

1 CPU Scheduling

Suppose that there are currently three processes in the Ready queue, waiting to be executed. Each process is accompanied by some information useful for the Scheduling Algorithm. More specifically, the matrix below shows the *Arrival Time*, the first *CPU-burst*, the first *I/O-burst* and the second *CPU-burst* for each process (the time is in milliseconds). Assume that the context switch time is negligible.

Process	Arrival Time	CPU	I/O	CPU
P_1	0	15	5	10
P_2	5	35	10	20
P_3	10	70		

- (a) Assume we deploy a simple scheduling algorithm, according to which the entire execution (both CPU and I/O bursts) of a process should be completed before the execution of the next process starts. The order of processes' execution is according to the order of processes' arrival. How much is the average waiting and turnaround time? By what time the executions of all processes have been completed?
- (b) If we deploy the SJF Scheduling Algorithm (non-preemptive), how much is the average waiting and turnaround time? By what time the executions of all processes have been completed?
- (c) Repeat question (b) for the Round Robin Algorithm with $Quantum = 15$ and $Quantum = 40$.
- (d) Compare the average waiting time, the average turnaround time and the total execution time of all processes for each algorithm (questions a,b,c), and explain the differences (i.e. why one is greater than the other?).
- (e) How long should the $Quantum$ be in order the Round Robin Algorithm to become FCFS Algorithm, for this scenario?
- (f) If the process P_3 has its first response 5 milliseconds after the execution of the first command, which one from the algorithms above decreases its response time? Which $Quantum$ would minimize its response time (suppose that $Quantum \geq 1$)? What conclusion can we draw from this?

2 Programming Part

Programs p1.py, p2.py, p3.py, p4.py and p5.py need the insertion of synchronization primitives (semaphores, monitors) in order to work properly. Please, look at the programs, follow the instructions and try to fill the code-gaps.