Anas Ahmad Ilyas Ahmad

Roll No : 222010001 Batch:1

Subject: OS Practical 2: Priority CPU Scheduling

**Source Code:**

#include <bits/stdc++.h>

using namespace std;

class Process{

public:

int p\_id; //process id

int at; //arrival time

int bt; //burst time

int pri; //priority

int st; // first time cpu alloted time

int ct; //completion time

int tat; // turnaround time

int wt; //waiting time

int rt; //response time

};

int main(){

int n;

cout << "Enter number of processes:";

cin >> n;

Process p[n];

bool is\_completed[n] = {false};

int total\_tat = 0;

int total\_wt = 0;

int total\_rt = 0;

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

cout << "Enter arrival time,burst time and priority of process " <<i+1<< ": ";

cin >> p[i].at>>p[i].bt>>p[i].pri;

p[i].p\_id = i + 1;

cout << endl;

}

int cur\_time = 0;

int completed = 0;

while (completed != n){

int cur\_pro = -1;

int max\_priority = INT\_MIN;

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

if (p[i].at <= cur\_time && is\_completed[i] == false){

if (p[i].pri > max\_priority){

max\_priority = p[i].pri;

cur\_pro = i;

}

if (p[i].pri == max\_priority){

if (p[i].at < p[cur\_pro].at){

max\_priority = p[i].pri;

cur\_pro = i;

}

}

}

}

if (cur\_pro != -1){

p[cur\_pro].st = cur\_time;

p[cur\_pro].ct = p[cur\_pro].st + p[cur\_pro].bt;

p[cur\_pro].tat = p[cur\_pro].ct - p[cur\_pro].at;

p[cur\_pro].wt = p[cur\_pro].tat - p[cur\_pro].bt;

p[cur\_pro].rt = p[cur\_pro].st - p[cur\_pro].at;

total\_tat = total\_tat + p[cur\_pro].tat;

total\_wt = total\_wt + p[cur\_pro].wt;

total\_rt = total\_rt + p[cur\_pro].rt;

completed++;

is\_completed[cur\_pro] = true;

cur\_time = p[cur\_pro].ct;

}else{

cur\_time++;

}

}

cout << "#P\t"<< "AT\t"<< "BT\t"<<"PT\t"<< "ST\t"<< "CT\t"<< "TAT\t"<< "WT\t"<< "RT\t"<< "\n"<< endl;

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

cout << p[i].p\_id << "\t" << p[i].at << "\t" << p[i].bt << "\t"<<p[i].pri<<"\t"<< p[i].st << "\t" << p[i].ct << "\t" << p[i].tat << "\t" << p[i].wt << "\t" << p[i].rt << "\t" << "\n"<< endl;

}

cout << "Average Turnaround Time: " << (float)total\_tat / n << endl;

cout << "Average Waiting Time: " << (float)total\_wt / n << endl;

cout << "Average Response Time: " << (float)total\_rt / n << endl;

return 0;

}

**Output:**

