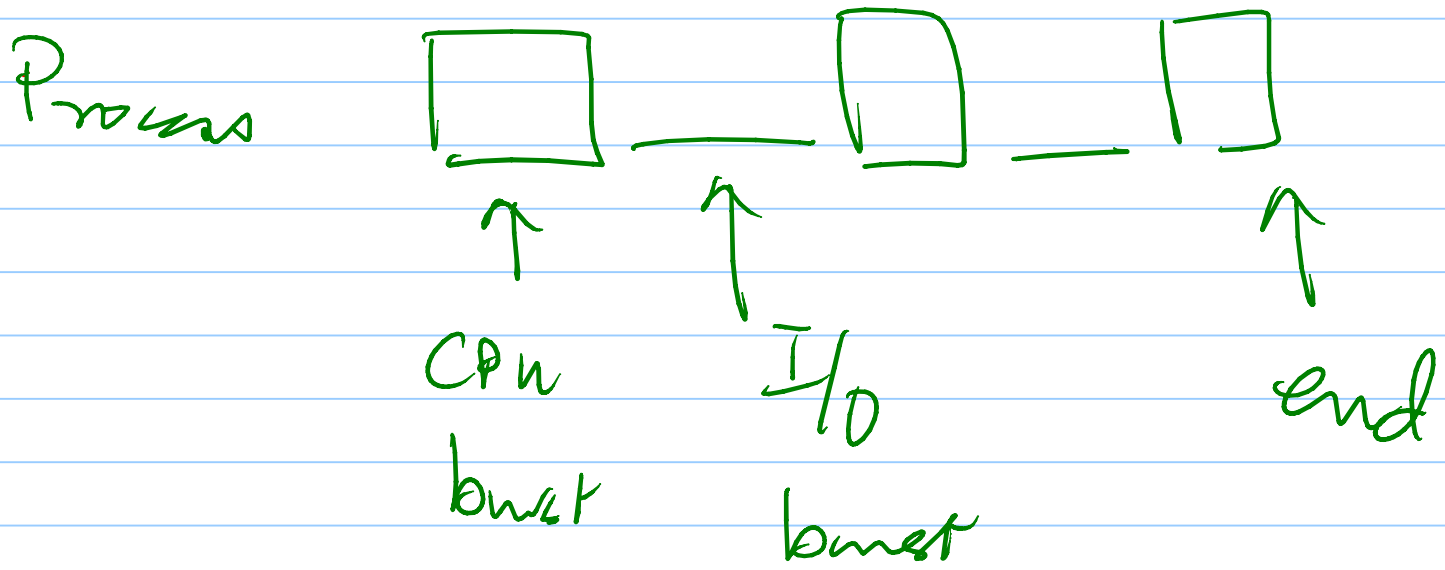
CSE 330: Operating Systems

Class: 07 **Date:** 1/30

Spring 2018

Note Title

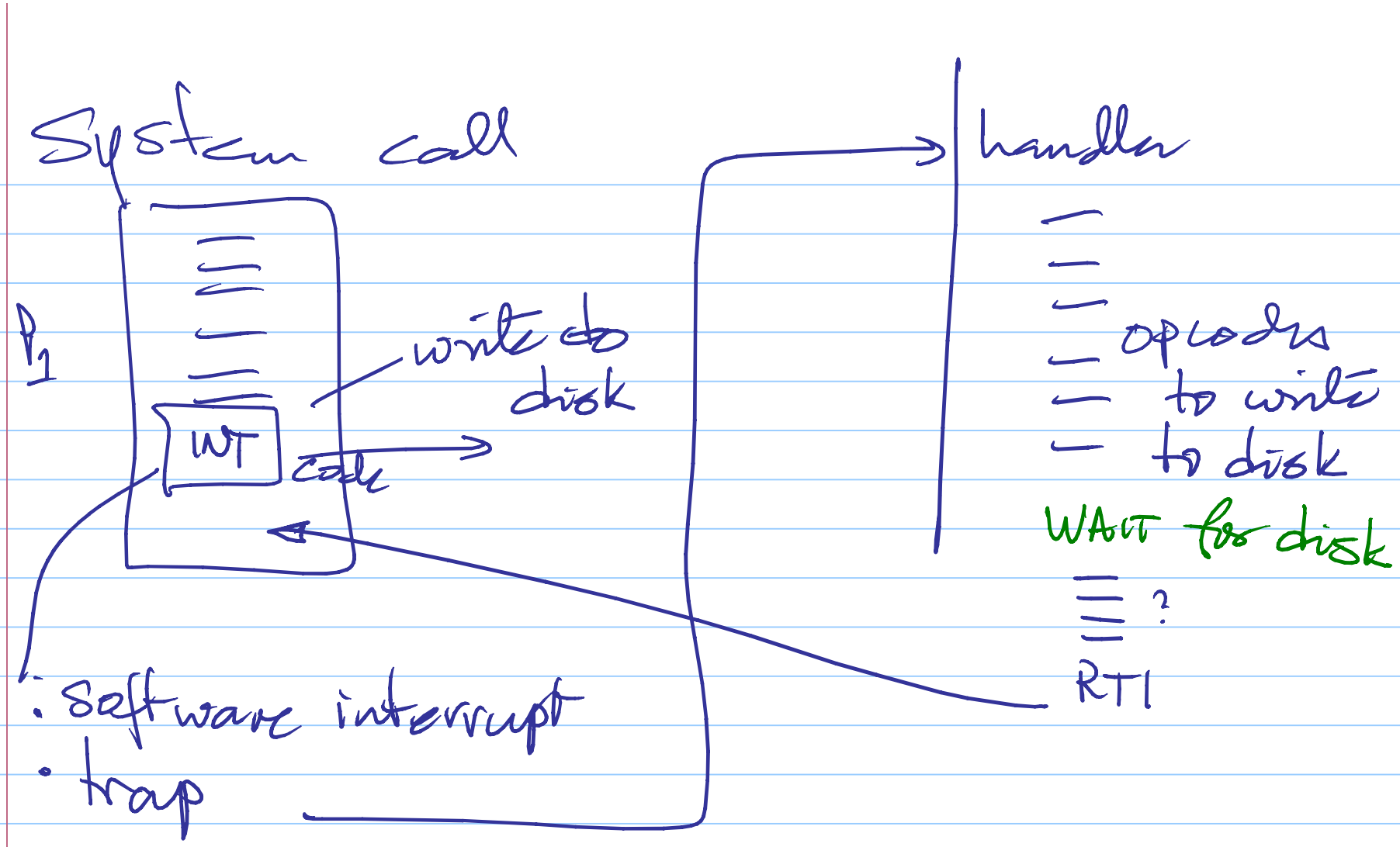
Non preemptive scheduling

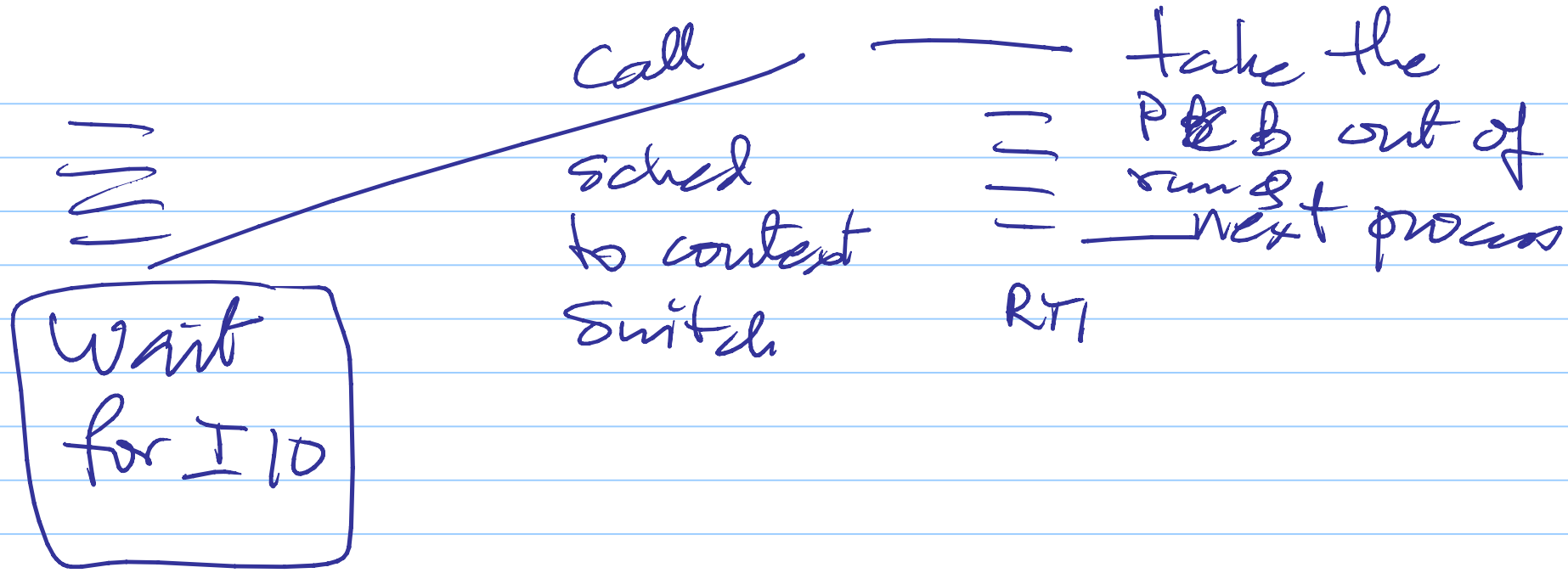


non - preemptive

