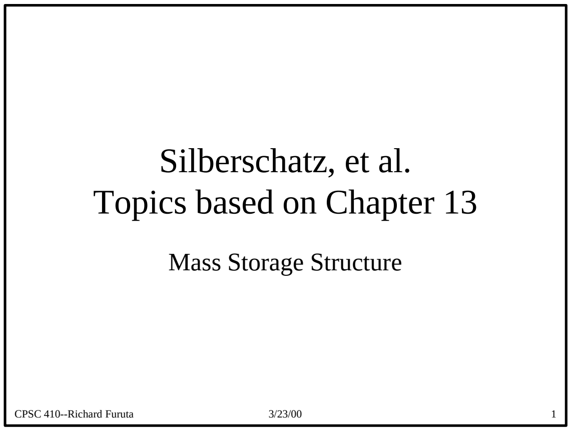
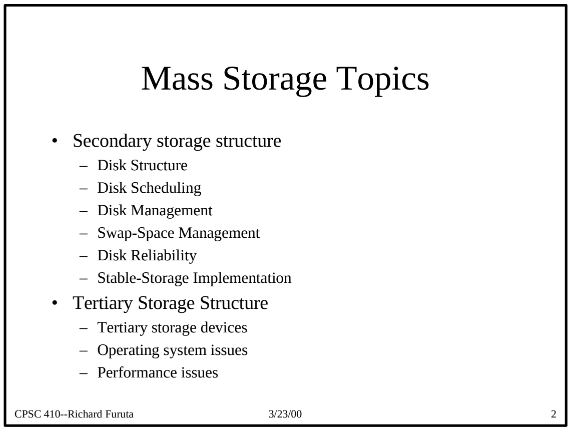


Silberschatz, et al.
Topics based on Chapter 13

Mass Storage Structure

CPSC 410--Richard Furuta 3/23/00 1

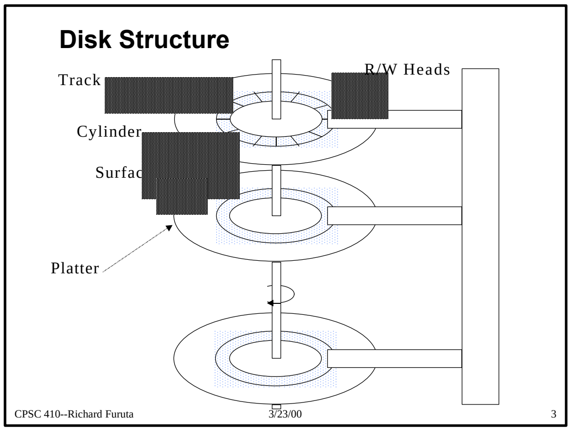


Mass Storage Topics

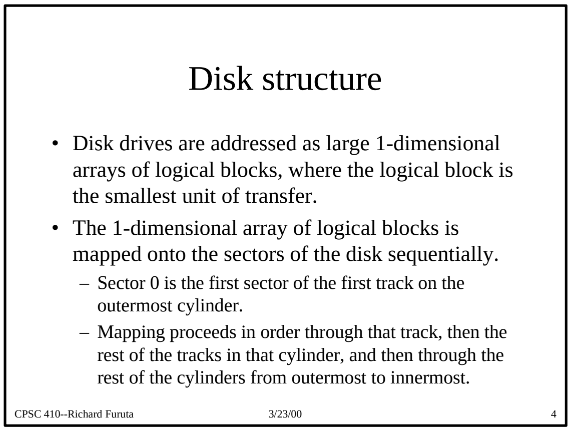
- Secondary storage structure
  - Disk Structure
  - Disk Scheduling
  - Disk Management
  - Swap-Space Management
  - Disk Reliability
  - Stable-Storage Implementation
- Tertiary Storage Structure
  - Tertiary storage devices
  - Operating system issues
  - Performance issues

