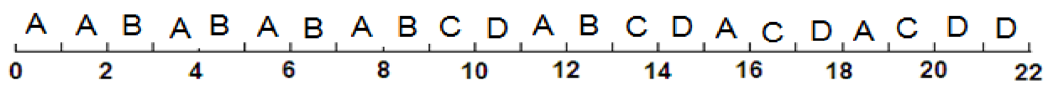
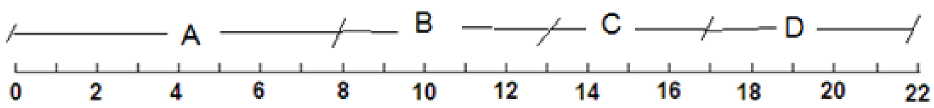
<u>Answers of Problem Set (1):</u>

**[P1]a)** <u>RR, q=1</u>



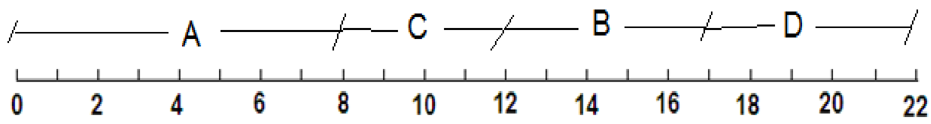

$W_{AV}$= (11+6+8+7)/4 = 8

**[P1]b)** <u>FCFS</u>



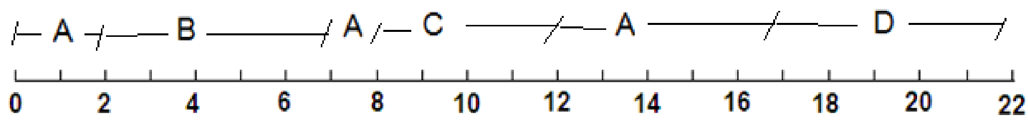

$W_{AV}$= (0+6+5+7)/4 = 4.5

<u>SPN</u>




$W_{AV}$= (0+10+0+7)/4 = 4.25

<u>SRT</u>




$W_{AV}$= (9+0+0+7)/4 = 4

**[P1] c)**

For RR with q=1, $W_{AV}$ becomes 8+18.5x

For FCFS, $W_{AV}$ becomes 4.5+2.5x

For SPN, $W_{AV}$ becomes 4.25+2.5x

For SRT, A works from x to 2 (for 2-x), B works from 2+x to 7+x and terminates, then A works from 7+2x to 8 (for 1-2x), then C runs from 8+x to 12+x and terminates. At 12+x, remaining time of A is 5+3x, while remaining time of D is 5, thus D runs from 12+2x to 17+2x and terminates. Finally, A runs from 17+3x to 22+6x. From the above, $W_{AV}$ becomes 4+2.5x

**[P2] a)** As q is very large, every process be blocked before the end of quantum. Thus the system will continuously run a process for time r, followed by a switching of time s. The CPU efficiency is thus given by:

$$\eta = \frac{r}{r+s}$$

**b)** As long as q > s, the process will stop before end of quantum, and the above result will not change (assuming immediate switching if a process stops).

**c)** In this case, process will run in $\left\lceil \frac{r}{q} \right\rceil$ quanta before blocking. Thus, the CPU efficiency becomes:

$$\eta = \frac{r}{r + \left\lceil \frac{r}{q} \right\rceil s}$$

**d)** As q=s, CPU efficiency will be 50% or less (since process may end within a quantum and a switching is corresponding to less than q of processing).

**e)** As q tends to 0, CPU efficiency will also tend to 0, as the switching time is fixed.

---

**[P3] a)** Each process will have one quantum for $[(p_a/q) - 1]$ times, then process A will have one quantum and then terminate. Thus termination time of A is:

$$t_a = 3q \left[ \left( \frac{p_a}{q} \right) - 1 \right] + q$$

and its waiting time is:

$$w_a = 2\,(p_a - q)$$

**b)** From the expression if $p_a$ = q waiting time is zero, then as q decreases the waiting time increases approaching $2p_a$. This decrease in q actually increases waiting time much further as the switching time becomes comparable to q.

