CSE 330: Operating Systems

Class: 24 **Date:** 4/10

Spring 2018

Note Title

Input Output / Storage devices

↳ hard disks ⟶

↳ SSD → solid state disks

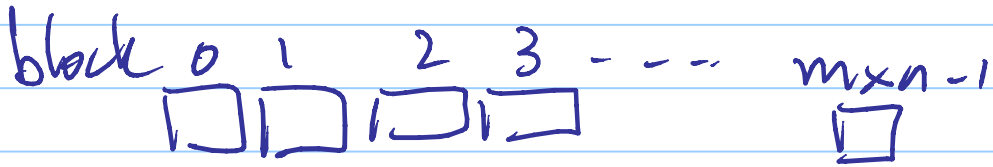
hard drive
tracks
head
platters
sectors

lots of sectors — N per track

lots of tracks → m per disk

Total sector m x N

Sector contains a block of data
p bytes



to write a block of data

① put data in mem

② Tell controller block # & mem address

③ controller does the rest (takes time)

④ "done"

SSD → non volatile RAM

→ internally uses RAM / physical addresses

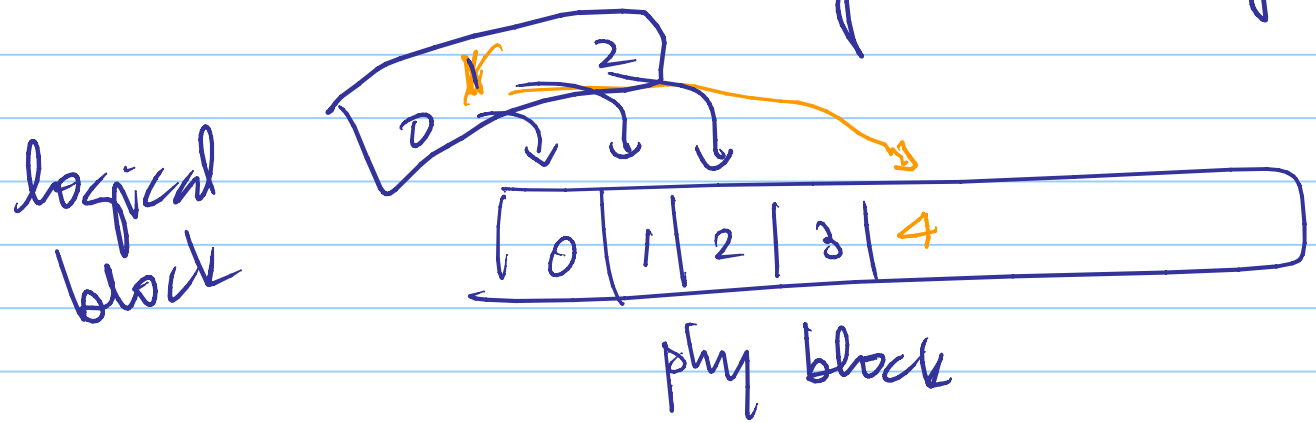
→ controller (hardware) makes the RAM look like a hard drive (read / write blocks)

write to SSD

① slow → erase then write

② destructive

load leveling (done by hardware)

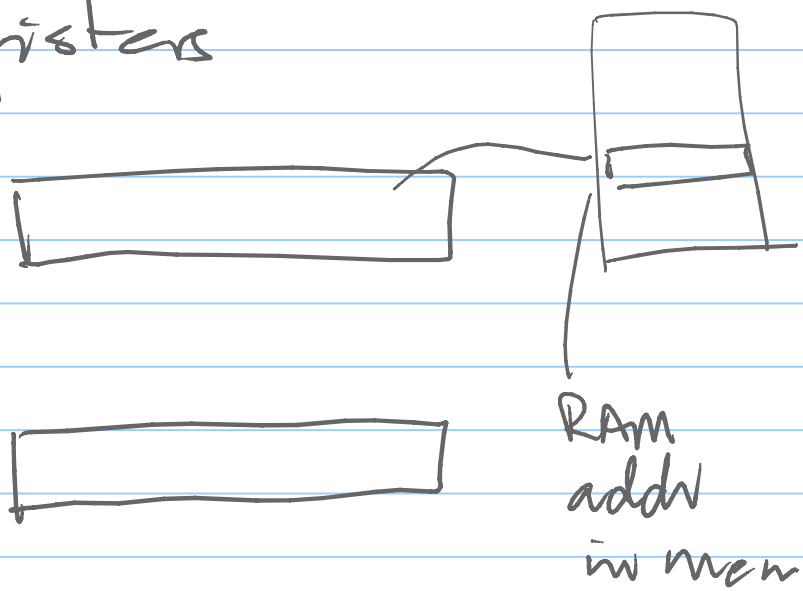


How to do I/O (talk to controllers)

