Name: Keshav Bansal
Roll Number : 2019101019

Performance Analysis:

1. The Round-robin is a good algorithm as equal weightage is given to both CPU bound and IO bound processes. Also it prevents starvation for any process.

2. MLFQ is the best algorithm as there are less chances of starvation because the processes which receive more CPU time are sent downwards and the I/o processes remain in the same queue.
Starvation is prevented by moving the processes in the lower priority queue to the higher priority queue.

3. Priority based scheduling is not very efficient because if the highest priority process takes a very long CPU time other processes might get starved.

4. FCFS is not efficient as the process with the least creation time takes the most amount of time thus leading to starvation for all the other processes.

EXPLOITATION OF VOLUNTARY RELINQUISH:

If a process is about to go to the lower queue, i.e. when it's time slice is about to get completed then it can relinquish the control and can again come back to the same queue, thus it can always stay in a queue with higher priority.

RESULTS:
After running the benchmark program, the below given values are the run times and wait times for running benchmark program :

	RUN TIME (ticks)	WAIT TIME(ticks)
1. FCFS	11	3516
2. PBS	16	2577
3. MLFQ	9	2674
4. ROUND ROBIN	12	2763

After running the benchmark program, the below given values are the run times and wait times for running an ‘ ls ‘ command :

	RUN TIME	WAIT TIME
1. FCFS	43	2
2. PBS	49	101
3. MLFQ	32	3
4. ROUND ROBIN	36	5

DIFFERENT PROCESSES FOR MLFQ:

Most IO bound processes mostly remain in the same queue whereas the CPU BOUND Processes keep on interchanging between different queues.

GRAPH(BONUS):

I have made three graphs for various sleeping times, aging time and run time slices per queue.

Observations for different processes:

1. FOR CPU BOUND PROCESSES:

   The PID 4,5 are CPU BOUND PROCESSES and they remain in the lower queues ,i.e. They remain in queue number 3 and queue number 4 and keep on interchanging between these queues.

2. The pid 7,8 are I/O bound processes and they remain in the higher priority queues for most of their time.