PERSISTENCE: DISK SCHEDULING

Shivaram Venkataraman

CS 537, Spring 2020

ADMINISTRIVIA

Project 4a is out! Due April 2th

More details in discussion section

Midterm grading in progress

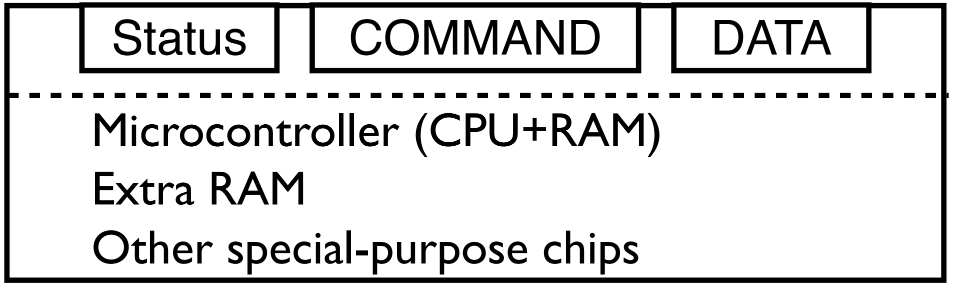
AGENDA / LEARNING OUTCOMES

How do you calculate sequential and random tput of a disk?

What algorithms are used to schedule I/O requests?

RECAP

EXAMPLE WRITE PROTOCOL



Status

COMMAND

DATA

Microcontroller (CPU+RAM)

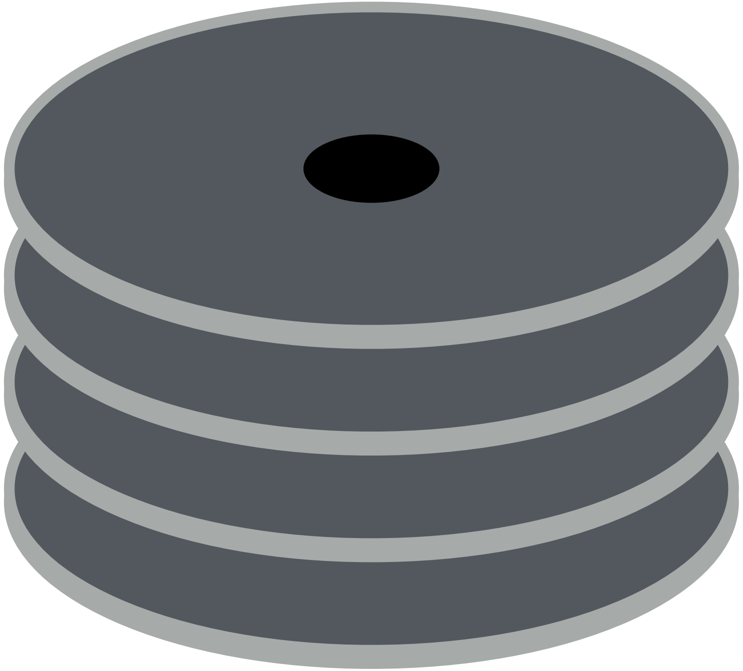
Extra RAM

Other special-purpose chips

```
while (STATUS == BUSY)
   ; // spin
Write data to DATA register
Write command to COMMAND register
while (STATUS == BUSY)
   ; // spin
```

RPM

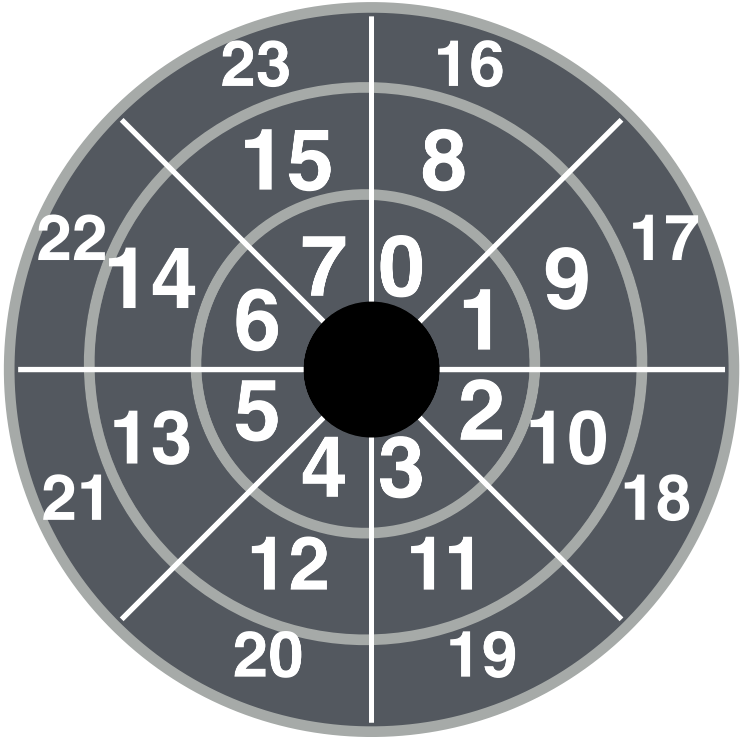
Motor connected to spindle spins platters



