

CPU scheduling metrics

Interactive
use

- CPU utilization

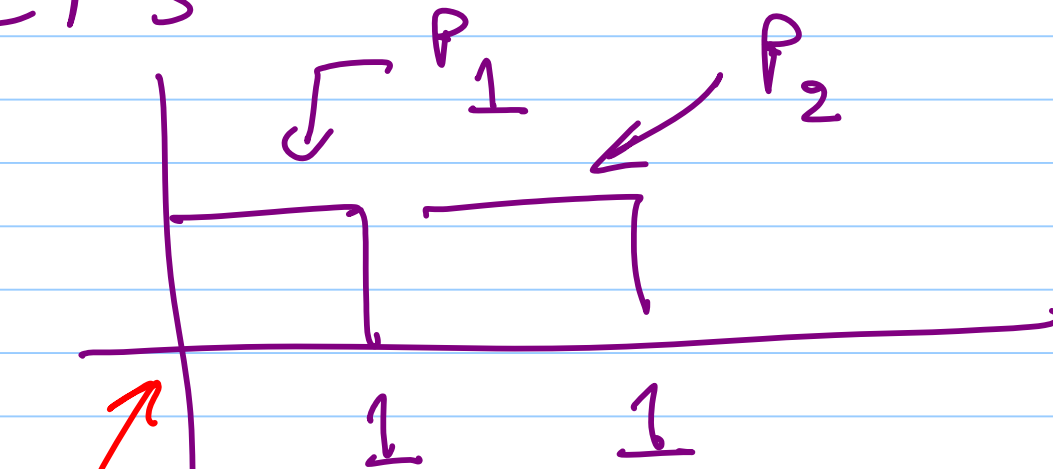
- Throughput \rightarrow # of jobs / time

\rightarrow - Turnaround time

\rightarrow - Response time

\rightarrow average
of how long
per job

FCFS

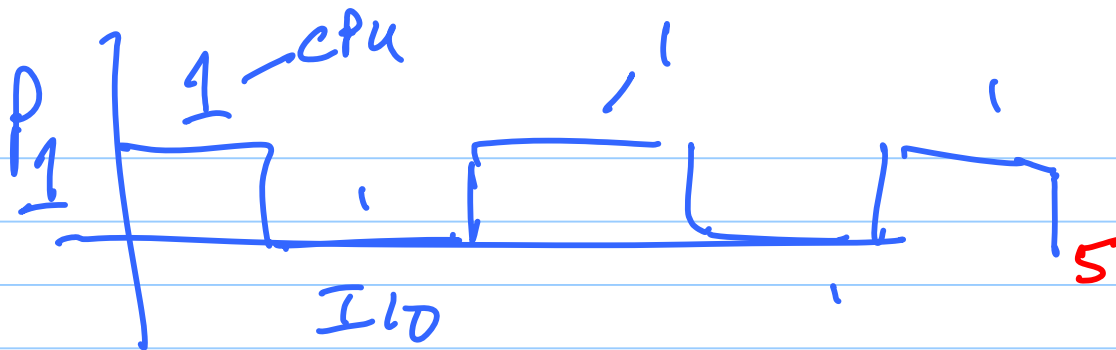


P_1 & P_2
arrived here

CPU bursts
no I/O.