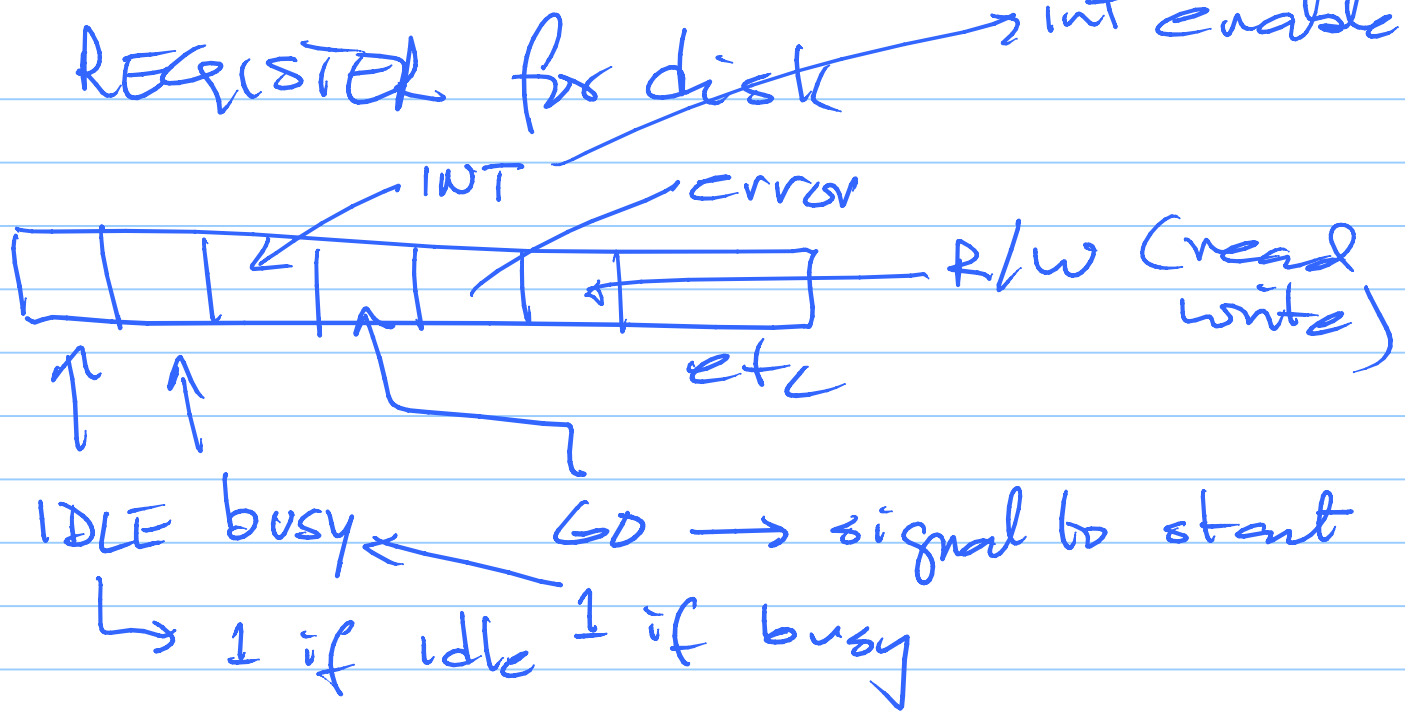
Controllers have registers

Status Register (& control)

