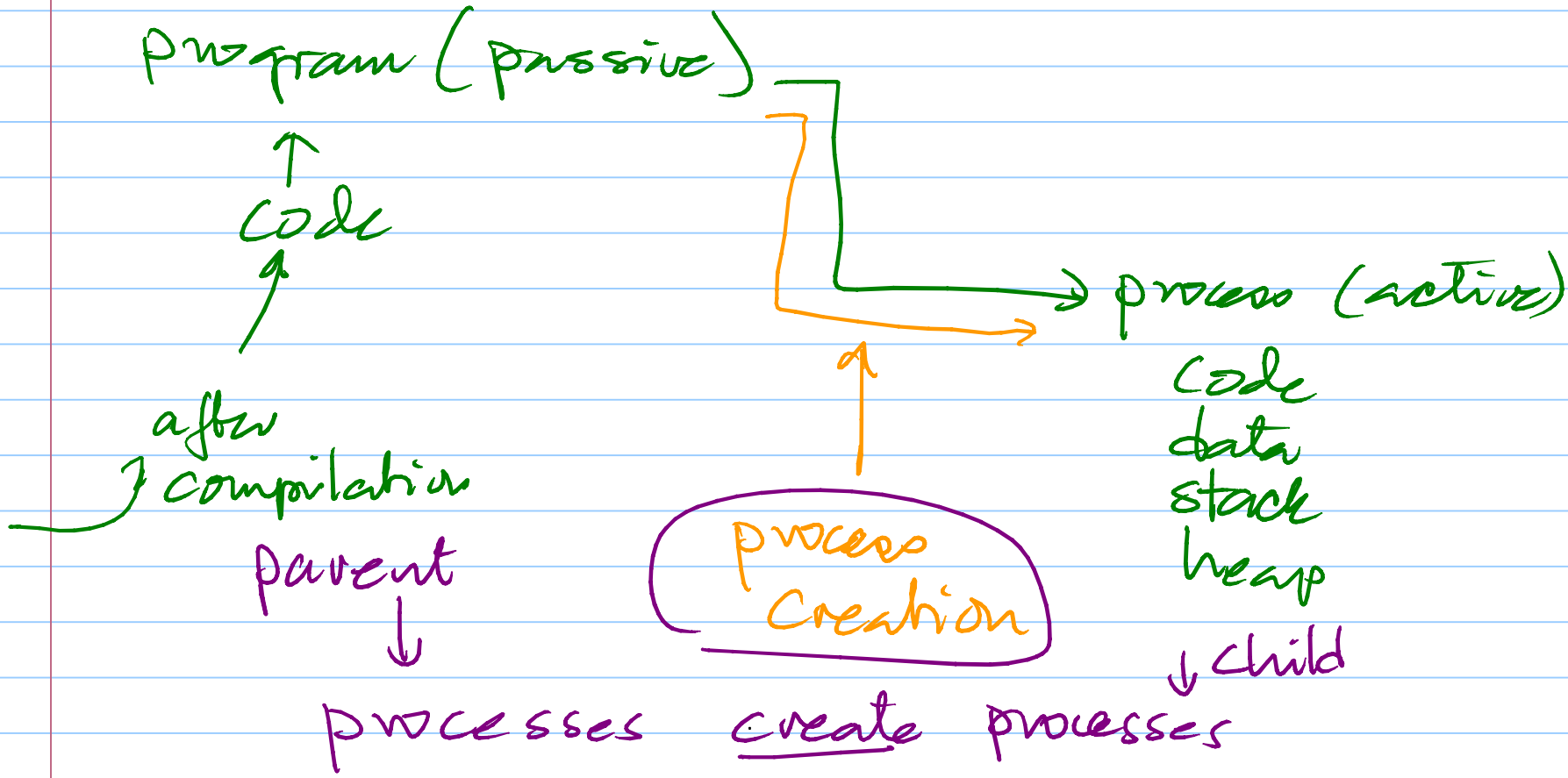
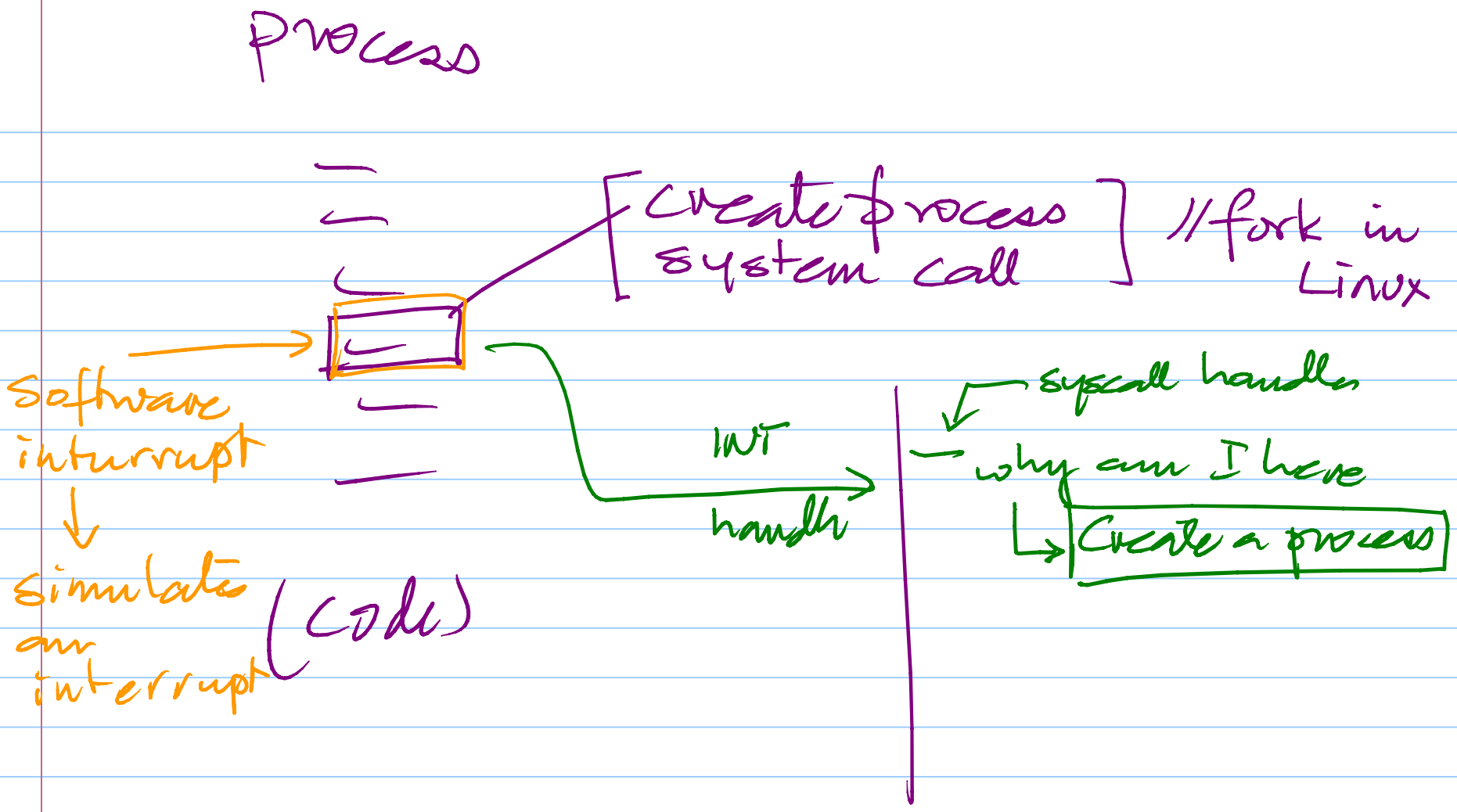
CSE 330: Operating Systems