**c)** Waiting time of process C is

$$w_c = (p_a + p_b)$$

independent of q.

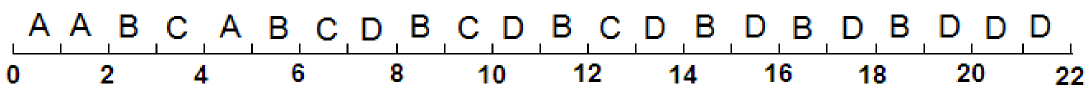
This is based on the assumption of negligible switching time

---

**[P4]** Priorities become 80, 69, and 65. Note that CPU usage may be measured by the number of quanta the process received in a given time period. A CPU-bound process is a process that spends most of its time using the CPU as opposed to an I/O-bound process that uses most of its time using or waiting for I/O (this is typical for processes interacting with the user). From the above, scheduler lowers the priority of CPU bound processes.

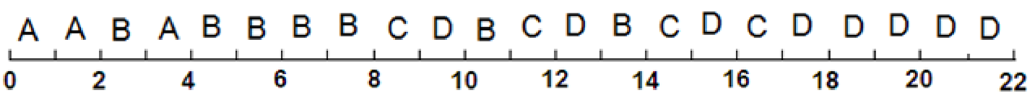
---

**[P5] a)**




$W_{AV}$= (2+10+6+9)/4 = 6.75

**[P5] b)**




$W_{AV}$= (1+5+10+9)/4 = 6.25

**[P5] c)**

Process B must have priority less than *both* C and D.

---

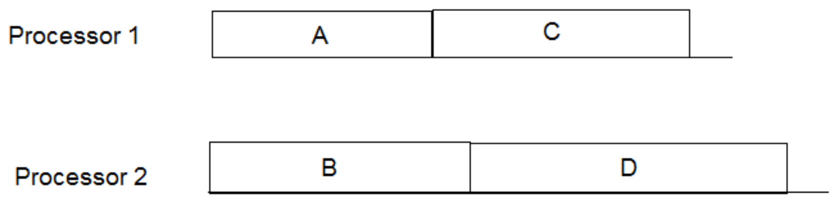
**[P6]**

**a)** The minimum average waiting time will be achieved by SPN (processes run in order A-B-C-D without pre-emption), and will be equal to

$$w_{av} = \frac{0 + p_A + (p_A + p_B) + (p_A + p_B + p_C)}{4}$$

$$= \frac{3p_A + 2p_B + p_C}{4}$$

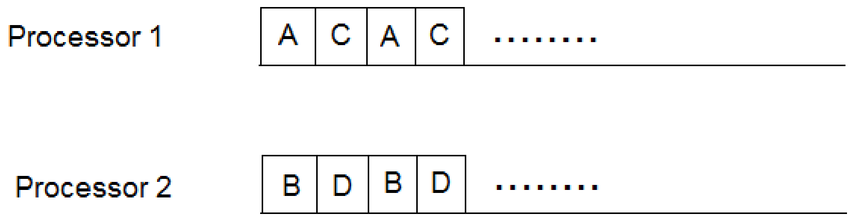
**b)**



$$w_{av} = \frac{(p_A + p_B)}{4}$$

No other order of execution can result in a lower value.

**c)**




Assuming for simplicity that all process times are multiple of q, then:

Termination time of A will be $2p_A - q$, and its waiting time will be $p_A - q$. Waiting time of C will be $p_A$. Similarly, waiting times of B and D will be $p_B - q$ and $p_B$.

Thus

$$w_{av} = \frac{(2p_A + 2p_B) - 2q}{4}$$

As a further exercise, find the average waiting time if process times are not multiple of q, both if immediate switching occurs when a process ends within a quantum or not.

---