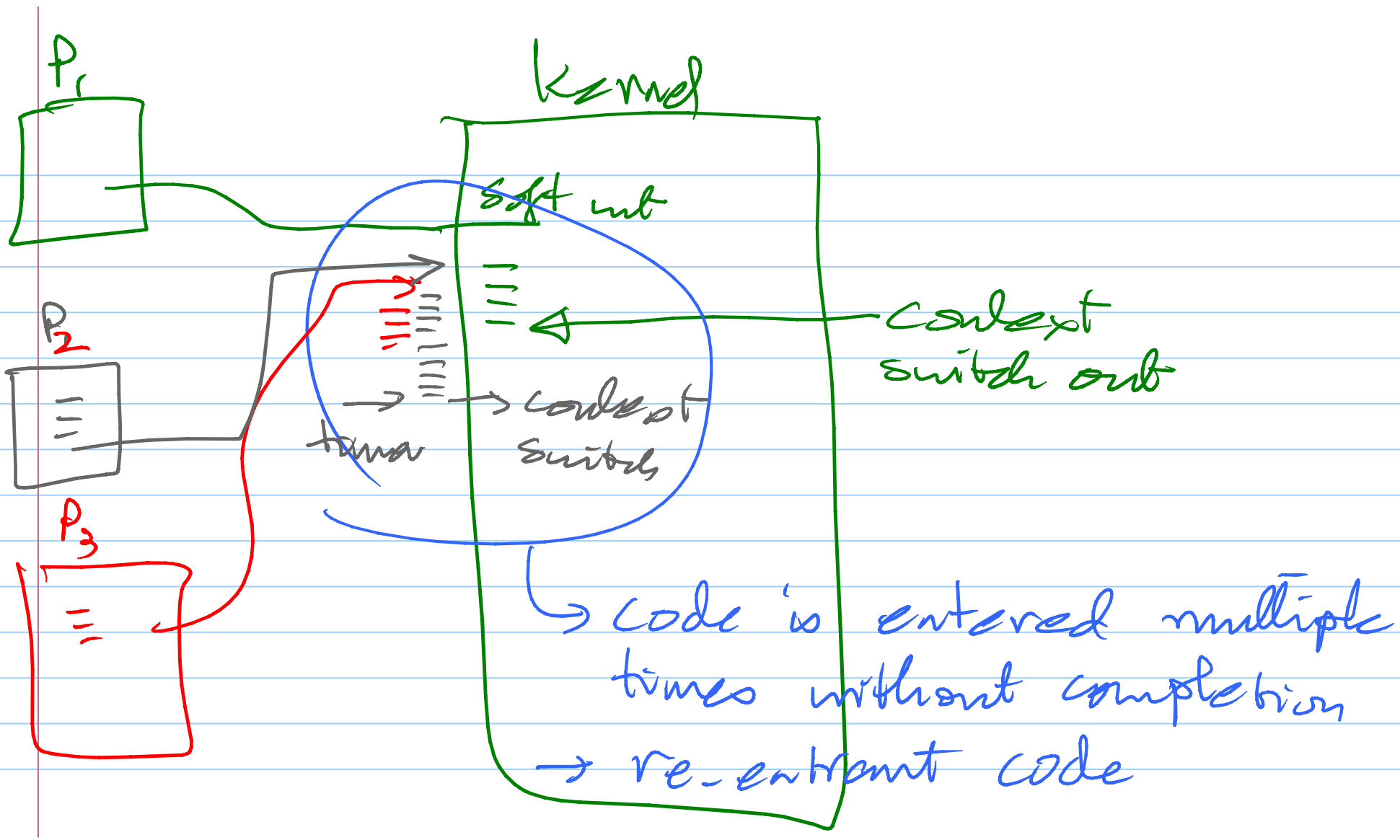
- FIFO
- SJF
- Priority

System calls
INT
this is still P1's execution
RTT
INT
P1
P2


System call
handler
P1
write to disk
INT
call
opcodes
to write
to disk
WAIT for disk
?
RTI
: software interrupt
• trap

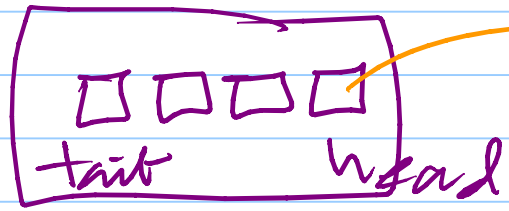


→ disk int handler will put P1
back into ready Q


P1
P2
P3
kernel
soft int
context switch out
timer
context switch
code is entered multiple times without completion
→ re-entrant code