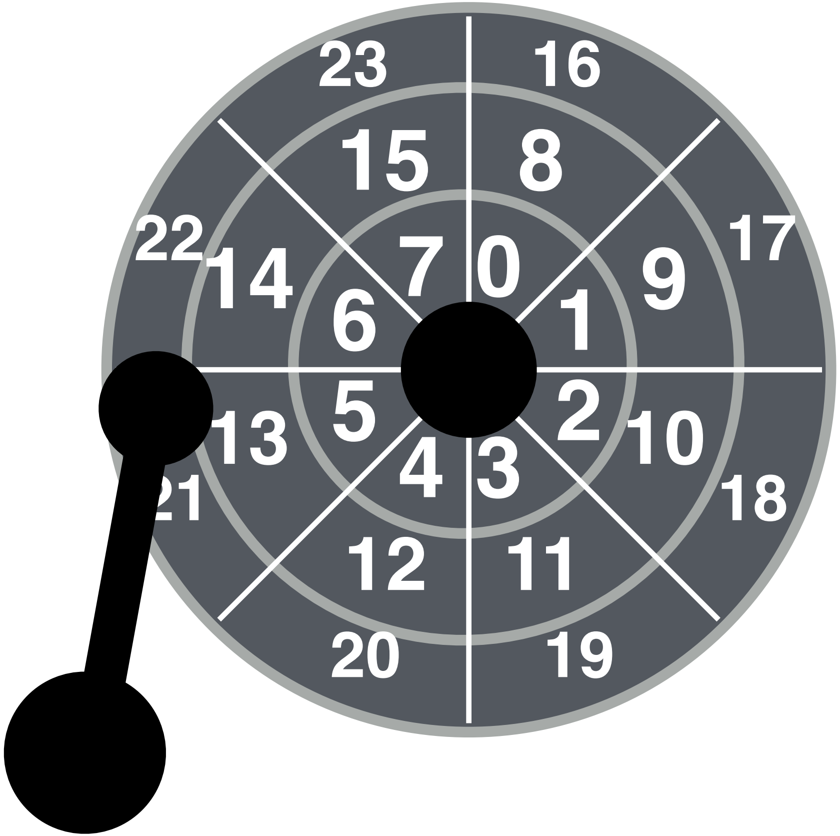
Rate of rotation: RPM

10000 RPM → single rotation is 6 ms

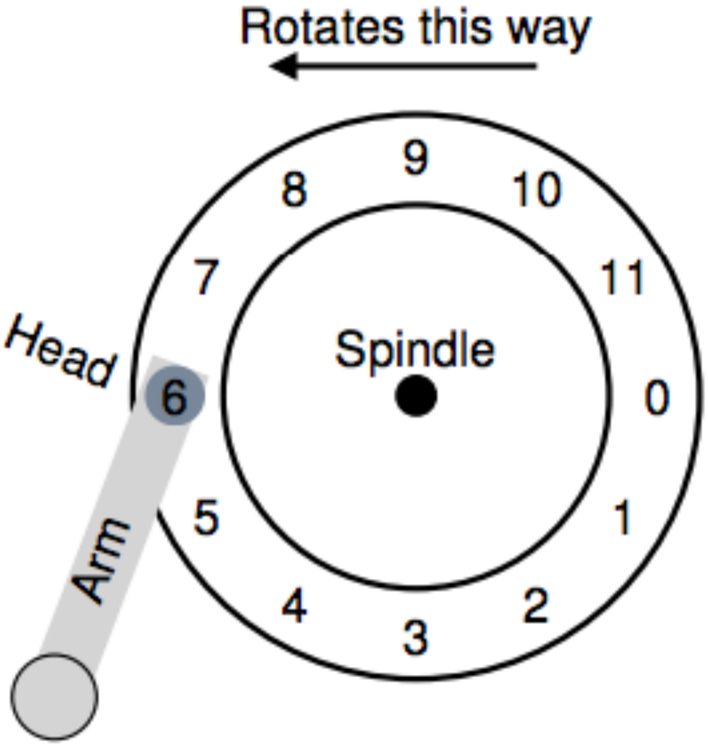
Tracks are divided into numbered sectors



