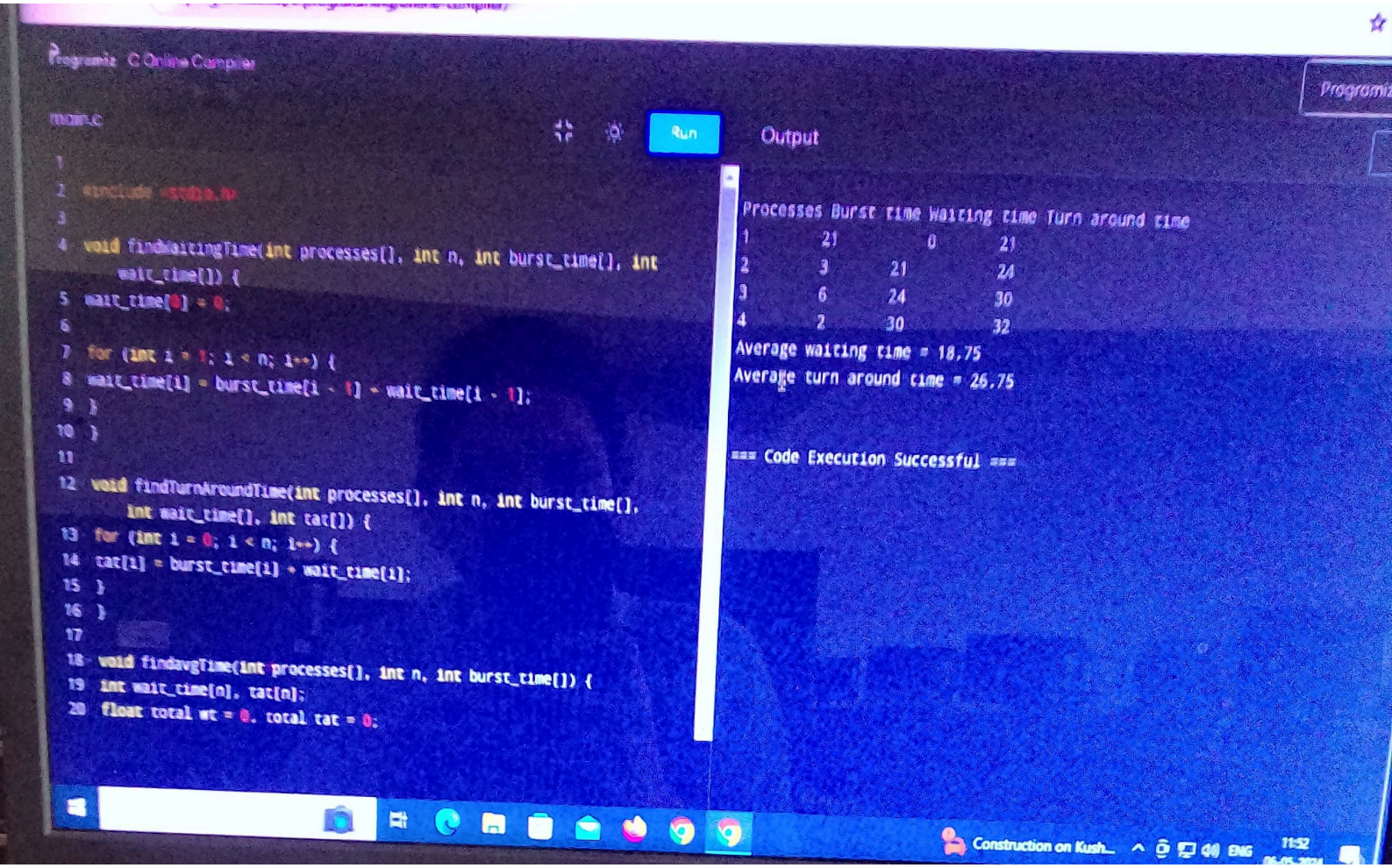
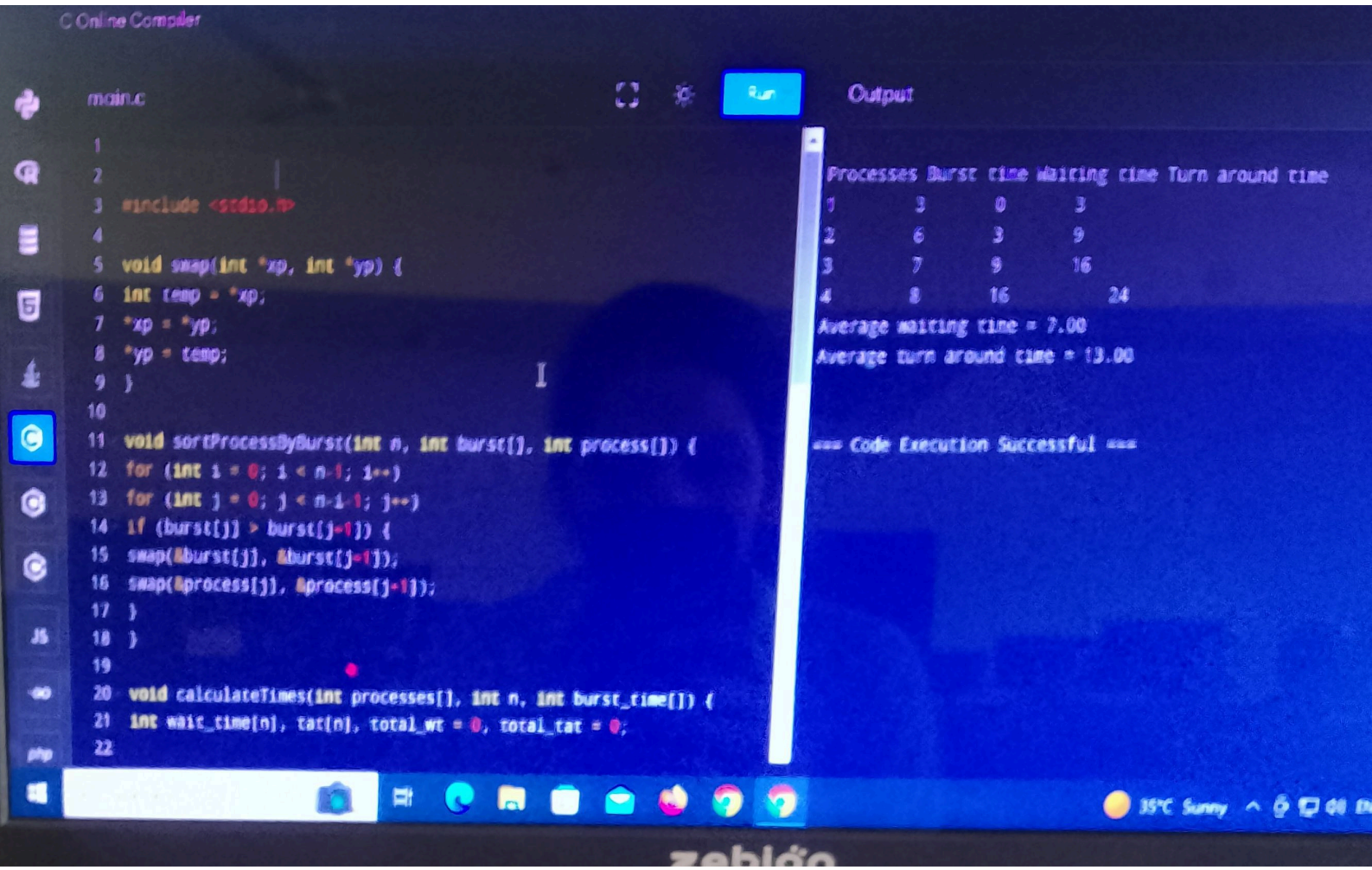
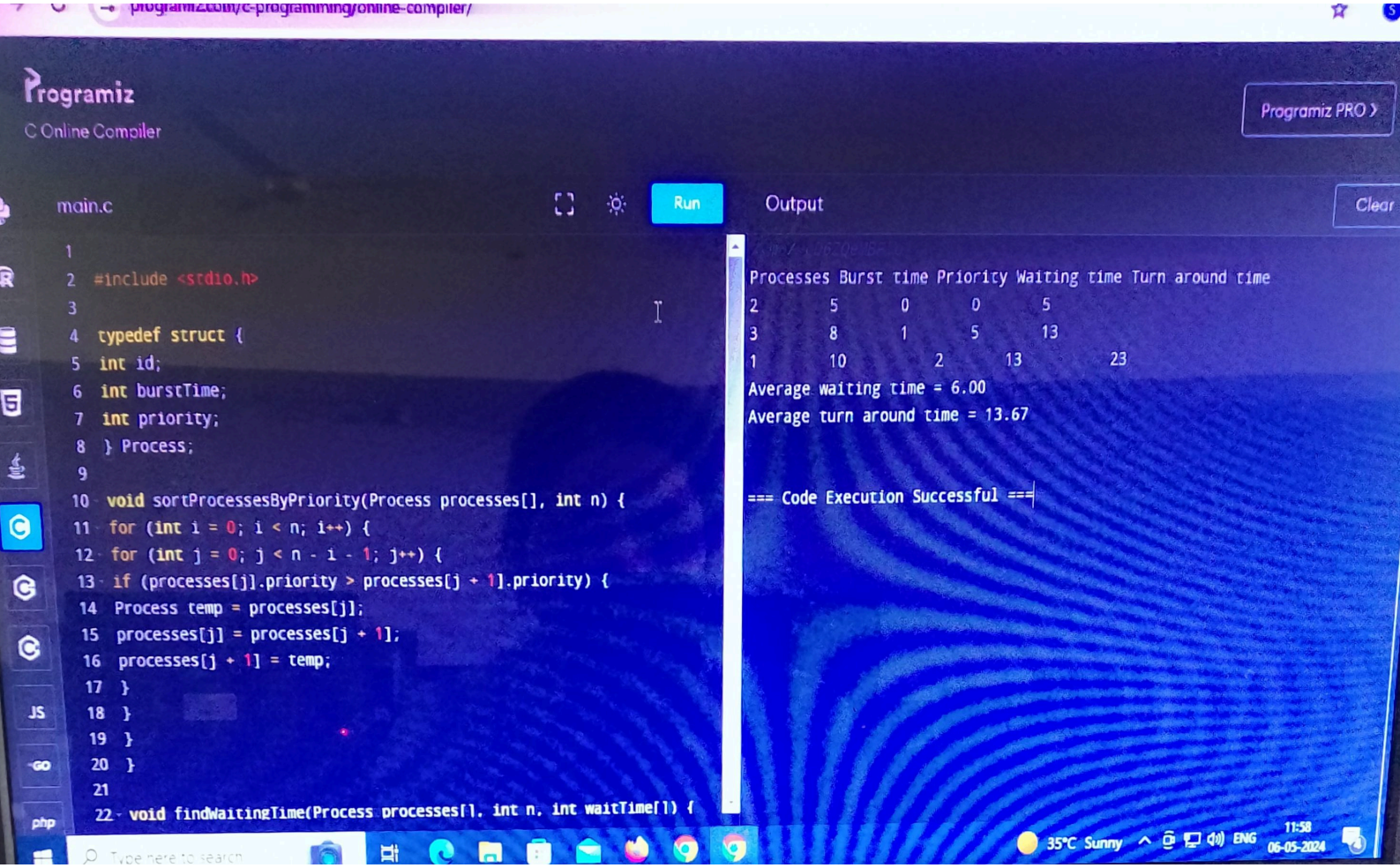
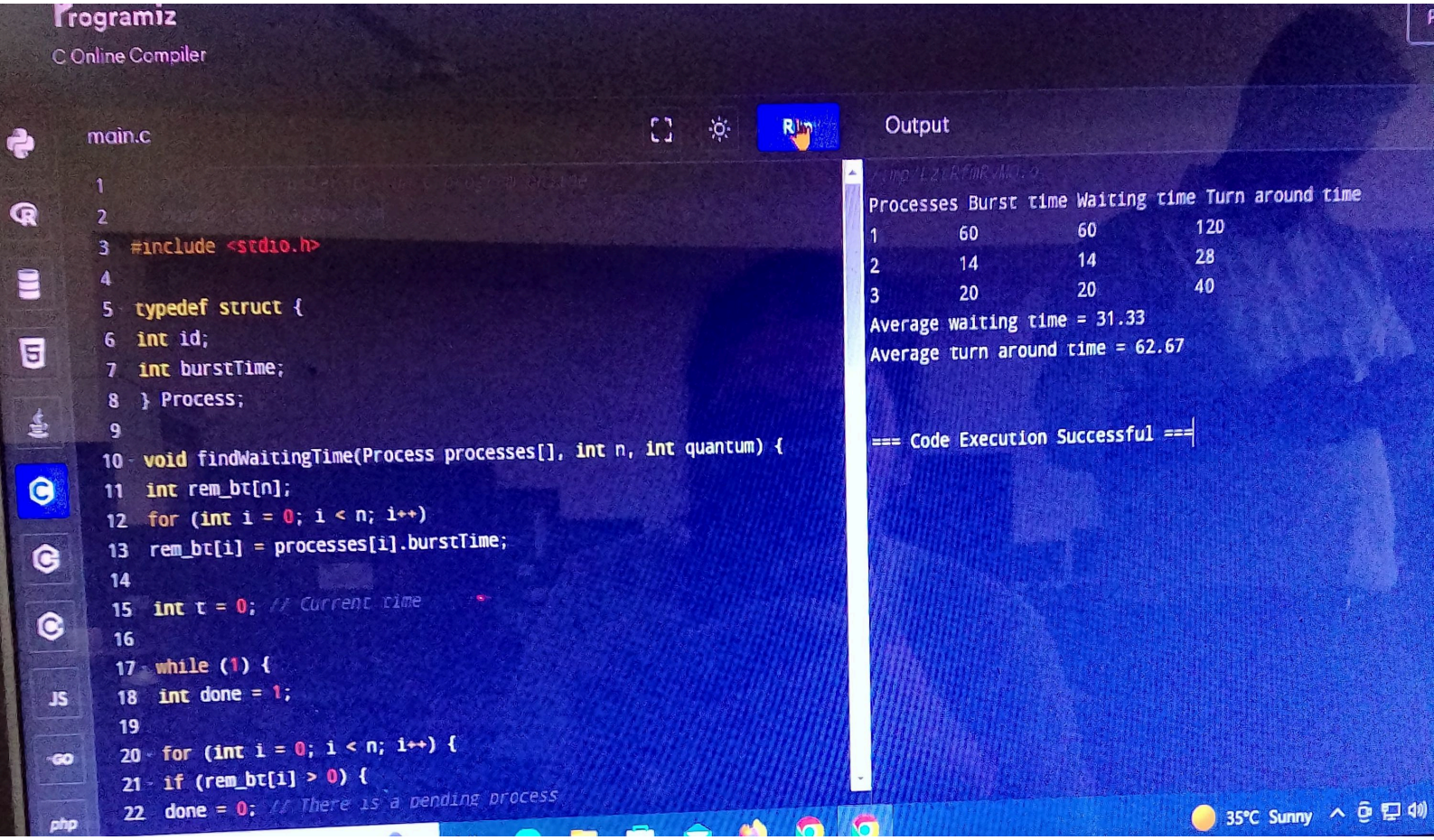
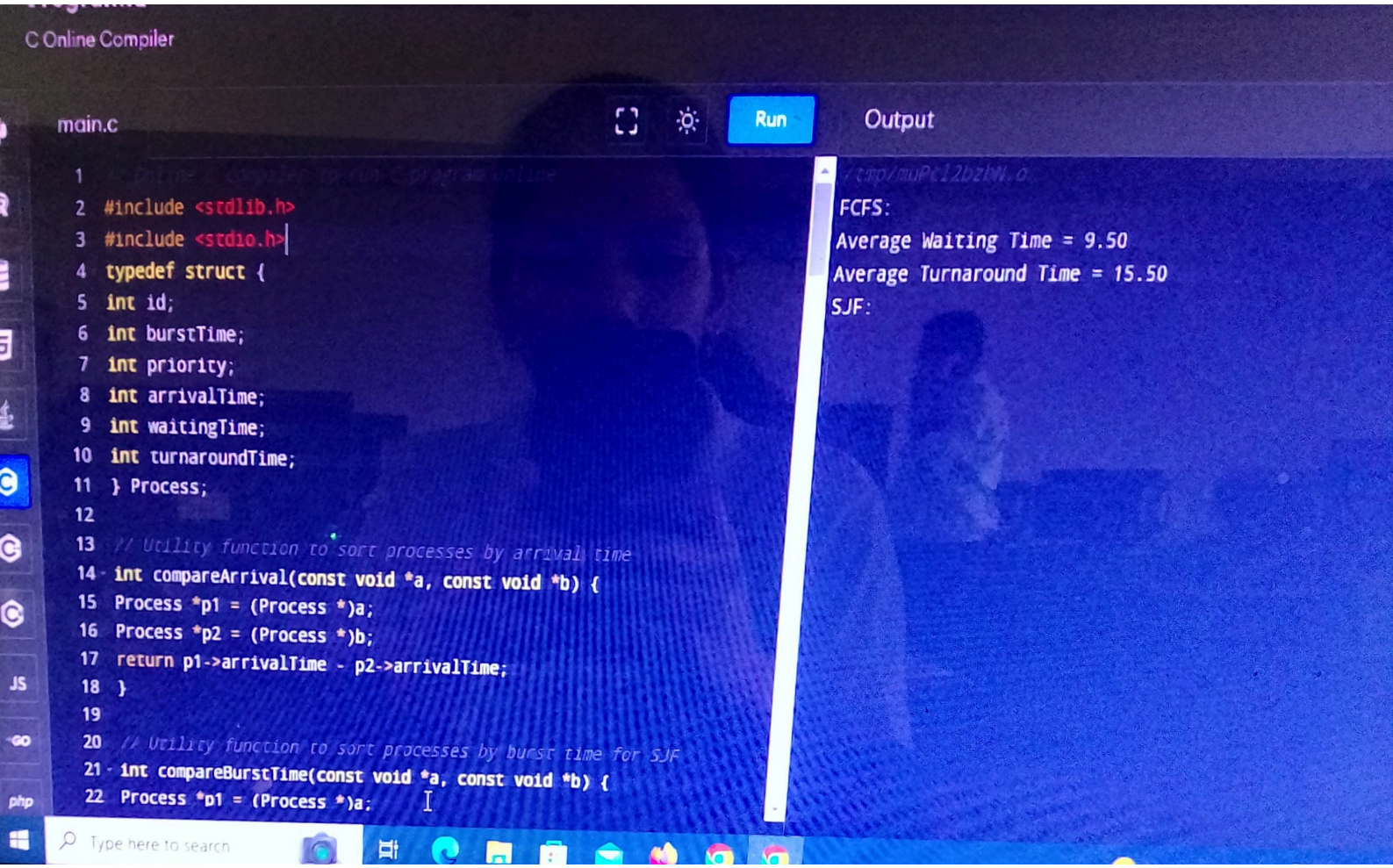

Programiz: C Online Compiler
main.c
Run
Output
#include <stdio.h>
void findWaitingTime(int processes[], int n, int burst_time[], int wait_time[]) {
wait_time[0] = 0;
for (int i = 1; i < n; i++) {
wait_time[i] = burst_time[i - 1] + wait_time[i - 1];
}
}
void findTurnAroundTime(int processes[], int n, int burst_time[], int wait_time[], int tat[]) {
for (int i = 0; i < n; i++) {
tat[i] = burst_time[i] + wait_time[i];
}
}
void findavgTime(int processes[], int n, int burst_time[]) {
int wait_time[n], tat[n];
float total wt = 0, total tat = 0;
Processes Burst time Waiting time Turn around time
1 21 0 21
2 3 21 24
3 6 24 30
4 2 30 32
Average waiting time = 18.75
Average turn around time = 26.75
=== Code Execution Successful ===
Construction on Kush...
