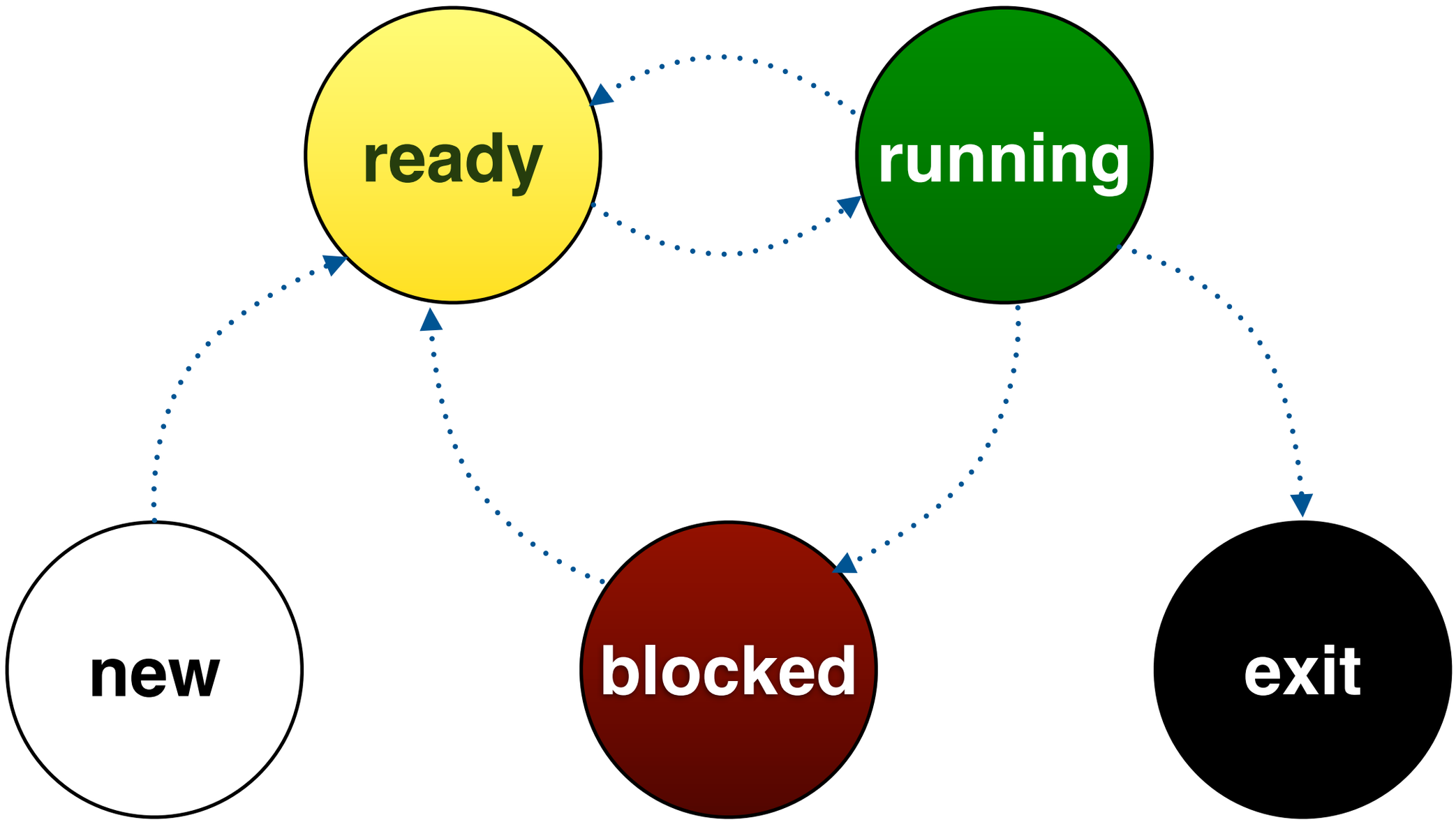
Lecture 6

CPU Scheduling

16 September, 2011


ready
running
new
blocked
exit

batch

vs.

interactive

vs.

real-time

CPU-bound

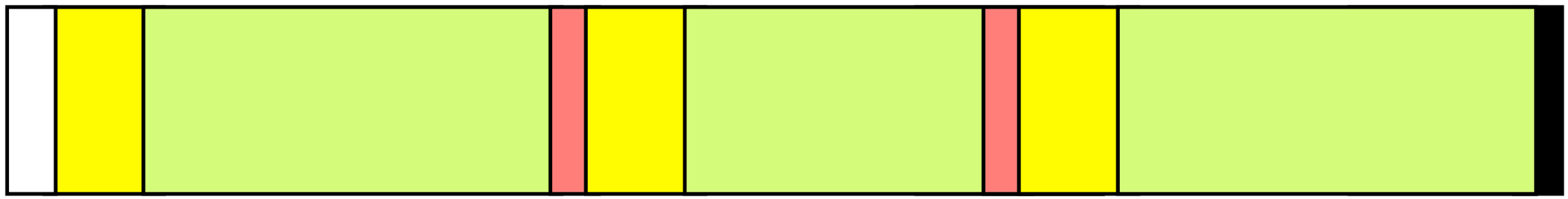
vs.

I/O-bound

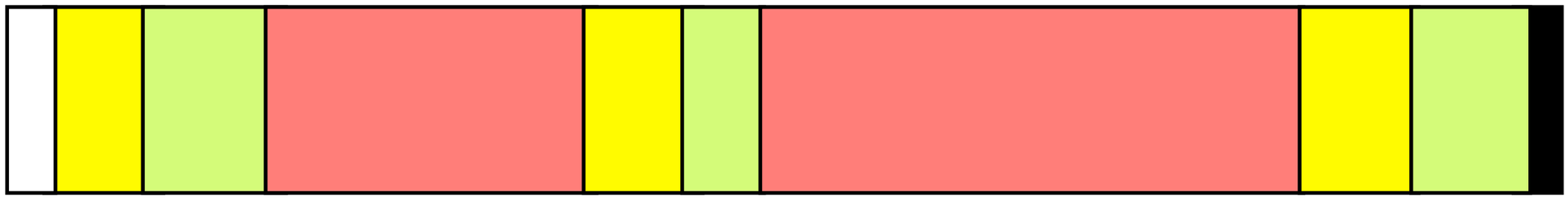
pre-emptive

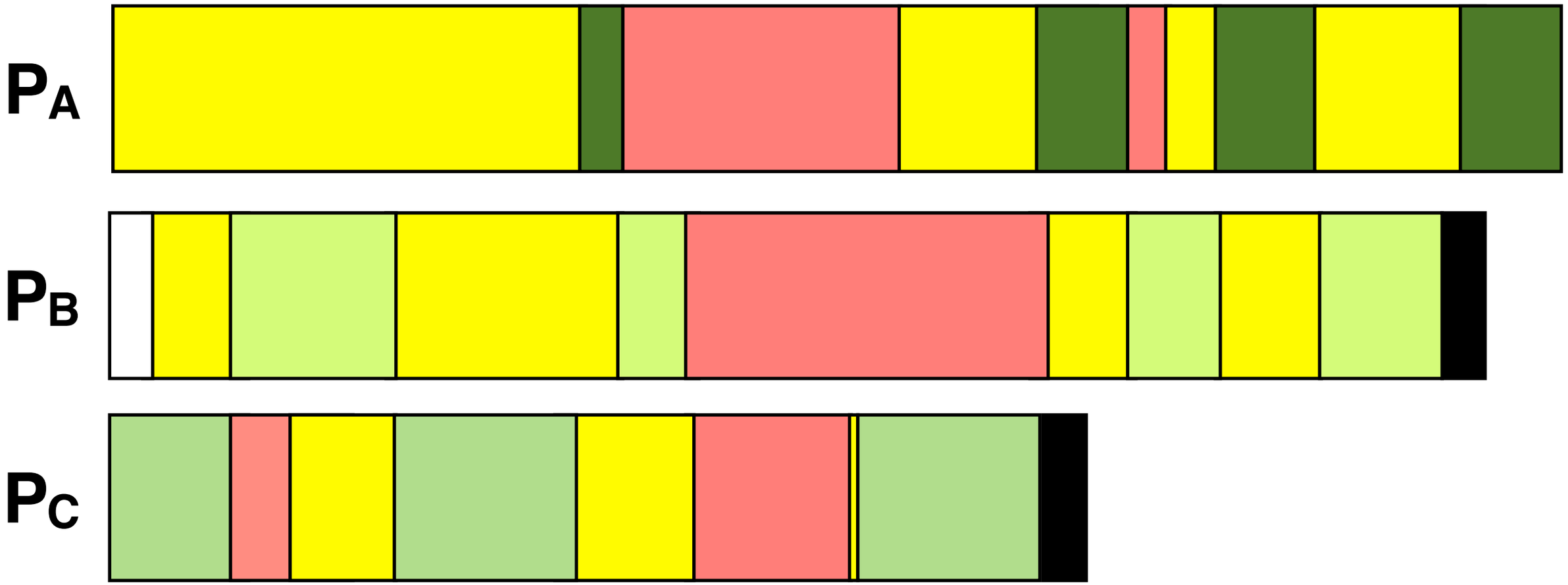
VS.

