**OPERATING SYSTEMS PROJECT**

**Report**

In short way, I will try to explain what I do in the given project. All project contains three main classes, that means three algorithms. First Come First Serve (FCFS) is a scheduling algorithm that executes automatically queued requests and processes in order of their arrival. It is simplest and easiest CPU scheduling algorithm. It is an non-preemptive scheduling,that is used when a process terminates or a process switches from running to a waiting state. Let me explain the code I had written.

All my code is based in five functions or methods. Starting with waiting time, in this method we declare and then we calculate the waiting time. The second one is to find the turnaround time by adding the formula CPU burst time + waiting time. After doing this calculation we will get the turnaround time. The third one is finding average time, inside this function we have some other functions such as finding waiting time, finding turnaround time then we calculate the total waiting time and total turnaround time. And finally, we will get the average waiting time and the average turnaround time also.

The next function is to set the priority scheduling, which means to set processes by priority. And the last one is the main method inside it we declare number of processes we want, in this case is three(n=3), and any process has own process id, CPU burst time and priority of process.

The second algorithm is the Shortest Job First (SJF) scheduling non preemptive. In this scheduling we should know that the execution time of each process before running. Let we base in our code and supposed we have set of processes are in ready queue. The SJF scheduling algorithm will choose the job which has shortest remaining time to complete. As we choose the non-preemptive version, and we have three number of processes, it means that we have three processes in our ready queue, and the SJF will schedule which is having least execution time or burst time. After that we calculate the completion time, waiting time and turnaround time of each process.

And the third scheduling algorithm is the Round Rubin scheduling algorithm. RR is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cycle way. It is preemptive as processes are assigned CPU only for a fixed slice of time at most. The newly created process is added to end of ready queue. Each process gets a change to reschedule after a particular quantum time in this scheduling. Let me explain how I did this program. It has three main functions, the first one is the completion time which means that time at which process completes its execution. Turnaround time – time difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time. And the last one is Waiting Time – time difference between turnaround time and burst time. Waiting Time = Turn Around Time – Burst Time.

How I get the waiting times of all processes:

Create an array to keep track of remaining burst time of processes, this array is initially a copy of burst time array. Create another array to store waiting times of processes. Initialize this array as 0. Also initialize time as 0. Keep traversing all processes while all processes are not done. If burst time>quantum then time is t = quantum and burst time is = burst[i] -= quantum. And the last cycle for this process is, else time is t = t+burst[i] then waiting time is wait[i] = t – burst[i], finally this process is over burst[i] = 0.

After we have waiting time, we can compute turnaround time of a process as sum of waiting and burst time.

**SUAD NESIMI**

**100044162**