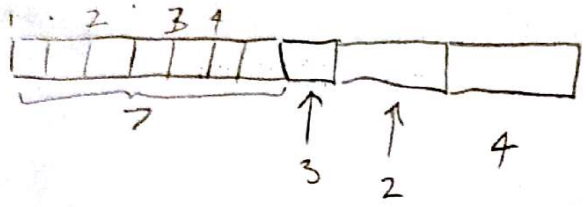
Shortest Job First (no preemption)

Process	Arrive	Processor time
1	0	7
2	2	4
3	4	1
4	5	4



Average wait = (0 + 3 + 6 + 7)/4 = 4

Advantage

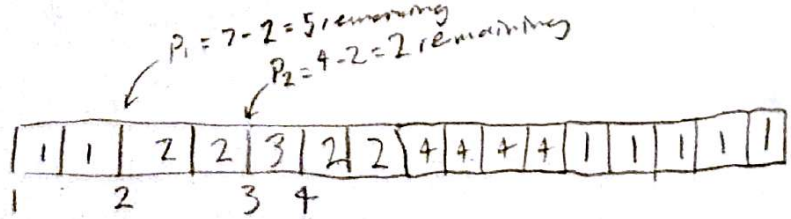
Minimizes average wait time

Disadvantage

- cannot predict processor time
- may starve long jobs (keep pushing them to back of line)

Shortest Remaining Time First SRTF

if new process arrives with shorter Processor time than remaining for current process schedule new process reduces average wait time



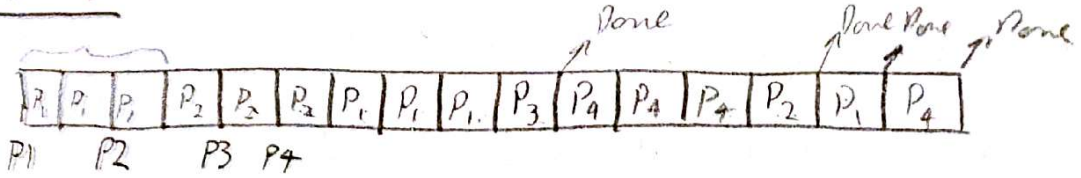
(9 + 1 + 0 + 2)/4 = 3

Round Robin (RR)

- practical approach to support time-sharing
- Run process for a time slice, then move back to FIFO queue
- preempted at end of time slice
- how to determine time slice?

P1	0	7
P2	2	4
P3	4	1
P4	5	4

Time slice = 3



Queue

P1	P1	P2	P2	P2	P1		P3	P4	P4	P4	P2	P1	P4
	P2	P1	P1	P1	P2		P4	P2	P2	P2	P1	P4	
			P3	P3	P4		P2	P1	P1	P1	P4		
				P4	P2		P1						

Average wait time

$$(8 + 8 + 5 + 7)/4 = 7$$

Average Response time

$$(0 + 1 + 5 + 5)/4 = 2.75$$

context switches = 7

R time slice = 1

	P1	P1	P2	P1	P2	P3	P1	P4	P2	P1	P4	P2	P1	P4	P1	P4
		P1	P2	P3	P1	P4	P2	P1	P4	P2	P1	P4	P1	P4		
				P1	P4	P2	P1	P4	P2	P1	P4					
					P2											

P#	A	P time
1	0	7
2	2	4
3	4	1
4	5	4

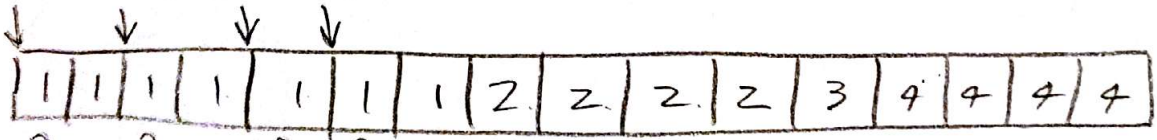
Av wait time (8 + 6 + 1 + 7)/4 = 5.5

Av response time (0 + 0 + 1 + 2)/4 = .75

(on up fast!)

context switches = 14 !! :(

RR time slice = 10



P1	P1	P1	P1
	P2	P2	P2
		P3	P3
			P4

Av wait (0 + 5 + 7 + 7)/4 = 4.75

Av res time = (0 + 5 + 7 + 7)/4 = 4.75

Context switches = 3

(looks like FCFS & will be same if ts > largest process time)

RR smaller time slice

0577

RR larger time slice

RR Advantages

- low response time
- fair CPU allocation
- low average waiting time (cause you are swapping)

Disadvantages

Performance depends on time slice

- too large FCFS
- too low too many context switches

Lots of