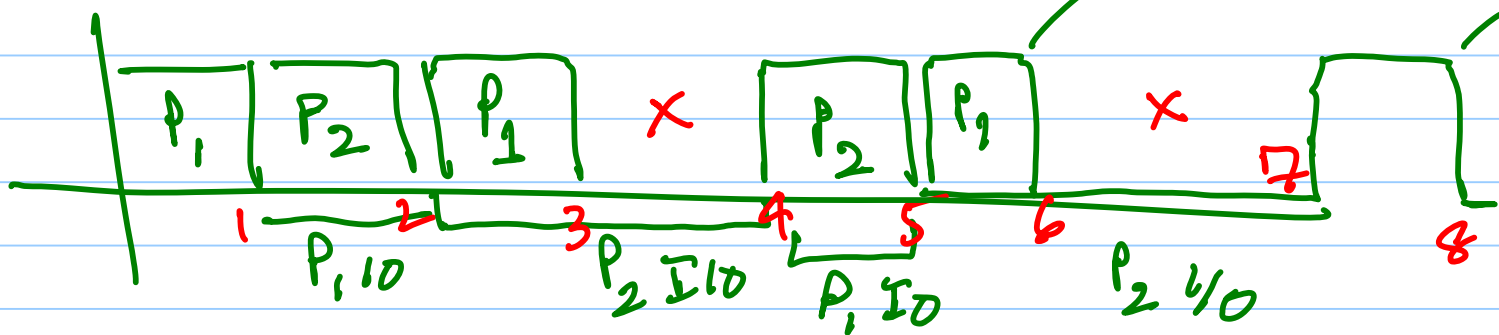
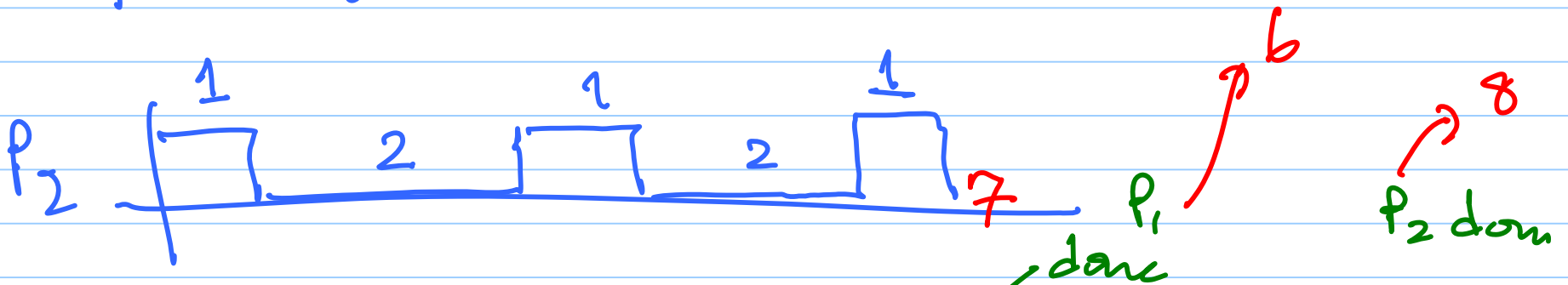
non
preemptive