Class: 06 **Date:** 1/25

Spring 2018

Note Title




process
create process system call
// fork in Linux
software interrupt
simulate an interrupt (code)
INT handler
syscall handler
why am I here
Create a process

Create a process (system call)

→ find space in memory

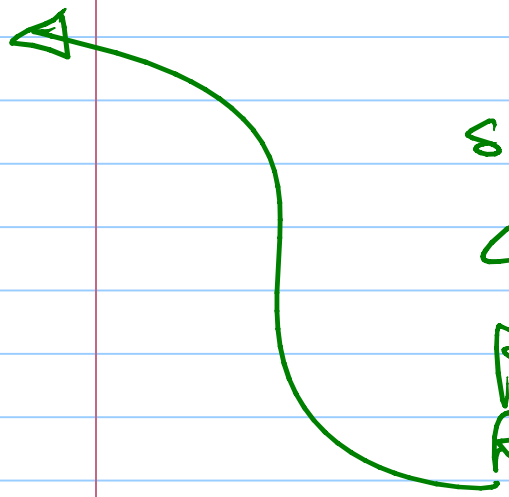
copy executable code into memory ← LOADER

set up stack & heap

create a PCB

put PCB into scheduling Q

RTI



Threads create threads
(or processes)

→ system call

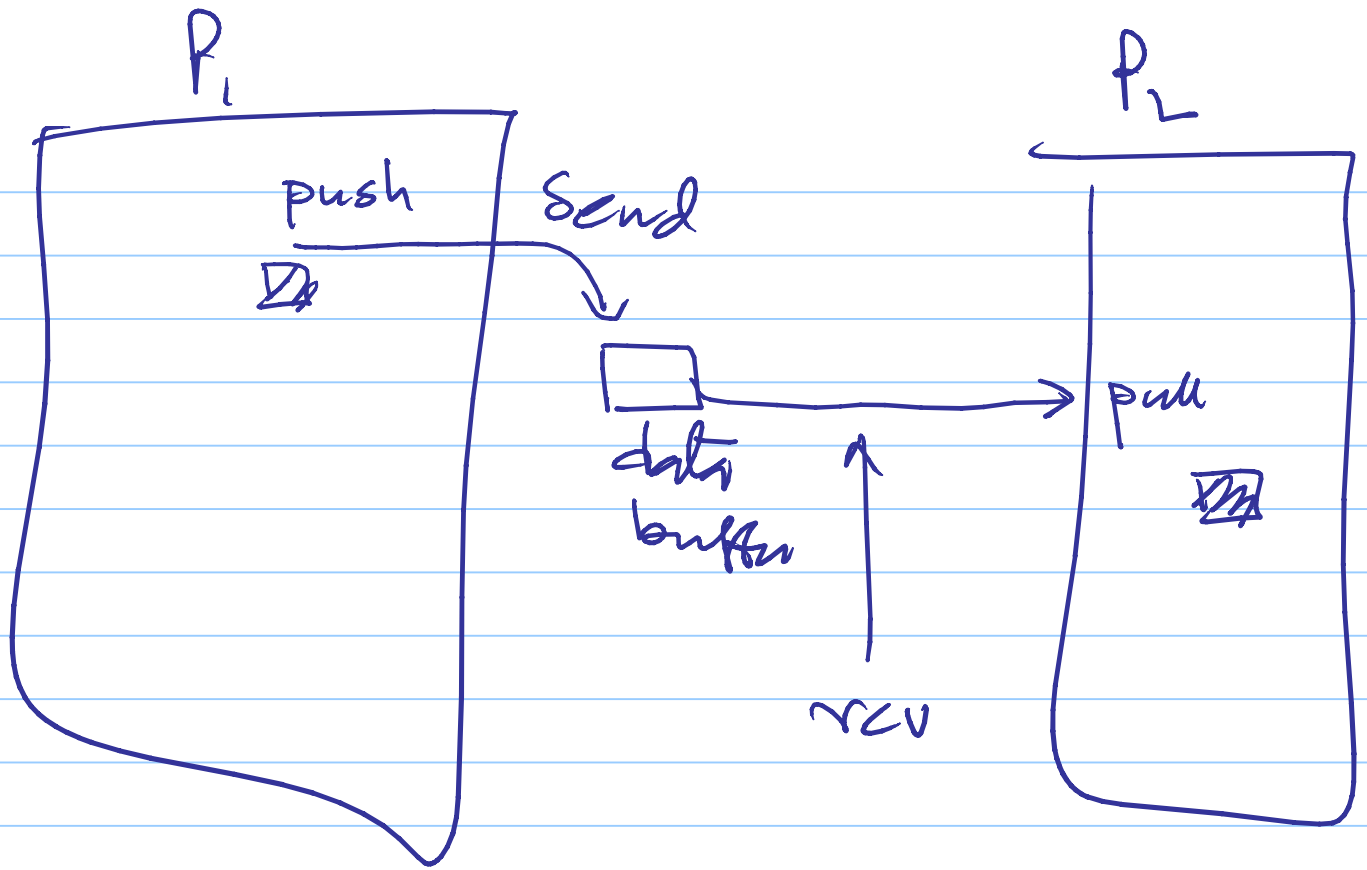
↓

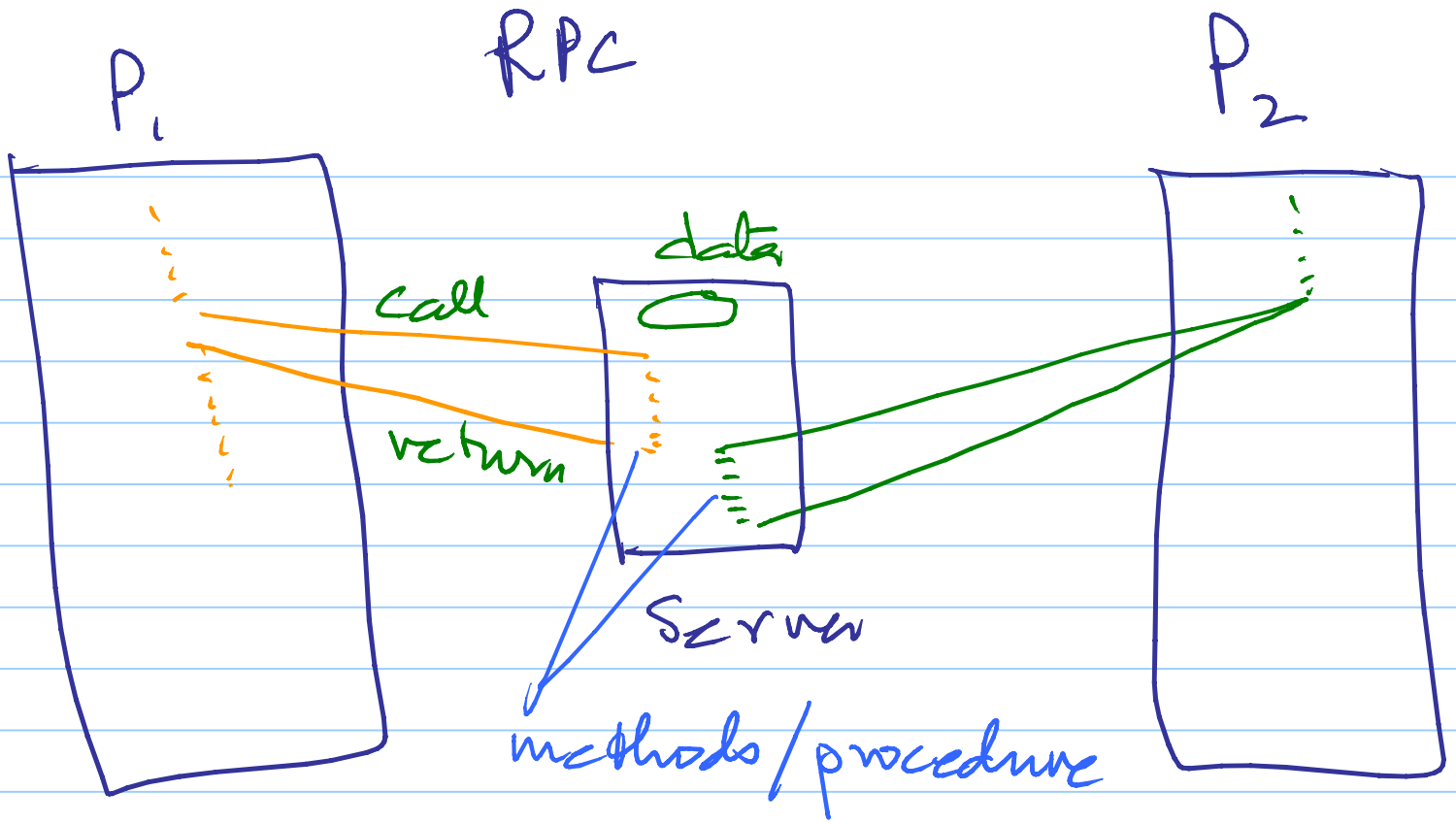
create a new stack
create a " PCB
add to scheduler

Cooperating processes →

↳ inter process communication

- message passing
- remote proc calls (RPC)
  " method invocation (RMI)

