rrb.c

#include <stdio.h>

int main()

{

int n, tq, i, time = 0, remain;

int pid[15], bt[15], rt[15]; // pid: process IDs, bt: burst times, rt: remaining times

float awt = 0.0, att = 0.0; // Average waiting time and turnaround time

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

scanf("%d", &n);

remain = n; // Remaining processes to execute

printf("Enter process ID of all processes: ");

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

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

}

printf("Enter burst time of all processes: ");

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

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

rt[i] = bt[i]; // Initialize remaining times

}

printf("Enter the time quantum: ");

scanf("%d", &tq);

printf("Process ID Burst Time Waiting Time TurnAround Time\n");

int wt[15] = {0}; // Initialize waiting times

while (remain > 0) {

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

if (rt[i] > 0) {

if (rt[i] > tq) {

time += tq;

rt[i] -= tq;

} else {

time += rt[i];

wt[i] = time - bt[i]; // Waiting time = Total time - Burst time

rt[i] = 0; // Process is finished

remain--;

}

}

}

}

// Calculate turnaround times and display results

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

int tat = wt[i] + bt[i]; // Turnaround time = Waiting time + Burst time

awt += wt[i]; // Accumulate total waiting time

att += tat; // Accumulate total turnaround time

printf("%d\t\t%d\t\t%d\t\t%d\n", pid[i], bt[i], wt[i], tat);

}

// Calculate averages

awt /= n;

att /= n;

printf("Avg. waiting time = %f\n", awt);

printf("Avg. turnaround time = %f\n", att);

return 0;

}

Enter the number of processes: 5

Enter process ID of all processes: 1

2

3

4

5

Enter burst time of all processes: 5

6

7

8

9

Enter the time quantum: 2

Process ID Burst Time Waiting Time TurnAround Time

1 5 16 21

2 6 17 23

3 7 23 30

4 8 24 32

5 9 26 35

Avg. waiting time = 21.200001

Avg. turnaround time = 28.200001