CPU Scheduling - Priority Scheduling

#include <stdio.h>

struct Process {

int id;

int burst\_time;

int priority;

int waiting\_time;

int turnaround\_time;

};

void sort\_by\_priority(struct Process p[], int n) {

struct Process temp;

for (int i = 0; i < n - 1; i++) {

for (int j = 0; j < n - i - 1; j++) {

if (p[j].priority > p[j + 1].priority) {

temp = p[j];

p[j] = p[j + 1];

p[j + 1] = temp;

}

}

}

}

int main() {

int n;

struct Process p[100];

int total\_waiting\_time = 0, total\_turnaround\_time = 0;

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

scanf("%d", &n);

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

p[i].id = i + 1;

printf("Enter burst time for Process %d: ", p[i].id);

scanf("%d", &p[i].burst\_time);

printf("Enter priority for Process %d (lower number = higher priority): ", p[i].id);

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

}

sort\_by\_priority(p, n);

p[0].waiting\_time = 0;

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

p[i].waiting\_time = p[i - 1].waiting\_time + p[i - 1].burst\_time;

}

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

p[i].turnaround\_time = p[i].waiting\_time + p[i].burst\_time;

total\_waiting\_time += p[i].waiting\_time;

total\_turnaround\_time += p[i].turnaround\_time;

}

printf("\nProcess\tBurst Time\tPriority\tWaiting Time\tTurnaround Time\n");

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

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

p[i].id, p[i].burst\_time, p[i].priority,

p[i].waiting\_time, p[i].turnaround\_time);

}

float avg\_waiting\_time = (float)total\_waiting\_time / n;

float avg\_turnaround\_time = (float)total\_turnaround\_time / n;

printf("\nAverage Waiting Time: %.2f\n", avg\_waiting\_time);

printf("Average Turnaround Time: %.2f\n", avg\_turnaround\_time);

return 0;

}

