

CISC 7310X

C11: Mass Storage

Hui Chen

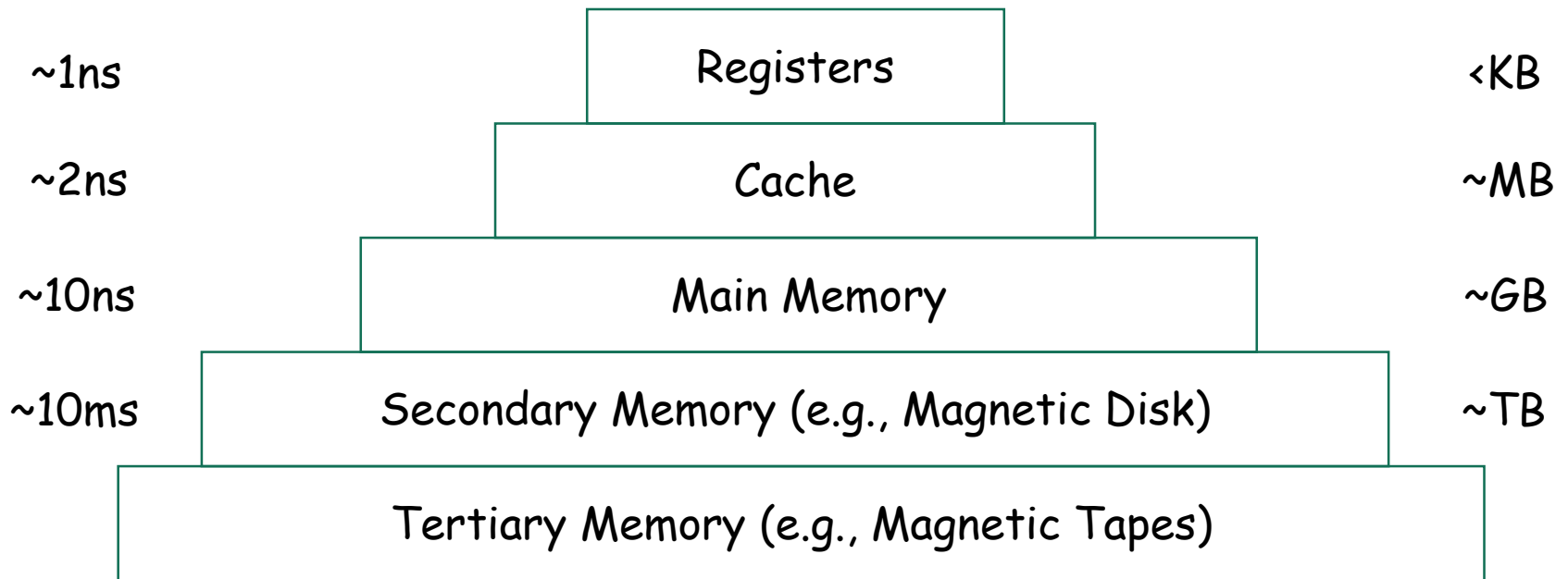
Department of Computer & Information Science

CUNY Brooklyn College

Outline

- Review of memory hierarchy
- Mass storage devices
- Reliability and performance
 - Redundancy and parallelism
 - Disk arm scheduling
- Error handling and stable storage
- Further reading
- Assignment

Memory Hierarchy

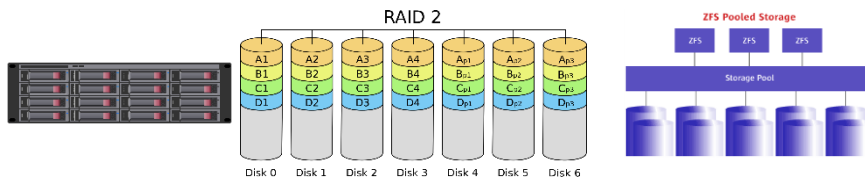


I/O Design

- User and programming interfaces
- Internal data structures and algorithms
- Secondary and tertiary storage structures