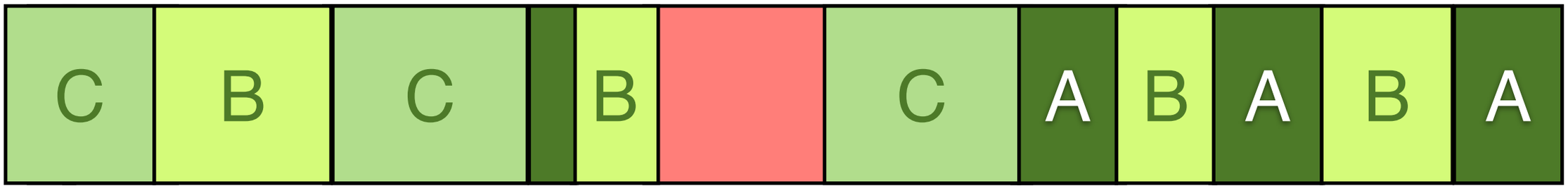
non pre-emptive



(process states)

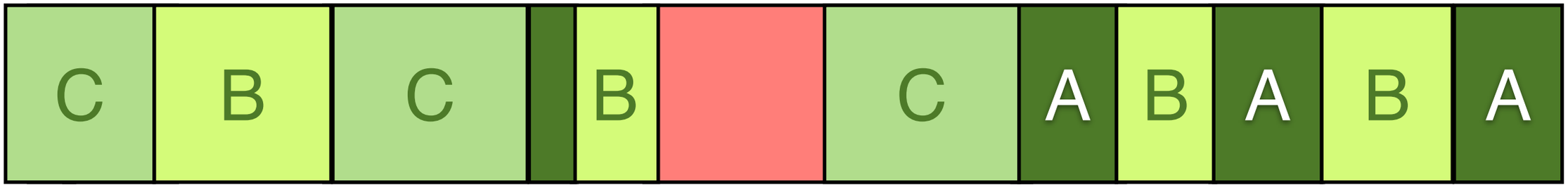



P_A
P_B
P_C

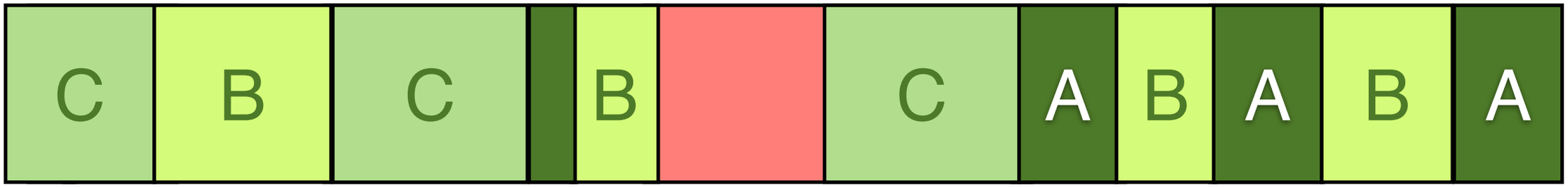


(CPU states)