Heads on a moving arm can read from each surface.

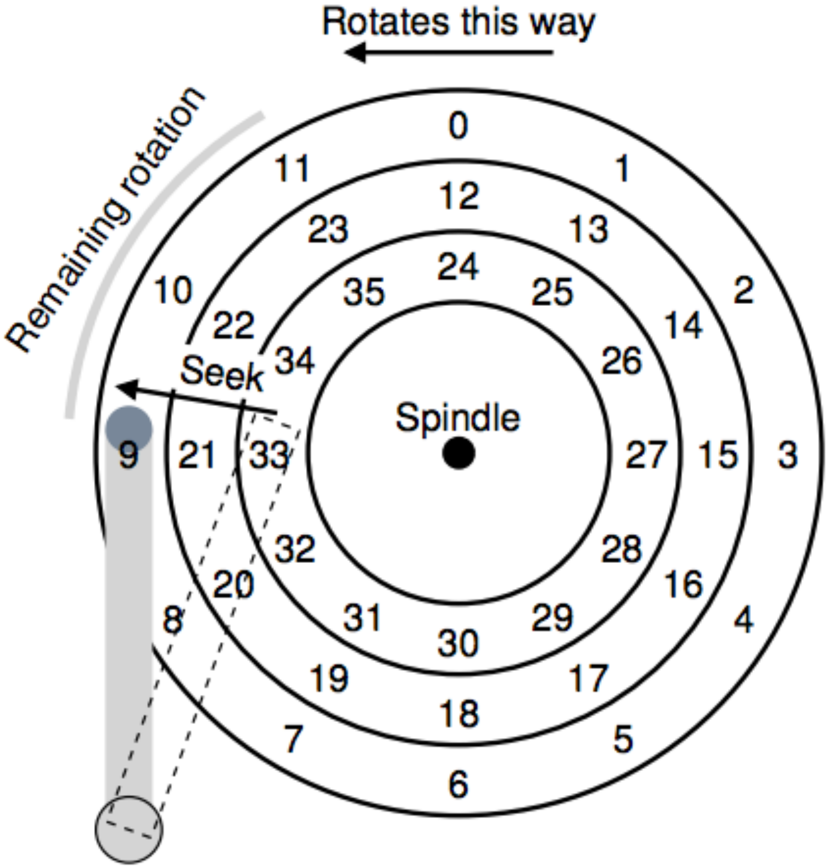


READING DATA FROM DISK



Rotational delay

READING DATA FROM DISK



Seek Time

TIME TO READ/WRITE

Three components:

Time = seek + rotation + transfer time

SEEK, ROTATE, TRANSFER

Seek cost: Function of cylinder distance

- Not purely linear cost
- Must accelerate, coast, decelerate, settle
- Settling alone can take 0.5 - 2 ms

Entire seeks often takes 4 - 10 ms

Average seek = 1/3 of max seek

Depends on rotations per minute (RPM)

- 7200 RPM is common, 15000 RPM is high end

Average rotation?