Data structure & algorithms



Focus of the discussion

Design Goals

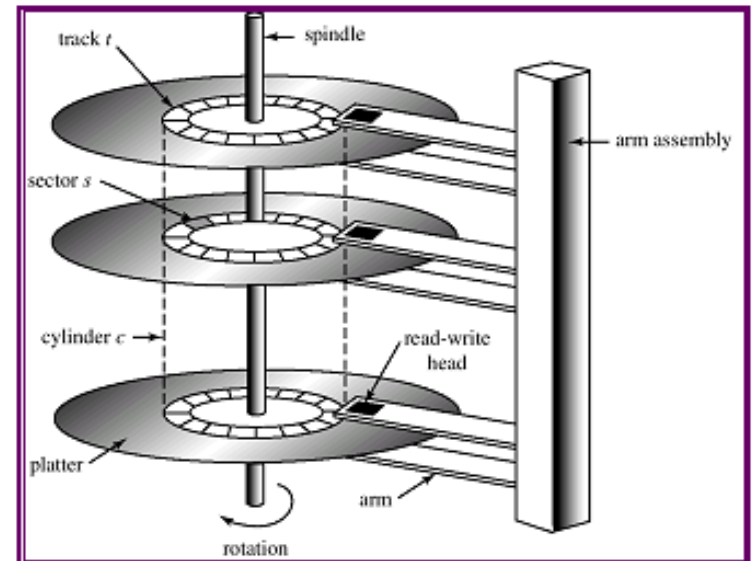
- Function
 - They need to work, read & write
- Reliability
 - Murphy's law
 - "Anything that can go wrong will go wrong"
 - How do we make it appearing reliable?
- I/O Efficiency (performance)
 - We need it to be "fast"
- Energy efficiency

Mass Storage Devices

- Disks
 - Magnetic disks
 - Solid state drives
 - Optical disks

Magnetic Disks

- Disk arm
- Read-write head
- Platter
 - 60 ~ 200 rounds/second
 - 1.8 ~ 5.25 inches
- Tracks
 - Sectors
- Cylinder



Magnetic Disks: Characteristics

- Size and rotation speed
- Transfer rate
 - The rate at which data flow between the drive and the host
- Random-access time (or position time)
 - Random-seek time (or seek time)
 - time required to move the arm to the desired cylinder
 - Rotational latency
 - Time required for the desired sector to rotate to the disk head
- Head crash
 - Read-write head make contact with the platter surface
- Bit rot
 - Degradation of data, such as, as the result of gradually lose of magneticity

Magnetic Disks: Improvement

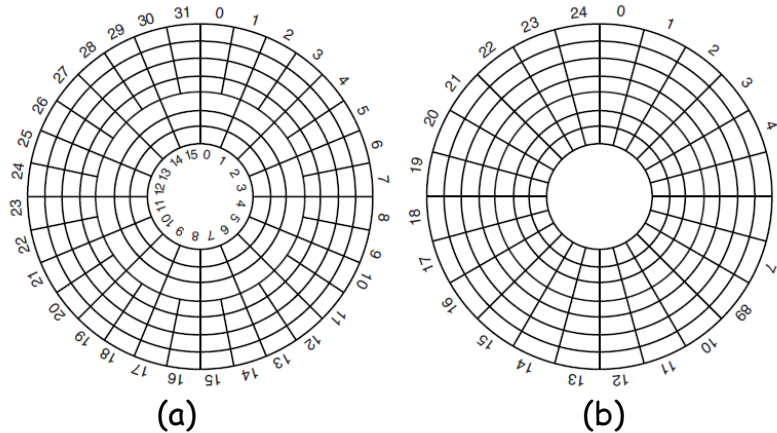
- Improvement over time
 - Gradual improvement of moving parts and more rapid improvement of bit densities

	1978	2008
Parameter	IBM 360-KB floppy disk	WD 3000 HLFS hard disk
Number of cylinders	40	36481
Tracks per cylinder	2	255
Sectors per track	9	63 (avg)
Sectors per disk	720	586,072,368
Bytes per sector	512	512
Disk capacity	360 KB	300 GB
Seek time (adjacent cylinders)	6 msec	0.7 msec
Seek time (average case)	77 msec	4.2 msec
Rotation time	200 msec	6 msec
Time to transfer 1 sector	22 msec	1.4 μ sec

- [Figure 5-18 in Tanenbaum & Bos, 2014]

Magnetic Disks: Zones

- Logical (virtual) geometry and physical geometry are different
 - Traditionally, (x, y, z) : (cylinders, heads, sectors), i.e., CHS
 - PC: (65535, 16, 63), a sector is typically 512 bytes
 - Modern approach: logical block addressing (LBA), disk sectors numbered consecutively starting at 0



- [Figure 5-19 in Tanenbaum & Bos, 2014] Figure 5-19. (a) Physical geometry of a disk with two zones. (b) A possible virtual geometry for this disk

Disk Structure

- Logical blocks
 - To the host, the disks are a one-dimensional array of logical blocks
 - Smallest unit of data transfer between the disk and the host
- Disk sector
 - A logical block mapped to one or more disk sectors
 - A sector is typically $2^9 = 512$ bytes
 - LBA or CHS
- Constant linear velocity
 - Optical disks
- Constant angular velocity
 - Magnetic disks

I/O Bus

- Interface to host via I/O bus
 - Enhanced integrated drive electronics (EIDE)
 - Advanced technology attachment (ATA)
 - Serial ATA (SATA)
 - Universal serial bus (USB)
 - Fiber channel (FC)
 - Small computer-systems interface (SCSI)

Host and Disk Controllers

- Controllers, responsible for data transfer via the I/O bus
 - Host controller (Host Bus Controller/Adaptor, HBA, e.g., on the motherboard of a PC)
 - CPU places a command to the controller, typically using memory-mapped I/O ports
 - LBA: present the disk to the host as a consecutively numbered sectors
 - Disk controller (inside the disk drive)
 - Host controller sends the command via messages to disk controller
 - Build-in memory buffer/cache
 - Data transfer between platters and cache, limited by speed of physical movements
 - Data transfer between the host controller and the cache, at faster electronic speeds

Questions?