CPSC 410--Richard Furuta 3/23/00 2

Disk Structure



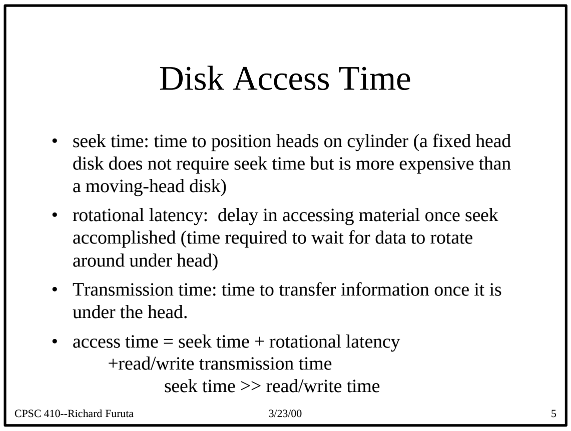
CPSC 410--Richard Furuta 3/23/00 3



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 onto 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.

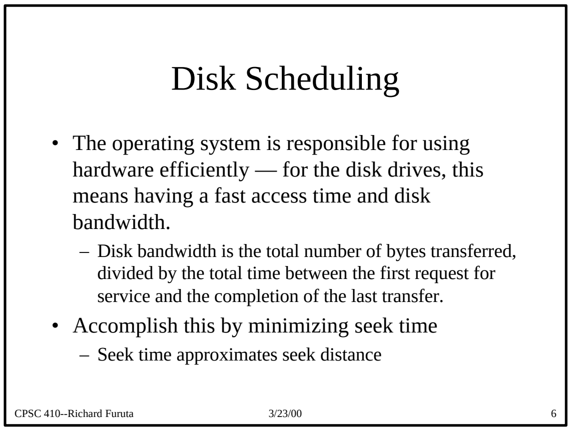
CPSC 410--Richard Furuta 3/23/00 4



Disk Access Time

- seek time: time to position heads on cylinder (a fixed head disk does not require seek time but is more expensive than a moving-head disk)
- rotational latency: delay in accessing material once seek accomplished (time required to wait for data to rotate around under head)
- Transmission time: time to transfer information once it is under the head.
- access time = seek time + rotational latency
  +read/write transmission time
  seek time >> read/write time

CPSC 410--Richard Furuta 3/23/00 5



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.
  - 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.
- Accomplish this by minimizing seek time
  - Seek time approximates seek distance

CPSC 410--Richard Furuta 3/23/00 6

Disk I/O Request

- Disk I/O request specifies
  - whether the operation is input or output
  - disk address (block number, which is translated into drive, cylinder, surface, and sector coordinates)
  - memory address to copy to or from
  - byte count giving the amount of information to be transfered

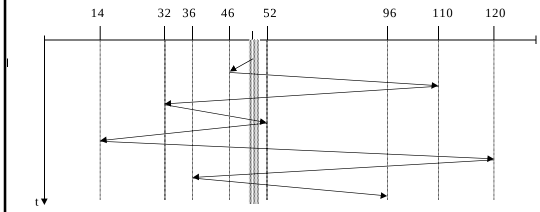
Disk Scheduling

- Many requests may be pending at once. Which should be handled first?
- Head moving strategy developed
- Attempting to manage the overall disk seek time. Latency is not controllable and transfer time depends on the size of the transfer request
- Different strategies:
  - FCFS
  - SSTF
  - SCAN
  - LOOK

FCFS Scheduling

- Simplest form
- First-come, first-served scheduling
- Requests served in order of arrival
- Advantage: simple queueing
- Disadvantage: does not provide the "best" seek time

Arrival order: 46, 110, 32, 52, 14, 120, 36, 96 (track addresses)
Head current position: 50



Total head movement = 454 tracks

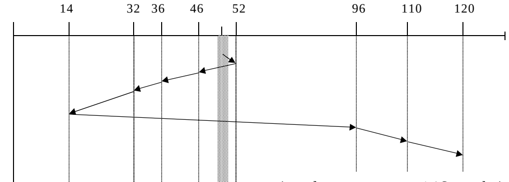
SSTF Scheduling

- Shortest-seek-time-first

Shortest-seek-time-first (SSTF)

- Better performance but not optimal.
- Starvation problem.