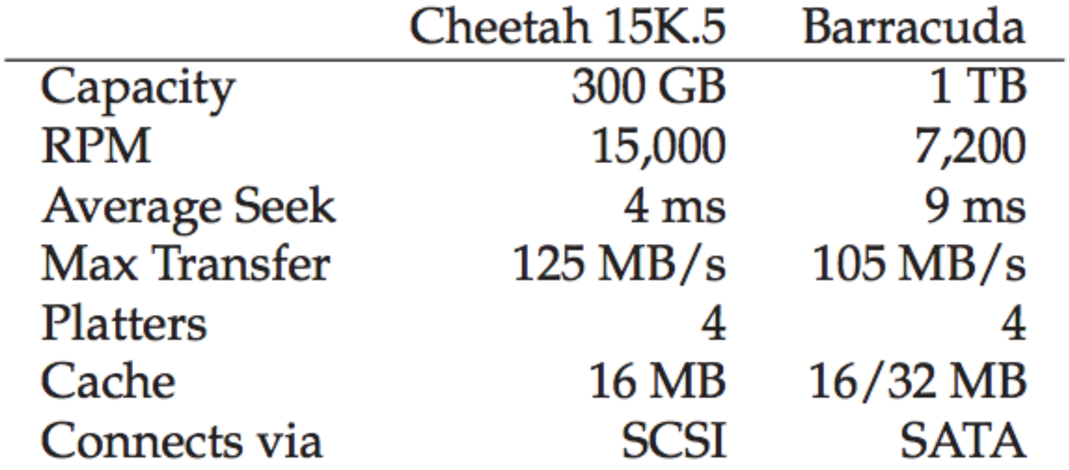
Pretty fast: depends on RPM and sector density.

100+ MB/s is typical for maximum transfer rate

QUIZ 21

https://tinyurl.com/cs537-sp20-quiz21

What is the time for 4KB random read with Cheetah?

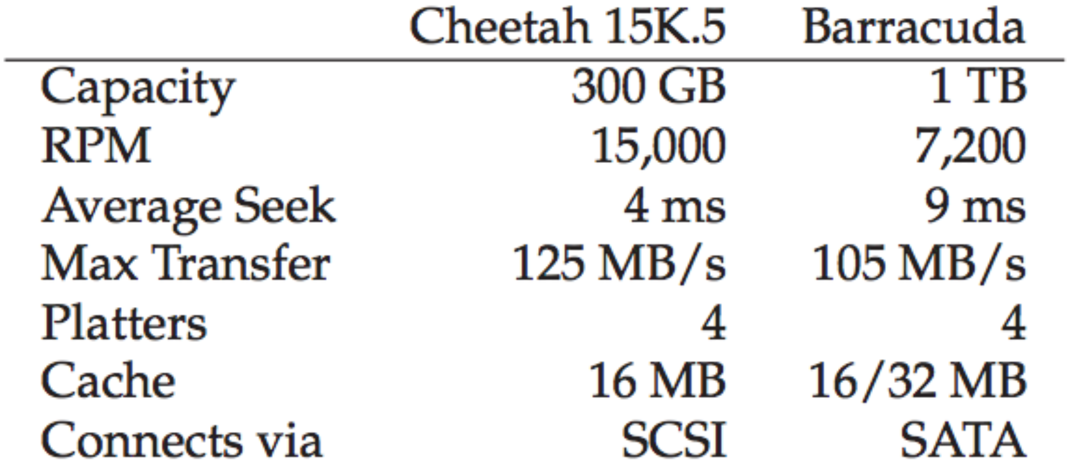


	Cheetah 15K.5	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Average Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	16/32 MB
Connects via	SCSI	SATA

QUIZ 21

https://tinyurl.com/cs537-sp20-quiz21

What is the time for 4KB random read with Barracuda?



	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

WORKLOAD PERFORMANCE

WORKLOAD PERFORMANCE

So…

- seeks are slow
- rotations are slow
- transfers are fast

How does the kind of workload affect performance?

Sequential: access sectors in order

Random: access sectors arbitrarily

DISK SPEC

	Cheetah	Barracuda
Capacity	300 GB	1 TB
RPM	15,000	7,200
Avg Seek	4 ms	9 ms
Max Transfer	125 MB/s	105 MB/s
Platters	4	4
Cache	16 MB	32 MB

Sequential workload: what is throughput for each?