Preemptive Scheduling

① Round Robin



Ready Q

FIFO Q

Run → till

A fixed small amount of time

① CPU burst ends → I/O Q

OR

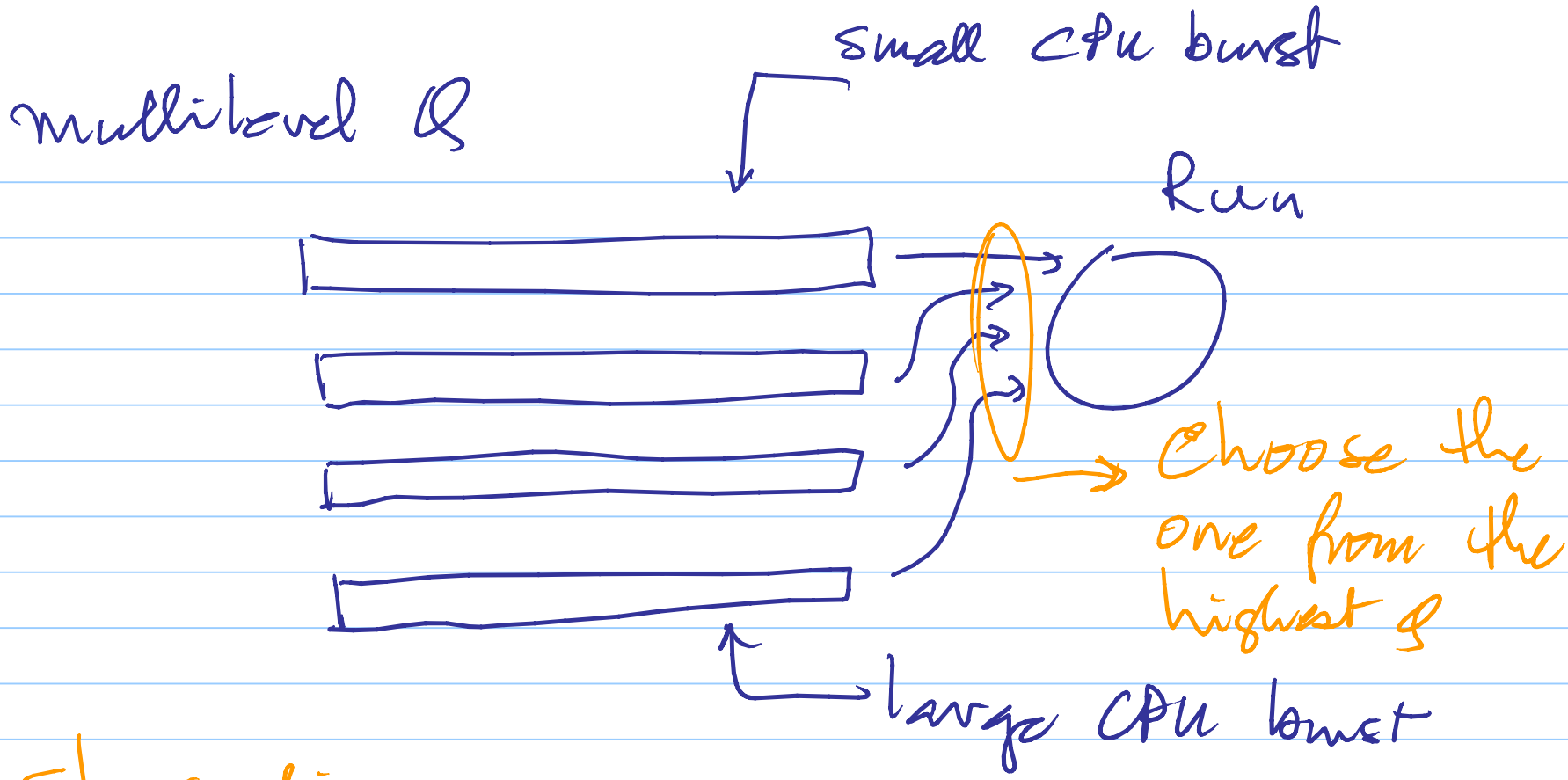
