Worksheet #1

Calculate the average wait time for a list of process with service times 16, 12, 3, 5, 10, 1 all arriving at 0.

$P_0 \Rightarrow 16$

$P_1 \Rightarrow 12$

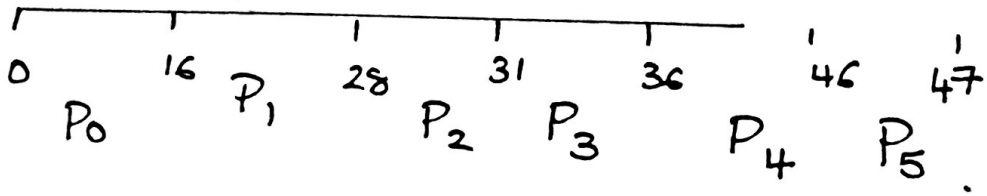
$P_2 \Rightarrow 3$

$P_3 \Rightarrow 5$

$P_4 \Rightarrow 10$

$P_5 \Rightarrow 1$

FCFS

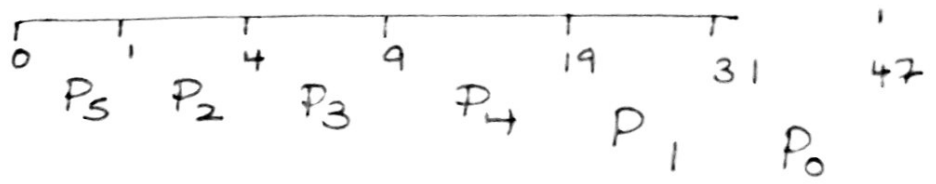


Average Wait Time $= \frac{0+16+28+31+36+46}{6} = \frac{157}{6} = 26.17$

SJF

Re-order Process ⇒ Note the process names also get re-ordered.

P_5	1
P_2	3
P_3	5
P_4	10
P_1	12
P_0	16

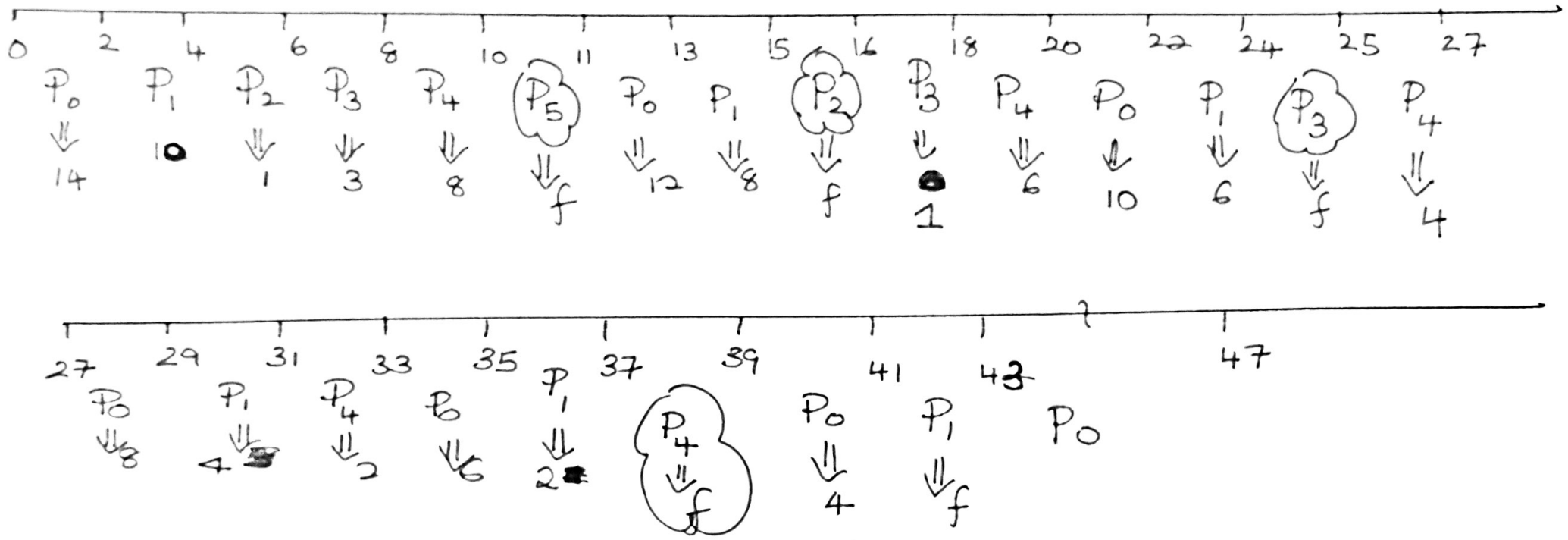


Average Wait Time

$$= \frac{0 + 1 + 4 + 9 + 19 + 31}{6}$$

$$= \frac{64}{6} = 10.66$$

Q = 2 Round Robin



$P_0 = 0 + (11 - 2) + (20 - 13) + (27 - 22) + (33 - 29) + (39 - 35) + (43 - 41) = 31$

$P_1 = 2 + (13 - 4) + (22 - 15) + (29 - 24) + (35 - 31) + (41 - 37) = 31$

$P_2 = 4 + (1 - 6) = 13$

$P_3 = 6 + (16-8) + (24-18) = 20$

$P_4 = 8 + (18-10) + (25-20) + (31-27) + (37-33) = 29$

$P_5 = 10$

Average Wait Time $= (31 + 31 + 13 + 20 + 29 + 10)/6$

$= 134/6 = \boxed{22.33}$

Alternatively

$P_1 = 41 - (5 \times 2) = 31$

$P_0 = 43 - (6 \times 2) = 31$

$P_2 = 15 - (1 \times 2) = 13$

$P_3 = 24 - (2 \times 2) = 20$

$P_4 = 37 - (4 \times 2) = 29$

$P_5 = 10$

- FCFS performs the worst in this situation because a long process can delay all processes (even very short processes).

- SJF gives optimal average wait times but it is difficult to predict service times and can cause starvation

- Round robin performs between SJF & FCFS. It removes the starvation issue of SJF. If the quantum is too long it will perform closer to FCFS. Reducing the quantum will generate excess context switches.