- Review of memory hierarchy
- Concepts about magnetic disk

Reliability and Performance

- Performance
 - Parallel processing
- Reliability
 - Redundancy

Redundancy and Data Loss

- Failure = lost of data; Mean time to failure (MTBF) of a single disk drive and many disk drives
 - Example
 - MTBF of a single disk drive: 1,000,000 hours
 - 100 disk drives: $1,000,000/100 = 10,000$ hours = 416.7 days
- Mirroring: duplicate disk drive
 - Example: two physical disk drives are presented as a logical disk drive (mirrored volume)
 - Failure: failure of two physical disk drives
- Mean time to repair (MTBR): time required to replace the failure disk drive
- With mirroring
 - $MTBF = 1,000,000$ hours; $MTBR = 10$ hours
 - Mean time to data loss: failure of the mirrored volume (the 2nd disk drive also failed before the 1st could be replaced) : $1,000,000^2 / (2 \times 10) = 5 \times 10^{10}$ hours = 5.7×10^6 years!

Practical Consideration

- Disk failures are not independent
- Example
 - Large number of disk drive failures can be the result of a power failure and a tremor of a minor earthquake
 - Manufacturing defects in a batch of disk drives can lead to correlated failures
 - The probability of failure grows as disk drives age
- Mean time to data loss is smaller

Parallelism and Performance

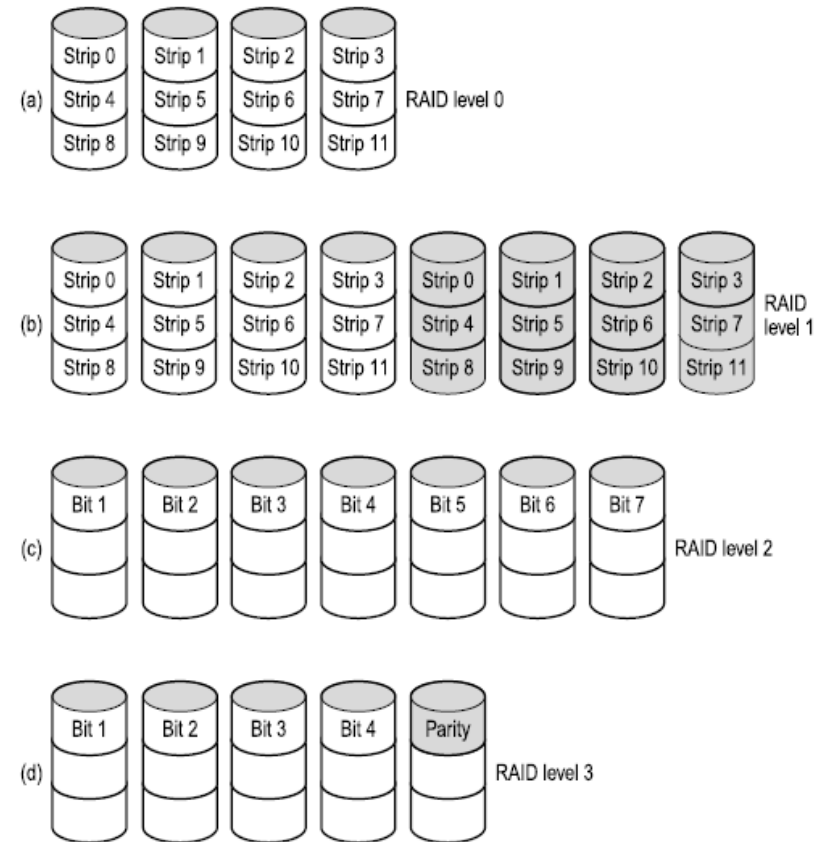
- Observation
 - Duplicating disk drives doubles the rate at which read requests can be handled
- Data stripping
 - Splitting data across multiple disk drives
 - Bit-level stripping
 - Splitting the bits of each byte across multiple disk drives
 - Example: using an array of 8 disks (or a factor of 8 or a multiple of 8)
 - Block-level stripping
 - Splitting blocks of a file across multiple disk drives
- Benefits
 - Increase the throughput of multiple small accesses (page accesses)
 - Reduce the response time of large accesses

Redundant Array of Inexpensive Disks (RAID)

