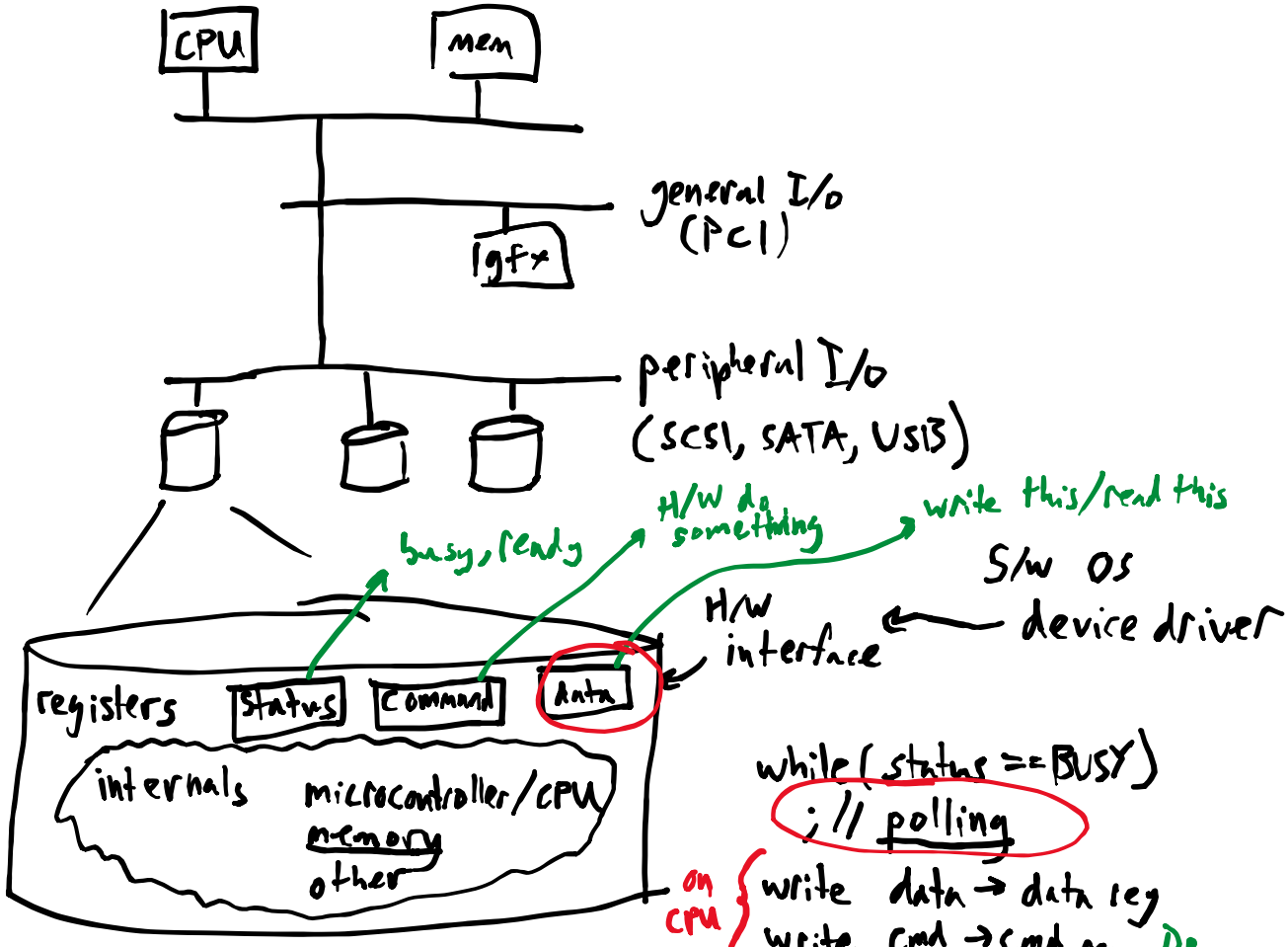
Concurrency: 4/15 9AM EDT

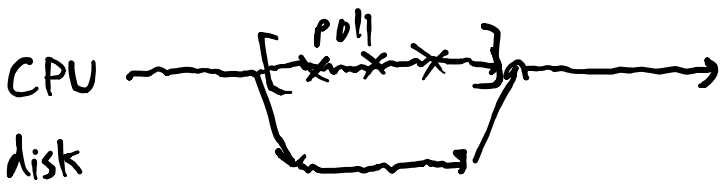
Recap: OS : private view of CPU/mem

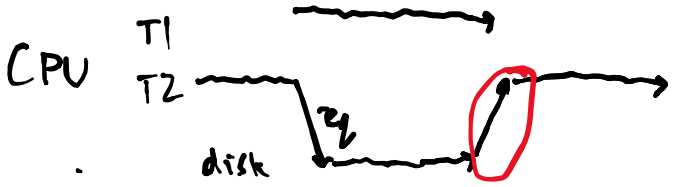
1. (virtualization)
   sharing address space
2. (concurrency)
   files persist
3. (persistance)



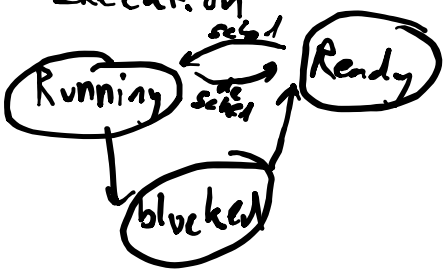
Problems

1. data xfer: 1 reg at a time ✓ DMA
2. CPU is used up for polling ✓ interrupts