Arrival order: 46, 110, 32, 52, 14, 120, 36, 96
Head current position: 50



(total movement = 146 tracks)

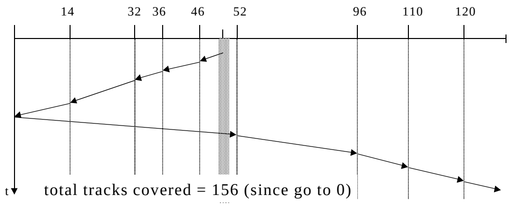
SCAN Scheduling

- Also called the “elevator” algorithm
- Continue in the direction of first movement until reach end then reverse and head in the other direction (moves completely to the ends of the disk)
- Need to know direction of head movement

Scan Algorithm

• Goes from one direction to another
• No starvation problem.
• Waiting time and its variance are position dependent.

Arrival order: 46, 110, 32, 52, 14, 120, 36, 96
Head current position: 50, moving toward to 0.



total tracks covered = 156 (since go to 0)

SCAN Scheduling

- Note that if there is a uniform distribution of (arriving) requests, the density of the requests near the head is the lowest (just been serviced). Hence when reverse direction the heaviest density of requests is at the other end of the disk and these have also waited the longest. (Uneven waiting time based on position.)

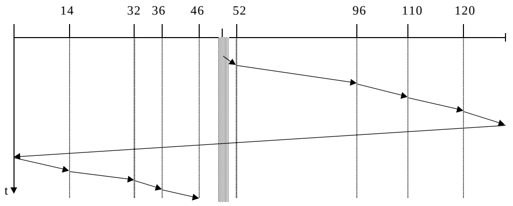
C-SCAN Scheduling

- Circular Scan scheduling
- Treats disk as if it were circular with the last track adjacent to the first one
- As it reaches the end of the disk, restarts again on the other side
- Does move head all the way to the end of the disk

Circular Scan Algorithm (C-SCAN)

• Always goes in one direction
• No starvation problem. Uniform average waiting time.

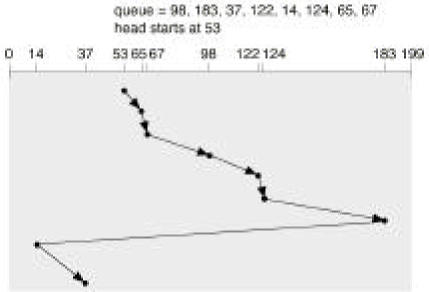
Arrival order: 46, 110, 32, 52, 14, 120, 36, 96
Head current position: 50, moving direction 0 --> 140



LOOK and C-LOOK Scheduling

- Improvements of SCAN and C-SCAN
- Only moves the head as far as the last request in each direction (rather than to the physical end of the disk).

C-LOOK Example



Disk Scheduling Algorithms

- If disk queue seldom has more than one request, all scheduling algorithms are effectively equivalent. FCFS attractive because of low overhead in implementing.
- SCAN and C-SCAN are appropriate for heavy load situations
- Relationship with allocation method: are file's blocks clustered or dispersed across disk.
- Location of directories in middle of disk (rather than on edges) can reduce amount of head movement.
- Different algorithms best for different situations--modular design helps designer adjust algorithm used.

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".

Disk management

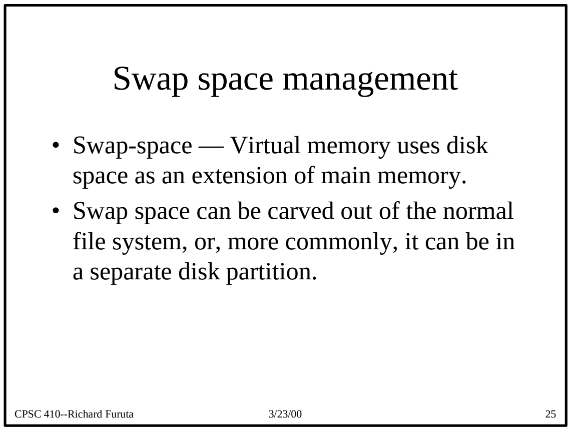
- Boot block initializes system.
  - The bootstrap is stored in ROM.
  - Bootstrap loader program brings in full bootstrap program from disk.
    - Bootstrap program stored at fixed location on disk (boot blocks)
    - Allows updating bootstrap program.