- 6 levels (not counting 0)
- RAID Level 0
- RAID Level 1
- ...
- RAID Level 6

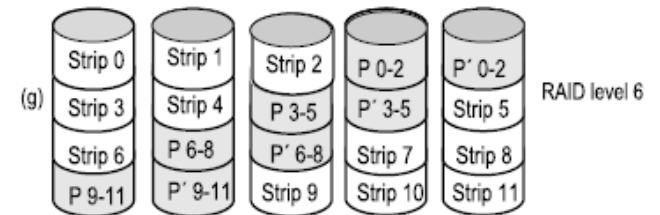
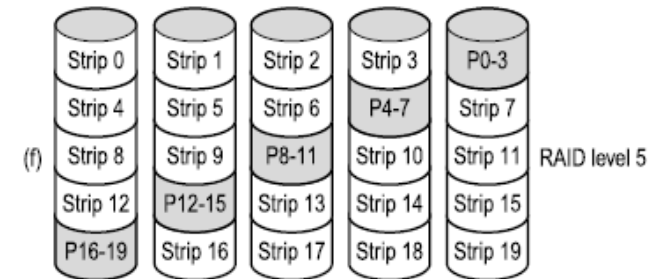
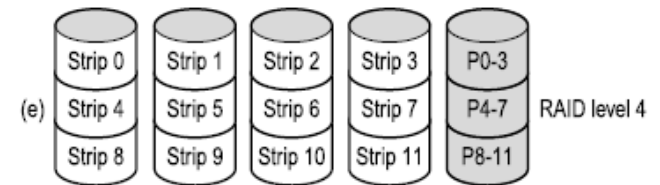
RAID Level 0 - 3

- Level 0 (not a true RAID): Striping only
- Level 1: Mirror + striping
- Level 2: Memory-style error-correction-code (ECC) organization
 - Example
 - Split a byte into 4-bit nibbles
 - Add Hamming code (3 bits) to each
 - One bit per drive onto 7 disk drives
- Level 3: Bit-interleaved parity organization
 - Compute a parity bit
 - Driver detects error, a parity bit is sufficient to correct error (unlike memory)
- Backup and parity drives are shown shaded.



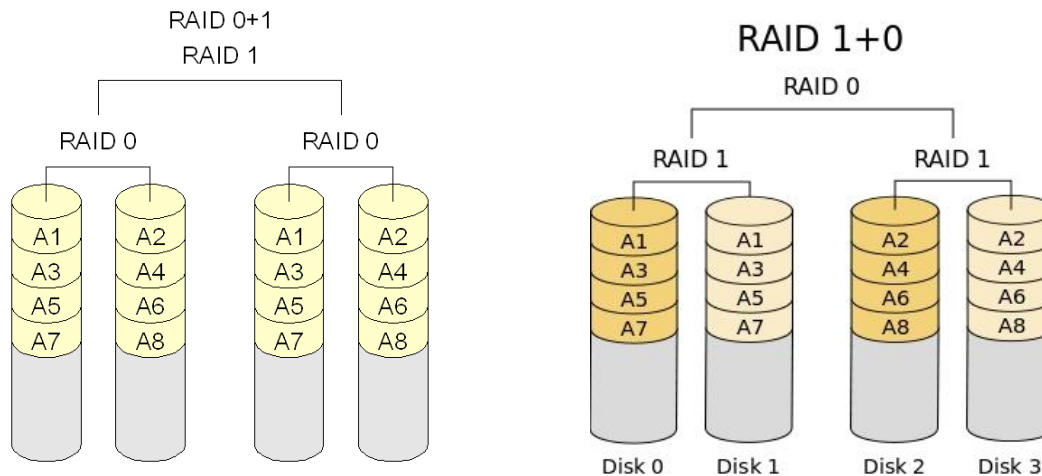
RAID Level 4 - 6

- Level 4: block-interleaved parity organization
 - Stripping
 - Compute parity for blocks
 - One disk for parities
- Level 5: block-interleaved distributed parity
 - Stripping
 - Compute parity for blocks
 - Parities are distributed
- Level 6: P+Q redundancy scheme
 - Extra redundant information to guard multiple disk failures
- Backup and parity drives are shown shaded.



RAID 0+1 and 1+0

- Combination of RAID levels 0 and 1
- RAID 0+1: striping and then mirroring
- RAID 1+0: mirroring and then striping

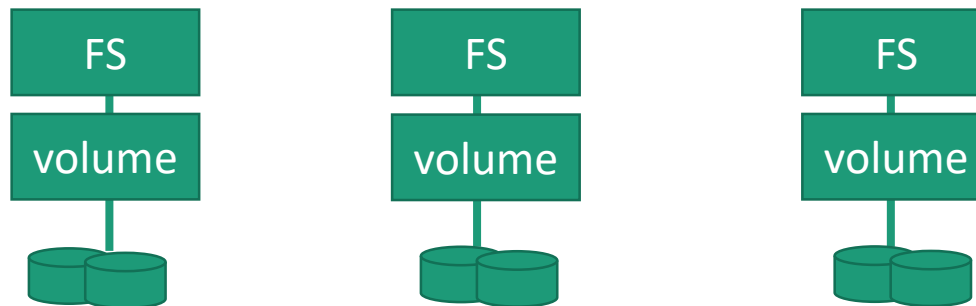


Selecting RAID Level

- RAID 0: high performance, data loss is not critical
- RAID 1: high reliability with fast recovery
- RAID 0+1 & 1+0: both performance and reliability
 - Expense: 2 for 1
- RAID 5:
 - Often preferred for large volumes of data
- Required number of disks?