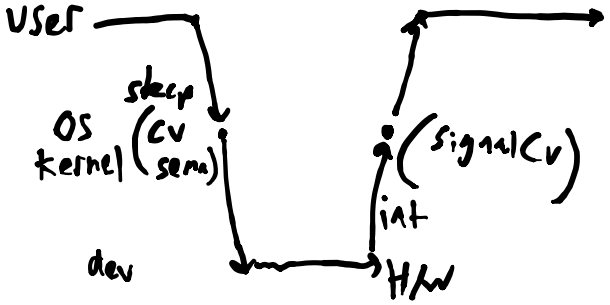


Recall: limited direct execution



OS @ boot

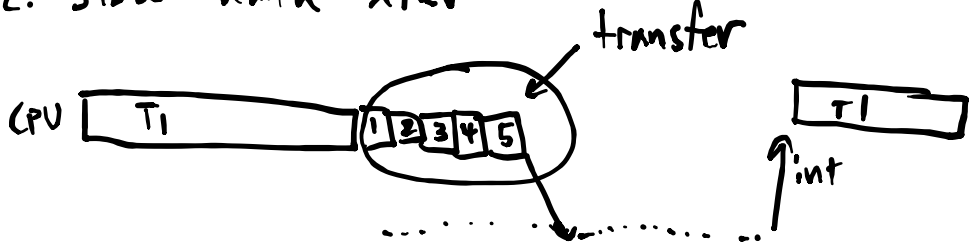
set interrupt hdlr
(h/w interrupts on IRQ Line)



Are interrupts always better? NO!

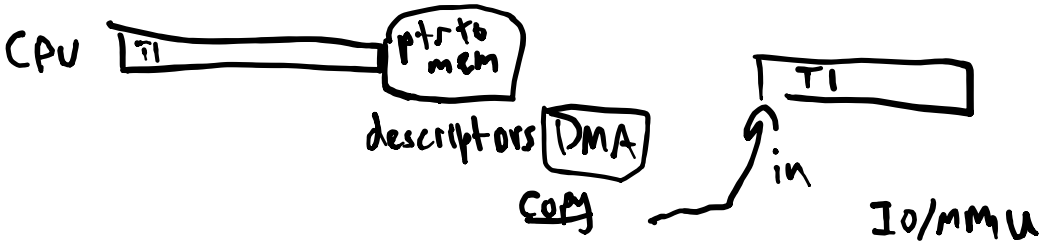
- polling for high traffic
- NAPI → hybrid — infreq → interrupt
  freq → polling

2. slow data xfer



instead: DMA: Direct Memory Access

device can read/write mem on its own!