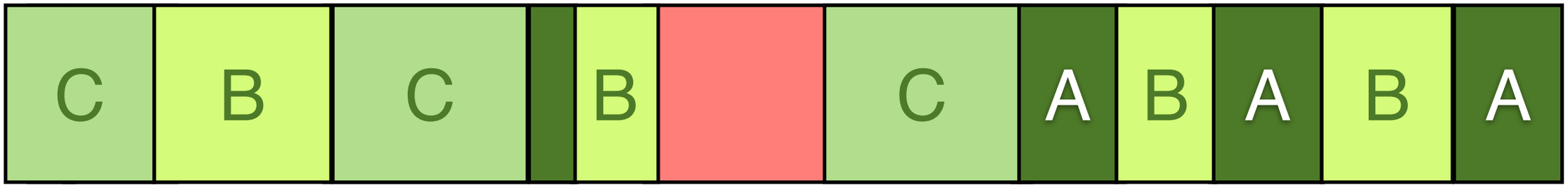
CPU Utilization



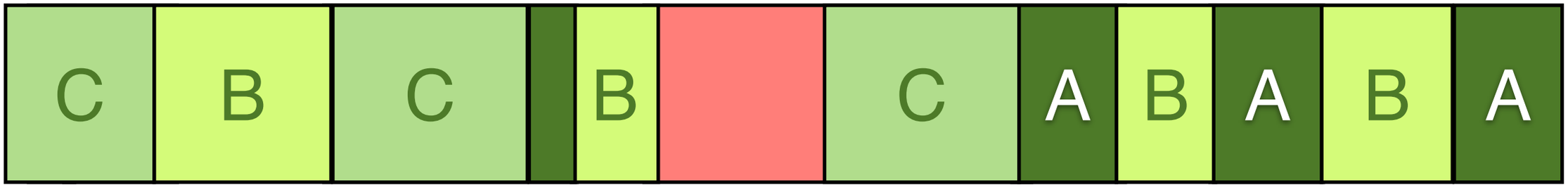
Throughput



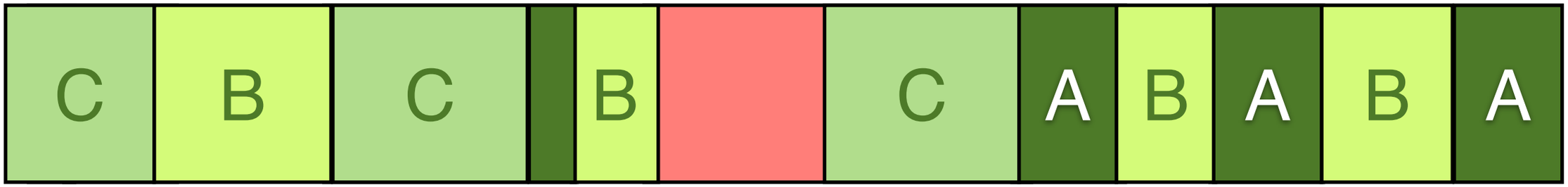
Response Time

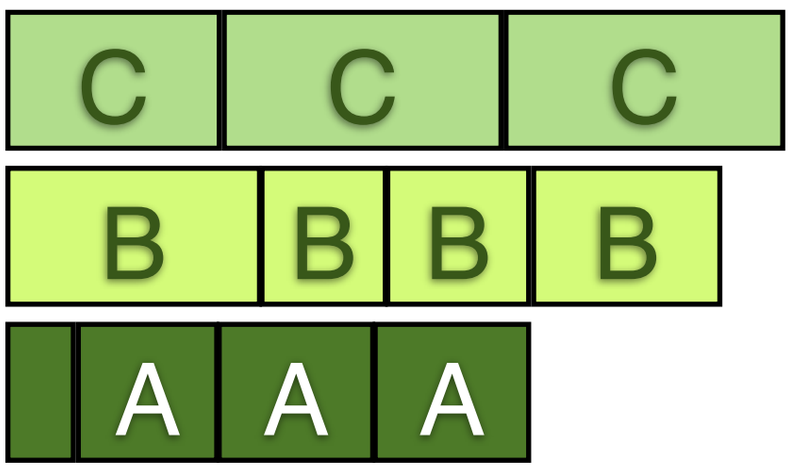


Turnaround Time

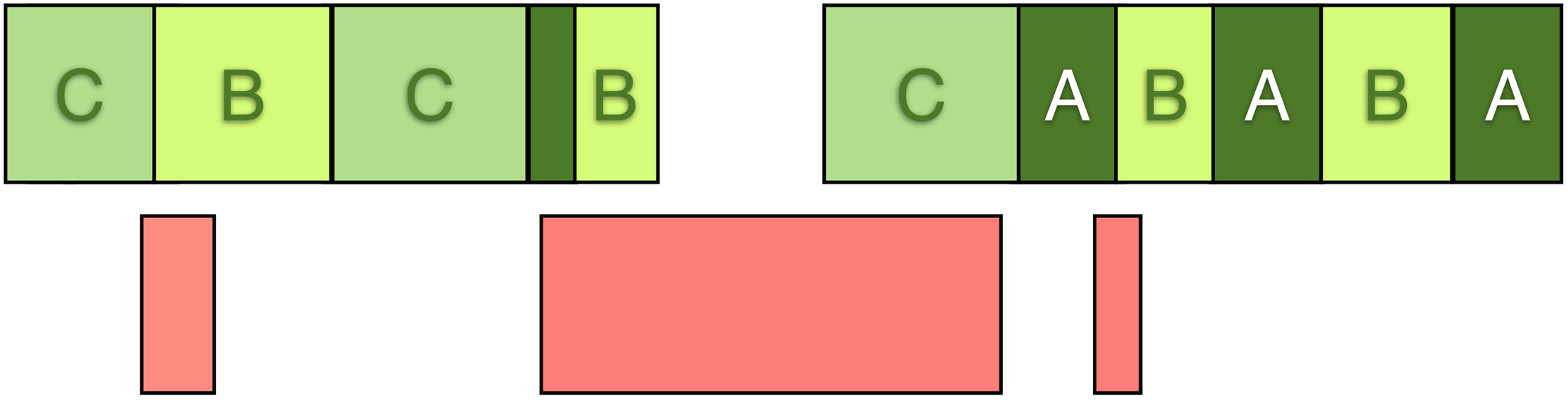


Fairness





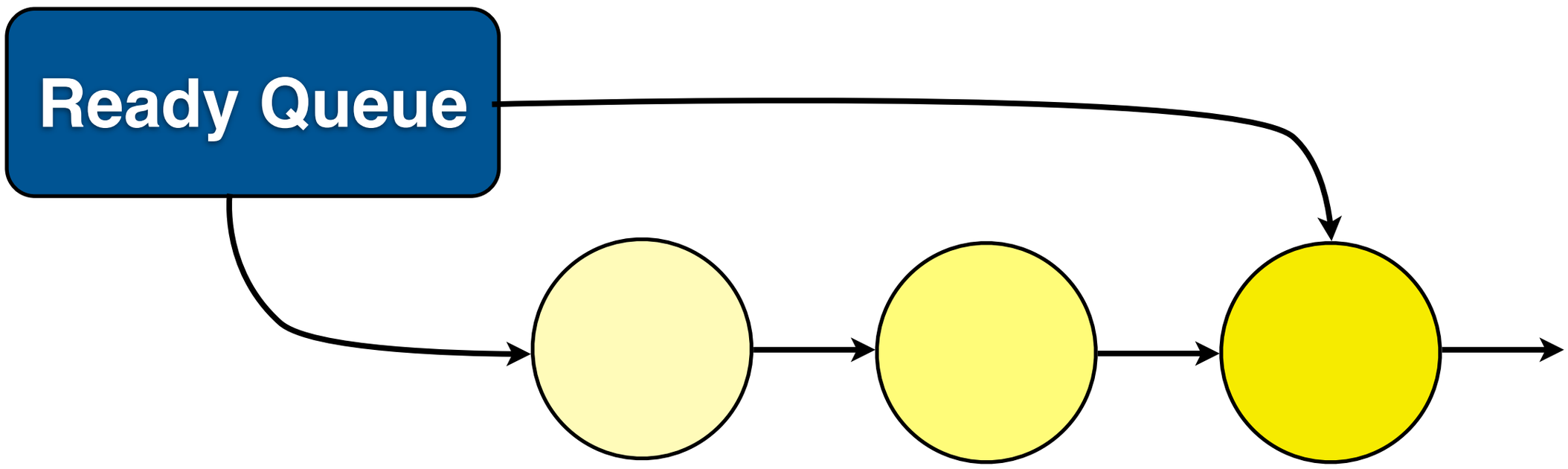
Balance

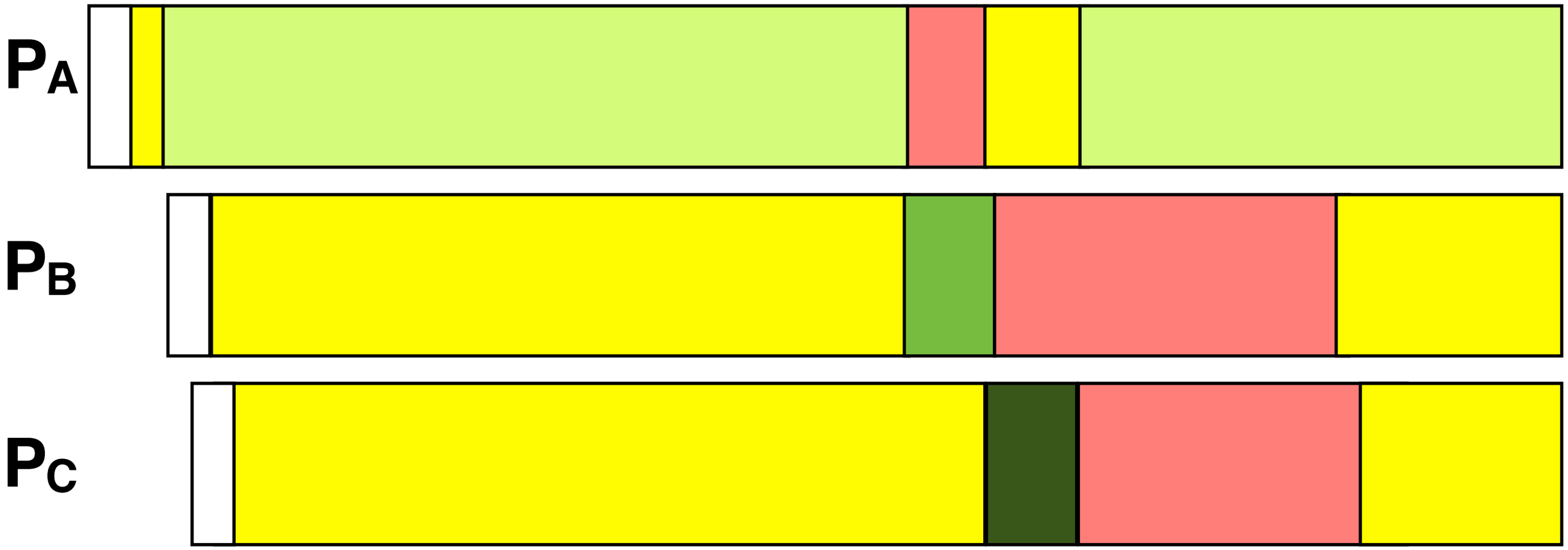


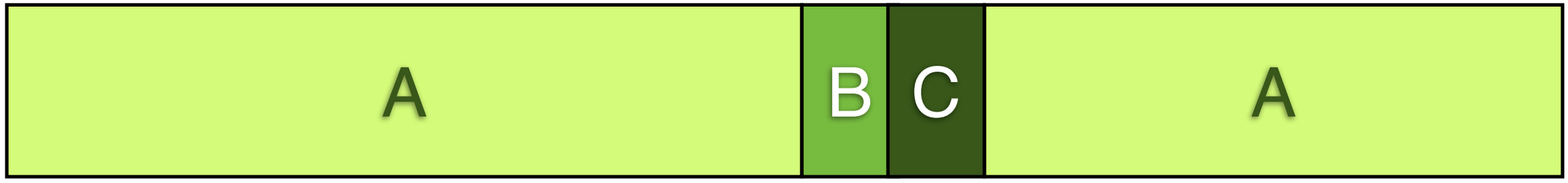
Scheduling Algorithms

FCFS

(non-preemptive)