② Process terminates → exit Q

OR

③ Time Quanta Expires

move to Ready Q

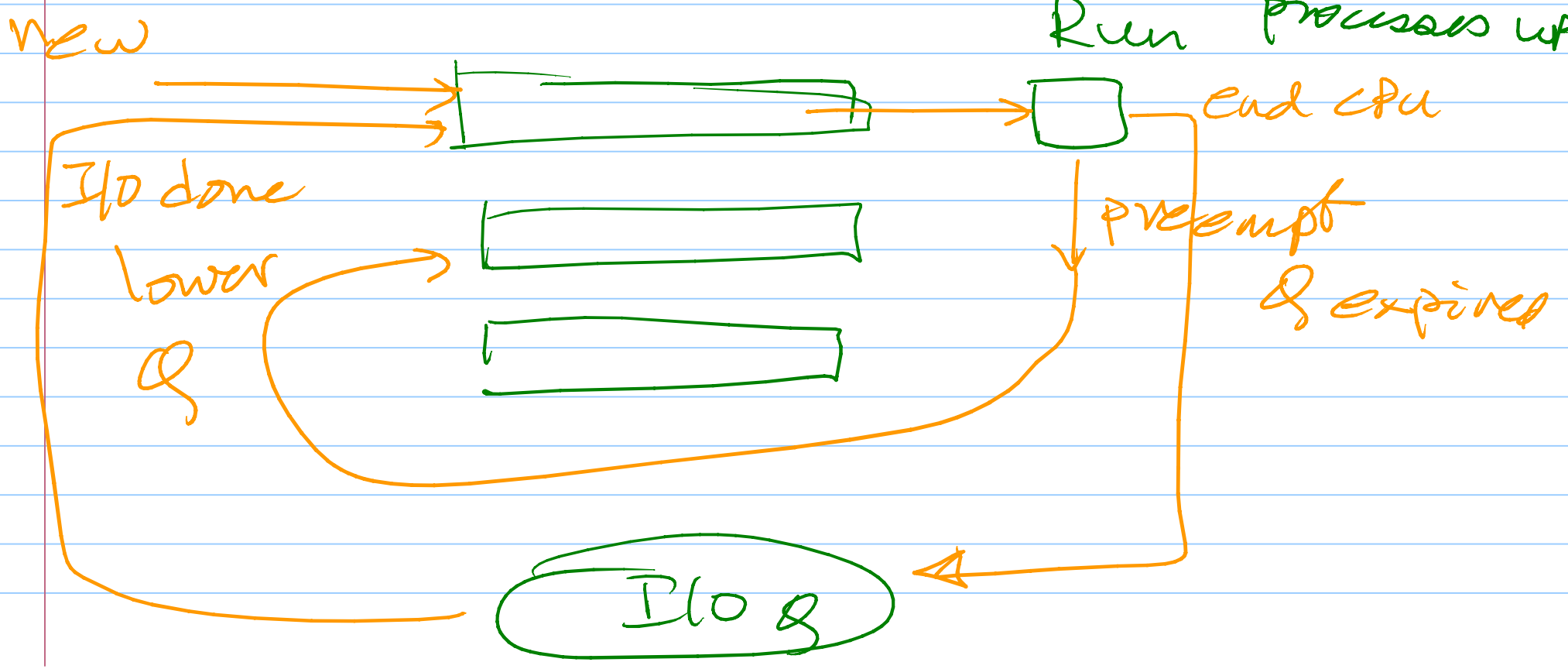
P1
P2
P3
P4
P5
interactive
(short cpu)
long cpu
burst
head