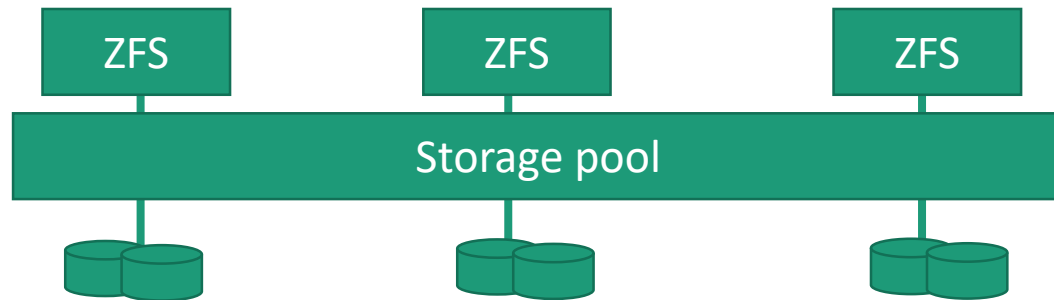
Limitation of RAID

- RAID only protects physical media errors, but not other hardware and software errors
- RAID is not flexible
 - Present a disk array as a volume
 - What if a file system is small, or large, or change over time?



Solaris ZFS

- Combines file-system management and volume management into one unit
 - Maintains a checksums of all blocks and data structures to support error detection and recovery
 - A storage pool holds one or more ZFS file systems
 - ZFS uses "allocate" and "free" model makes the storage pool available to each file system



Questions

- Reliability and performance
- Performance via parallelism
- Reliability via redundancy
- RAID
 - Which level to use? How many disks?

Disk Formatting

- Low-level formatting
- Partition and Logical formatting

Low-level Formatting

- Each disk must receive a low-level format done by software
 - To form a series of concentric tracks
 - each consists of some number of sectors
 - with short gaps between sectors
- Typically performed at the factory as the part of the manufacturing process

Disk Sector

- Preamble
 - A bit pattern that allows the hardware to recognize the start of the sector
- Data
 - Typically 512 bytes, some allows a few sizes (e.g., 256, 512, 1024)
- Error-correction code (ECC)
 - Computed based on the data to correct bit errors in data



A Disk Sector

Partition and High-level Formatting

- Partition
 - Divide the disk into one or more groups of cylinders
 - OS treats each partition as though it were a separate disk
- Logical formatting
 - Creating a file system on a partition, i.e., storing the initial file-system data structures onto the partition
 - Example
 - Maps of free and allocated space, an initial empty directory
 - Most file systems group blocks together into larger chunks, called clusters
 - Disk I/O: minimal unit is a block
 - File system I/O: minimal unit is a cluster

Boot Block

- An OS must have an initial bootstrap program
- Typical design
 - The bootstrap program is in a read-only memory (ROM) with a fixed address
 - It loads and runs another small program stored at a fixed location on the disk, called the "boot block"
 - Example
 - Windows places its boot code and a partition table in the first sector of the hard disk (sector 0, called Master Boot Record, or MBR)
 - The boot code loads another small program at the first sector (called the boot sector) of a designated partition, called the boot partition
 - The code in the boot sector loads the OS and the device drivers in the boot partition.

Questions?

- Disk formatting
 - Low-level formatting
 - Partition and logical formatting
 - Formatting and boot block

Seek Time and Disk Arm Scheduling

- Recall the factors of a disk block read/write performance
 - Seek time (the time to move the arm to the proper cylinder).
 - Rotational delay (how long for the proper sector to come under the head).
 - Actual data transfer time.
- The seek time often dominates.
- Can we reduce the seek time?

Disk Arm Scheduling Problem

- Assumption
 - A disk driver accepts bursts of access requests
- Problem
 - In which order should these requests be carried out to minimize the seek time?
 - Average seek time?
 - Overall seek time?