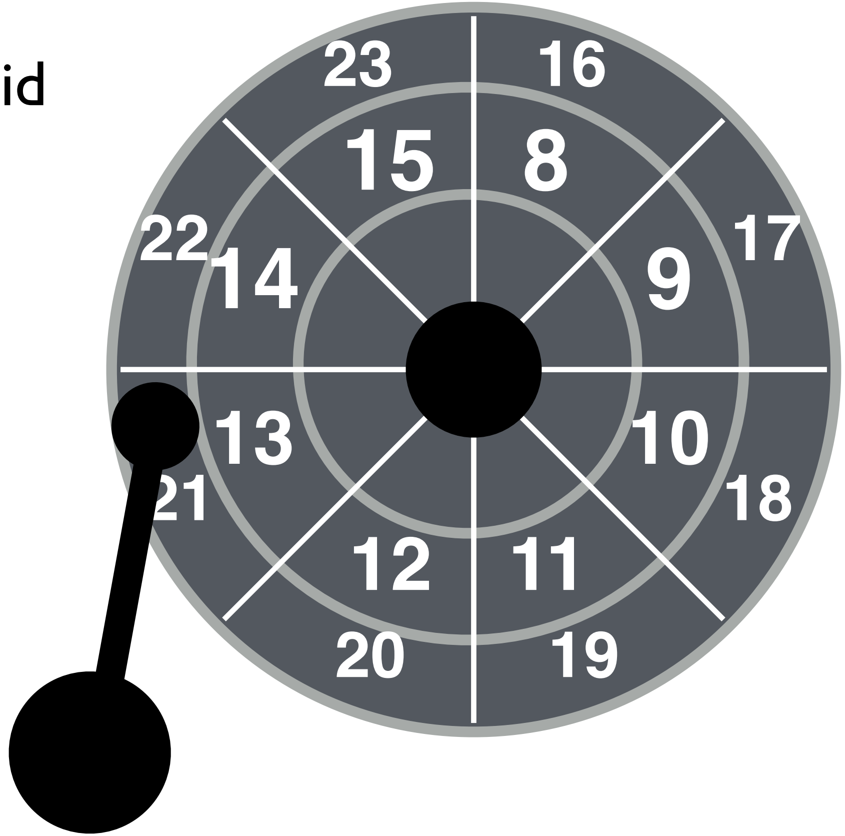
OTHER IMPROVEMENTS

Track Skew

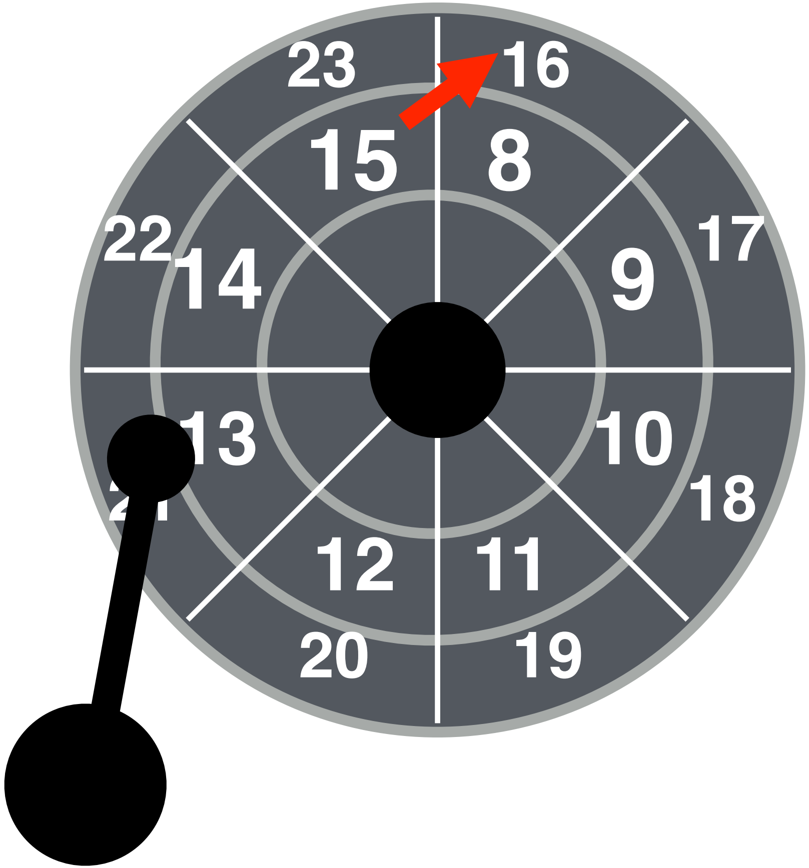
Zones

