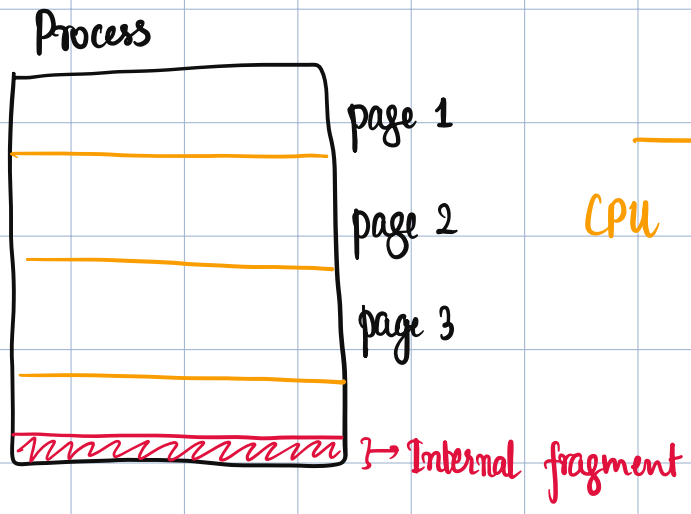
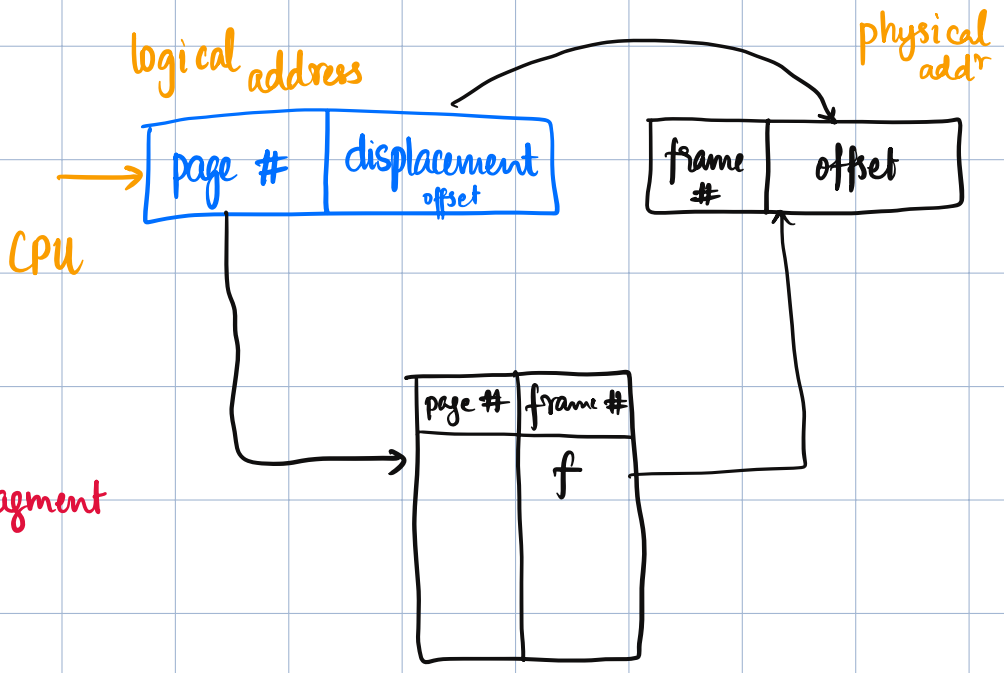
19 Nov 2024 - Operating Systems - I - Week 05