$P_1 \rightarrow 1s$
 $P_2 \rightarrow 2s$ } 1.5 sec



$$UTIL \rightarrow 6/8$$

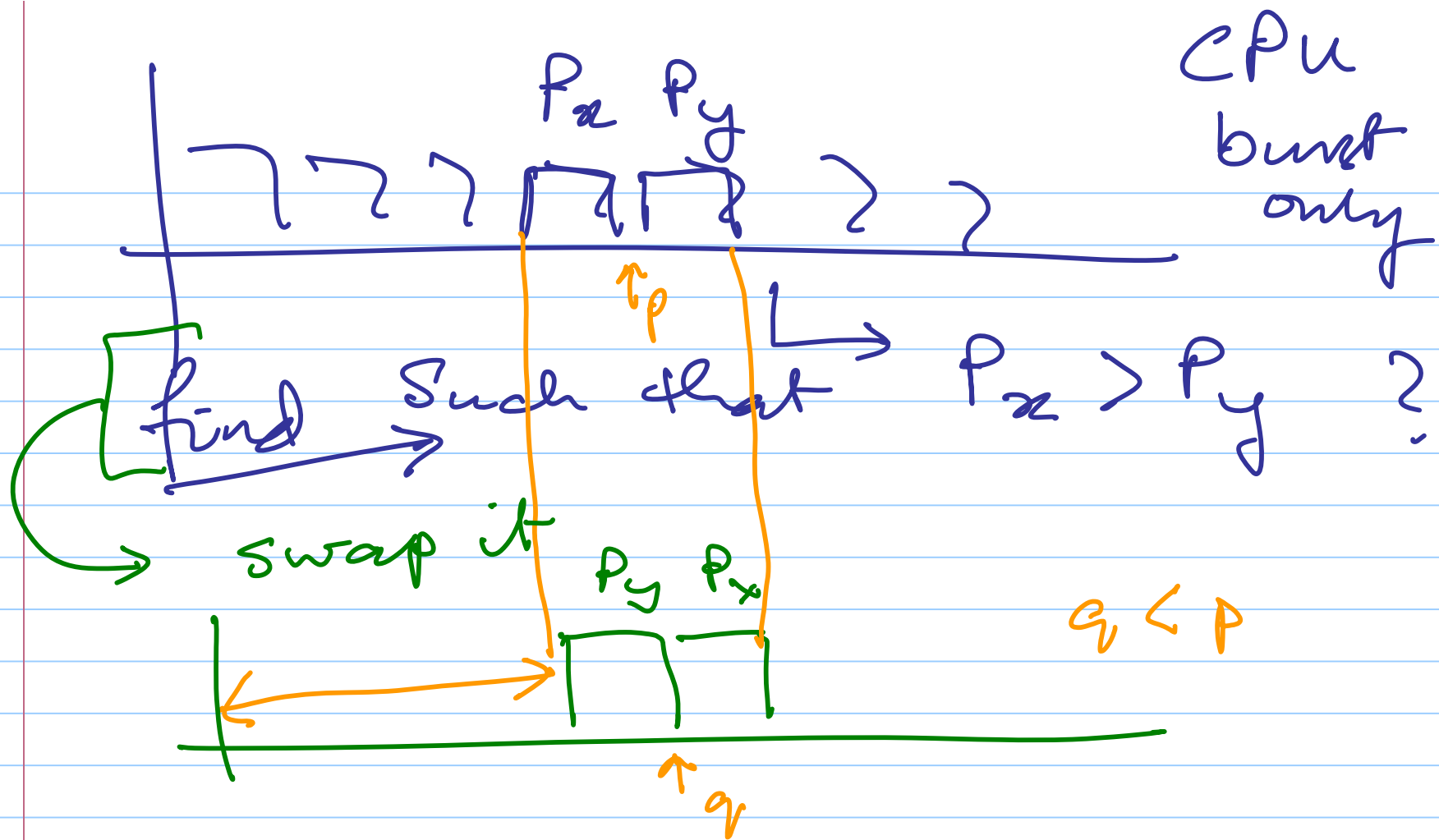
$$TT \rightarrow \frac{6+8}{2} = 7$$



SJF \rightarrow shortest "job" first

\rightarrow for each CPU burst

\rightarrow optimal



SJF \rightarrow

- ① NOT implementable
- ② starvation prone

Priority scheduling

↳ highest priority first

FCFS → arrival time is priority

SJF → CPU burst length "

Need preemptive ...

→ long running / infinite loops need to be interrupted

→ Interactive usage ...

