Lab-2

#include <stdio.h>

#define MAX 10

typedef struct {

int pid, at, bt, pt, remaining\_bt, ct, tat, wt, rt, is\_completed, st;

} Process;

void nonPreemptivePriority(Process p[], int n) {

int time = 0, completed = 0;

while (completed < n) {

int highest\_priority = 9999, selected = -1;

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

if (p[i].at <= time && !p[i].is\_completed && p[i].pt < highest\_priority) {

highest\_priority = p[i].pt;

selected = i;

}

}

if (selected == -1) {

time++;

continue;

}

if (p[selected].rt == -1) {

p[selected].st = time;

p[selected].rt = time - p[selected].at;

}

time += p[selected].bt;

p[selected].ct = time;

p[selected].tat = p[selected].ct - p[selected].at;

p[selected].wt = p[selected].tat - p[selected].bt;

p[selected].is\_completed = 1;

completed++;

}

}

void preemptivePriority(Process p[], int n) {

int time = 0, completed = 0;

while (completed < n) {

int highest\_priority = 9999, selected = -1;

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

if (p[i].at <= time && p[i].remaining\_bt > 0 && p[i].pt < highest\_priority) {

highest\_priority = p[i].pt;

selected = i;

}

}

if (selected == -1) {

time++;

continue;

}

if (p[selected].rt == -1) {

p[selected].st = time;

p[selected].rt = time - p[selected].at;

}

p[selected].remaining\_bt--;

time++;

if (p[selected].remaining\_bt == 0) {

p[selected].ct = time;

p[selected].tat = p[selected].ct - p[selected].at;

p[selected].wt = p[selected].tat - p[selected].bt;

completed++;

}

}

}

void displayProcesses(Process p[], int n) {

float avg\_tat = 0, avg\_wt = 0, avg\_rt = 0;

printf("\nPID\tAT\tBT\tPriority\tCT\tTAT\tWT\tRT\n");

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

printf("%d\t%d\t%d\t%d\t\t%d\t%d\t%d\t%d\n",

p[i].pid, p[i].at, p[i].bt, p[i].pt, p[i].ct, p[i].tat, p[i].wt, p[i].rt);

avg\_tat += p[i].tat;

avg\_wt += p[i].wt;

avg\_rt += p[i].rt;

}

printf("\nAverage TAT: %.2f", avg\_tat / n);

printf("\nAverage WT: %.2f", avg\_wt / n);

printf("\nAverage RT: %.2f\n", avg\_rt / n);

}

int main() {

Process p[MAX];

int n, choice;

printf("Enter the number of processes: ");

scanf("%d", &n);

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

p[i].pid = i + 1;

printf("\nEnter Arrival Time, Burst Time, and Priority for Process %d:\n", p[i].pid);

printf("Arrival Time: ");

scanf("%d", &p[i].at);

printf("Burst Time: ");

scanf("%d", &p[i].bt);

printf("Priority : ");

scanf("%d", &p[i].pt);

p[i].remaining\_bt = p[i].bt;

p[i].is\_completed = 0;

p[i].rt = -1;

}

while (1) {

printf("\nPriority Scheduling Menu:\n");

printf("1. Non-Preemptive Priority Scheduling\n");

printf("2. Preemptive Priority Scheduling\n");

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch (choice) {

case 1:

nonPreemptivePriority(p, n);

printf("Non-Preemptive Scheduling Completed!\n");

displayProcesses(p, n);

break;

case 2:

preemptivePriority(p, n);

printf("Preemptive Scheduling Completed!\n");

displayProcesses(p, n);

break;

case 3:

printf("Exiting...\n");

return 0;

default:

printf("Invalid choice! Try again.\n");

}

}

return 0;

}

**Output:**