→ Paging

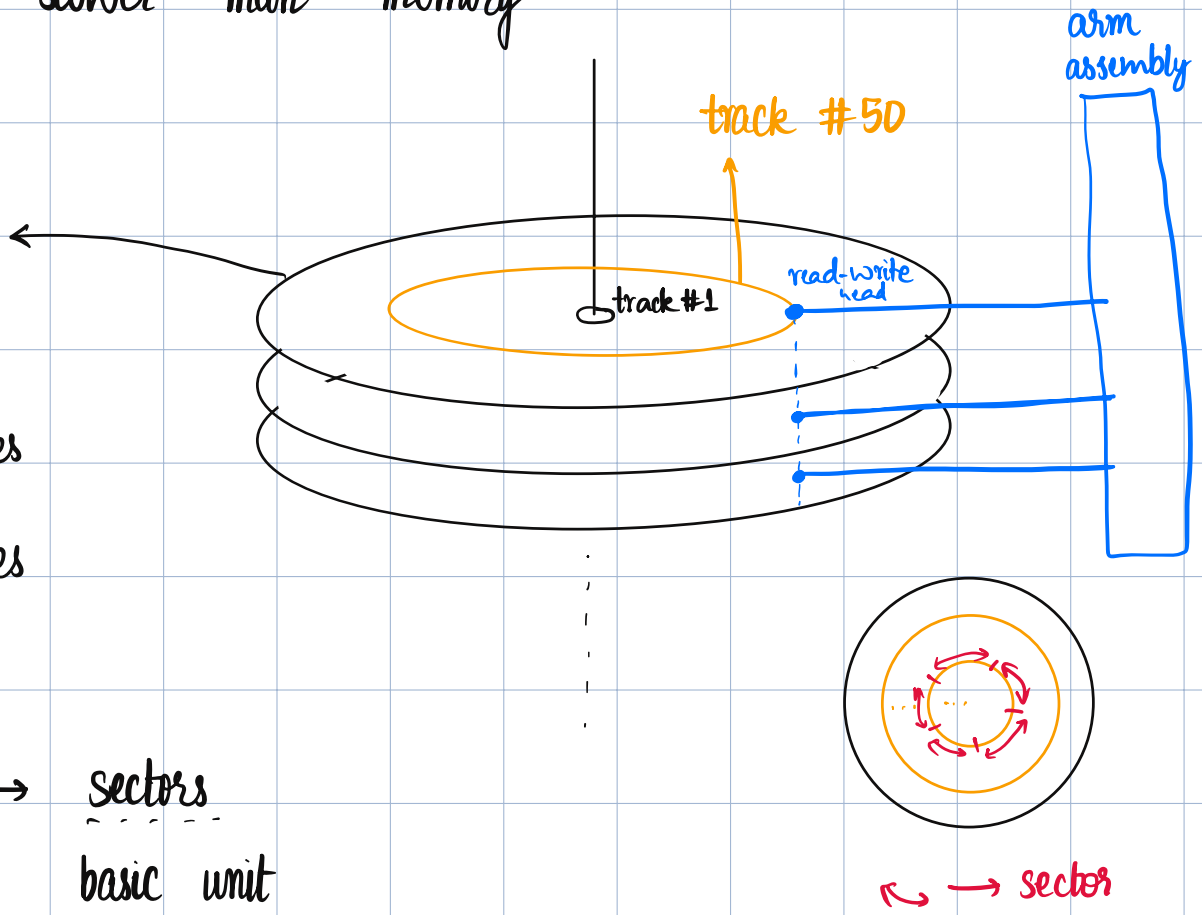
→ process need not be allocated contiguously in memory.





→ Hard disk → electromechanical parts

→ much slower than memory



1 platter → 2 surfaces

3 platters → 6 surfaces

platter → tracks → sectors

basic unit

file ⇝ 10 sectors, or
4 tracks, etc.

same track # across all the surfaces = cylinder

→ conversion between { surface #, track #, sector # } ↔ linear address

→ 1s and 0s are stored in a magnetic pattern

→ read - write heads float above the platters

  → never touch the platters (wear and tear).

  → detect the charge.

1) Rotational delay / latency

Spindle rotates at 5,000 RPM – 15,000 RPM

why rotate spindle and not arm?

Moment of inertia ?

average time to get the desired ~~cylinder~~

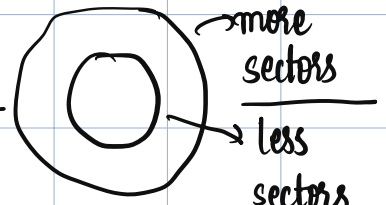
= ⟨ rotational delay ⟩ = $\frac{1}{2}$ rotation time = 2 - 6 ms

2) Seek delay : delay for read-write head to move to the desired track #.