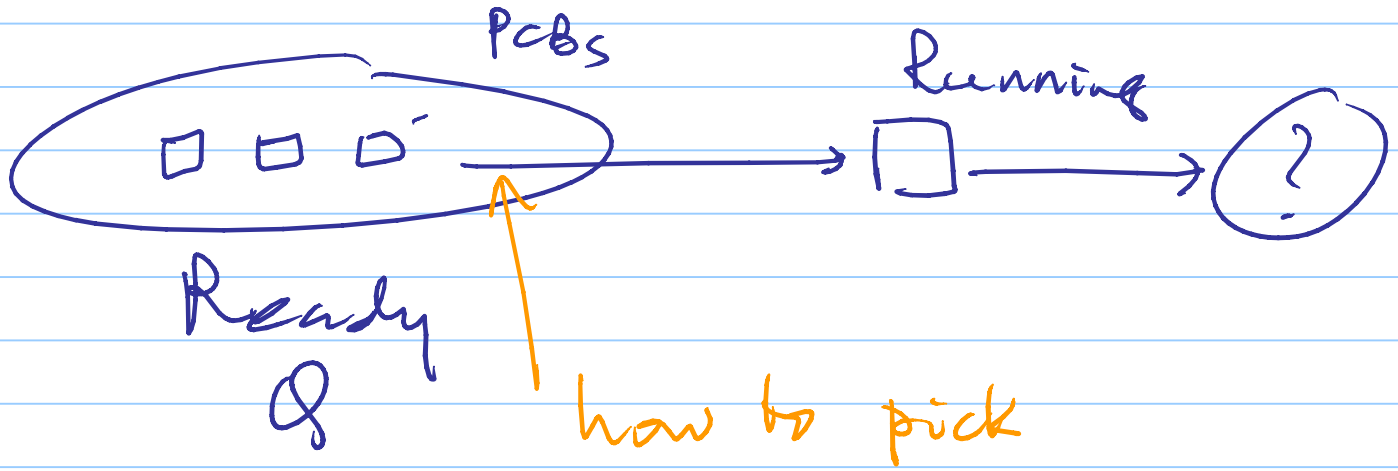

P1
push
send
P2
pull
data
buffer
rcv


RPC
P1
P2
data
call
return
Server
methods/procedure

why msg passing etc

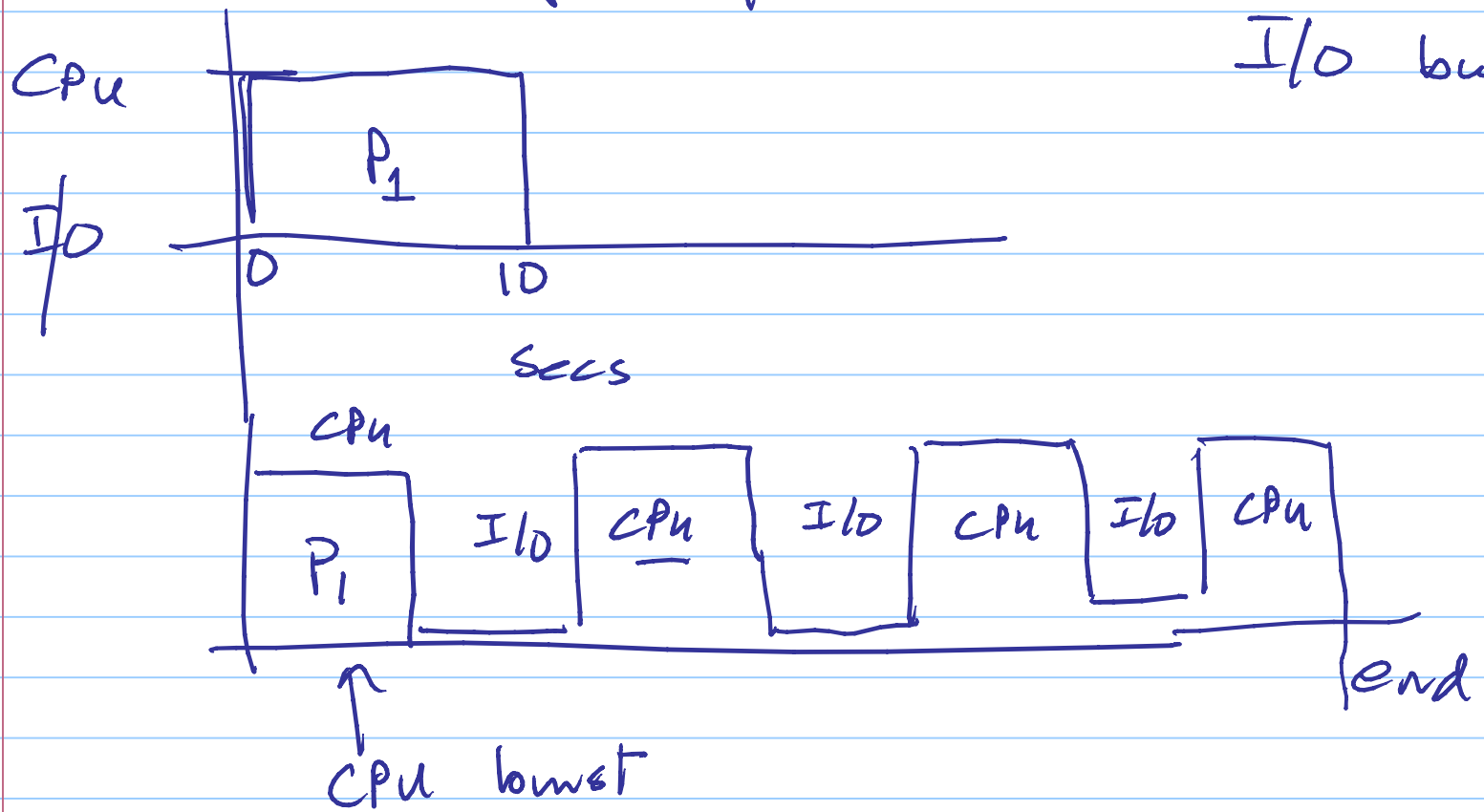
1. Information sharing
2. Computation speedup
3. Modularity
4. Convinience