Disk management:
Bad blocks

- Methods such as sector sparing (also known as forwarding) used to handle bad blocks.
  - Spare sectors set aside on low-level formatting
  - Controller told to replace a bad sector logically with one of the spare sectors
  - To retain effectiveness of disk-scheduling optimization, provide spare sectors in each cylinder and also provide some spare cylinders. Use spare sector from same cylinder if possible.

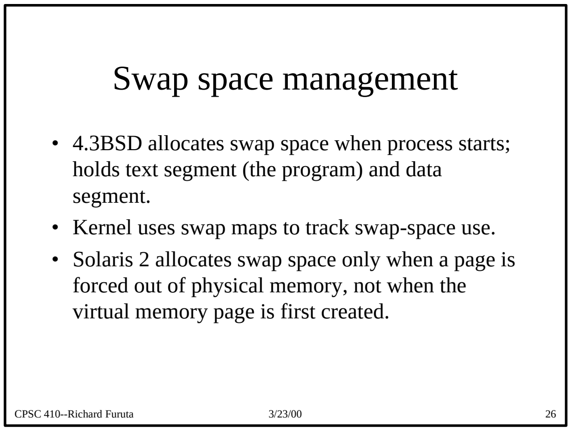
Disk management:
Bad blocks

- Sector slipping
  - moves blocks following bad block downward (occupying spare sector) to free up block following bad block
  - skips bad block, using freed up block to hold that sector's information.



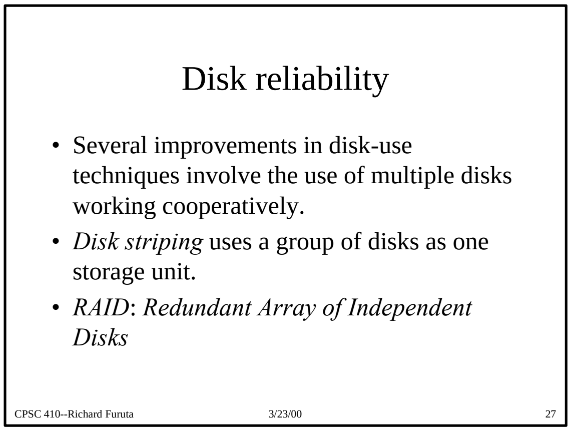
Swap space management

- 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.



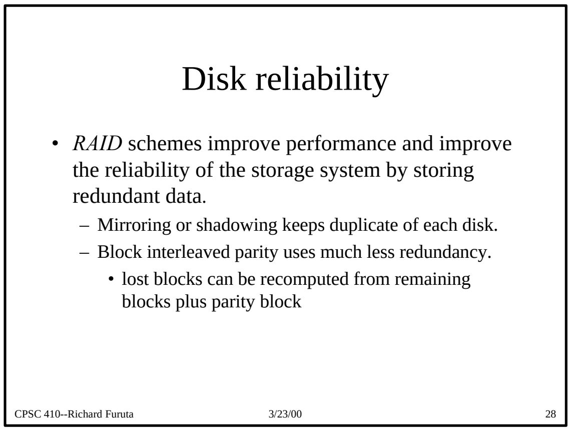
Swap space management

- 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 page is forced out of physical memory, not when the virtual memory page is first created.