Disk Arm Scheduling Algorithms

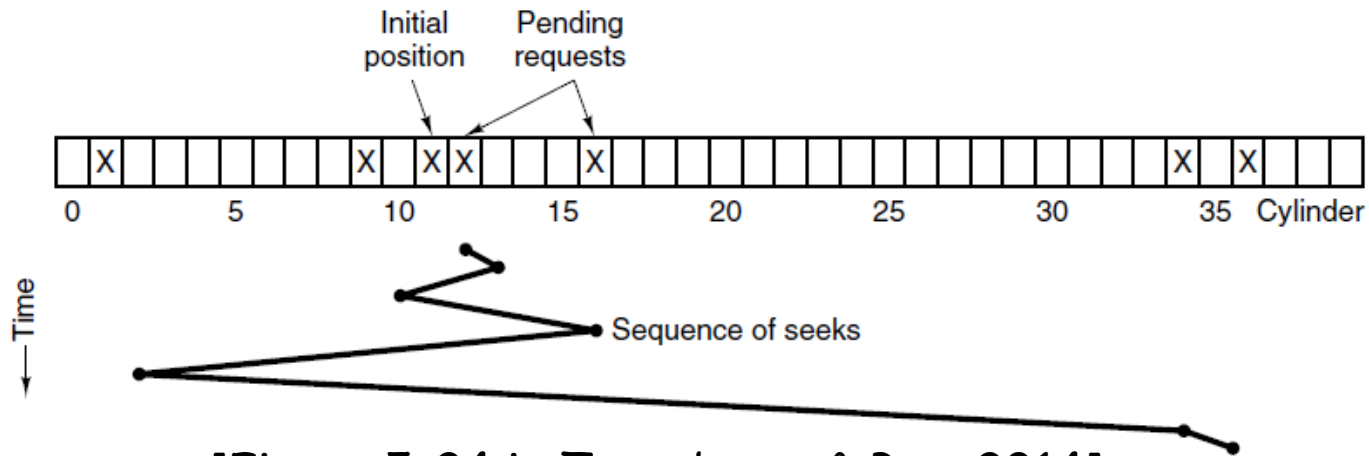
- First-Come, First Served (FCFS)
- Shortest-Seek-First/Shortest-Seek-Time-First (SSF/SSTF)
- The elevator algorithm
 - LOOK

First-Come, First Served

- Requests: when 11 being sought to, request to seek to cylinders 1, 36, 16, 34, 9, and 12 arrived
- Total arm movements
 - $|11 - 1| = 10$
 - $|1 - 36| = 35$
 - ...

Shortest-Seek-First

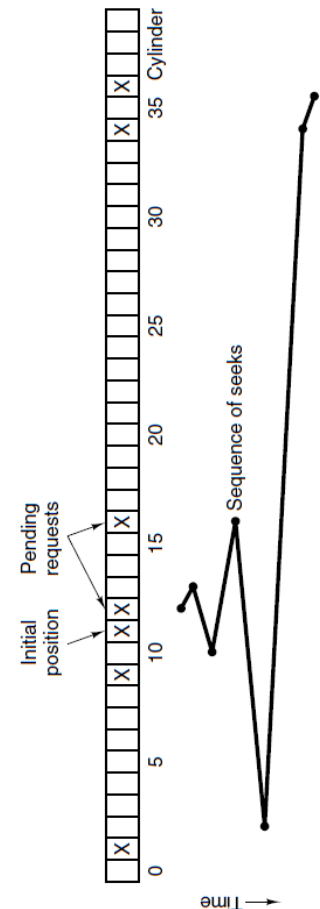
- Requests: when 11 being sought to, request to seek to cylinders 1, 36, 16, 34, 9, and 12 arrived
- Resulting sequence: 12, 9, 16, 1, 34, 36



- [Figure 5-24 in Tanenbaum & Bos, 2014]

Elevator in a Tall Building

- The arm scheduling is like scheduling an elevator in a tall building
- Shortcoming of STFS
 - Fairness
 - In a busy system, the arm tends to stay at the middle of the disk
- Elevator scheduling ...