ENG
11:52


C Online Compiler
main.c
Run
Output
#include <stdio.h>
void swap(int *xp, int *yp) {
int temp = *xp;
*xp = *yp;
*yp = temp;
}
void sortProcessByBurst(int n, int burst[], int process[]) {
for (int i = 0; i < n-1; i++)
for (int j = 0; j < n-i-1; j++)
if (burst[j] > burst[j+1]) {
swap(&burst[j], &burst[j+1]);
swap(&process[j], &process[j+1]);
}
}
void calculateTimes(int processes[], int n, int burst_time[]) {
int wait_time[n], tat[n], total_wt = 0, total_tat = 0;
Processes Burst time Waiting time Turn around time
1 3 0 3
2 6 3 9
3 7 9 16
4 8 16 24
Average waiting time = 7.00
Average turn around time = 13.00
=== Code Execution Successful ===
35°C Sunny


Programiz
C Online Compiler
Programiz PRO
main.c
Run
Output
Clear
#include <stdio.h>
typedef struct {
int id;
int burstTime;
int priority;
} Process;
void sortProcessesByPriority(Process processes[], int n) {
for (int i = 0; i < n; i++) {
for (int j = 0; j < n - i - 1; j++) {
if (processes[j].priority > processes[j + 1].priority) {
Process temp = processes[j];
processes[j] = processes[j + 1];
processes[j + 1] = temp;
}
}
}
}
void findWaitingTime(Process processes[], int n, int waitTime[]) {
Processes Burst time Priority Waiting time Turn around time
2 5 0 0 5
3 8 1 5 13
1 10 2 13 23
Average waiting time = 6.00
Average turn around time = 13.67
=== Code Execution Successful ===
35°C Sunny
ENG
11:58
06-05-2024


Programiz
C Online Compiler
main.c
Run
Output
#include <stdio.h>
typedef struct {
int id;
int burstTime;
} Process;
void findWaitingTime(Process processes[], int n, int quantum) {
int rem_bt[n];
for (int i = 0; i < n; i++)
rem_bt[i] = processes[i].burstTime;
int t = 0; // Current time
while (1) {
int done = 1;
for (int i = 0; i < n; i++) {
if (rem_bt[i] > 0) {
done = 0; // There is a pending process
Processes Burst time Waiting time Turn around time
1 60 60 120
2 14 14 28
3 20 20 40
Average waiting time = 31.33
Average turn around time = 62.67
=== Code Execution Successful ===
35°C Sunny


C Online Compiler
main.c
Run
Output
#include <stdlib.h>
#include <stdio.h>
typedef struct {
int id;
int burstTime;
int priority;
int arrivalTime;
int waitingTime;
int turnaroundTime;
} Process;
// Utility function to sort processes by arrival time
int compareArrival(const void *a, const void *b) {
Process *p1 = (Process *)a;
Process *p2 = (Process *)b;
return p1->arrivalTime - p2->arrivalTime;
}
// Utility function to sort processes by burst time for SJF
int compareBurstTime(const void *a, const void *b) {
Process *p1 = (Process *)a;
FCFS:
Average Waiting Time = 9.50
Average Turnaround Time = 15.50
SJF:
Type here to search