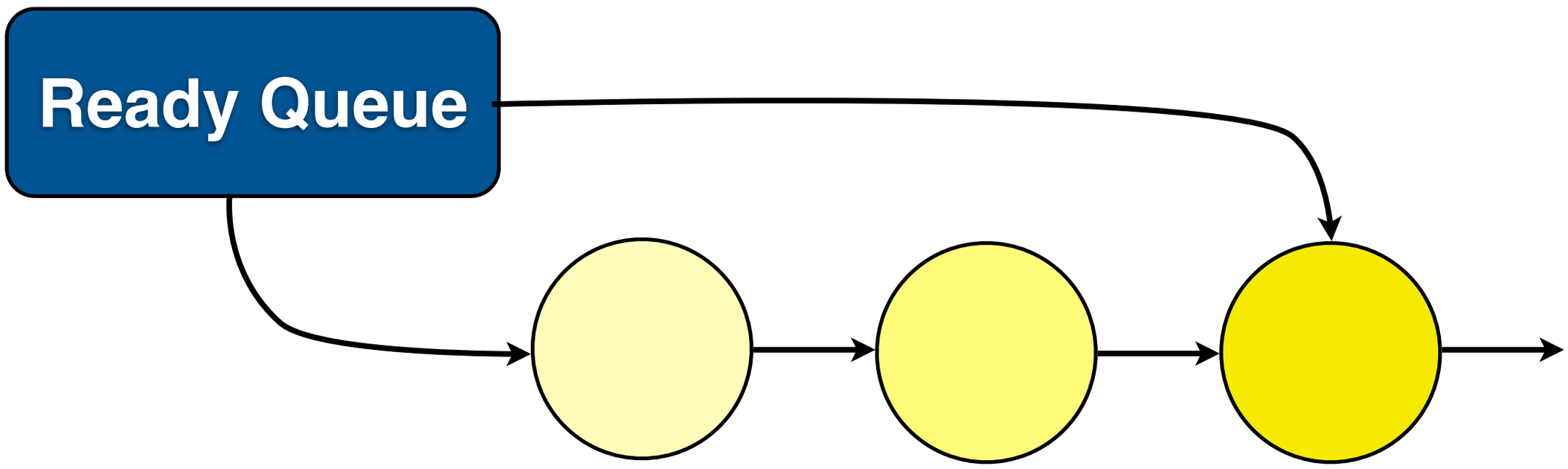


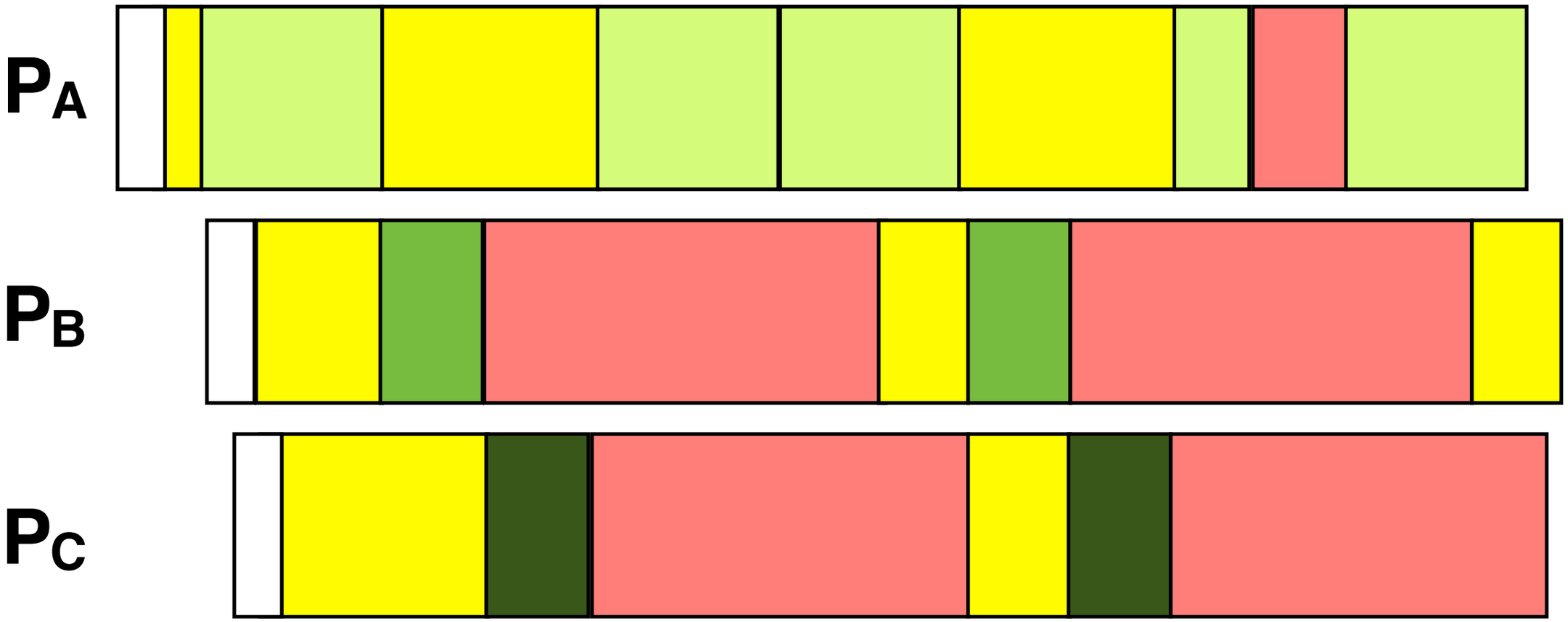

P_A
P_B
P_C

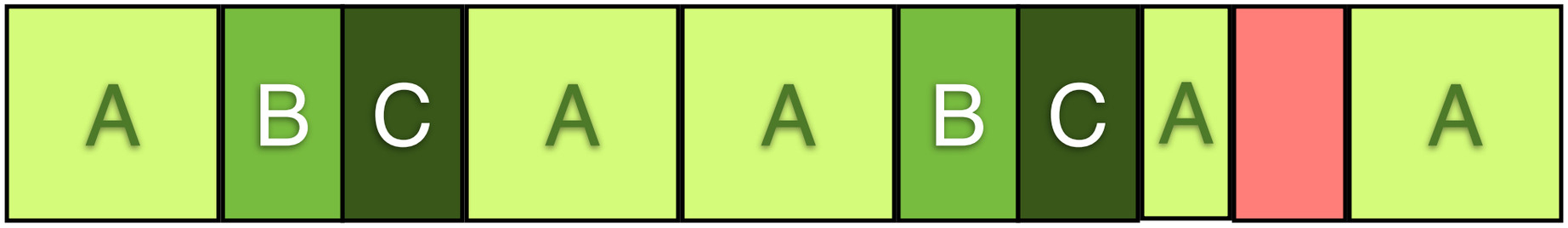

A
B
C
A

Round Robin

(preemptive)




P_A
P_B
P_C


A
B
C
A
A
B
C
A
A

Long Time Quantum

vs.