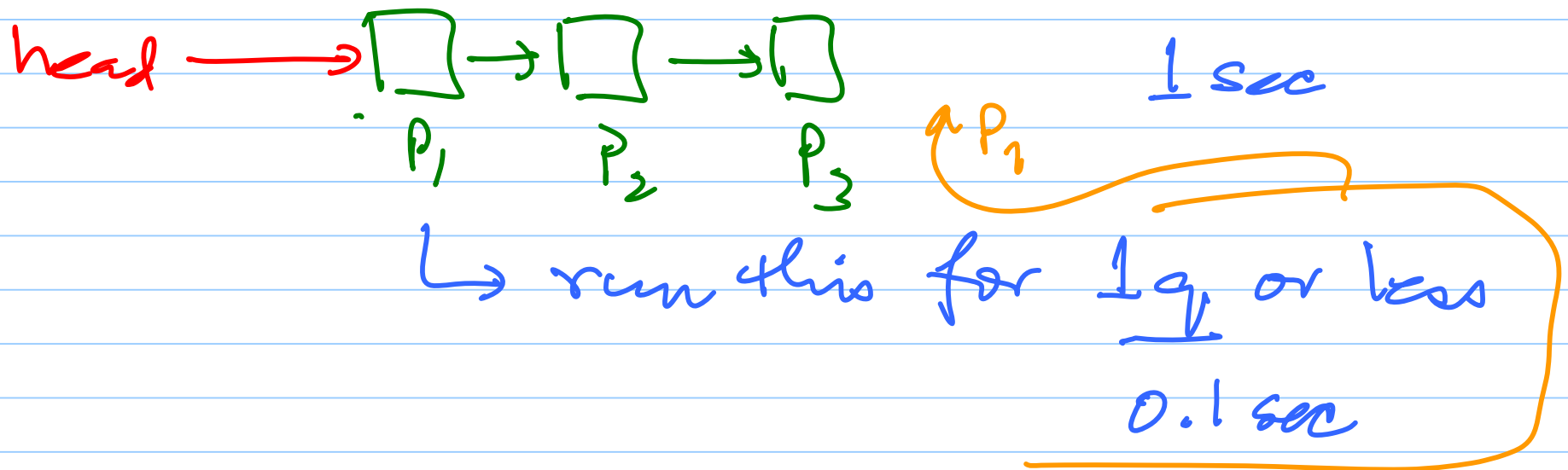
Round-Robin. (preemptive)