CPU scheduling basics



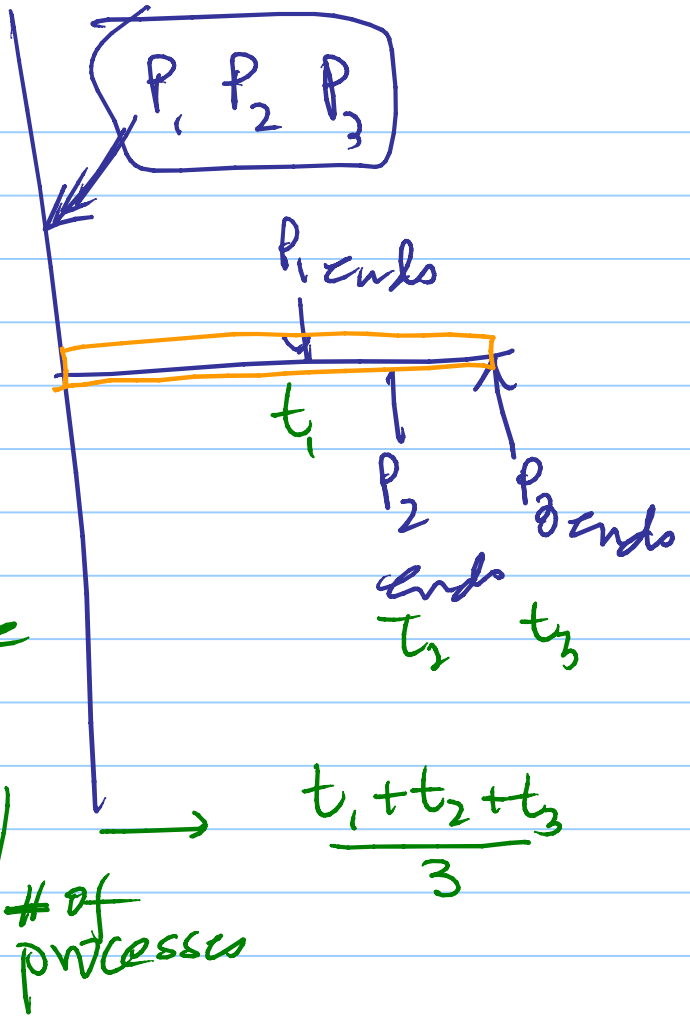
Assume 1 processor

Behavior of a process → CPU burst
I/O burst

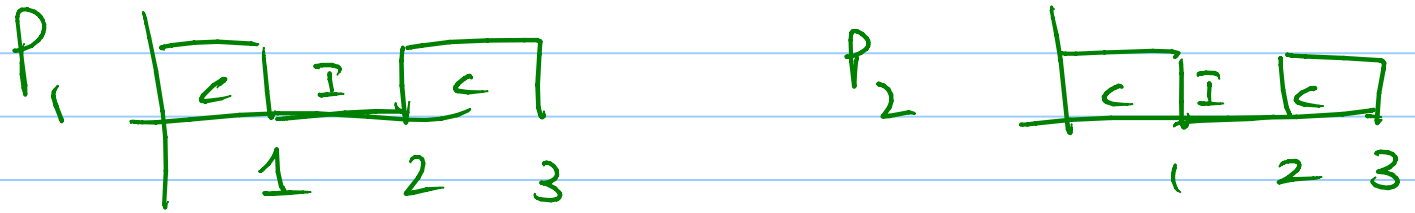


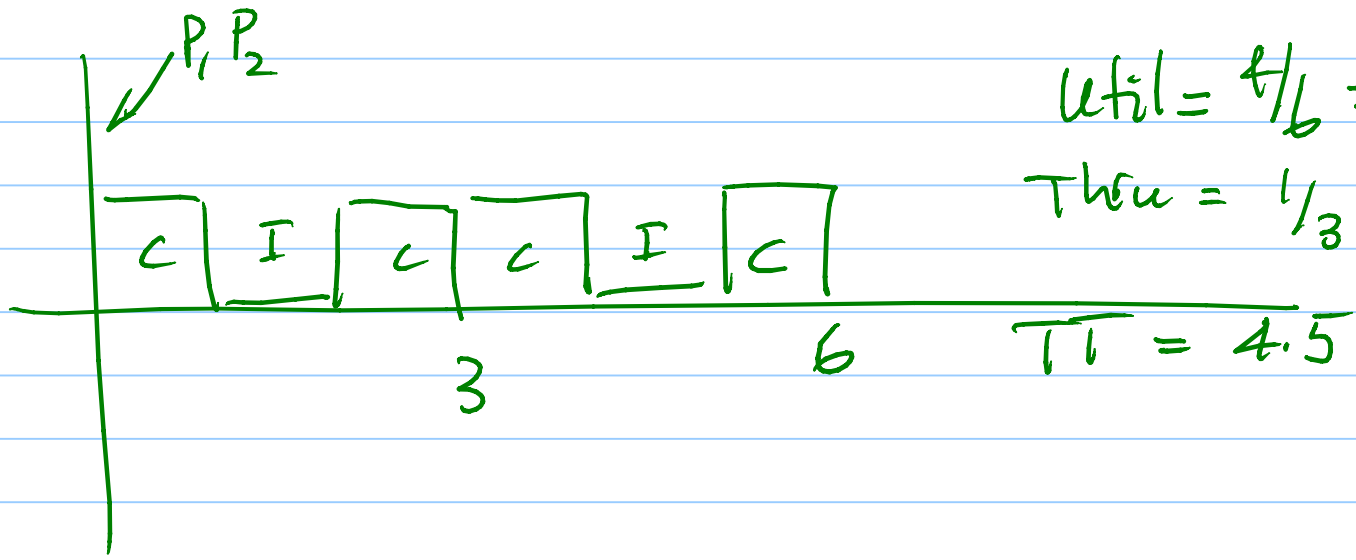
Scheduling metrics

- CPU utilization
  - ↳ (time CPU is busy) / total time
- Throughput
  - # of processes completed / total time
- Turnaround time
  - $\left(\sum_{\text{process}} \text{time taken for each}\right) \Big/ \#\text{ of processes}$