3) Transfer delay : transfer time (fixed)

→ writing to disks
↓
I/O asks
↓
OS ~→ priviledged instructions
↓
disk read/write

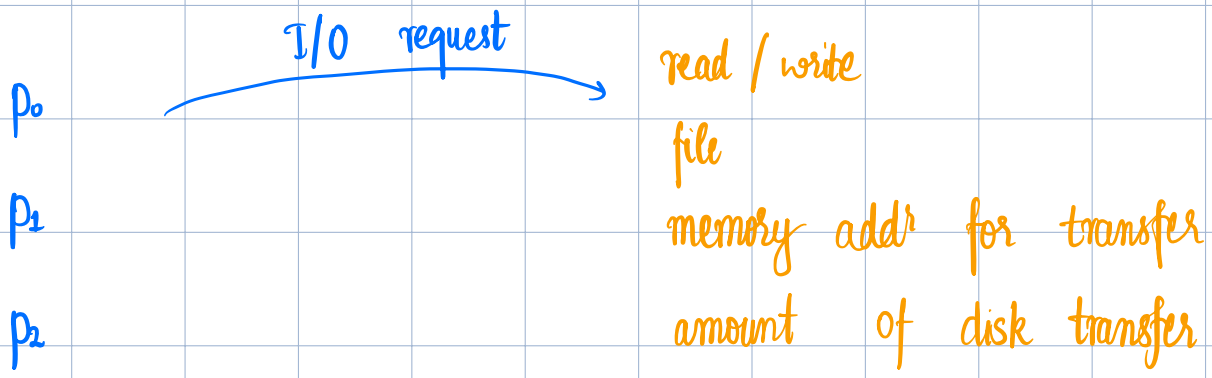
Disk access time = ① + ② + ③

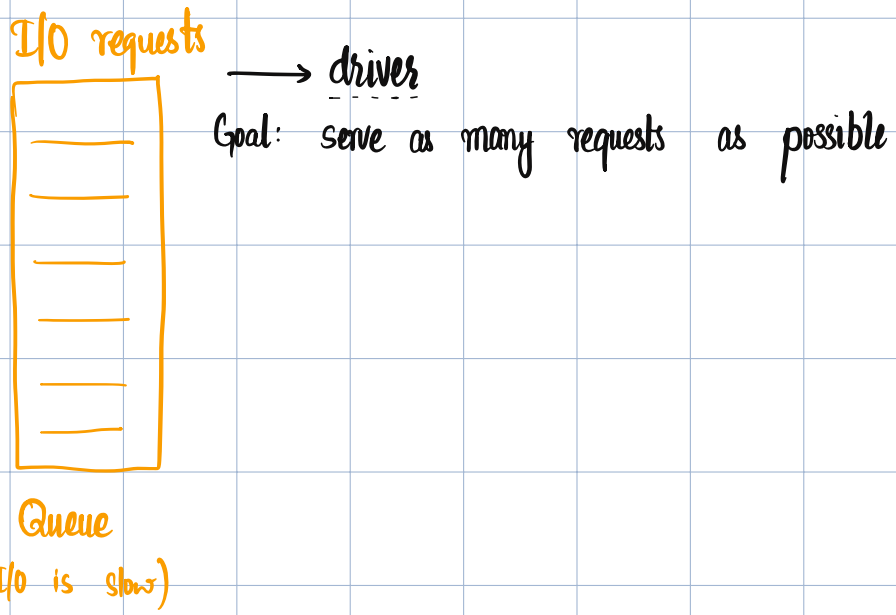
Problem : 

density of sectors is more in outer zone → transfer time fluctuates

??

Solution → ① vary rotation speeds of inner and outer tracks
↓
② space





→ How to service I/O requests?

