Ex. No.: 6c)

PRIORITY SCHEDULING

Aim:

To implement priority scheduling technique

Algorithm:

1. Get the number of processes from the user.
2. Read the process name, burst time and priority of process.
3. Sort based on burst time of all processes in ascending order based priority 4.
   Calculate the total waiting time and total turnaround time for each process 5.
   Display the process name & burst time for each process.
   6. Display the total waiting time, average waiting time, turnaround time

Program Code: #

```
Priority Scheduling

Algorithm (Non-

Preemptive) with User

Input



class Process:

   def __init__(self, pid,

arrival_time, burst_time,

priority):

       self.pid = pid

       self.arrival_time =

arrival_time
```

```
        self.burst_time =
burst_time
        self.priority =
priority
        self.complete_time
= 0

self.turnaround_time = 0
        self.waiting_time =
0

def
priority_scheduling(proc
esses):
    n = len(processes)
    completed = 0
    current_time = 0
    avg_turnaround_time
= 0
    avg_waiting_time = 0
    is_completed = [False]
* n
```

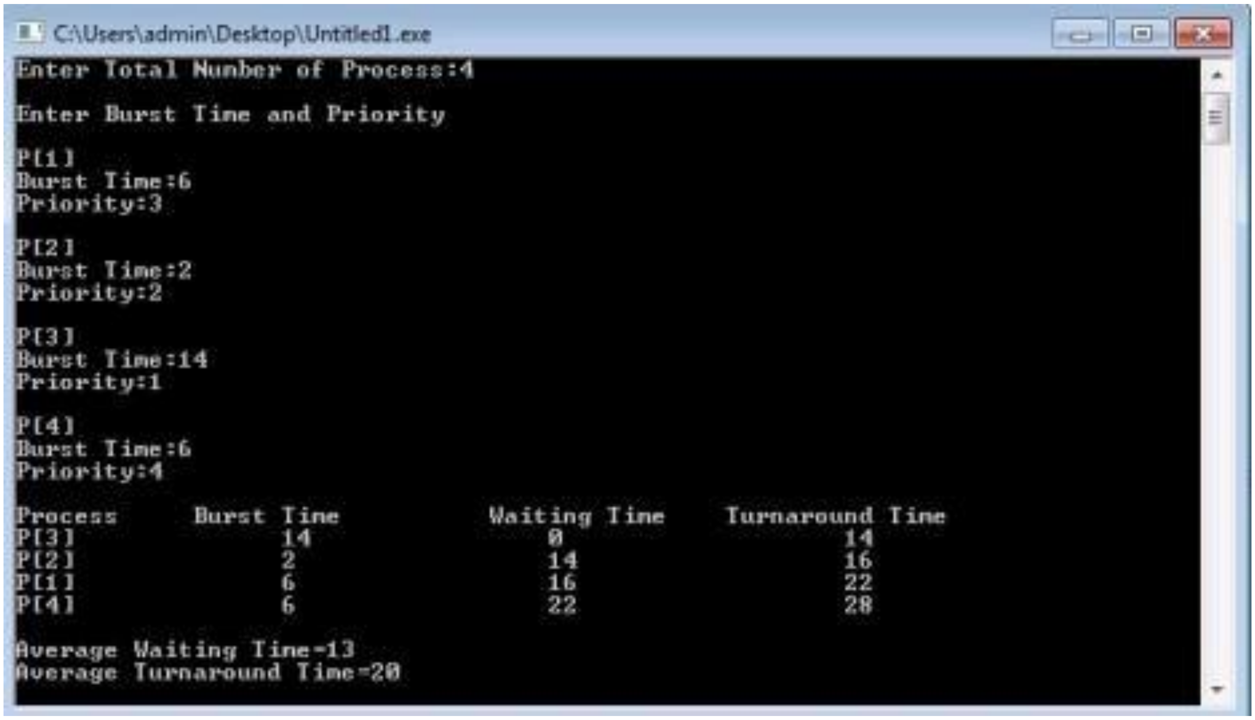
```
    while completed != n:
        idx = -1
        highest_priority =
float('inf')
        for i in range(n):
            p = processes[i]
            if (p.arrival_time
<= current_time) and
(not is_completed[i]):
                if p.priority <
highest_priority:

highest_priority =
p.priority
                    idx = i
                elif p.priority
== highest_priority:
                    if
p.arrival_time <
processes[idx].arrival_ti
me:
                        idx = i
        if idx != -1:
```

```
        p =
processes[idx]
        current_time +=
p.burst_time
        p.complete_time
= current_time

p.turnaround_time =
p.complete_time -
p.arrival_time
        p.waiting_time =
p.turnaround_time -
p.burst_time


avg_turnaround_time +=
p.turnaround_time
        avg_waiting_time
+= p.waiting_time

        is_completed[idx]
= True
        completed += 1
        else:

            current_time += 1



    avg_turnaround_time

/= n

    avg_waiting_time /= n





print(f"{'PID':<5}{'Arriv

al':<10}{'Burst':<8}{'Pri

ority':<10}{'Complete':<

10}{'Turnaround':<12}{'

Waiting':<8}")

    for p in processes:



print(f"{p.pid:<5}{p.arri

val_time:<10}{p.burst_ti

me:<8}{p.priority:<10}{

p.complete_time:<10}{p.

turnaround_time:<12}{p.

waiting_time:<8}")
```

```
    print(f"\nAverage
Turnaround Time:
{avg_turnaround_time:.2
f}")
    print(f"Average
Waiting Time:
{avg_waiting_time:.2f}"
)


if __name__ ==
"__main__":
    processes = []
    n = int(input("Enter
the number of processes:
"))
    for i in range(n):
        print(f"\nEnter
details for Process
{i+1}:")
        arrival =
int(input("Arrival Time:
"))
```

```
        burst =
    int(input("Burst Time:
    "))
        priority =
    int(input("Priority (lower
    number = higher
    priority): "))

    processes.append(Proces
    s(i+1, arrival, burst,
    priority))


    priority_scheduling(proc
    esses)
```

Sample Output:



```
Enter Total Number of Process:4

Enter Burst Time and Priority

P[1]
Burst Time:6
Priority:3

P[2]
Burst Time:2
Priority:2

P[3]
Burst Time:14
Priority:1

P[4]
Burst Time:6
Priority:4

Process     Burst Time          Waiting Time    Turnaround Time
P[3]           14                   0                 14
P[2]           2                    14                16
P[1]           6                    16                22
P[4]           6                    22                28

Average Waiting Time=13
Average Turnaround Time=20
```

Output:

```
Enter total number of processes: 4
Enter burst time and priority for process 1: 10 3
Enter burst time and priority for process 2: 1 1
Enter burst time and priority for process 3: 2 4
Enter burst time and priority for process 4: 1 2
Process     Burst Time     Priority     Waiting Time     Turnaround Time
P2          1              1            0                1
P4          1              2            1                2
P1          10             3            2                12
P3          2              4            12               14
Average Waiting Time = 3.75
Average Turnaround Time = 7.25
```

Result:

The priority scheduling technique has been implemented successfully and the output has been verified.