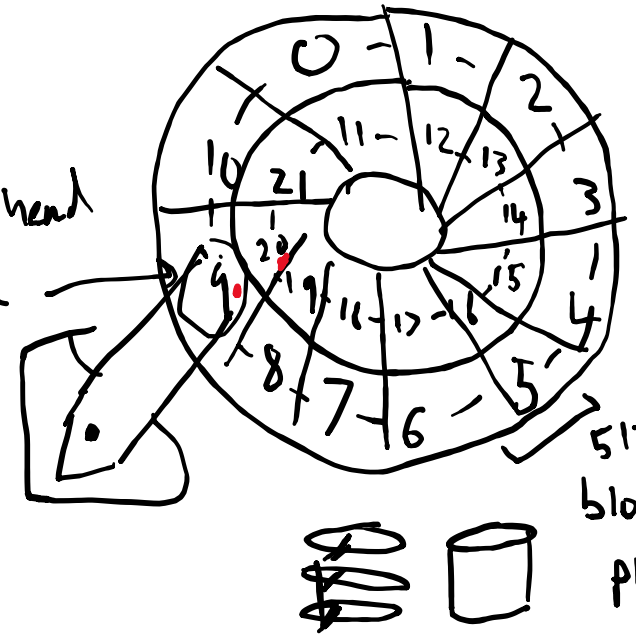


How to actually communicate with devices?

- special instruction (in/out) — in <port> / out <port>
- memory mapped I/O (mov)

Disks (magnetic)

Sector (512 bytes blocks)



rotates @ fixed speed RPM

track (concentric circles)

1000s blocks per track

100s of track within width of human

r/w head

induce surface charge or sense charge

512b blocks [4k]

platter may have >1 each has 2 surfaces

to read/write

seek: move arm to correct track
(speedup, coast, slow down, down)

rotation: wait for correct block

transfer: read/write sector

$T_{I/O}$: $T_{seek} + T_{rotate} + T_{transfer}$

($T_{seek} + T_{rotate}$: try to minimize)

[skew, caching

Sequentially vs. random

$R_{transfer}$: 100 MB/s

T_{seek}: 10ms
Size: 10MB

$= 10ms + \frac{10MB}{100MB/s} \cdot \frac{1000ms}{s}$

$= 110ms$

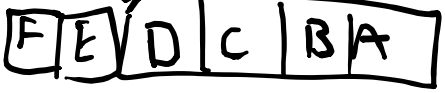
$R = \frac{10MB}{110ms} \sim 90.9 \frac{MB}{s}$

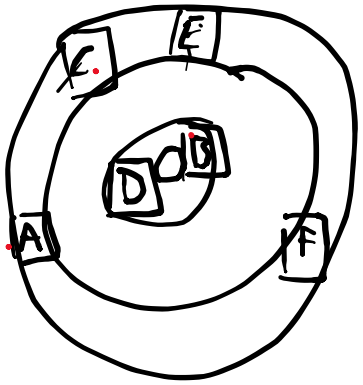
size 10 KB = 10 ms + $\frac{10\text{ KB}}{100\text{ MB/s}}$ [100x difference]

~ 10 ms

R $\simeq$ 1 MB/s

Disk scheduling: queue of requests





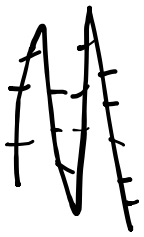
FIFO: first in, first out

~~SJF~~: SSF (shortest seek first) ACDB

problem: <u>starvation</u>

"elevator" SCAN algorithm)

how long should you wait?



FIFO