data Register

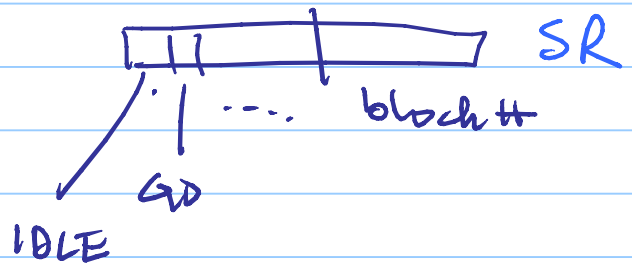


STATUS REGISTER for disk



OS wants to write block b from mem location x

↳ disk addr



↳ wait till idle bit is set

put block # in SR

put x in data reg

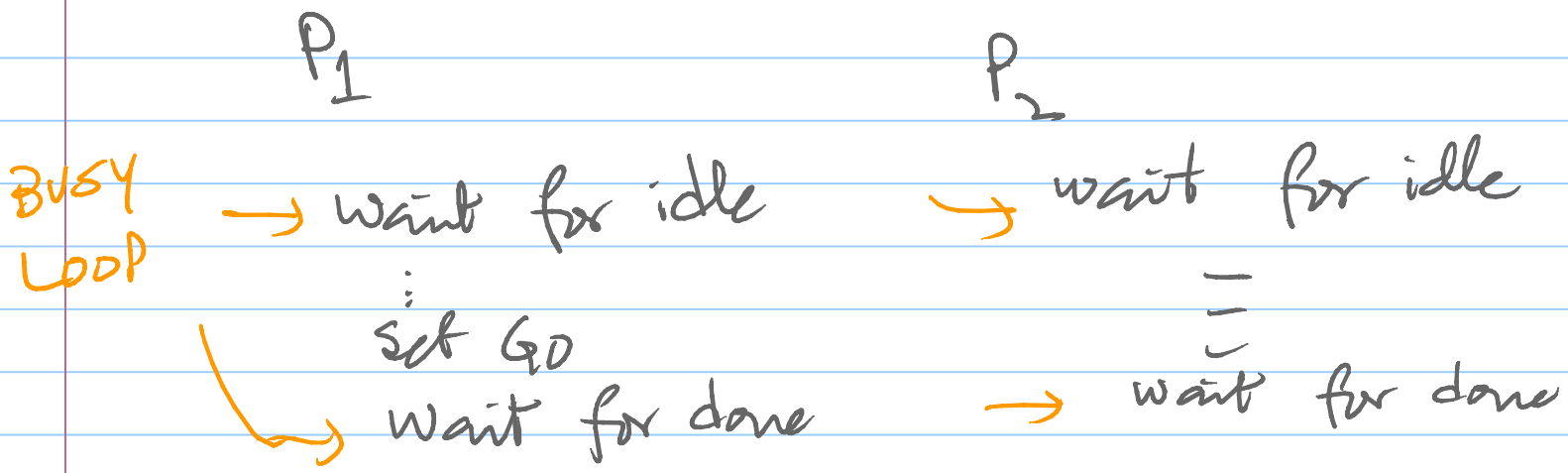
set R/W to W

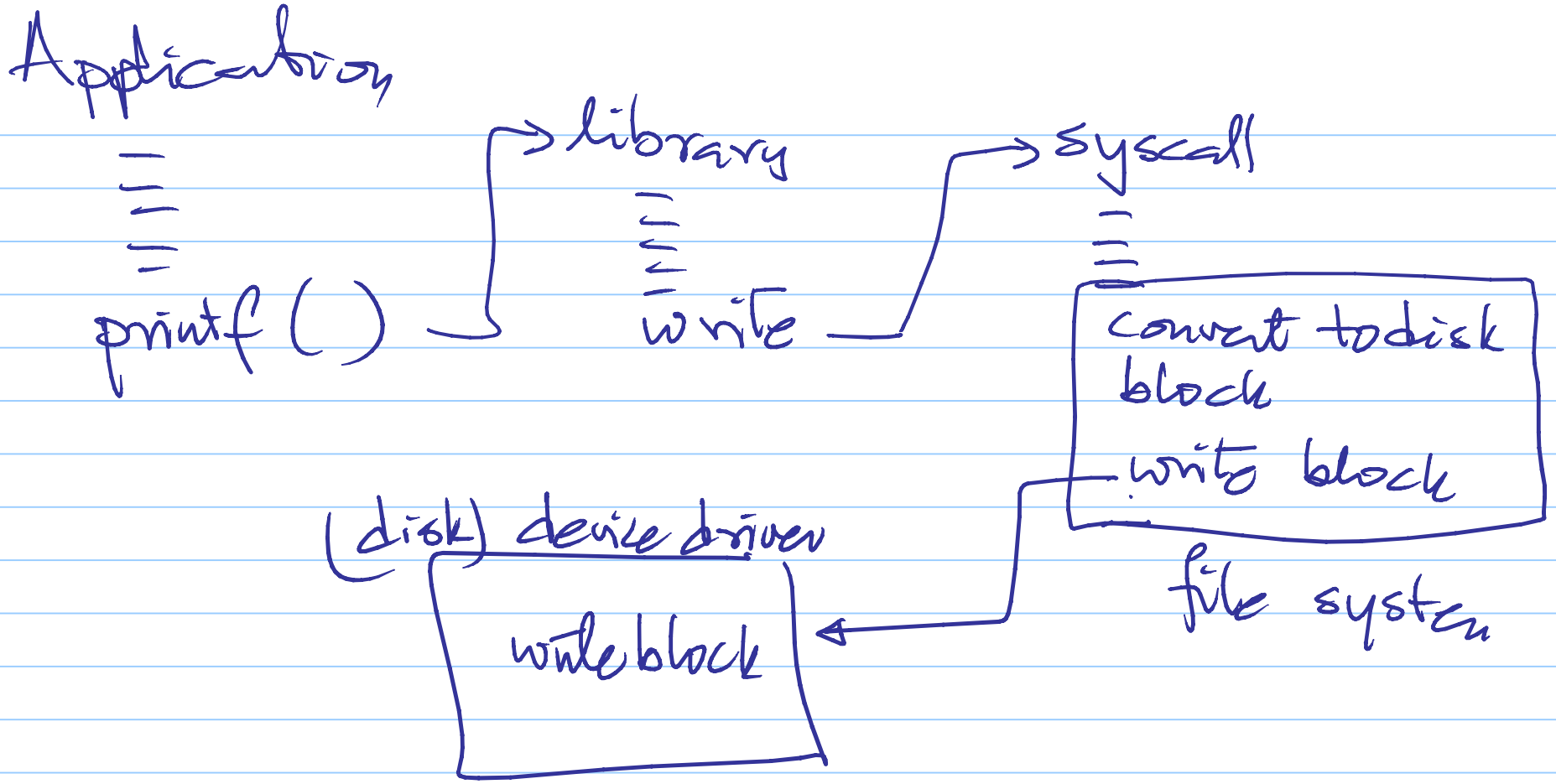
set 'go' bit to 1

Polled I/O

- wait for idle
- wait for done

→ busy wait




Application
printf ( )
library
write
syscall
convert to disk block
write block
file system
(disk) device driver
write block

interrupt driven I/O

→ every time the disk controller finishes an operation, it ~~signals~~ sends an interrupt to the CPU

→ device driver has 2 halves
- → user half
- → interrupt half

write a block (block#, addr)

{ P(disk_mutex) ← I/O q

= put block# → SR

= put addr → DR

set go bit

P(done) ←→ I/O burst

V(disk_mutex)

}

disk_mutex = 1

done = 0

INIT

INT

{

V(done)

}

P1
P2
P(writer)
P(done)
Pwriter
P(done)
Idwait
Ild burst
Ild burst
V(done)
V(writer)
V(done)
V(writer)

I/O scheduling

P_1, P_2, P_3 → all want to write to disk

→ what order (assume P_1 came first & then P_2 & then P_3)

→ FCFS if fair semaphores are used →

P_1 → gets $P(\text{mutex}_{\text{disk}})$

P_1 using disk

P_2
P_3

waiting

→ when P_1 is done should we pick P_2 ?

head
p2
trade
p3
p1
Spindle

Disk I/O scheduling

1) Disk head scheduling — SEEK Time

2) Disk latency scheduling — Latency

rotational

time taken to move head from t_1 to t_2

time taken to wait for sector to come to head

Seek optimization

1. FCFS → not good
2. SSTF → shortest seek time first