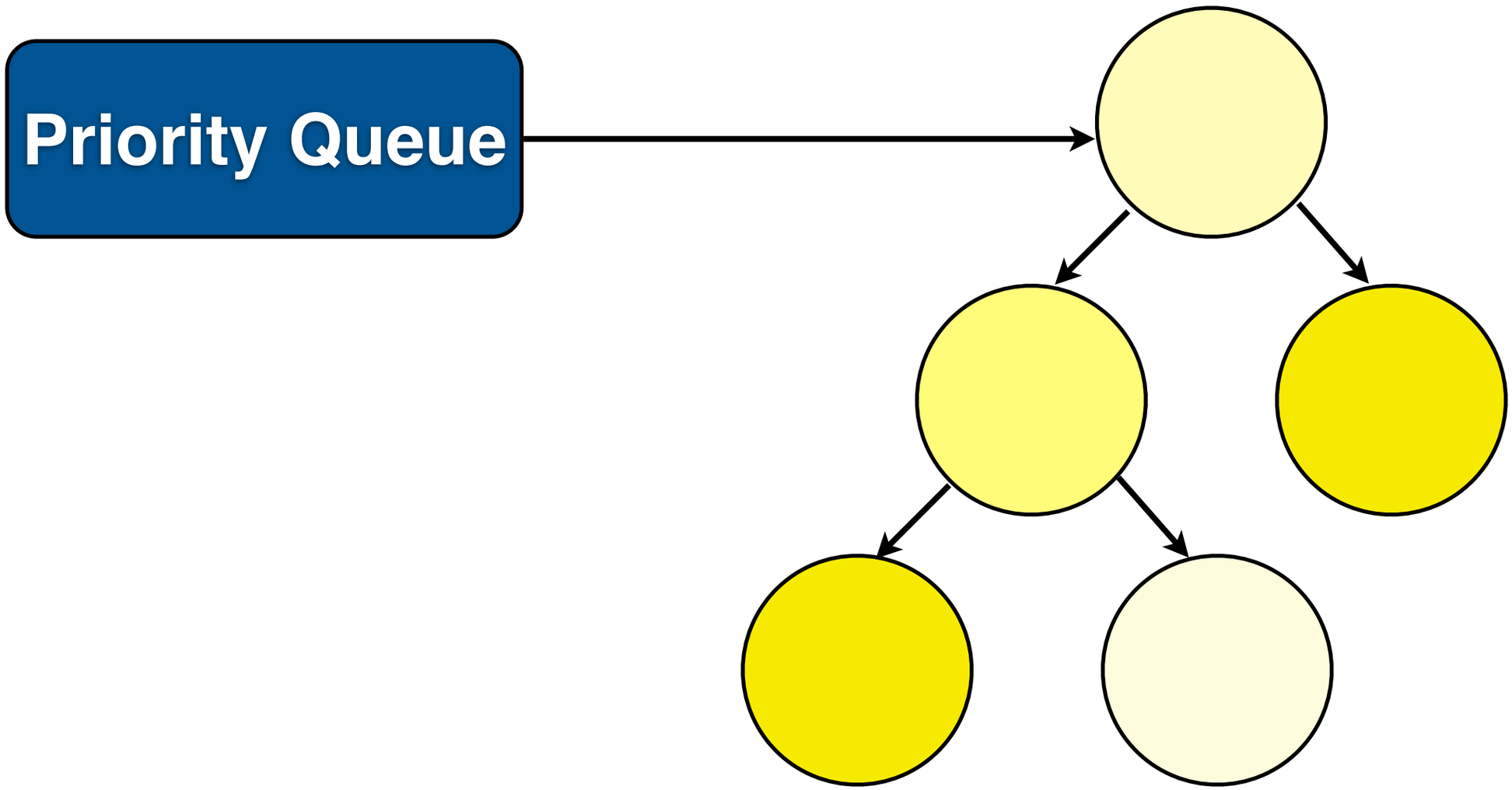
Short Time Quantum

Time Quantum

20 - 50ms

Shortest Job First

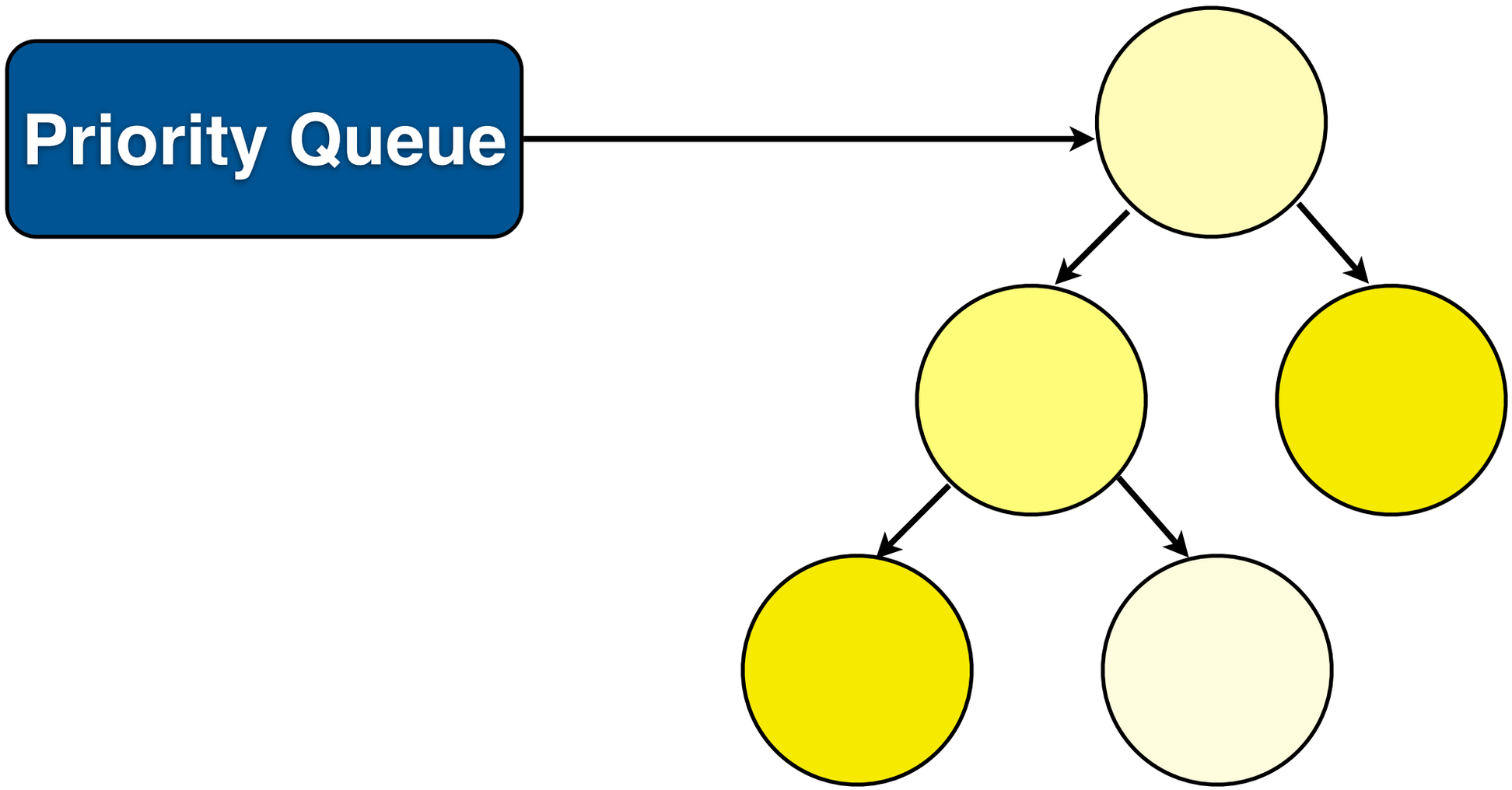
(non-preemptive)



FCFS

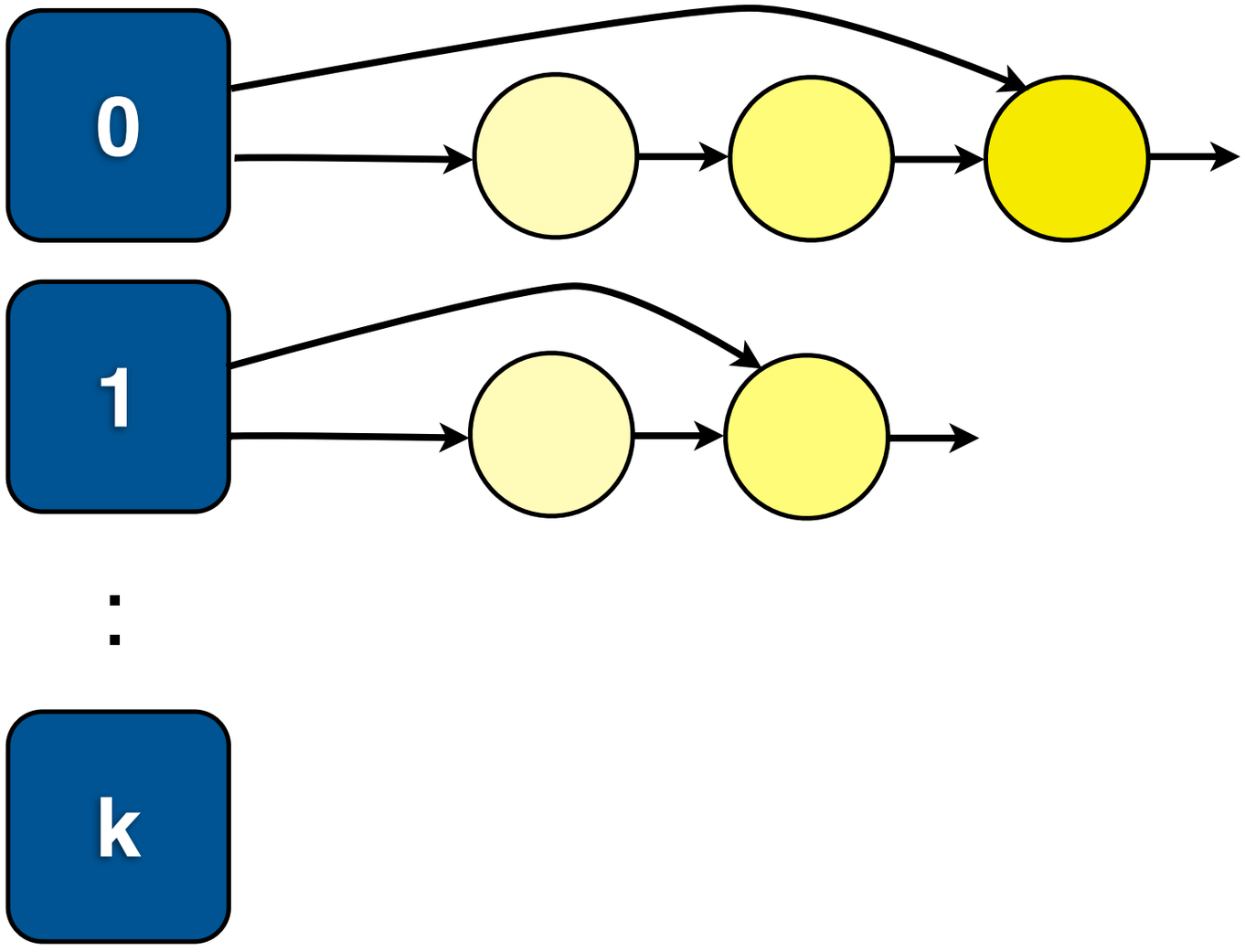
SJF

Shortest Remaining Time First (preemptive)



Estimating Remaining Time

Priority Scheduling (preemptive)



Static

vs.

Dynamic Priority

Guidelines for Good Schedulers

1.
interactive jobs should have higher priority than CPU bound jobs

2.

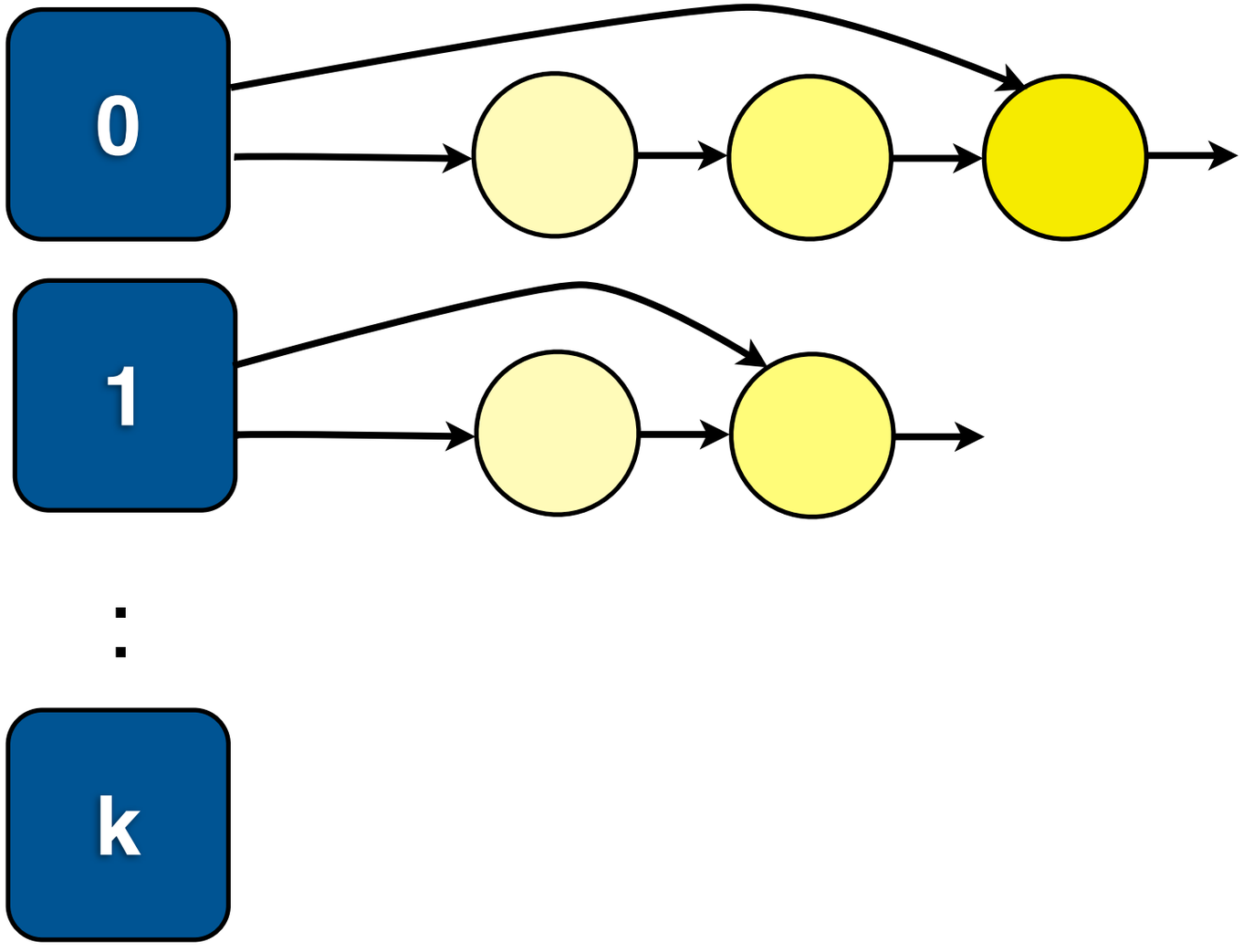
CPU intensive jobs should be given larger time quantum