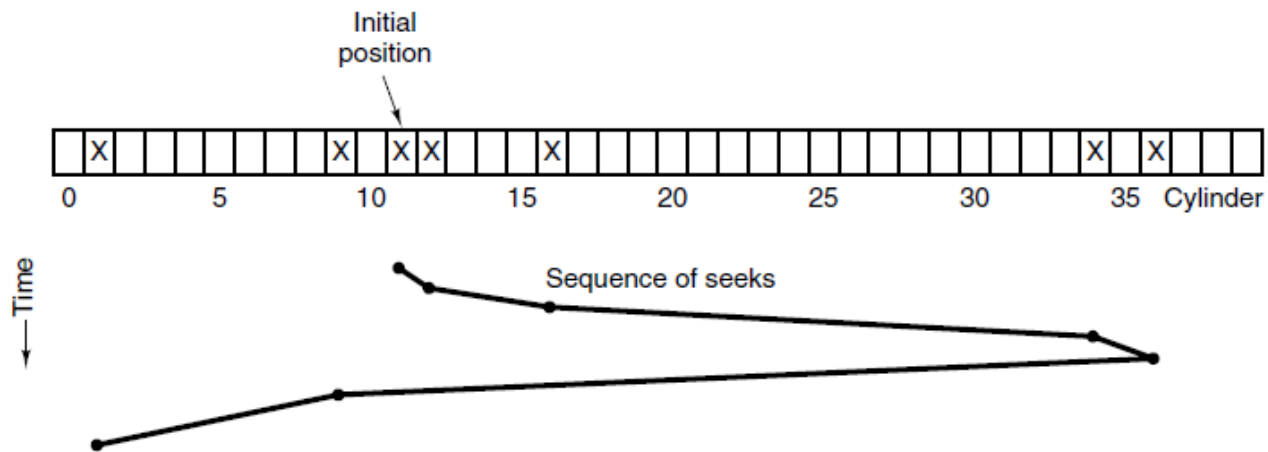


LOOK

- Maintains a flag whose value is up or down
- If the flag is up
 - If having higher request (cylinder bigger)
 - Seek to the cylinder
 - Else
 - Reverse the flag
- If the flag is down
 - If having lower request (cylinder lower)
 - Seek to the cylinder
 - Else
 - Reverse the flag

LOOK: Example

- Requests: when 11 being sought to, request to seek to cylinders 1, 36, 16, 34, 9, and 12 arrived



- [Figure 5-25 in Tanenbaum & Bos, 2014]

Selecting Scheduling Algorithm

- Performance: SSF is common, as it increases performance over FCFS
- Fairness: The Elevator algorithm balances fairness and performance for busy systems
- Computation: It must finish computing the schedule before the seek is completed
- Number and types of requests
- File allocation method: Are files continuously allocated on the disk?
- Locations of directories and index blocks
 - To read the file, we need first read the directory/index blocks data to locate the file
 - Caching the directories and index blocks?
- Disk-scheduling algorithms should be written as a separate module, and can be replaced if necessary

Questions?

- Disk arm scheduling algorithms
 - No optimization
 - FCFS
 - Reduce seek time
 - SSF
 - Balance fairness and performance
 - LOOK

Bad Blocks and Sectors

- Murphy's Law
 - "Anything that can go wrong will go wrong".
- How do we handle error?
- A disk can have bad blocks or bad sectors
 - A read does not produce the data just written to it
 - ECC can correct small number of bits errors

Error Handling of Bad Blocks

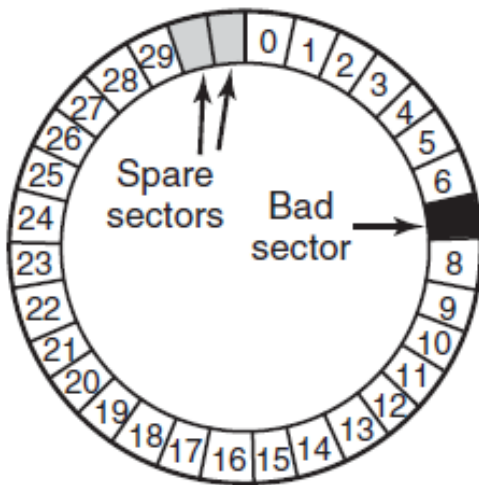
- Deal with them in the controller
- Deal with them in the OS

Spare Substitution by Disk Controller

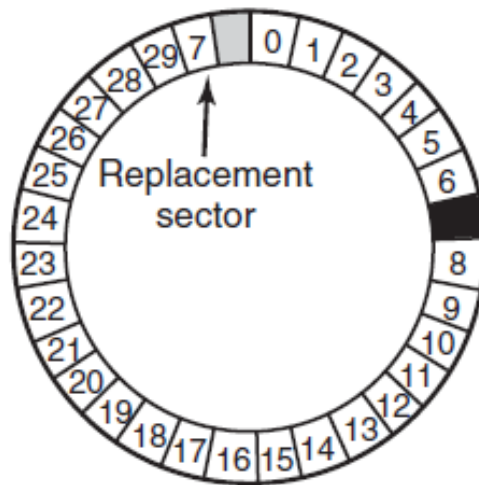
- Disk controller handles the bad sectors
 - During the testing at the factory as the manufacturing process
 - Disk comes with spare sectors
 - Replace each bad sector by a spare
 - These steps are also performed during normal operation
 - Transient errors are overcome by repeated read attempts
 - Repeated errors trigger spare substitution

Spare Substitution

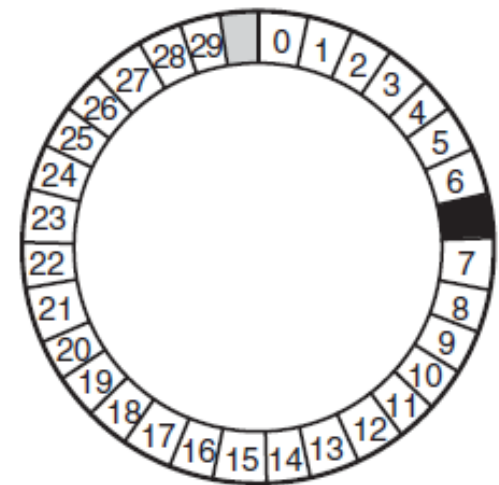
- Two approaches shown in (b) and (c)



(a)



(b)



(c)

- [Figure 5-26 in Tanenbaum & Bos, 2014]

Questions?

