CPU Efficiency

Measurements of a certain computer system have shown that the average process time runs for a time T before blocking on I/O. A process switch requires a time S, which is effectively wasted (overhead). For round-robin scheduling with quantum Q, give a formula for the CPU efficiency (defined as the percentage of CPU time used for useful work) for each of the following:

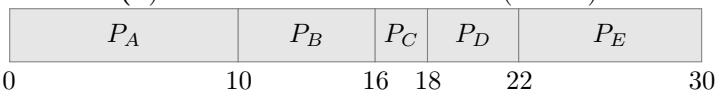
When $T > Q$ Then the formula is $\frac{Q}{Q+S}$
When $T < Q$ Then the formula is $\frac{T}{T+S}$
When $Q = S$ Then the formula is $\frac{Q}{Q+Q}$ or $\frac{Q}{Q+S} \rightarrow \frac{1}{2}$
When $Q \approx 0 \rightarrow Q \rightarrow \lim_{Q \rightarrow 0} \rightarrow 0$
When $Q \approx \infty \rightarrow T$ is used

CPU Scheduling

Five tasks A through E, arrive at a computer system at almost the same time. They have estimated running times of 10, 6, 2, 4 and 8. For each of the following scheduling algorithms, determine the **AVERAGE WAITING TIME.** Ignore process-switching overhead, you need to draw the gantt chart to show the schedule/running behavior of the five tasks.

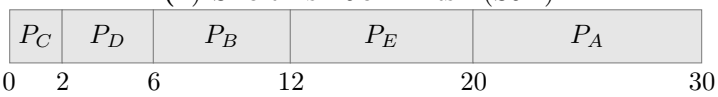
- First-come, first-served (run in order 10, 6, 2, 4, 8).
- Shortest job first.
- Longest job first: the runnable process with the longest estimated running time (CPU burst) will be scheduled to run.
- Priority scheduling: each process is assigned a priority, and the runnable process with the highest priority is allowed to run. In this question, the five tasks' priorities are 3, 5, 2, 1 and 4, respectively, with 5 being the highest priority.

(A) First-come first-served (FCFS)



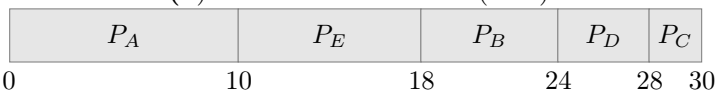
Average waiting time: $(0 + \{10 - 0\} + \{16 - 0\} + \{18 - 0\} + \{22 - 0\}) \div 5 = 13.2$

(B) Shortest Job First (SJF)



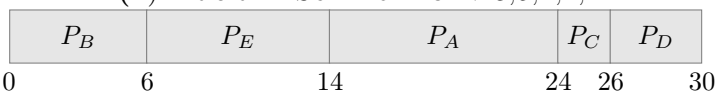
Average waiting time: $(\{20 - 0\} + \{6 - 0\} + 0 + \{2 - 0\} + \{12 - 0\}) \div 5 = 8$

(C) Longest Job First (SJF)



Average waiting time: $(0 + \{18 - 0\} + \{28 - 0\} + \{24 - 0\} + \{10 - 0\}) \div 5 = 16$

(D) Priority Scheduling → 3,5,2,1,4



Average waiting time: $(\{14 - 0\} + 0 + \{24 - 0\} + \{26 - 0\} + \{6 - 0\}) \div 5 = 14$

Synchronization

```
while true do
    wait(wrt);
    // writing is performed
    signal(wrt);
end
```

Algorithm 1: Reader

```
while true do
    wait(mutex);
    readCount++;
    if readCount == 1 then
        wait(wrt);
    end
    signal(wrt);
    // reading is performed
    wait(mutex);
    readCount−−;
    if readCount == 0 then
        signal(wrt);
    end
    signal(mutex);
end
```

Algorithm 2: Writer