Disk Scheduling Algorithms

→ First come first serve

I/O blocks on cylinder # :

head ——————————————— tail

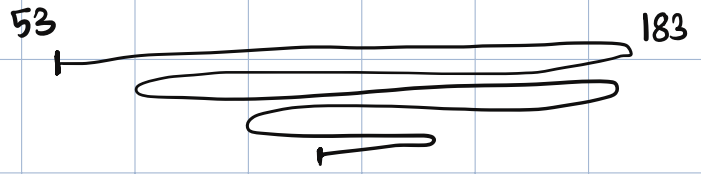
98, 183, 37, 122, 14, 124, 65, 67

→ read-write head is at cylinder # 53

initially

distance travelled = (98 − 53) + (183 − 98) + (183 − 37)

+ (122 − 37) + ...



Unnecessary extra movement

21 Nov 2024

Disk scheduling algorithm

→ FCFS ⇝ fairness (whichever is first gets serviced)

→ seek delay

→ SCAN

→ starts at one end of the disk and moves towards the other end

I/O requests : 98, 183, 37, 122, 14, 124, 65, 67

initial position 53, moving towards zero.

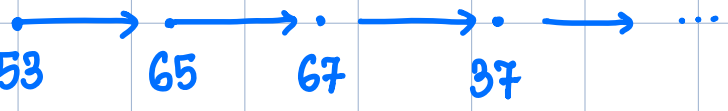
total head movement : 53 → 14 → 183

$$\begin{array}{r} 53 - 14 \\ + \quad 183 - 14 \\ \hline 208 \end{array}$$

→ also called as elevator algorithm

→ not fair, but average response time is not much different.

→ Shortest seek time first (SSTF)

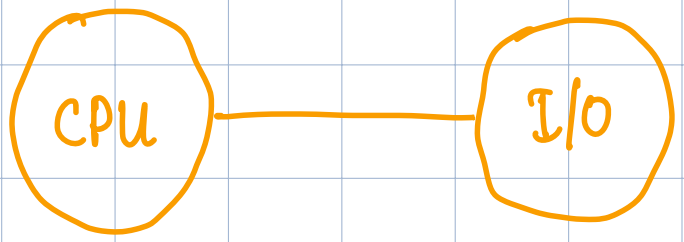


priority same
time same
seek time same
} first come first serve

→ no fairness

→ starvation (not in SCAN)