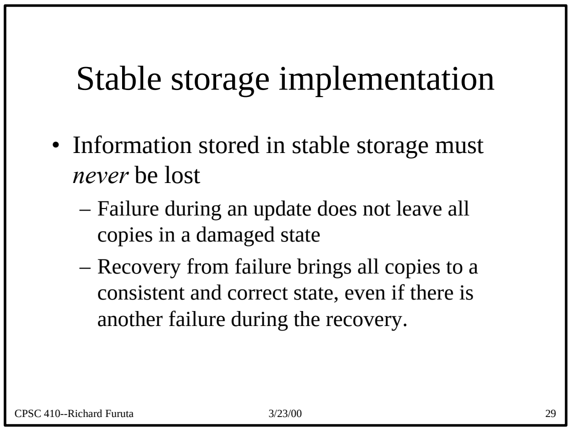
Disk reliability

- 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*: *Redundant Array of Independent Disks*



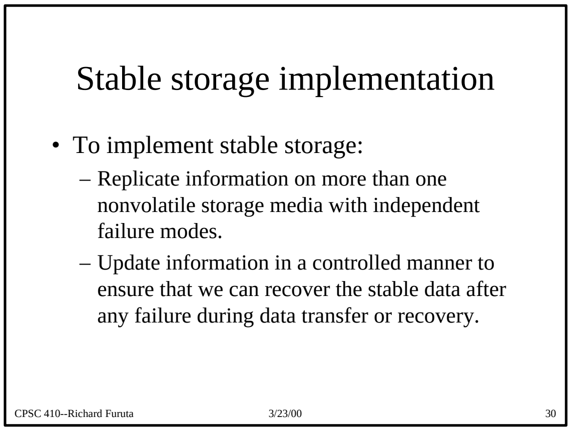
Disk reliability

- *RAID* schemes improve performance and improve the reliability of the storage system by storing redundant data.
  - Mirroring or shadowing keeps duplicate of each disk.
  - Block interleaved parity uses much less redundancy.
    - lost blocks can be recomputed from remaining blocks plus parity block



Stable storage implementation

- Information stored in stable storage must *never* be lost
  - Failure during an update does not leave all copies in a damaged state
  - Recovery from failure brings all copies to a consistent and correct state, even if there is another failure during the recovery.



Stable storage implementation

- 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.

Stable storage implementation example

- Disk write results in one of these outcomes
  - Successful completion
  - Partial failure (middle of transfer)
  - Total failure (previous values remain intact)
- Maintain two copies of block and follow this protocol for writes
  - Write information on first physical block
  - When write completes successfully, write same information onto second physical block
  - Declare the operation complete only after the second write completes successfully

Stable storage implementation example

- Recovery from failure
  - inspect each pair of physical blocks
    - both the same and no detectable error exists: no further action required
    - one block contains detectable error: replace with other
    - both blocks have no detectable error but differ in content: replace content of first with second
  - write to stable storage either succeeds completely or results in no change

Tertiary Storage Devices

- Defining characteristic: low cost
- Generally built using *removable media*
- Examples: floppy disks, CD-ROM, …
  - Floppy disks: thin flexible disk coated with magnetic material, enclosed in a protective plastic case
  - Optical disks: materials that are altered by laser light to have spots that are relatively light and dark
    - Phase-change disk: crystalline or amorphous state
    - Dye-polymer disk: laser heat makes bumps, warms bumps to remove them

Magneto-optic disk

- Magnetic material covered with protective layer of plastic or glass; head much farther from disk than with magnetic disk; less susceptible to head crashes
- Laser heat makes spot susceptible to magnetic field (records)
- Laser light polarization when bouncing off of magnetic spot used for reading (Kerr effect)

Removable disks

- Read-write disks
  - Magnetic disks, magneto-optic disks, optical disks
- Write-once, read many (WORM)
  - One example: thin aluminum film sandwiched between two glass or plastic platters; holes burnt through aluminum; information can be destroyed but not altered
  - Another example: CD-R
- Read-only disks
  - Examples: CD-ROM and DVD

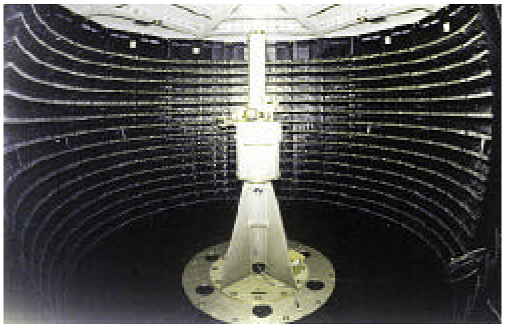
Robotic jukebox for Magneto-optical disks