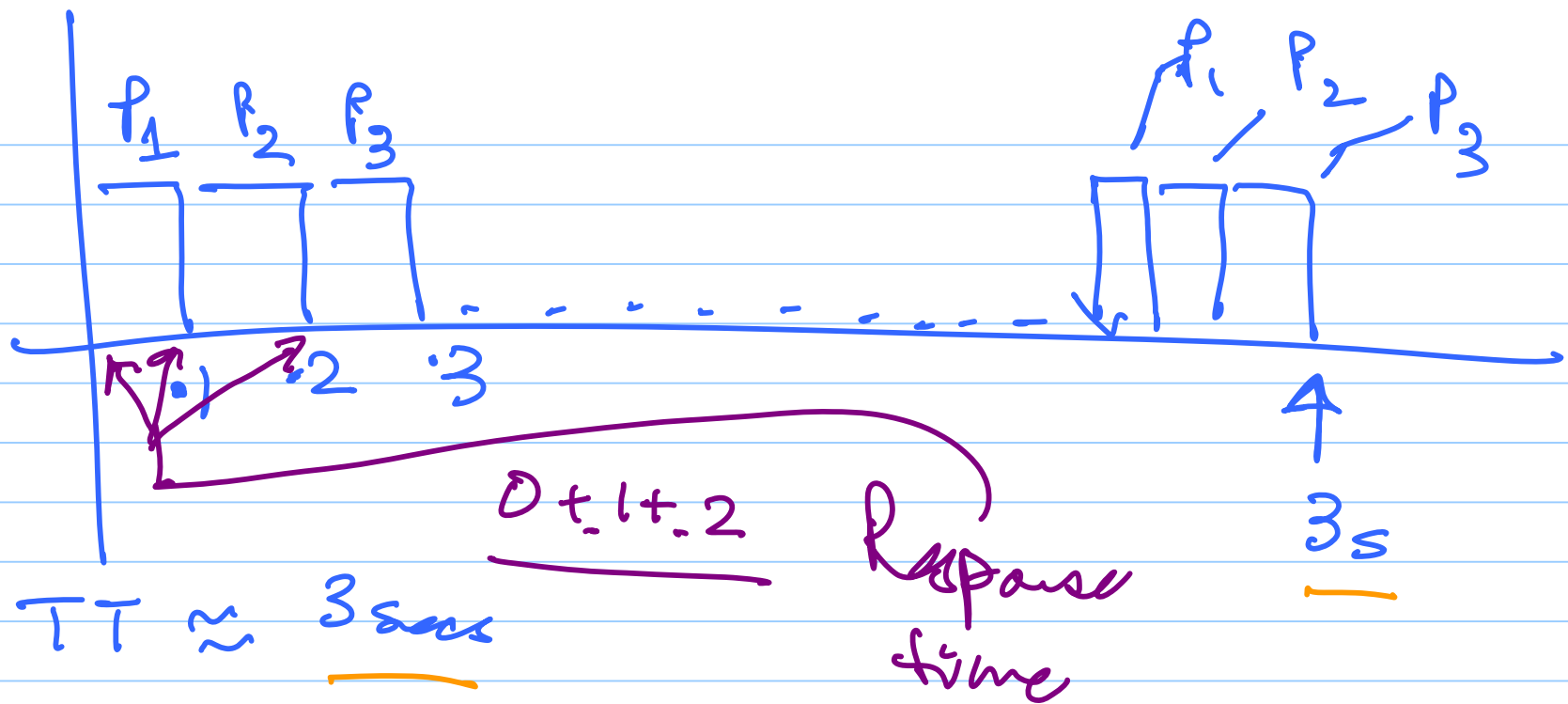
↳ time quanta

↳ the longest interval of time a job is allowed to use the CPU

↳ FCFS

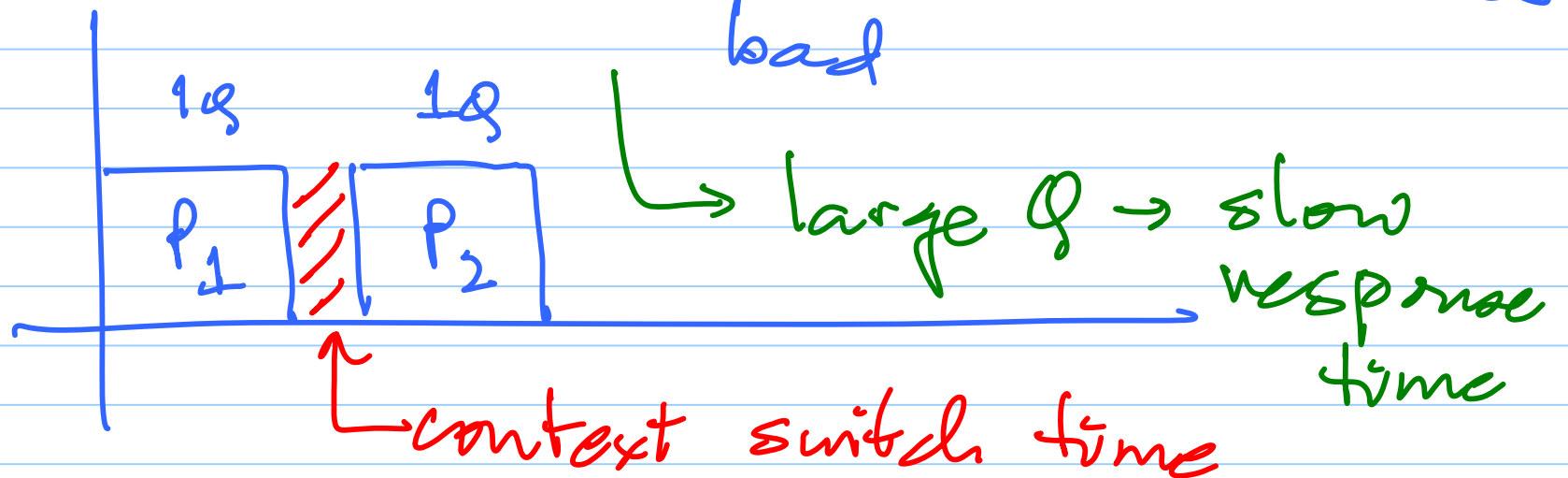
Q of ready processes

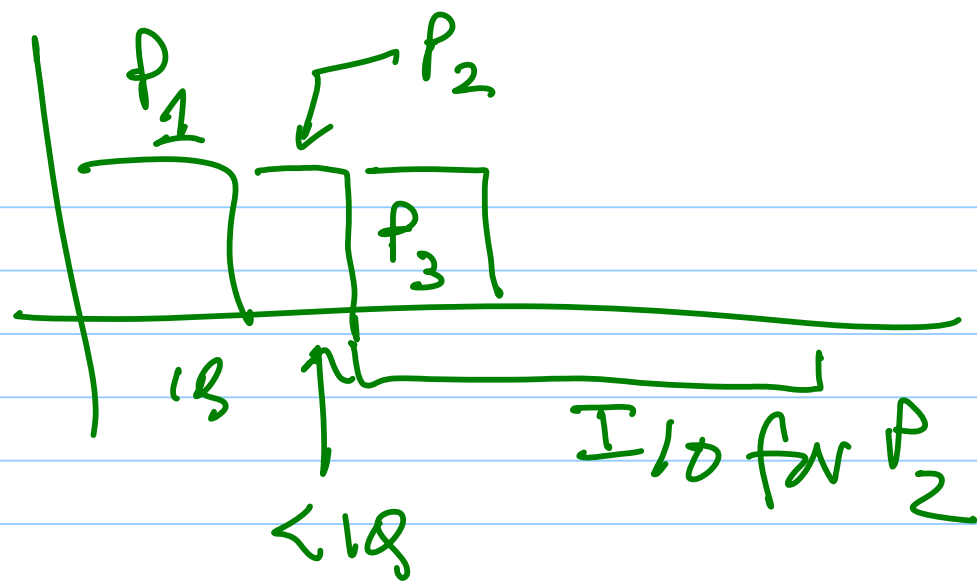




