#include <stdio.h>

#include <stdlib.h>

struct process {

int pid;

int arrival\_time;

int burst\_time;

int remaining\_time;

int waiting\_time;

int turnaround\_time;

};

int main() {

int n = 4; // number of processes

struct process p[] = {{1, 0, 5, 5, 0, 0},

{2, 1, 3, 3, 0, 0},

{3, 2, 3, 3, 0, 0},

{4, 4, 1, 1, 0, 0}};

int current\_time = 0;

int completed = 0;

int shortest = -1;

int i, j;

float total\_waiting\_time = 0;

float total\_turnaround\_time = 0;

while (completed < n) {

shortest = -1;

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

if (p[i].arrival\_time <= current\_time && p[i].remaining\_time > 0) {

if (shortest == -1 || p[i].remaining\_time < p[shortest].remaining\_time) {

shortest = i;

}

}

}

if (shortest == -1) {

current\_time++;

continue;

}

p[shortest].remaining\_time--;

if (p[shortest].remaining\_time == 0) {

completed++;

p[shortest].turnaround\_time = current\_time + 1 - p[shortest].arrival\_time;

p[shortest].waiting\_time = p[shortest].turnaround\_time - p[shortest].burst\_time;

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

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

}

current\_time++;

}

printf("PID\tArrival Time\tBurst Time\tWaiting Time\tTurnaround Time\n");

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

printf("%d\t%d\t\t%d\t\t%d\t\t%d\n", p[i].pid, p[i].arrival\_time, p[i].burst\_time, p[i].waiting\_time, p[i].turnaround\_time);

}

printf("Average waiting time: %.2f\n", total\_waiting\_time/n);

printf("Average turnaround time: %.2f\n", total\_turnaround\_time/n);

return 0;

}

OUTPUT Of SRPT Algorithm

PID Arrival Time Burst Time Waiting Time Turnaround Time

1 0 5 7 12

2 1 3 0 3

3 2 3 3 6

4 4 1 0 1

Average waiting time: 2.50

Average turnaround time: 5.50

--------------------------------

Process exited after 0.03223 seconds with return value 0

Press any key to continue . . .