Serial Scheduling





Util = 4/6 = 66%

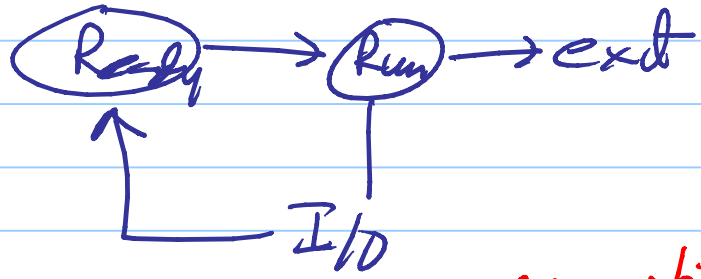
Thru = 1/3

TT = 4.5

2 forms of scheduling

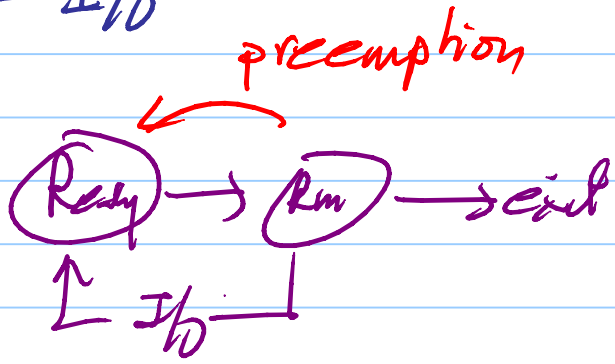
1. non preemptive

   [do not stop CPU burst]



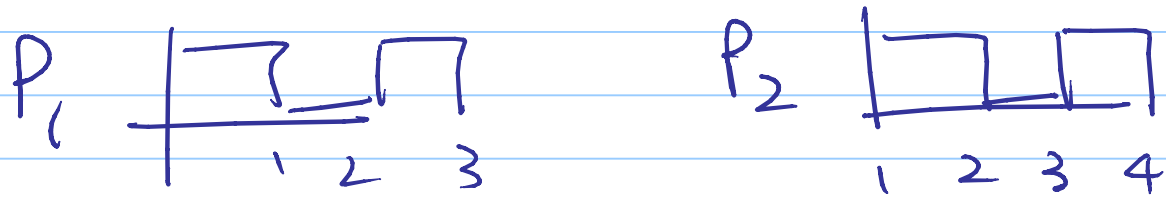
2. preemptive

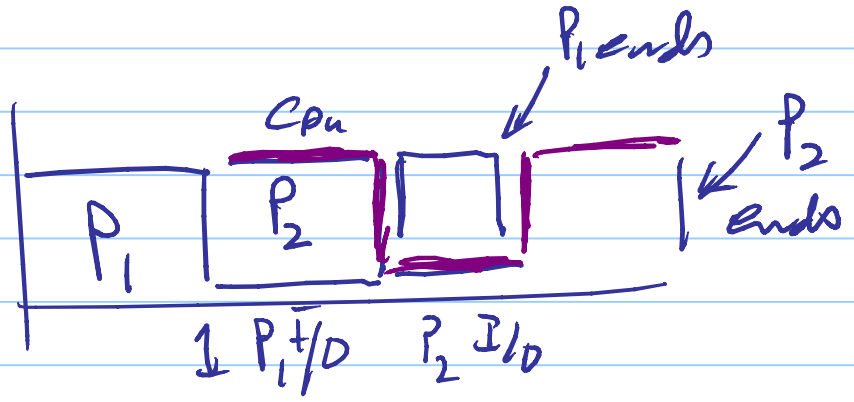
   [can stop a CPU burst
   ↳ preemption]