Cache

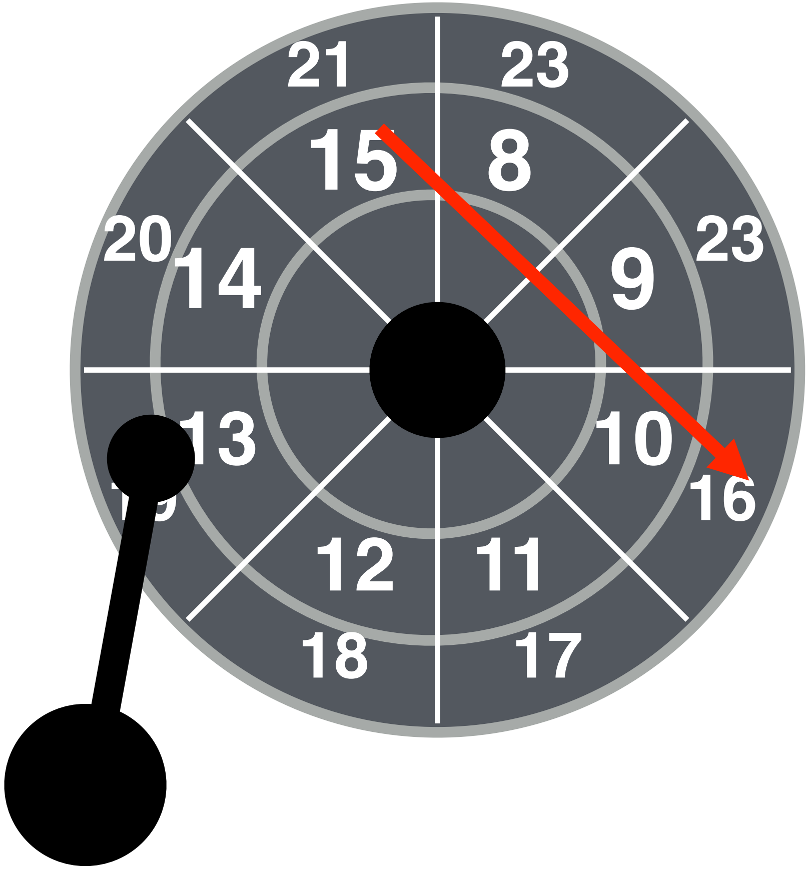
Imagine sequential reading,
how should sectors numbers be laid
out on disk?