- starvation
- classification of apps

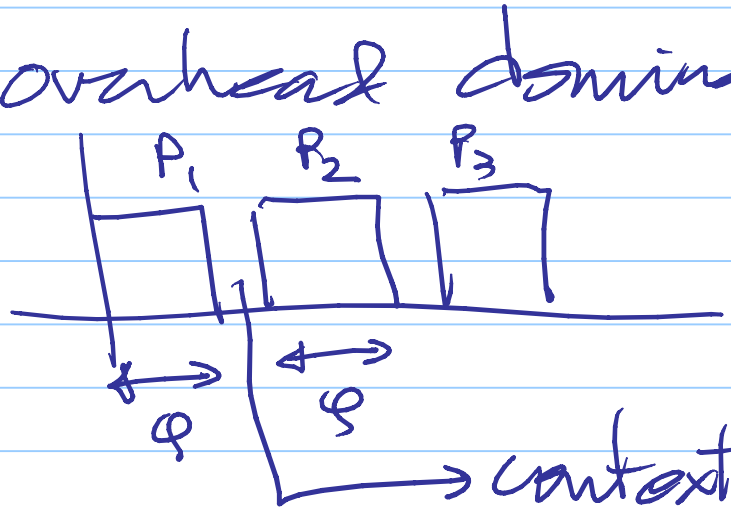
Multilevel Feedback Q

→ add aging
↳ move old



Time Quanta

Small → overhead dominates



Large → → context switch

Response time

→ average time taken for a process to start running on CPU after it gets ready

→ (avg length of Ready Q * (quanta)

Windows Scheduler

32 level MLFQ — 1–15 for regular

16–32 for real time

Unix Scheduler

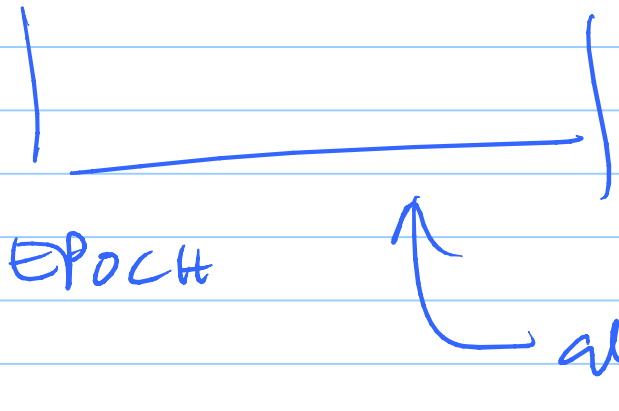
NICE → −20 to +19

↑ high ↑ 0 is normal

↓ Low

Linux 2.4 Epoch scheduler

3 classes interactive, batch, real time



all processes get
1 time ϕ ⇒ also time ϕ
is different...

Linux 2.6 (original)

- 140 qms 100 r.t, 40 oth
- Active / Expired qs (switch)
- priority recalculations

Linux 2.6 CFS

↳ Completely Fair Scheduler

↳ red black tree

O(log n) to pick / rebalance tree

key = CPU time used

?

Scheduler Evaluations

1. traces
2. queueing models
3. simulation
4. implement (+ instrument)