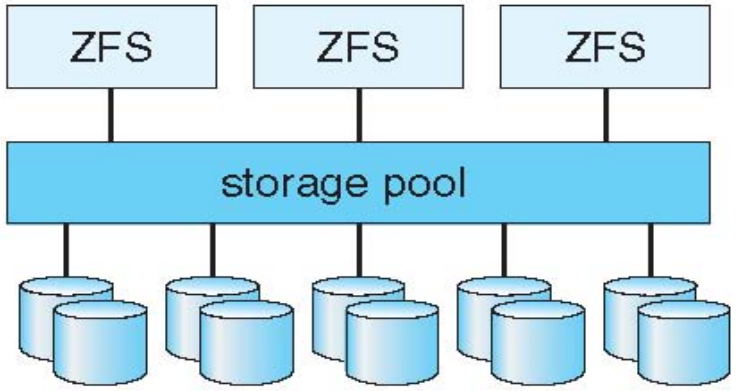
- Solaris ZFS adds **checksums** of all data and metadata
- Checksums kept with pointer to object, to detect if object is the right one and whether it changed
- Can detect and correct data and metadata corruption
- ZFS also removes volumes, partititions
  - Disks allocated in **pools**
  - File systems with a pool share that pool, use and release space like “malloc” and “free” memory allocate / release calls



Traditional and Pooled Storage



(a) Traditional volumes and file systems.



(b) ZFS and pooled storage.

Stable-Storage Implementation

- 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 we can recover the stable data after any failure during data transfer or recovery
  - NVRAM reduces the wait for disk writes, can be considered a stable storage, and improves performance

Tertiary Storage Devices

- Low cost is the defining characteristic of tertiary storage
- Generally, tertiary storage is built using **removable media**
- Common examples of removable media are floppy disks and CD-ROMs; 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
  - 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-optic 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-optic 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

WORM Disks

- The data on read-write disks can be modified over and over
- **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 by not altered
- Very durable and reliable
- **Read-only disks**, such ad CD-ROM and DVD, come from the factory with the data pre-recorded

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.

Operating System Support

- Major OS jobs are to manage physical devices and to present a virtual machine abstraction to applications
- For hard disks, the OS provides two abstraction:
  - 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., and 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 generally only be used by the program that created it

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 EOT mark is placed after a block that is written

File Naming

- The issue of naming files on removable media is especially difficult when we want to write data on a removable cartridge on one computer, and then use the cartridge in another computer.
- 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.
- 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
- HSM is usually found in supercomputing centers and other large installations that have enormous volumes of data.

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

Speed (cont.)

- **Access latency** – amount of time needed to locate data
  - Access time for a disk – move the arm to the selected cylinder and wait for the rotational latency; < 35 milliseconds
  - Access on tape requires winding the tape reels until the selected block reaches the tape head; tens or hundreds of seconds
  - Generally say that random access within a tape cartridge is about a thousand times slower than random access on disk
- The low cost of tertiary storage is a result of having many cheap cartridges share a few expensive drives
- A removable library is best devoted to the storage of infrequently used data, because the library can only satisfy a relatively small number of I/O requests per hour