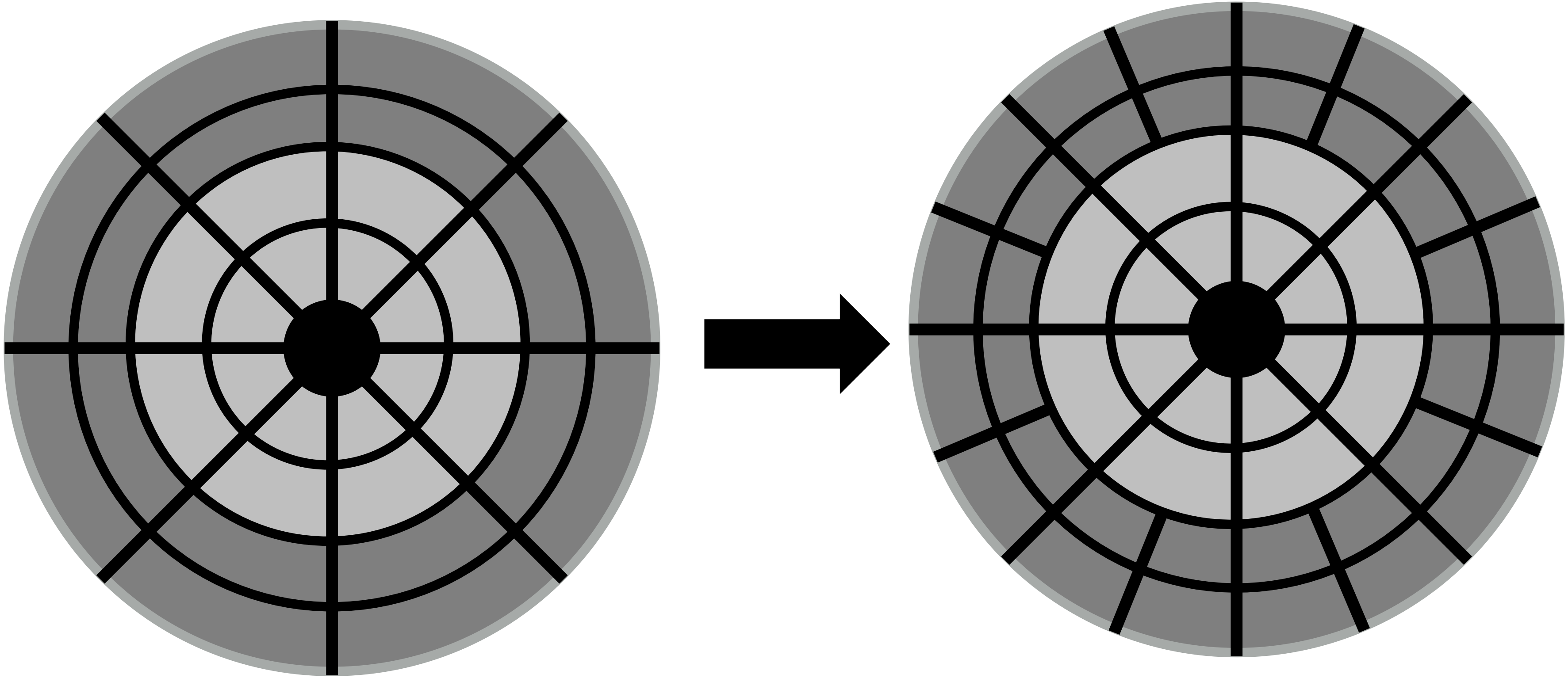


When reading 16 after 15, the head won’t settle
quick enough, so we need to do a rotation.



Track Skew





ZBR (Zoned bit recording): More sectors on outer tracks

DRIVE CACHE

Drives may cache both reads and writes. (In addition to OS cache)

What advantage does caching in **drive** have for reads?

What advantage does caching in **drive** have for writes?

BUFFERING

Disks contain internal memory (2MB-16MB) used as cache

Read-ahead: "Track buffer"

- Read contents of entire track into memory during rotational delay

Write caching with volatile memory

- Immediate reporting: Claim written to disk when not
- Data could be lost on power failure

Tagged command queueing

- Have multiple outstanding requests to the disk
- Disk can reorder (schedule) requests for better performance

I/O SCHEDULERS

I/O SCHEDULERS

Given a stream of I/O requests, in what order should they be served?

Much different than CPU scheduling

Position of disk head relative to request position matters more than length of job

FCFS (FIRST-COME-FIRST-SERVE)

Assume seek+rotate = 10 ms for random request

How long (roughly) does the below workload take?Requests are given in sector numbers

300001, 700001, 300002, 700002, 300003, 700003

300001, 300002, 300003, 700001, 700002, 700003

SSTF (SHORTEST SEEK TIME FIRST)

Strategy always choose request that requires least seek time (approximate total time with seek time)

Greedy algorithm (just looks for best NEXT decision)

How to implement in OS?

Disadvantages?

SCAN

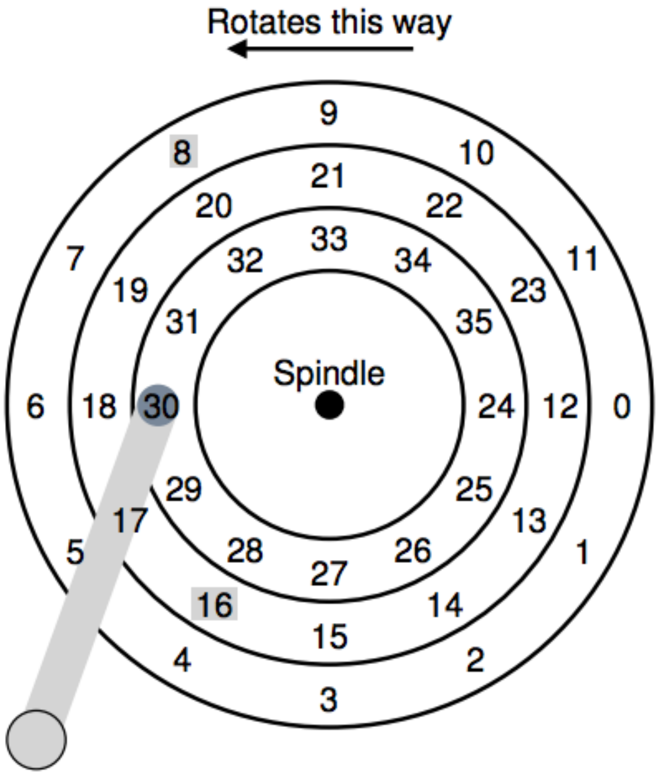
SCAN or Elevator Algorithm:

- Sweep back and forth, from one end of disk other, serving requests as pass that cylinder
- Sorts by cylinder number; ignores rotation delays

C-SCAN (circular scan): Only sweep in one direction

Pros/Cons?