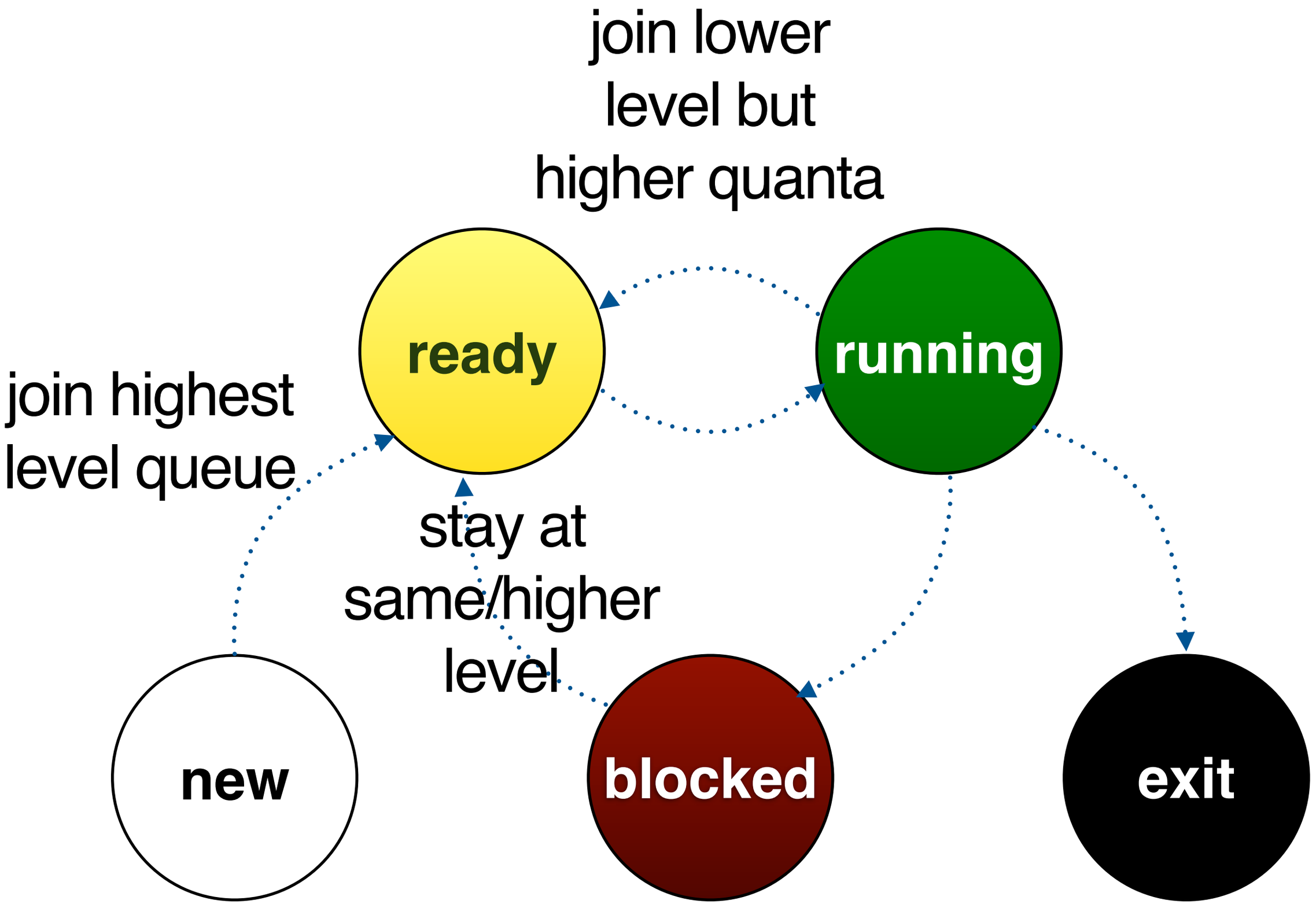
3.
round robin is good for response time but bad for turnaround time

4.
shortest job first reduces the turnaround time

(how to know if a job is interactive? remaining time on the job?)

Multilevel Feedback Queue




join lower
level but
higher quanta
ready
running
join highest
level queue
stay at
same/higher
level
new
blocked
exit

interactive jobs remains on top

CPU intensive jobs sink to bottom

1.
interactive jobs should have higher priority than CPU bound jobs

2.

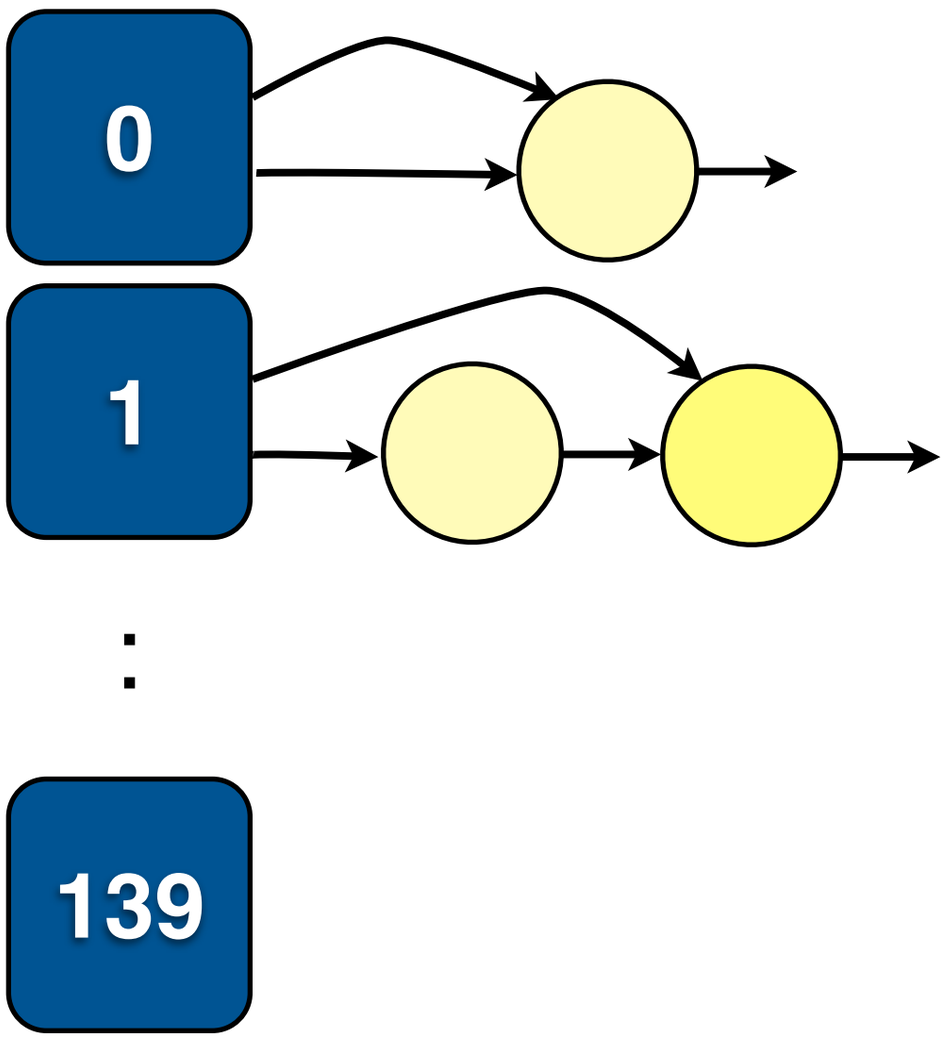
CPU intensive jobs should have be given larger time quantum