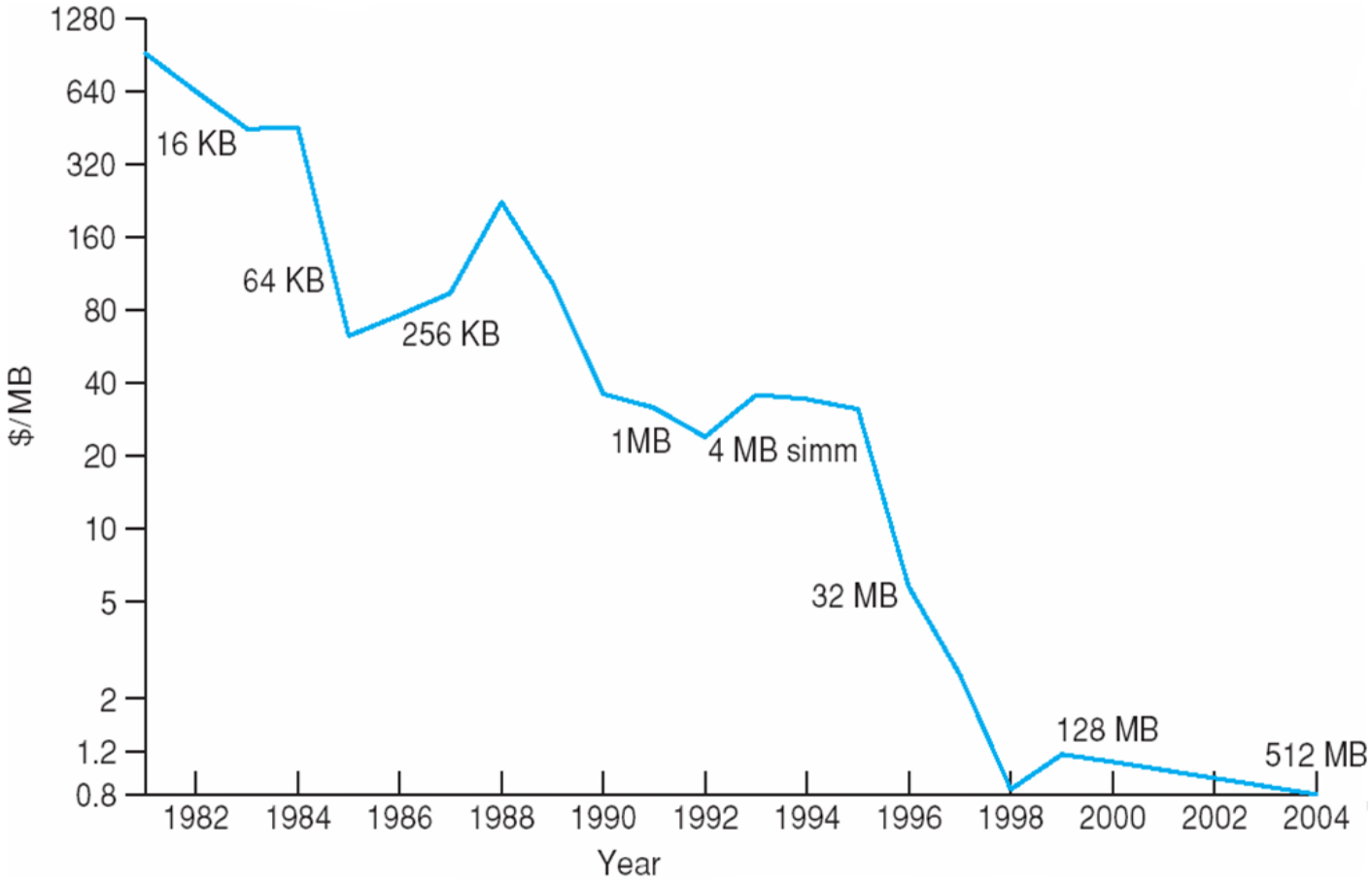
Reliability

- 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

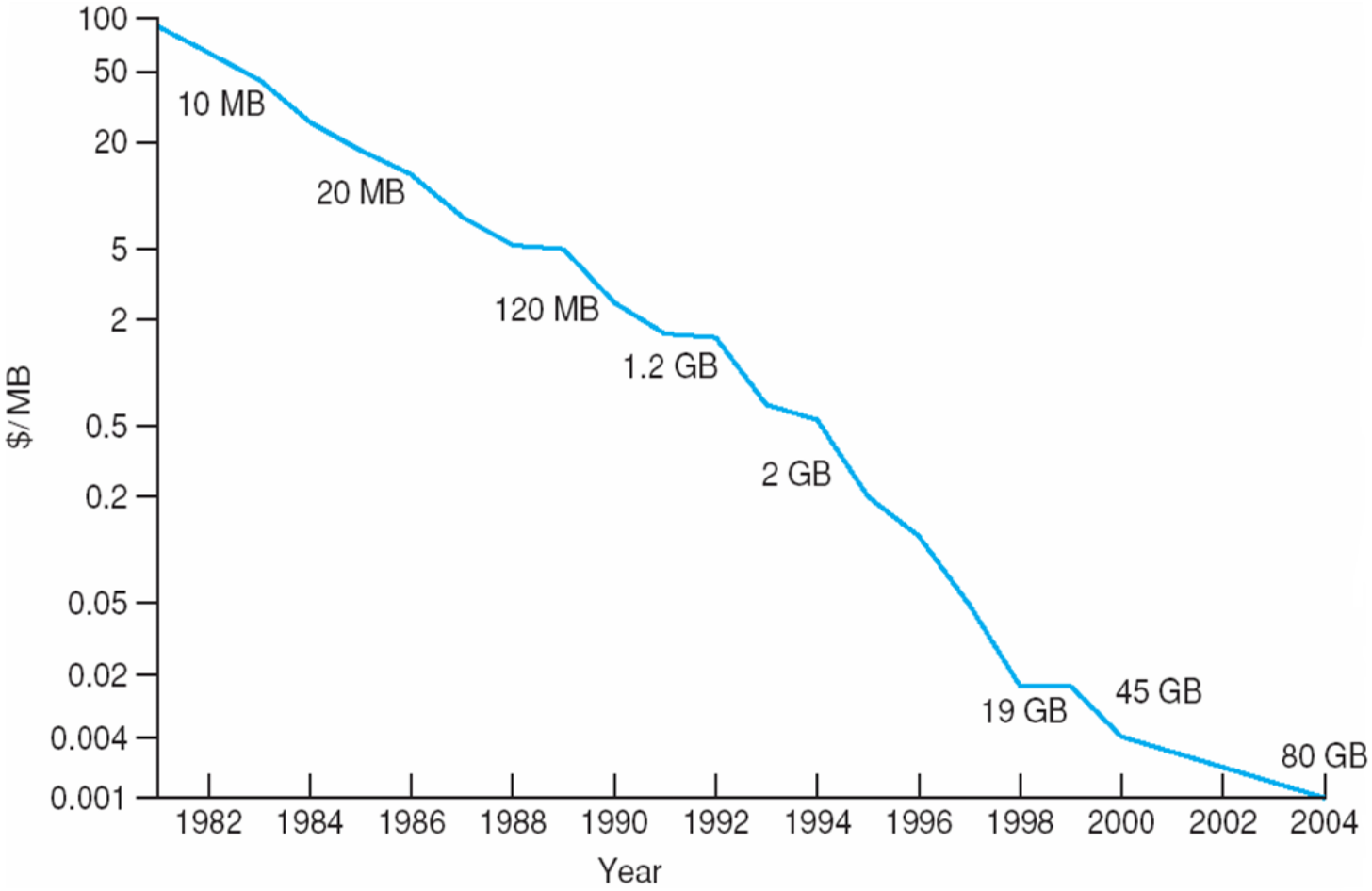
Cost

- 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

