Progressive Education Society’s

**MODERN COLLEGE OF ENGINEERING, Pune -05.**

(An Autonomous Institute Affiliated to Savitribai Phule Pune University)

MCA Department

**PRACTICAL SUBMISSION RECORD- A.Y. 2024-25**

|  |  |  |  |
| --- | --- | --- | --- |
| **Class: FYMCA Div: A**  **Semester: II** | **Course Code: MCA01554**  **Course Name: Laboratory Practice - II** | | **Batch: F2** |
| **Name: Pranav Raju Malwatkar** | | **Roll No: 51037** | |
| **CO No:** **CO517.1** | | **Assignment No: 1** | |

**Program Title: 1) Write a C / C++ program for implementation of scheduling algorithms a) FCFS b) SJF c)**

**Priority d) Round Robin.**

**Program Code:**

#include <iostream>

#include <algorithm>

using namespace std;

struct Process {

int pid, bt, at, wt, tat, priority;

};

void fcfs(Process proc[], int n) {

proc[0].wt = 0;

for (int i = 1; i < n; i++)

proc[i].wt = proc[i - 1].wt + proc[i - 1].bt;

for (int i = 0; i < n; i++)

proc[i].tat = proc[i].wt + proc[i].bt;

}

bool compareSJF(Process a, Process b) { return a.bt < b.bt; }

void sjf(Process proc[], int n) {

sort(proc, proc + n, compareSJF);

fcfs(proc, n);

}

bool comparePriority(Process a, Process b) { return a.priority < b.priority; }

void priorityScheduling(Process proc[], int n) {

sort(proc, proc + n, comparePriority);

fcfs(proc, n);

}

void roundRobin(Process proc[], int n, int quantum) {

int rem\_bt[n], t = 0;

for (int i = 0; i < n; i++)

rem\_bt[i] = proc[i].bt;

bool done;

do {

done = true;

for (int i = 0; i < n; i++) {

if (rem\_bt[i] > 0) {

done = false;

if (rem\_bt[i] > quantum) {

t += quantum;

rem\_bt[i] -= quantum;

} else {

t += rem\_bt[i];

proc[i].wt = t - proc[i].bt;

rem\_bt[i] = 0;

}

}

}

} while (!done);

for (int i = 0; i < n; i++)

proc[i].tat = proc[i].wt + proc[i].bt;

}

void display(Process proc[], int n) {

cout << "PID\tBT\tWT\tTAT\n";

for (int i = 0; i < n; i++)

cout << proc[i].pid << "\t" << proc[i].bt << "\t" << proc[i].wt << "\t" << proc[i].tat << "\n";

}

int main() {

int n, choice, quantum;

cout << "Enter number of processes: ";

cin >> n;

Process proc[n];

cout << "Enter Burst Time and Priority:\n";

for (int i = 0; i < n; i++) {

proc[i].pid = i + 1;

cout << "P" << i + 1 << " BT: ";

cin >> proc[i].bt;

cout << "P" << i + 1 << " Priority: ";

cin >> proc[i].priority;

}

cout << "Choose Scheduling Algorithm:\n1. FCFS\n2. SJF\n3. Priority\n4. Round Robin\n";

cin >> choice;

switch (choice) {

case 1: fcfs(proc, n); break;

case 2: sjf(proc, n); break;

case 3: priorityScheduling(proc, n); break;

case 4:

cout << "Enter Time Quantum: ";

cin >> quantum;

roundRobin(proc, n, quantum);

break;

default: cout << "Invalid Choice"; return 0;

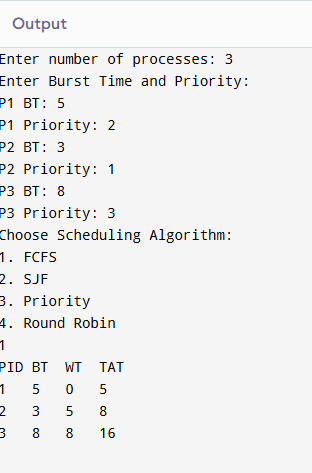
}

display(proc, n);

return 0;

}

**Output:**

****