TT, FCFS non pre \Rightarrow 2 secs.

time $Q \rightarrow$ should be very small
 $\rightarrow 0$
 \hookrightarrow overhead becomes bad





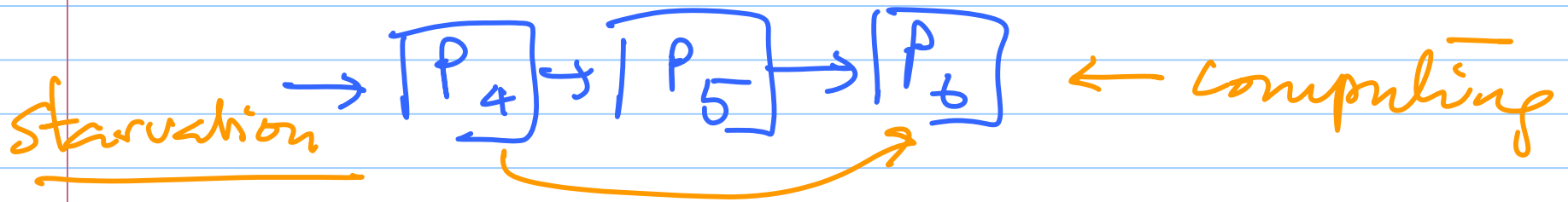
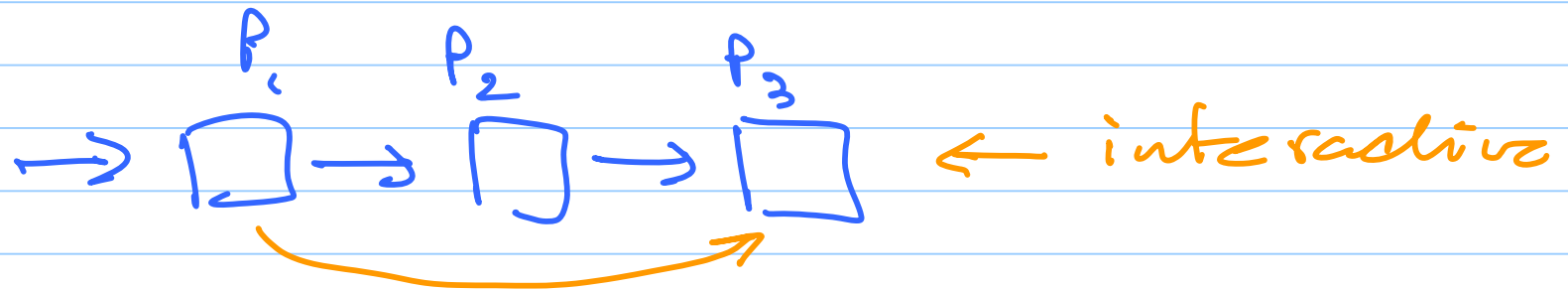
RR has poor performance