3.
round robin is good for response time but bad for turnaround time

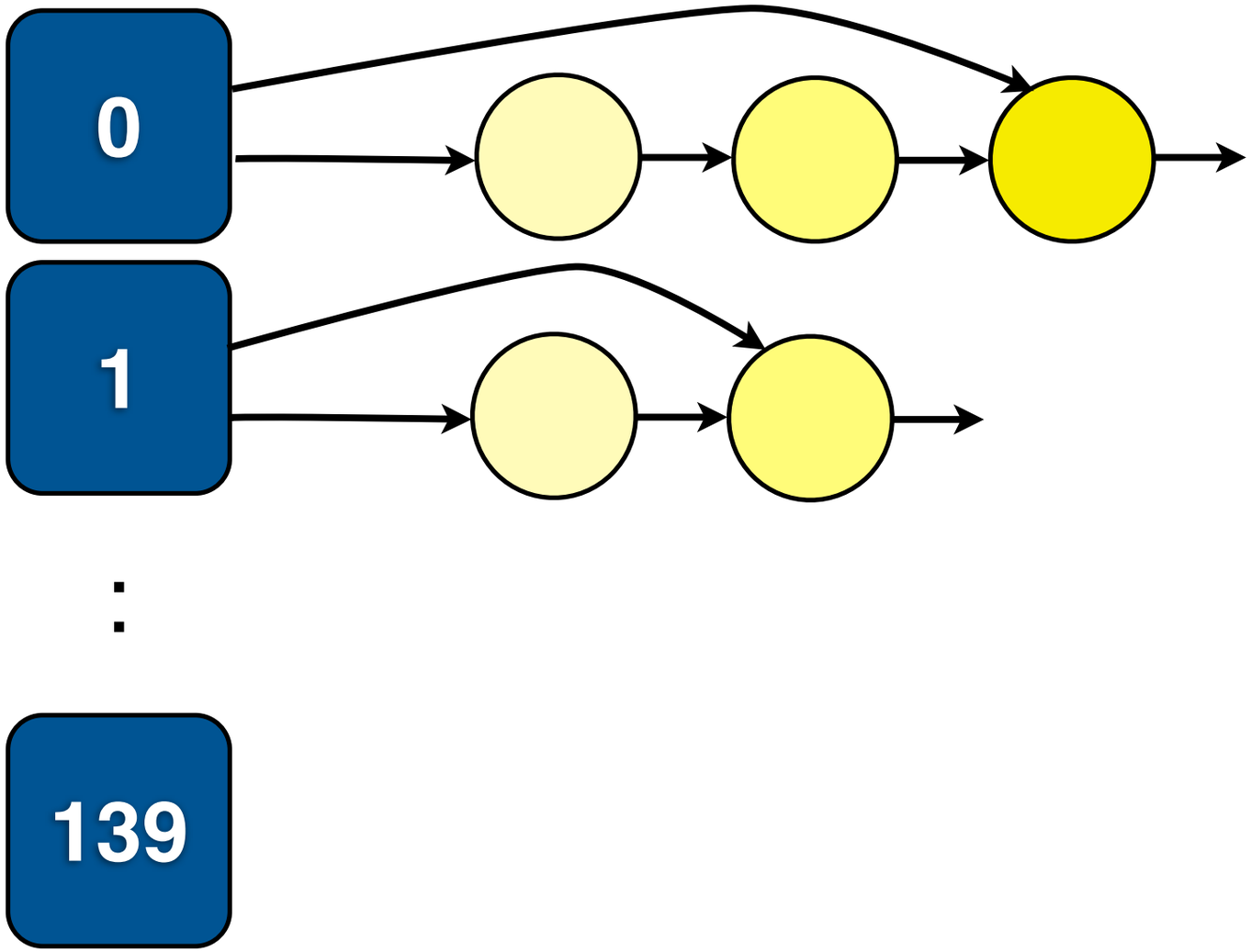
4.
shortest job first reduces the turnaround time

