Mawlana Bhashani Science and Technology University

Lab Report

Report No: 09

Course Code: ICT – 3110

Course title: Operating Systems Lab

Date of Performance:

Date of Submission: 12/09/2020

Submitted by

Name: Pritom Saha

ID: IT – 17010

3rd year 1st semester

Dept. of ICT

MBSTU

Submitted to

Nazrul Islam

Assistant Professor

Dept. of ICT

MBSTU

Experiment no: 09

Experiment name: Implementation of Priority Scheduling Algorithm.

Theory:

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 highest priority is to be executed first and so on. Processes with 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.

Working process:

1. **First of all we have to take input for the processes with their burst time and priority.**
2. **Then we have to sort the processes, burst time and priority according to the priority.**
3. **After that we just have to apply FCFS algorithm.**

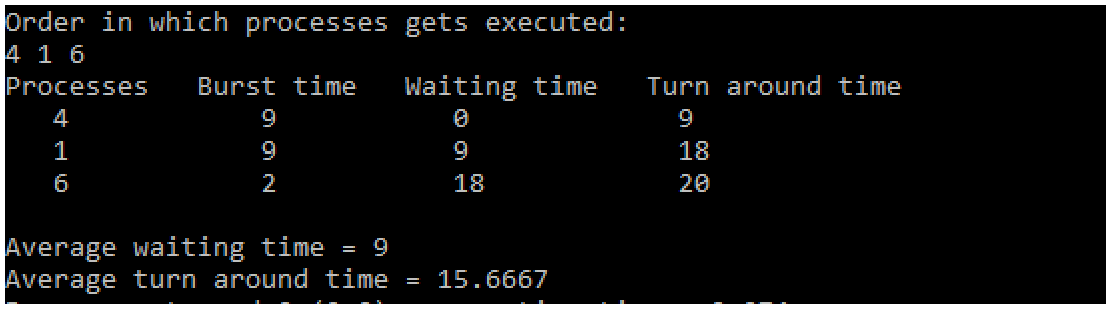
Code:

```
#include<bits/stdc++.h>

using namespace std;

vector<pair<int, pair<int, int> > > process;

signed main() {

    ios :: sync_with_stdio(false);

    cin.tie(0); cout.tie(0);

    process.push_back({5, {9, 4}});
```

```
	process.push_back({2, {2, 6}});
	process.push_back({4, {9, 1}});
	int n = 3;
	sort(process.rbegin(), process.rend());
	cout << "Order in which processes gets executed: " << endl;
	for (auto u : process) {
		cout << u.second.second << " ";
	}
	int wt[n], tat[n], total_wt = 0, total_tat = 0;
	wt[0] = 0;
	for (int i = 1; i < n; i++) {
		wt[i] = wt[i - 1] + process[i - 1].second.first;
	}
	for (int i = 0; i < n; i++) {
		tat[i] = process[i].second.first + wt[i];
	}
	cout << "\nProcesses  " << " Burst time  " << " Waiting time  " << " Turn
around time" << endl;
	for (int i = 0; i < n; i++) {
		total_wt += wt[i];
		total_tat += tat[i];
		cout << "   " << process[i].second.second << "\t\t" <<
process[i].second.first << "\t    " << wt[i] << "\t\t  " << tat[i] <<endl;
```

```
    }
    cout << "\nAverage waiting time = " << (double)total_wt / (double)n;
    cout << "\nAverage turn around time = "<< (double)total_tat / (double)n;
    return 0;
}
```

<u>Output</u>:



```
Order in which processes gets executed:
4 1 6
Processes    Burst time   Waiting time   Turn around time
   4            9            0              9
   1            9            9              18
   6            2            18             20

Average waiting time = 9
Average turn around time = 15.6667
```

<u>Discussion</u>:

A major problem with priority scheduling is indefinite blocking or starvation. A solution to the problem of indefinite blockage of the low – priority process is aging.