Recap

Bus system to connect components inside a computer.

I/O device's generic interface: status, command and data registers.

Protocols to interact with I/O devices: polling (programmed I/O), interrupt, DMA (direct memory access).

Device drivers as part of OS.

Hard Disk Drives

(based on Ch. 37)

Slides revised from
Matthew Tancreti
Iowa State University

Hard Disk Drives (HDD)

HDD have been main form of persistent data storage for decades

Most of file systems were developed around their behavior

How to store and access data on disk?

Interface

The interface for a drive is virtualized by the drive's controller

From OS perspective the **address space** of the drive is divided into 512-byte blocks called **sectors**

Writing to a block is **atomic**

- Either write to block fully succeeds or it doesn't change anything
- No partial write to block, even if power goes out in the middle

Multi-sector reads and writes are supported, but no guarantee of atomicity

- If power goes out during multi-sector write, **torn write** possible
- It is software's responsibility to handle torn writes

Geometry

Platter – circular hard surface where data is stored using magnetic persistence

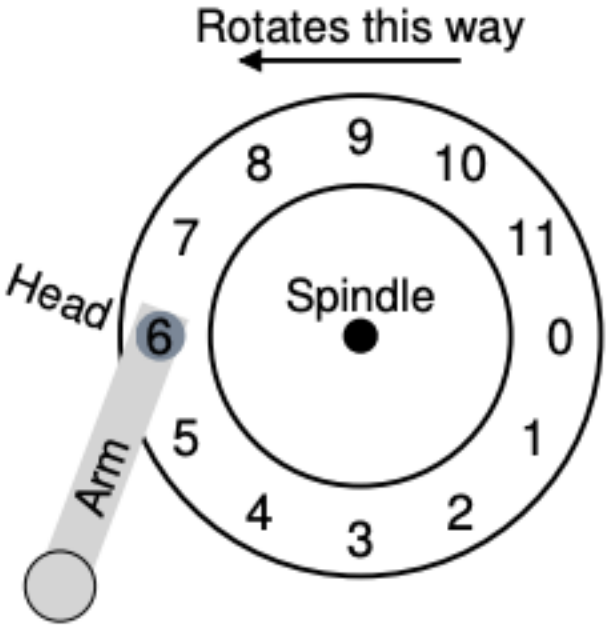
Surface – a platter has a top and bottom surface, both can store data

Track – each circle around the center (a surface consists of thousands of tracks)

Spindle – the center of the platter which connects to a motor

Disk Arm – moves the head across the platter to select a specific track

Disk Head – transforms the magnetic field into electrical current

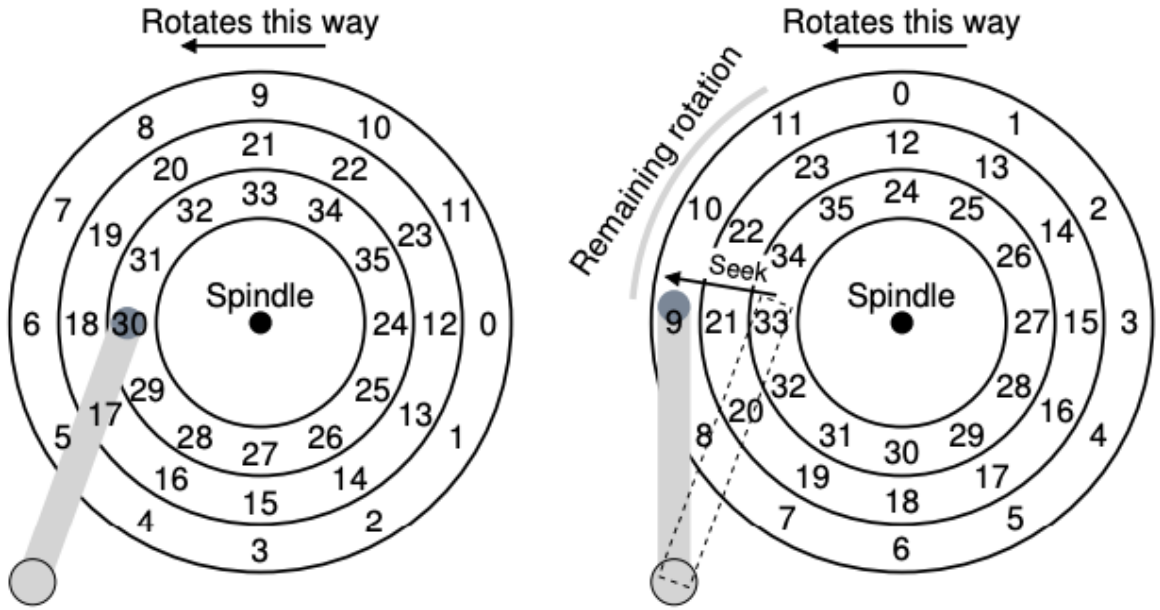


Latencies

Seek time – time for disk arm to change position to the correct track

Rotational delay – time for sector to rotate under the disk head

$$T_{I/O} = T_{seek} + T_{rotation} + T_{transfer}$$



Examples

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects via	SCSI	SATA

Random vs. Sequential

Random – software requests addresses in any order it choses

1, 20, 5, 6, 25, 10

can result in very high seek time

Sequential – software requests addresses in increasing

1, 2, 3, 4, 5, 6

only need to seek for the first access

Examples

	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects via	SCSI	SATA

	Cheetah	Barracuda
$R_{I/O}$ Random	0.66 MB/s	0.31 MB/s
$R_{I/O}$ Sequential	125 MB/s	105 MB/s

Scheduling

Multiple processes making concurrent requests can result in frequent seeking

Order of requests by an application can also increase seek time

Solution is to use a scheduling policy

- Maintain an ordered queue of I/O requests by track
- **Shortest Seek Time First (SSTF)** – pick requests on the nearest track to complete first

Requests (track #s)	**1, 20, 5**	**6**	**25**	**4**	**10**	**3**	**2**	**40**		
Requests serviced	1	5	6	4	10	3	2	20	25	40

Scheduling

Solution is to use a scheduling policy (cont')

- Maintain an ordered queue of I/O requests by track
- **SCAN (Elevator)** – the head moves across the tracks in order (sweep 0 -> n, sweep n -> 0, sweep 0 ->n, …) serving requests along the way

Requests (track #s)	1, 20, 5	6	25	4	10	3	2	40		
Requests serviced	1	5	6	20	25	10	4	3	2	40

- **F-SCAN** – a variant of SCAN, which freezes the queue to be serviced when it is doing a sweep (i.e., moving from track 0->n or from n->0); more fair

Requests (track #s)	1, 20, 5	6	25	4	10	2	3	40		
Requests serviced	1	5	20	25	6	4	10	3	2	40

- **C-SCAN (Circular SCAN)** - only sweeps from outer-to-inner and then resets at the outer track to begin again; more fair

Requests (track #s)	1, 20, 5	6	25	4	10	3	2	40		
Requests serviced	1	5	6	20	25	3	4	10	40	2