$P_1 P_2 P_3 \rightarrow$ interactive

$P_4 P_5 P_6 \rightarrow$ compute
intensive

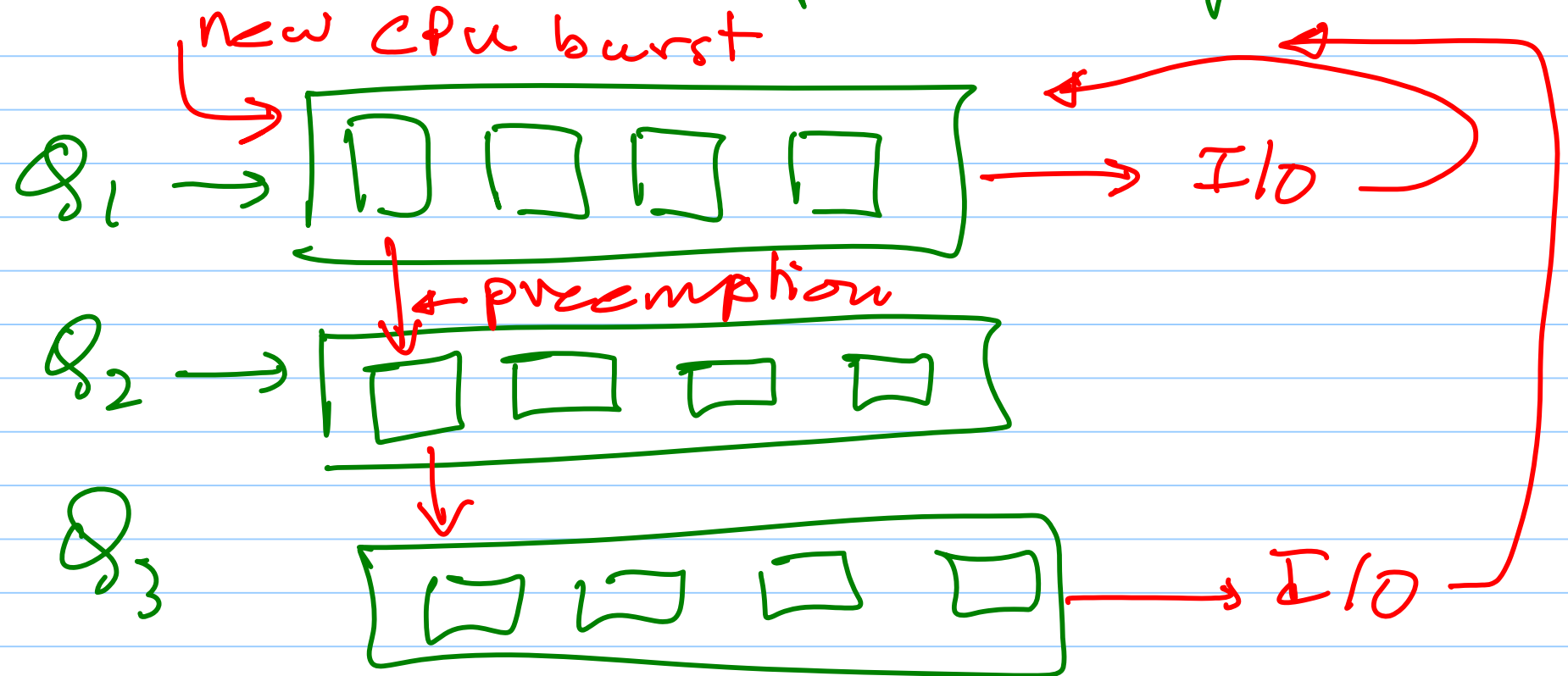
\rightarrow slows down response for $P_1 P_2 P_3$

MLQ \rightarrow multilevel Q's



time q different for diff q memo.

