Student Name: kartikeya chaurasia

Student ID:11806332

Email Address:ckartikeya43@gmail.com

GitHub Link: https://github.com/kartkey/Operating-system

Code: Mention solution code assigned to you

- Priority Scheduling Algorithm:

Priority scheduling is a non-preemptive algorithm and one of the most common scheduling algorithms in batch systems. Each process is assigned a priority. Process with the highest priority is to be executed first and so on. Processes with the same priority are executed on first come first served basis. Priority can be decided based on memory requirements, time requirements or any other resource requirement.

Algorithm:

- Priority Scheduling Algorithm:

In this, each process is assigned a priority and processes are executed on the basis of their priority. We can either choose to set priority of the lowest number to be the first priority or vice versa. No other process can execute until the process with the highest priority has fully executed. If two processes have same priority, then process is executed on the basis of their arrival time.

- Priority Scheduling Algorithm

```
def waitingTime(processes, n, wt):
    wt[0] = 0
  for i in range(1, n):
        wt[i] = processes[i - 1][1] + wt[i - 1]
def turnAroundTime(processes, n, wt, tat):
      for i in range(n):
      tat[i] = processes[i][1] + wt[i]
```

```
def findavgTime(processes, n):
 wt = [0] * n     tat = [0] * n
 waitingTime(processes, n, wt)
 turnAroundTime(processes, n, wt, tat)
print("\nProcesses Burst Time Waiting", "Time Turn-Around Time")
total_wt = 0
 total_tat = 0
for i in range(n):
        total_wt = total_wt + wt[i]
         total_tat = total_tat + tat[i]
         print(" ", processes[i][0], "\t\t",
                processes[i][1], "\t\t",
                wt[i], "\t\t", tat[i])
           print("\nAverage waiting time = %.5f "%(total_wt /n))
           print("Average turn around time = ", total_tat / n)
def priorityScheduling(proc, n):
    proc = sorted(proc, key = lambda proc:proc[2],
                                reverse = True);
print("Order in which processes gets executed")
for i in proc:
        print(i[0], end = " ")
findavgTime(proc, n)
 if __name__ =="__main__":
        process = [[1, 10, 1],
           [2, 20, 0],
           [3, 15, 1],
```

```
        [4, 11, 2]]
    n = 4
    priorityScheduling(process, n)
```

- Constraints for Priority Scheduling Algorithm:

1. Process with highest priority gets preference first.

2. If two or more processes have same priority, then process if executed on their arrival time.

- Priority Scheduling Algorithm:

1. Process 1 comes -> starts getting executed

2. Next process comes with lower priority -> Waits till execution finishes

3. Another process comes with higher priority -> Pre-empts running process and starts executing

4. If two process have similar priority -> process that arrives first, gets executed

GitHub Link: https://github.com/kartkey/Operating-system