Chapter 10 – Outline

- Overview of the disk structure
- Disk scheduling and management
- RAID Structure
- Operating System Issues
Disks

- Over 30 year old storage technology.
- Very complicated
  - modern drive uses over 250,000 lines of micro code
Disk Access Time

- **Seek time**
  - the time to position disk heads (~8msec on average)

- **Rotational latency**
  - time to rotate the track sector under the disk head
  - assume 7,200 rotations per minute (7,200 RPM)
  - \( \frac{7,200}{60} = 120 \) rotations per second
  - \( \frac{1}{120} = \) ~8msec per rotation
  - average rotational delay is ~4msec

- **Transfer time**
  - the time to transfer bytes from disk to controller
  - for 58 Mbytes/sec and 4 Kbyte disk blocks
    - time to transfer a block is 0.07 msec
Disk Performance Metrics

- Disk access time
  - Seek time + Rotational latency + Transfer time

- Latency
  - Seek Time + Rotational latency

- Bandwidth
  - bytes transferred / disk access time
Example of Disk Bandwidth

- If disk blocks are randomly accessed
  - average disk access time = ~12 msec
  - assume 4 kbyte blocks
  - bandwidth is: $4 \text{ kbyte} / 12 \text{ msec} = \sim 340 \text{ kbyte/sec}$
- If disk blocks of the same cylinder are randomly accessed without disk seeks
  - average disk access time = ~4 msec
  - bandwidth is: $4 \text{ kbyte} / 4 \text{ msec} = \sim 1 \text{ Mbyte/sec}$
- If disk blocks are accessed sequentially
  - without seeks and rotational delays
  - bandwidth is: 58 Mbytes/sec

*Key to good disk performance is to minimize seek time and rotational delay.*
Disk Structure

- Disk drives are addressed as large one-dimensional arrays of *logical blocks*.
- Logical block is the smallest unit of transfer.
- The one-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
Solid-State Disks

• Nonvolatile memory used like a hard drive
  • many technology variations
• Can be more reliable than HDDs
  • more expensive per MB
  • maybe have shorter life span
  • less capacity
  • but much faster
• Busses can be too slow
  • connect directly to PCI for example
• No moving parts, so no seek time or rotational latency
Host and disk controllers facilitate data transfer from disk to computer on an I/O bus.

Few popular disk controller standards
- IDE (Integrated Device Electronics)
- Parallel and Serial ATA (AT Attachment Interface)
- SCSI (Small Computer Systems Interface)

Differences
- performance
  - IDE < PATA < SATA < SCSI
- parallelism
  - IDE, PATA are parallel interfaces
  - SATA is serial
  - SCSI can be parallel or serial
Disk Scheduling

- Flashback
  - *seek time* is the time for the disk to move the read/write head to the cylinder containing the desired sector
  - *rotational latency* is the additional time waiting for the disk to rotate the desired sector to the disk head
  - seek time and rotational latency dominate disk access time!
  - seek time $\approx$ seek distance

- Algorithms to schedule the servicing of disk I/O requests
  - only attempt to minimize seek time
  - schedule a request from a *queue* of pending disk I/O requests
  - consider example disk queue with requests to blocks on cylinders 98, 183, 37, 122, 14, 124, 65, 67
  - current location of head pointer: cylinder 53
First Come First Serve (FCFS)

- Requests are served in order of arrival
  - fair among requesters
  - poor for accesses to random disk blocks
  - for example: total head movement of 640 cylinders
Shortest Seek Time First (SSTF)

- Picks request that is closest to the current disk arm position
  - good at reducing seeks
  - may result in starvation
  - not optimal (unlike SJF)
  - example requires total head movement of 236 cylinders.

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

Diagram showing seek operations with labels on the horizontal axis.
SCAN

- Start disk arm at one end of the disk, and move toward the other end
  - servicing closet request in the direction of traversal
  - proceed backwards along same path (like elevator)
  - no starvation
  - a new request can wait for almost two full scans of the disk
  - example requires total head movement of 236 cylinders
queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
Circular – SCAN

- Disk arm always serves requests by scanning in one direction
  - once the arm finishes scanning for one direction
  - returns to the 0th track for the next round of scanning
  - provides a more uniform wait time than SCAN
LOOK Scheduling

- Modified version of 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
Redundant Array of Independent Disks (RAID)

- Used to improve disk I/O performance and reliability
  - using additional or redundant disks
- Reliability
  - maintain extra information to allow recovery from disk failures
- Performance
  - store information on multiple smaller disks
  - data transfer from disks in parallel
- Bit-level striping
  - split bits of each byte across multiple disks
  - with $N$ disks, bit $i$ and $N+i$ are stored on disk $i$
- Block-level striping
  - blocks are striped across multiple disks
RAID Levels

- Reliability-performance schemes with different cost-performance tradeoffs.
- RAID level 0
  - employ bit-level or block-level striping for performance
  - no redundancy for reliability
- RAID level 1 (disk *mirroring*)
  - duplicate every disk in RAID array
  - simple
  - expensive
  - can provide better performance as well as reliability
RAID Levels (2)

- RAID level 2 (memory-style ECC organization)
  - redundant disks hold *parity* information
  - single bit error detection requires 1-bit parity, single bit error correction needs two or more parity bits
  - uses bit-striping for improved performance
  - less expensive than RAID level 1

- RAID level 3 (bit-interleaved parity organization)
  - uses single parity bit (disk) for error correction
  - we can easily figure out which sector on disk is damaged
  - use single parity bit to figure each bit on the damaged disk
  - as good as level 2, but less expensive
  - less storage overhead than level 1
  - bit-interleaved, so every disk has to participate in each I/O request
RAID levels (3)

- RAID level 4 (block-interleaved parity organization)
  - similar to RAID level 3 with block-level striping scheme
  - supports more simultaneous I/O requests than level 3
  - however, data transfer rate for each access is slower
  - writes smaller than a block needs 2 block reads and 2 writes, why?
  - how about write of a complete block?

- RAID level 5 (block-interleaved distributed parity)
  - data and parity spread among all N+1 disks
  - avoids overuse of the parity disk
  - most commonly adopted RAID scheme

- RAID level 6 (P+Q redundancy scheme)
  - uses extra information to guard against multiple disk failures
RAID Levels (4)

(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.