Non preemptive scheduling

① FIFO



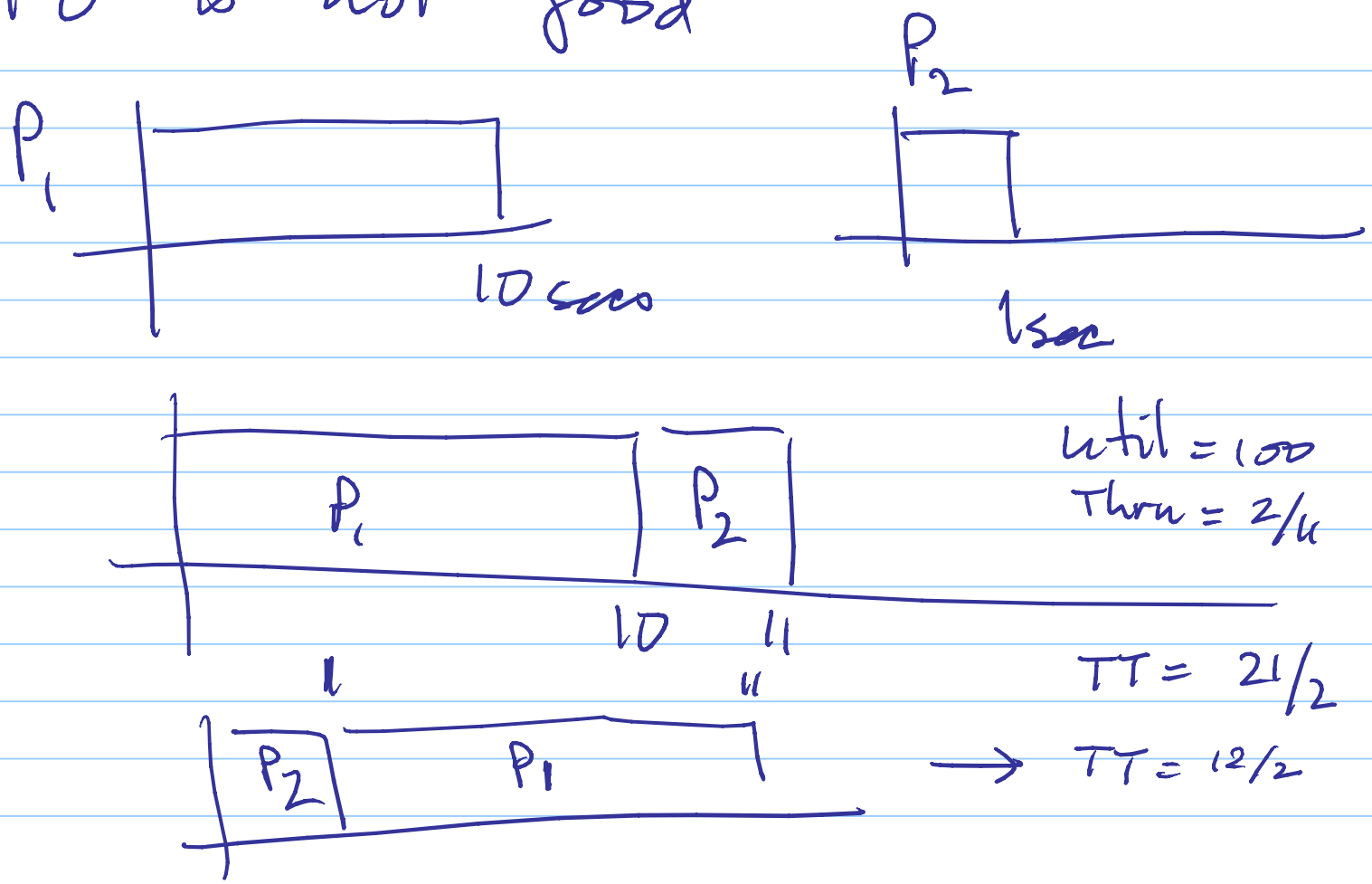


UTIL → 100

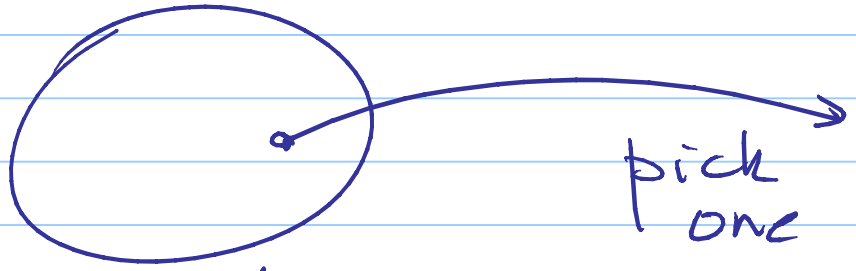
Thru → 2/4 = .5

TT → $\frac{3+4}{2} = 3.5$

FIFO is not good

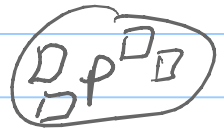


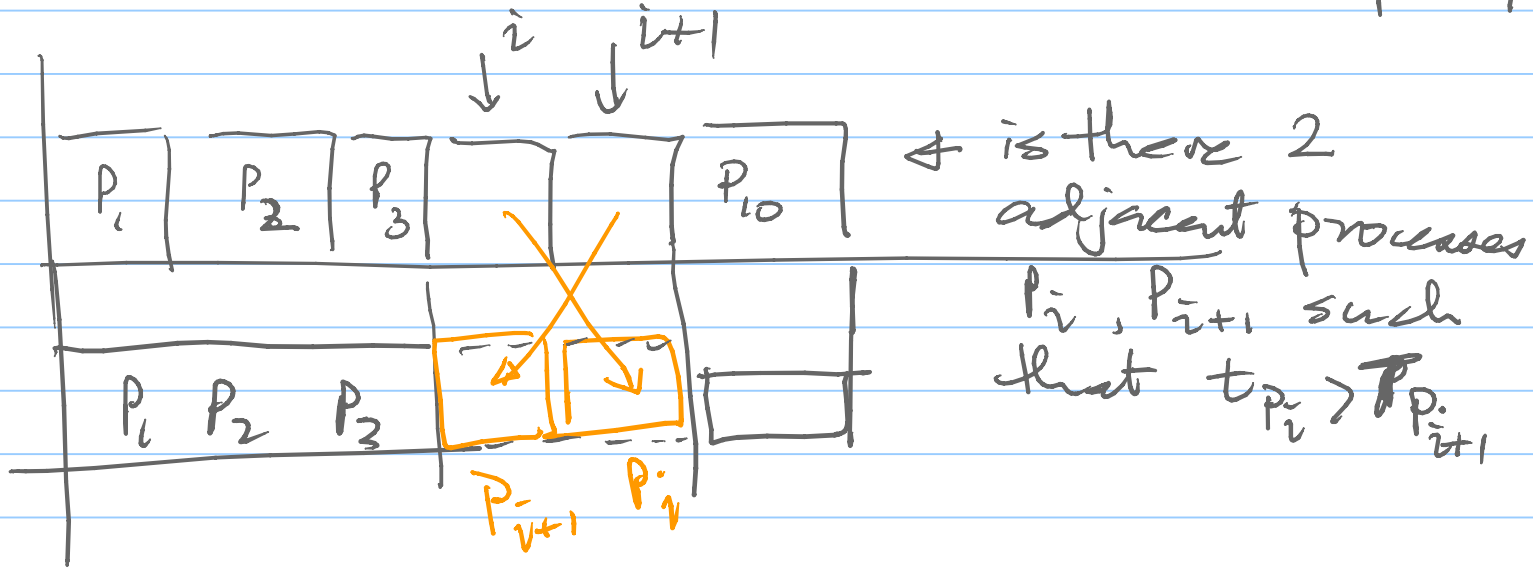
Shortest job first SJF



→ pick the one with the shortest CPU burst

SJF is an optimal CPU scheduling policy