- Error handling

Stable Storage

- An “all or none” property
- A property that a write request is issued to the disk, the disk either correctly writes the data or it does nothing (and reporting error)

Error Model

- The system detects an error via the value of the ECC field
- Example
 - 16-byte ECC, 512-byte sector
 - Data values: $2^{512 \times 8} = 2^{4096}$
 - ECC values: $2^{16 \times 8} = 2^{128}$
 - There is always a tiny chance that an error is not detected.

Design of Stable Storage

- Uses pair of identical disks
- Either can be read to get same results
- Operations defined to accomplish this:
 1. Stable Writes
 2. Stable Reads
 3. Crash recovery

Stable Writes

- Foreach disk drive d do
 - Set n as 0
 - While (Error && $n < MAX$)
 - First write the block on disk drive d
 - Read it back
 - Verify correctness via ECC (Error or not error)
 - Increment n
 - If Error, report error

Stable Readers

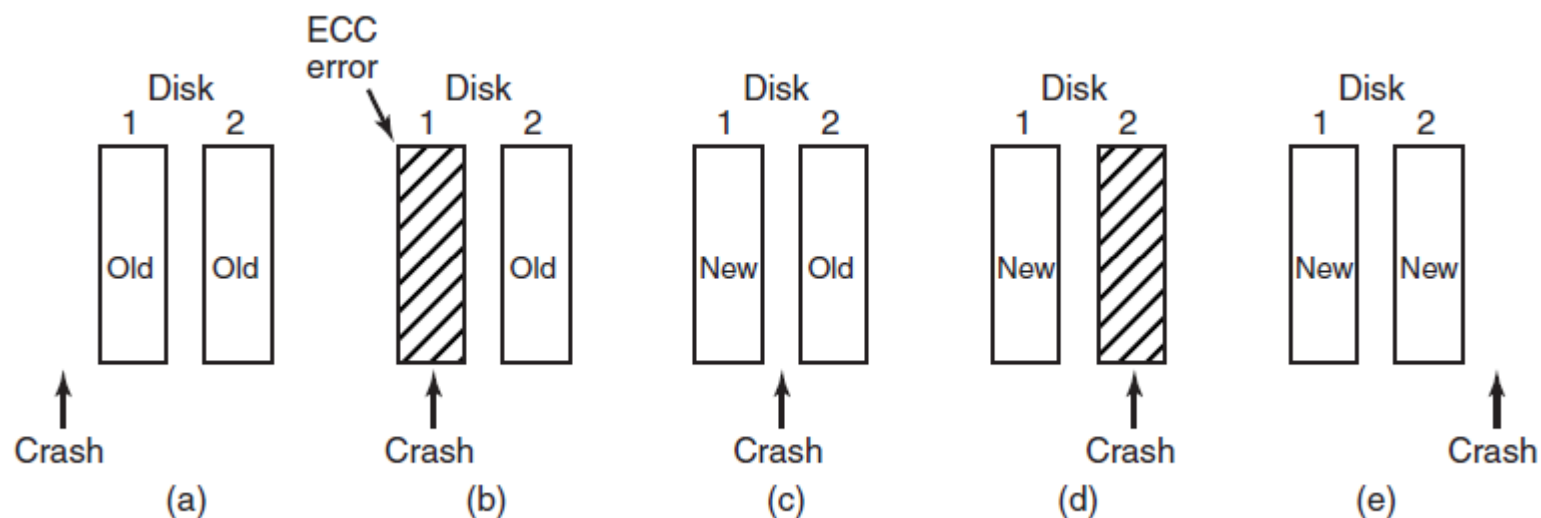
- Foreach disk drive d do
 - Set n as 0
 - While (Error && $n < MAX$)
 - First read the block on disk drive d
 - Verify correctness via ECC (Error or not error)
 - Increment n
 - If Error, report error

Crash Recovery

- After an OS crash, a recovery program scans both disks and comparing the blocks
- If a pair of blocks are identical, nothing is done
- If one of the pair has an ECC error, the bad block is overwritten with the corresponding good one
- If both have no ECC error, but are different, the block from drive 1 is written on drive 2 (the block on drive 1 is newer).

Influence of Crash

- Analysis indicates the storage is stable



- [Figure 5-27 in Tanenbaum & Bos, 2014]

Questions?

- Stable storage
- Design

Further Reading

- Disk Attachment
 - Host-attached storage
 - Network-attached storage
 - Storage-area network

Assignments

- Group and individual