MLFQ multilevel feedback queue



⇒ past predicts the future

Windows — 32 level MLFQ

0-15 → normal
higher → "real time"

UNIX → the NICE system

process has a 'nice' level
→
↳ number
-20 ... +19
↑ nice

$-20 \dots 0 \dots +19$

↔ formula driven ↔

↑ start here
for user processes

↑
~~all~~ only
priv
processes can use this

linux 2.4 → 3 classes

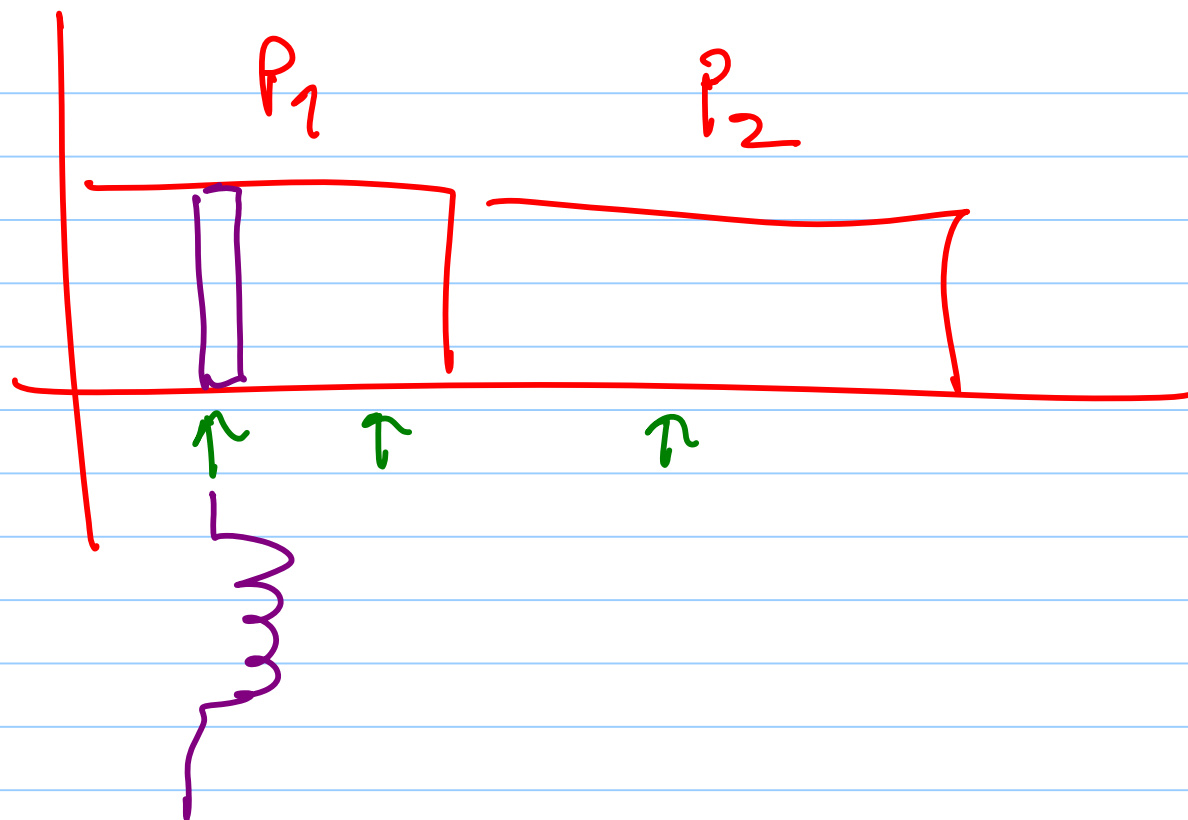
interactive, batch, real time

→ Global Φ

— EPOCH

Linux 2.6 v1 \rightarrow 140 qus. \sqrt{n}
Active / expired qs
:

Linux 2.6 v2 \rightarrow $(\log n)$ CFS scheduler
Completely Fair Sched
 \hookrightarrow balanced binary tree



Evaluation

- traces → keep history
- queuing models
- simulations
- actual implementation