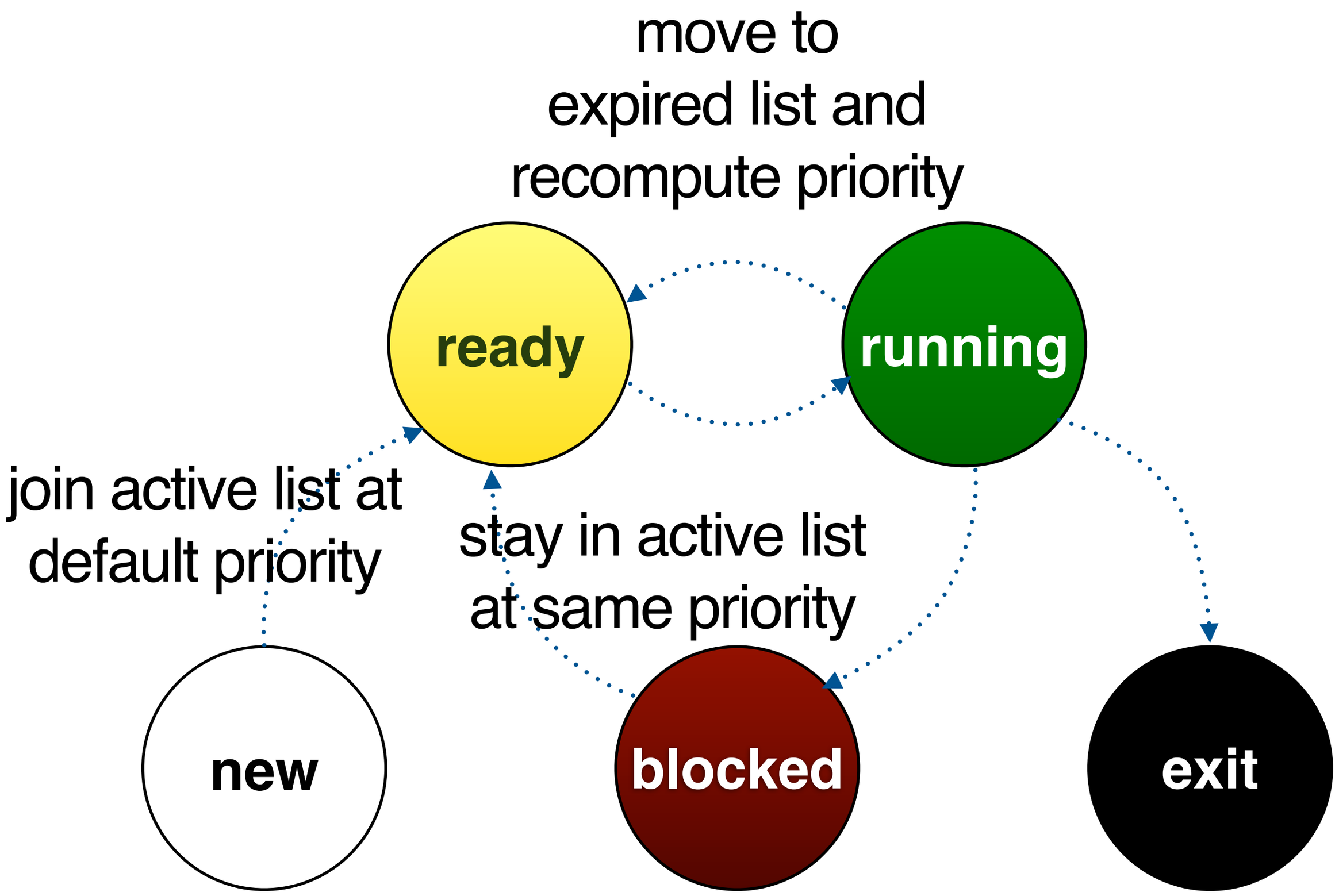
Linux Scheduler

Active

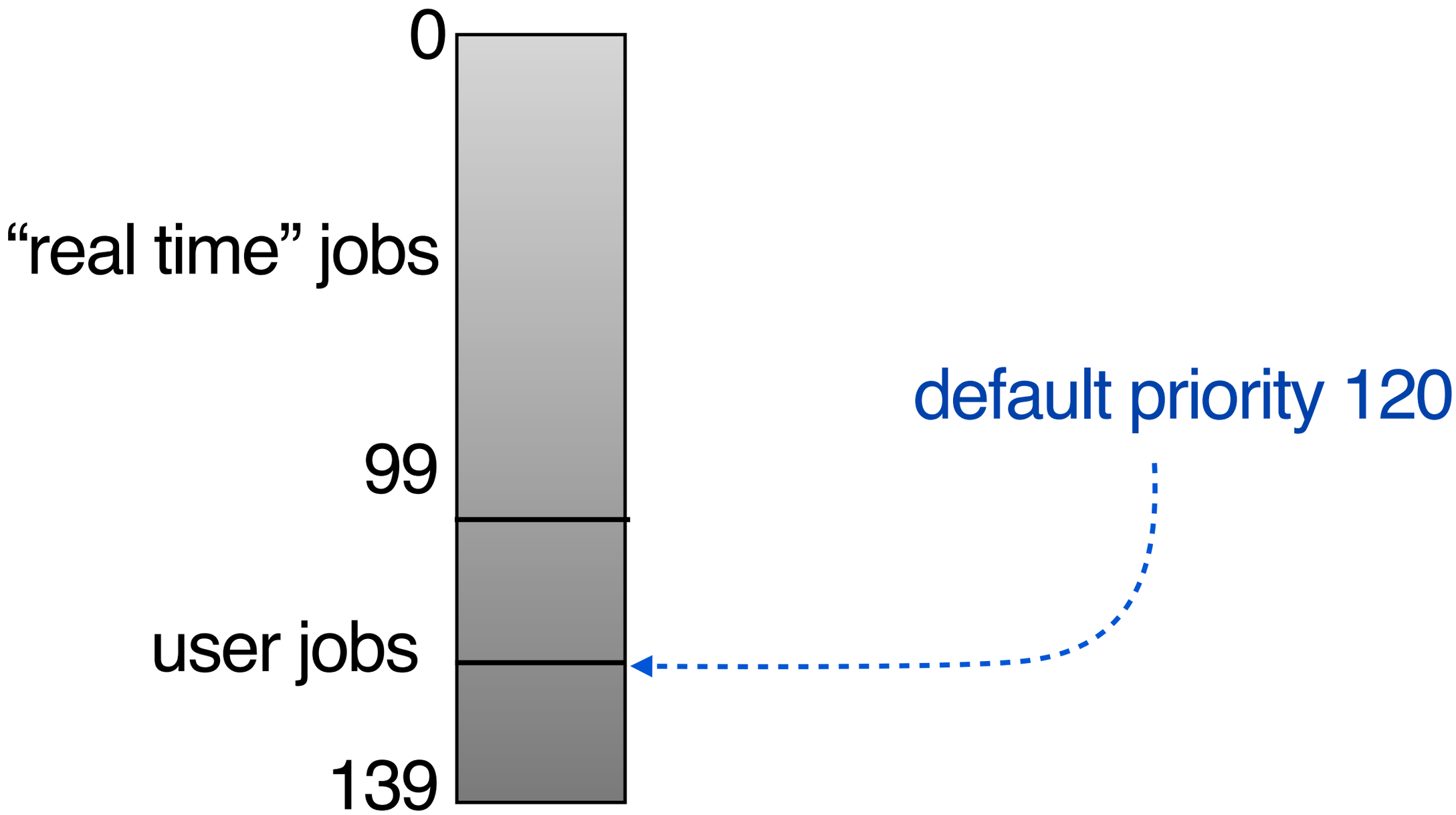


Expired

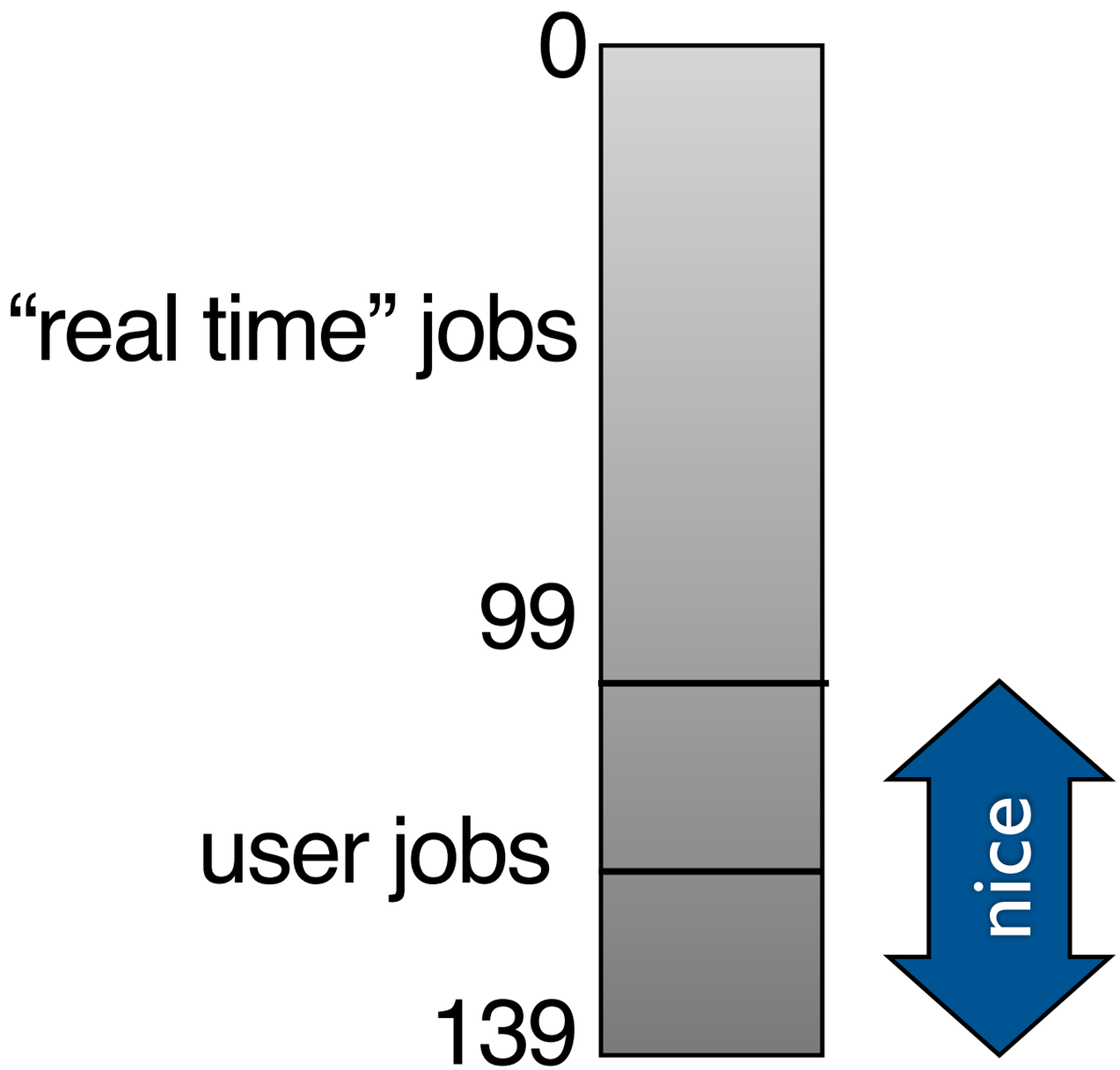



move to
expired list and
recompute priority
ready
running
join active list at
default priority
stay in active list
at same priority
new
blocked
exit

Linux Static Priority

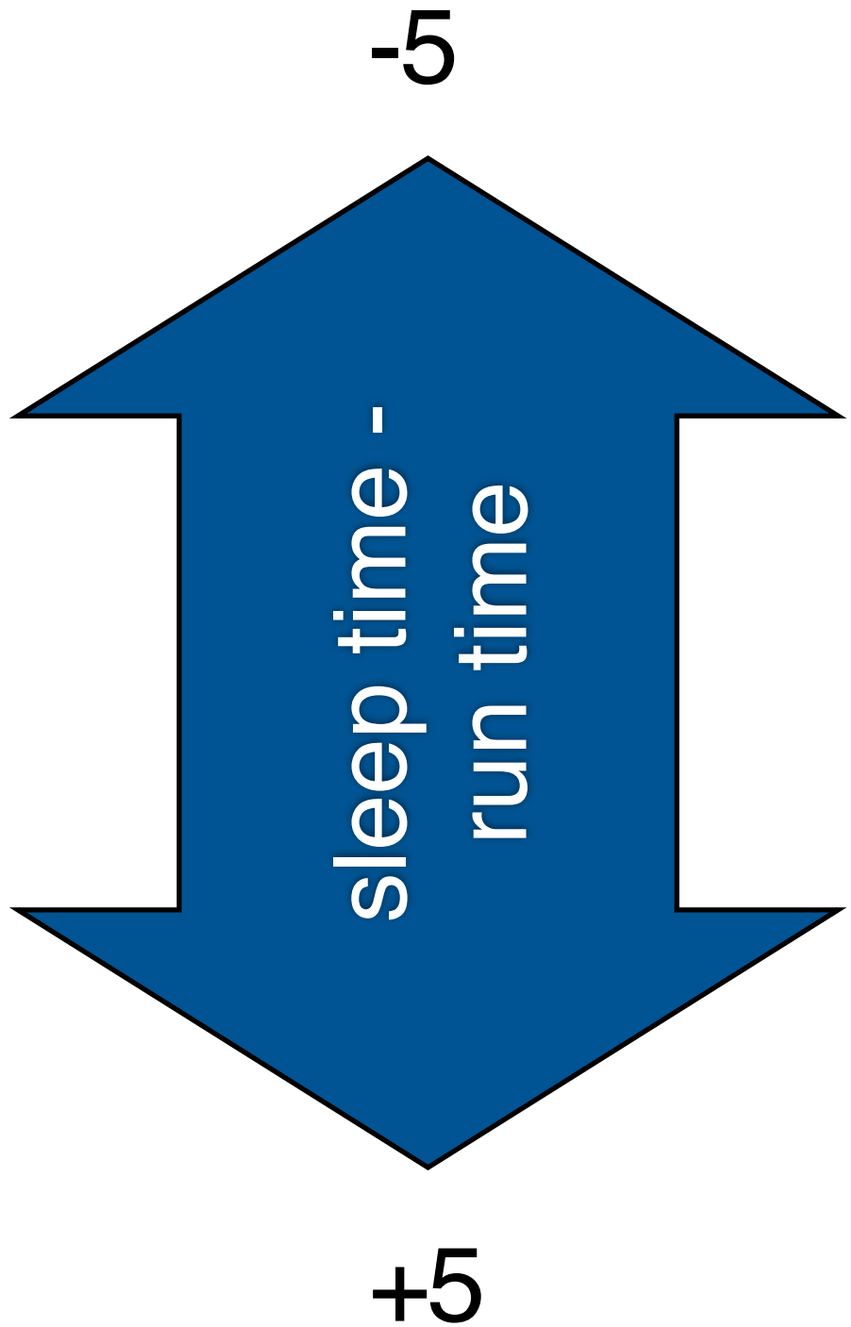


Linux Static Priority



	time quantum
-20	800ms
+19	5ms

Linux Dynamic Priority



Swap Active <-> Expired