↳ is there 2 adjacent processes P_i, P_{i+1} such that $t_{P_i} > t_{P_{i+1}}$

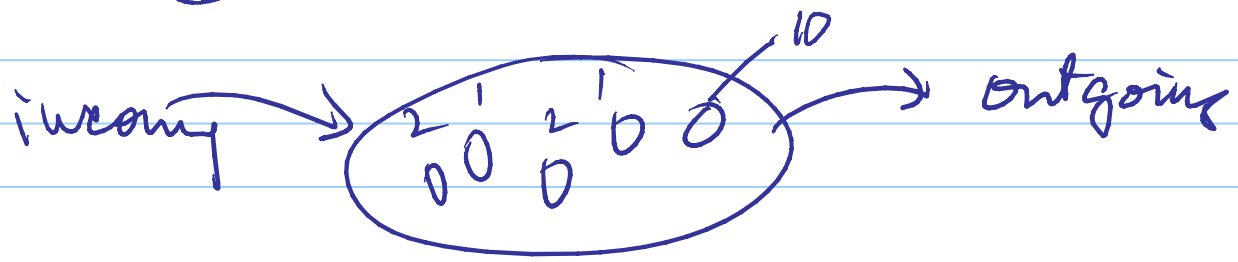
SJF is not useful / implementable ....

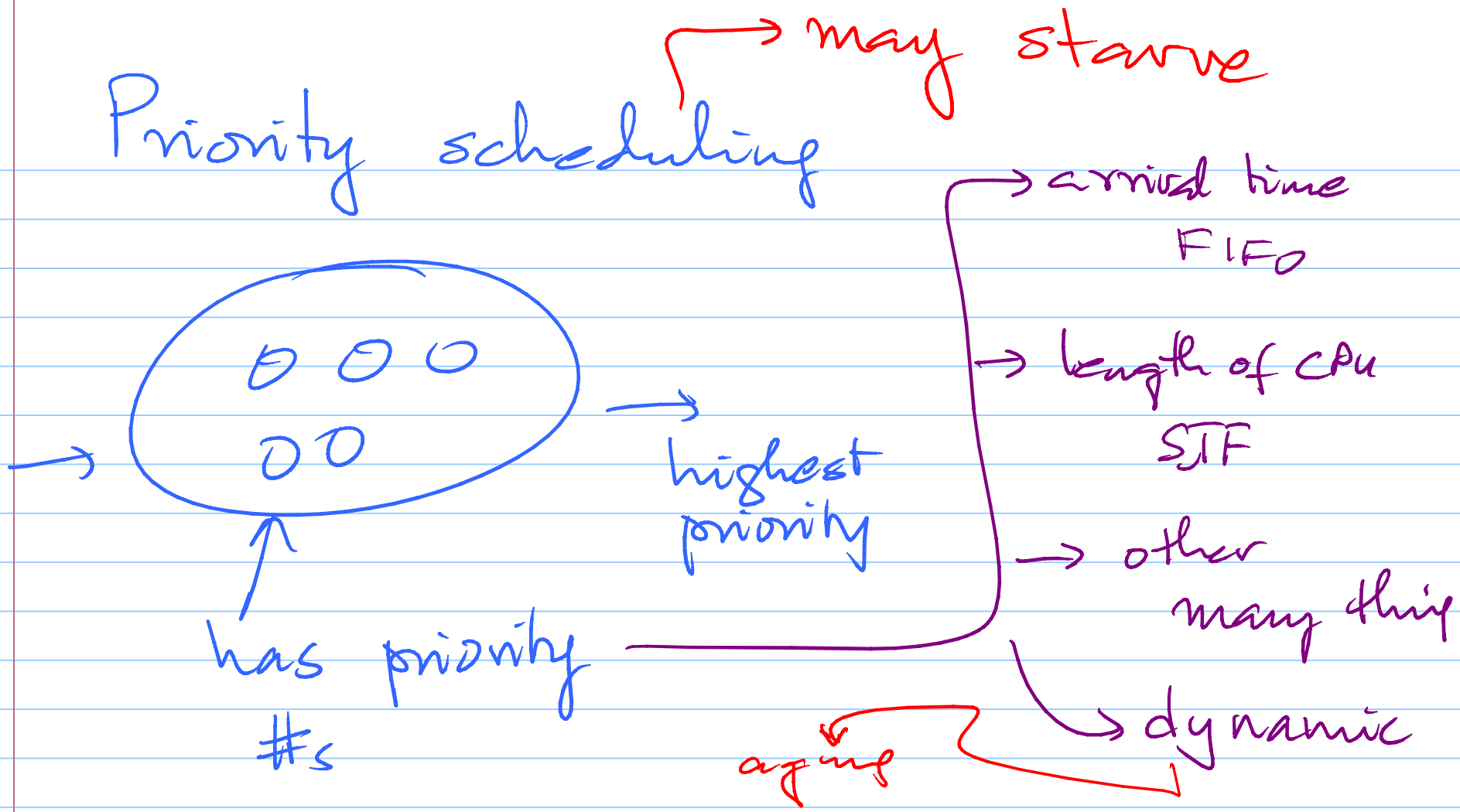
- optimal on turnaround time

But

① CPU burst times are not known in advance

② SJF leads to starvation




Priority scheduling
may starve
arrival time
FIFO
length of CPU
SJF
other
may think
dynamic
aging
highest
priority
has priority
#s