Magnetic tape

- Compared to disk:
  - Less expensive, holds more data, random access much slower
  - Robotic tape installations
    - Stacker: library that holds a few tapes
    - Silo: library that holds thousands of tapes
  - *Archive* disk resident tapes for low-cost storage. *Stage* back into disk storage for active use

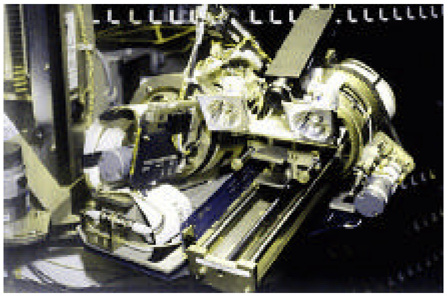
Magnetic tape

- Tape silo at Jefferson Lab (http://www.jlab.org/ccc/silo/info/)
  - One terabyte of data a day received
  - 6000 tapes
  - 50 gigabytes per tape at present
  - 300 terabytes total storage
  - Expected enhancements up to over a petabyte (1,000 terabytes) of "near-line" storage



Interior of the STK 4410 before the robotics arm was installed.

STK Personnel set up central robotics arm.



The robot's "head" contains two cameras and two hands for information retrieval.

Operating System Issues

- Major OS jobs
  - Manage physical devices
  - Present a virtual machine abstraction to applications
- Hard disk abstractions
  - Raw device: an array of data blocks
  - File system: 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: raw device. Application opens whole tape drive rather than file on tape.
  - Tape drive reserved for exclusive use of application
  - Application decides how to use the array of blocks
  - Tape's format is generally specific to the program that created it

Tape drives

- Basic operations for tape drives differ from those of a disk drive
- **Locate**: position tape to specific logical block (instead of **seek**)
  - Locate 0 is the same as rewinding
- **Read position**: current logical block
- **Space**: relative movement over logical blocks
  - Space -2: go back two logical blocks
- Append-only devices. Update effectively erases everything past that block
- EOT mark follows last block on tape

Speed

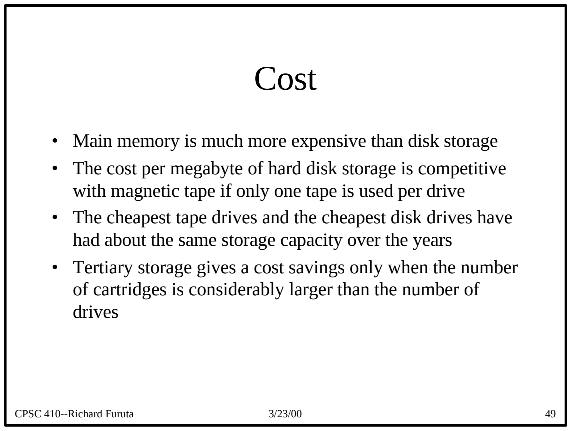
- Tertiary storage aspects of speed: *bandwidth* and *latency*
- Bandwidth: measured in bytes per second
  - *Sustained bandwidth*--average data rate during a large transfer; number of bytes/transfer time (this is the data rate when the data stream is actually flowing)
  - *Effective bandwidth*--average over the entire I/O time, including **seek** or **locate**, and cartridge switching (this is the drive's overall data rate)

Speed

- 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; generally less than 35 milliseconds
  - Access time on tape requires winding 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.
- Access times on jukebox or tape silo (robotic arm) also requires time to remove (including possibly a return to a consistent state), locate, and load media. Hence removable library best for infrequently used data.

Reliability

- Fixed disk drive is likely to be more reliable than removable disk or tape drive
- Optical cartridge is likely to be more reliable than a magnetic disk or tape
- 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
- Recently, much controversy over lifetimes of CD-ROM
  - Manufactured CD-ROMs versus CD-R (predictions in both directions)
  - Years or decades?



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