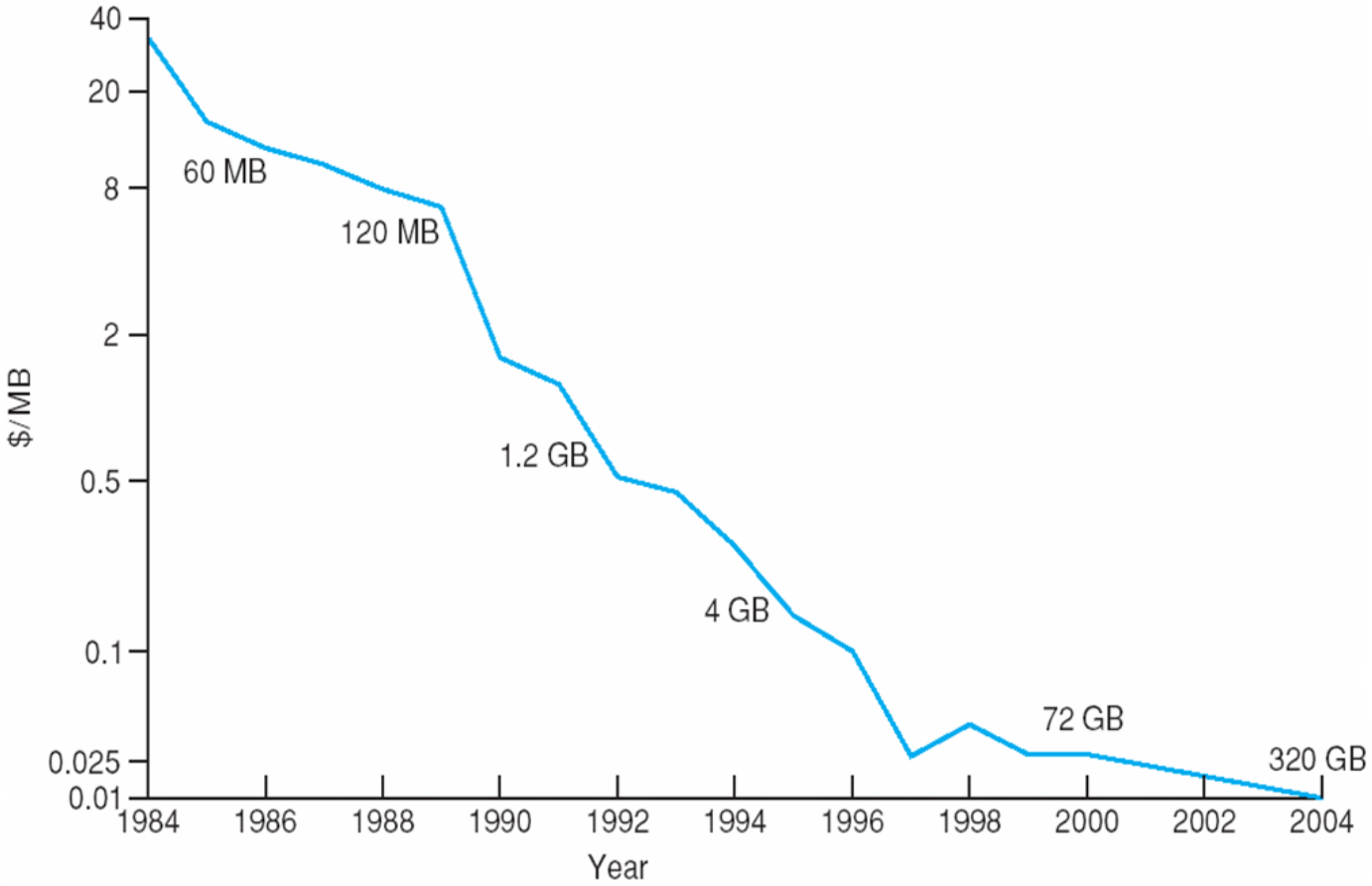
Price per Megabyte of DRAM From 1981 to 2004



Price per Megabyte of Magnetic Hard Disk From 1981 to 2004



Price per Megabyte of a Tape Drive From 1984-2000



End of Chapter 10