SPTF (SHORTEST POSITIONING TIME FIRST)



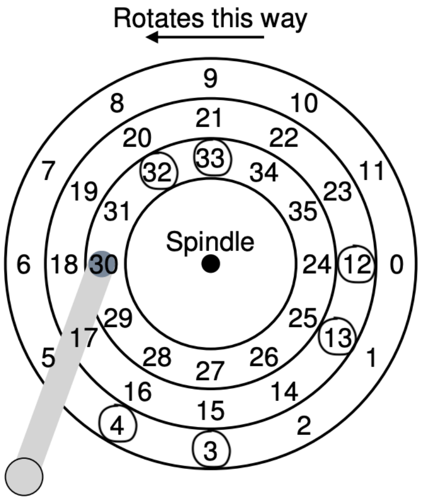
SATF

(SHORTEST ACCESS

TIME FIRST)

QUIZ 22

https://tinyurl.com/cs537-sp20-quiz22



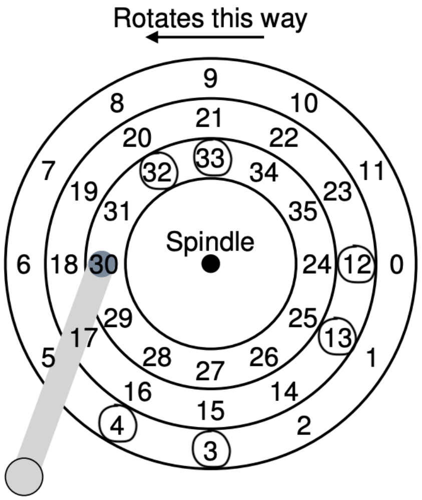
Disk accesses: 32, 12, 33, 3, 13, 4
Rotation Time = 2ms (non-adjacent reads)
Seek Time (for adjacent track) = 2ms.

What is the time taken to using (FCFS) scheduling?

Order in which requests will be serviced for Shortest Seek Time First (SSTF)?

QUIZ 22

https://tinyurl.com/cs537-sp20-quiz22



Disk accesses: 32, 12, 33, 3, 13, 4

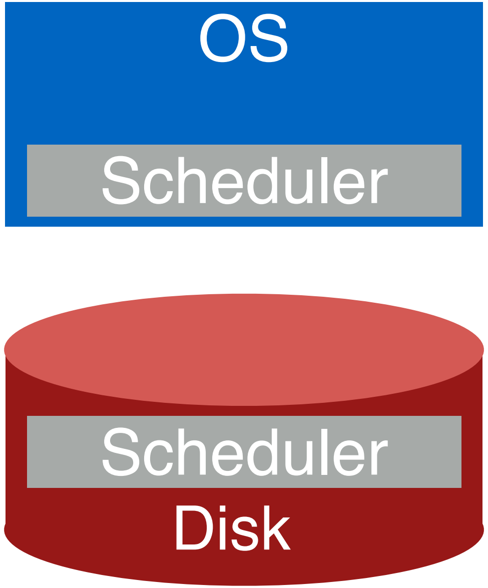
Rotation Time = 2ms (non-adjacent reads)

Seek Time (for adjacent track) = 2ms.

Order in which requests will be serviced for Shortest Seek Time First (SSTF)?

Time Taken

SCHEDULERS



Where should the scheduler go?

WHAT HAPPENS?

Assume 2 processes each calling read() with C-SCAN

```
void reader(int fd) {
    char buf[1024];
    int rv;
    while((rv = read(fd, buf)) != 0) {
        assert(rv);
        // takes short time, e.g., 1ms
        process(buf, rv);
    }
}
```

WORK CONSERVATION

Work conserving schedulers always try to do work if there's work to be done

Sometimes, it's better to wait instead if system anticipates another request will arrive

Possible improvements from I/O Merging

SUMMARY

Disks: Specific geometry with platters, spindle, tracks, sector

I/O Time: rotation_time + seek_time + transfer_time
Sequential throughput vs. random throughput

Advanced Techniques: Skewed layout, caching

Scheduling approaches: SSTF, SCAN, C-SCAN
Benefits of violating work conservation

NEXT STEPS

Next class: How to achieve resilience against disk errors

Project 4a in Discussion today