**EXPERIMENT-5**

5. Construct a scheduling program with C that selects the waiting process with the highest

priority to execute next.

#include <stdio.h>

int main() {

int n;

printf("Enter number of processes: ");

scanf("%d", &n);

int burst[n], priority[n], pid[n];

int waiting[n], turnaround[n];

int i, j;

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

pid[i] = i + 1;

printf("Process %d burst time: ", pid[i]);

scanf("%d", &burst[i]);

printf("Process %d priority (lower = higher): ", pid[i]);

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

}

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

int best = i;

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

if (priority[j] < priority[best]) {

best = j;

}

}

int tmp = priority[i]; priority[i] = priority[best]; priority[best] = tmp;

tmp = burst[i]; burst[i] = burst[best]; burst[best] = tmp;

tmp = pid[i]; pid[i] = pid[best]; pid[best] = tmp;

}

waiting[0] = 0;

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

waiting[i] = waiting[i-1] + burst[i-1];

}

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

turnaround[i] = waiting[i] + burst[i];

}

int totalW = 0, totalT = 0;

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

totalW += waiting[i];

totalT += turnaround[i];

}

printf("\nPID\tPri\tBurst\tWait\tTurnaround\n");

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

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

pid[i], priority[i], burst[i], waiting[i], turnaround[i]);

}

printf("\nAverage waiting time: %.2f\n", (float)totalW / n);

printf("Average turnaround time: %.2f\n", (float)totalT / n);

return 0;

}

**SAMPLE**

Enter number of processes: 4

Process 1 burst time: 5

Process 1 priority (lower = higher): 2

Process 2 burst time: 3

Process 2 priority (lower = higher): 1

Process 3 burst time: 8

Process 3 priority (lower = higher): 4

Process 4 burst time: 6

Process 4 priority (lower = higher): 3

PID Pri Burst Wait Turnaround

P2 1 3 0 3

P1 2 5 3 8

P4 3 6 8 14

P3 4 8 14 22

Average waiting time: 6.25

Average turnaround time: 11.75