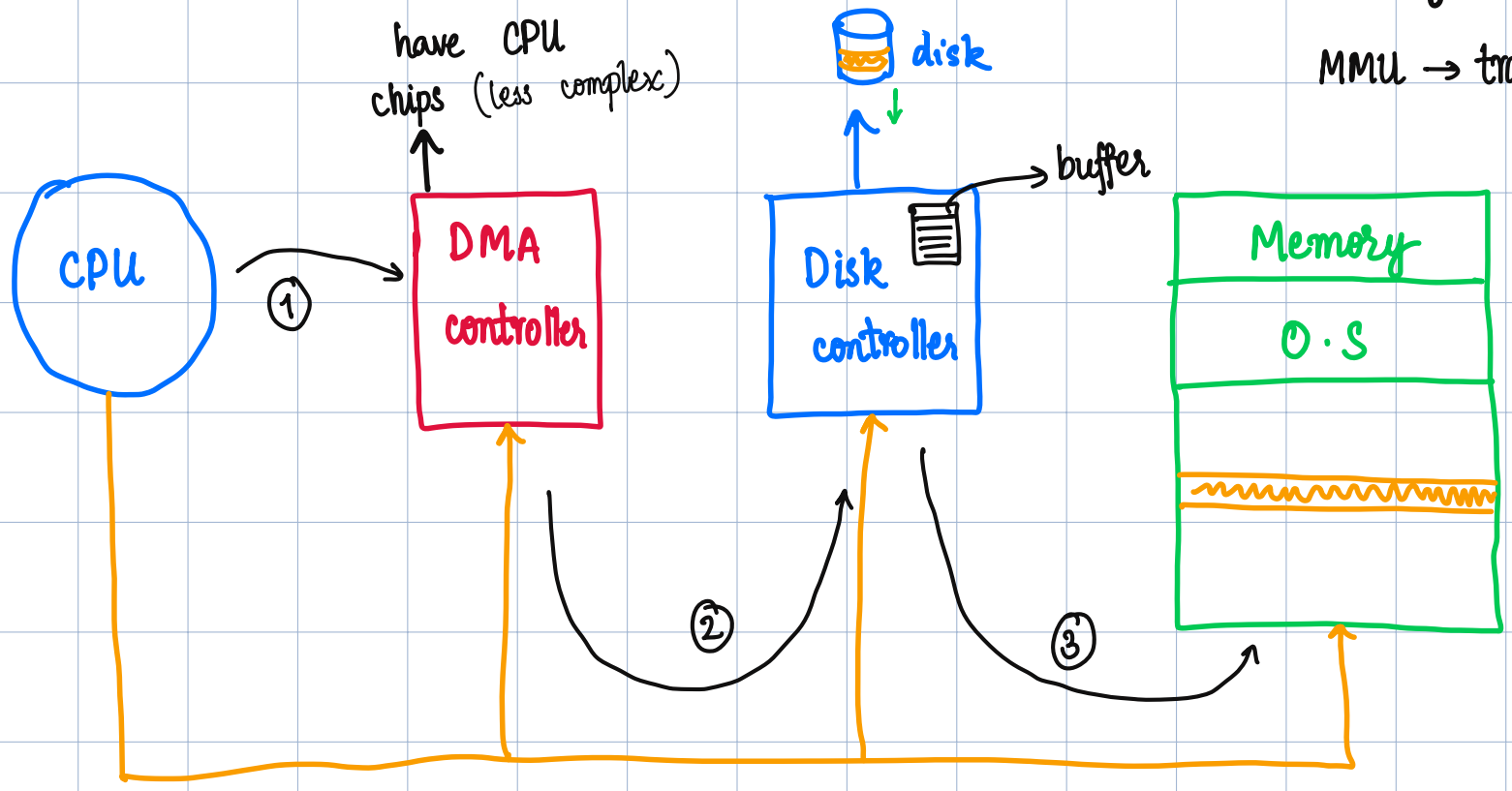
* variance is usually large



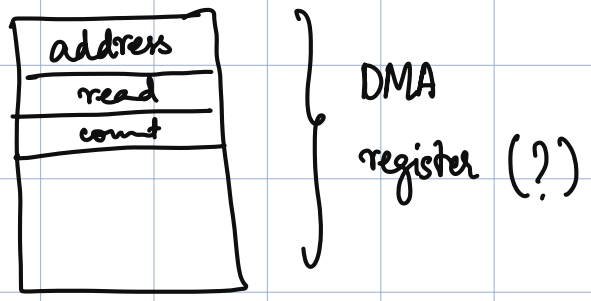
→ To keep both CPU and I/O both busy → multiprogramming

→ DMA : Direct Memory Access

MMU → translation



① CPU programs DMA controller



② DMA requests transfer to memory

③ Data transfer

CPU can work on other process now → Other process is cached

④ Acknowledgement

⑤ Interrupt once done

Q: Compute the time to read a random 1,024 byte sector

Disk rotation = 5400 RPM

transfer time = 10 MB/s.

average seek time = 9 ms

5400 r ⟶ 60 s

$\frac{1}{2}$ r ⟶ $\frac{1}{2} \times \frac{60}{5400}$

$$9\text{ ms} + \left(\frac{1}{2} \times \frac{60}{5400} = \frac{1}{180}\right) + \frac{1\text{ KB}}{10 \times 2^{10}\text{ KB/S}}$$

$$\frac{9}{1000} + \frac{1}{180}\text{ S} + 0.1 \times 2^{-10}\text{ S}$$

$$\left(\frac{9}{1000} + \frac{1}{180} + 0.1 \times 2^{-10} \right) s$$

$$\frac{18}{\frac{9}{2}} \quad 7$$

$$\frac{1620 + 1000}{180\,000} + 0.1 \times 2^{-10} \quad s$$

$$\frac{2620}{180000} \qquad \frac{262}{18,000} \quad s$$

$$\frac{262}{18} \quad ms$$

$$1.47 \quad ms$$