**Exercise-1**

Implement CPU Scheduling Algorithms: **First Come First Serve**

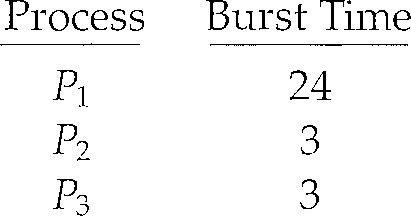
**AIM:**

To Implement CPU Scheduling Algorithms: **First Come First Serve**

**DESCRIPTION:**

## FCFS CPU scheduling algorithm:

* The simplest CPU-scheduling algorithm is the first-come, first-served (FCFS) scheduling algorithm.
* The process that requests the CPU first is allocated the CPU first.
* The implementation of the FCFS policy is easily managed with a FIFO queue.
* When a process enters the ready queue, its PCB is linked onto the tail of the queue.
* When the CPU is free, it is allocated to the process at the head of the queue.
* The running process is then removed from the queue.
* The average waiting time under the FCFS policy is often quite long.
* Consider the following set of processes that arrive at time 0, with the length of the CPU burst given in milliseconds:



* If the processes arrive in the order P1, P2, P3, and are served in FCFS order, we get the result shown in the following Gantt chart.
* Gantt chart is a bar chart that illustrates a particular schedule, including the start and finish times of each of the participating processes.
* The waiting time is 0 milliseconds for process P1, 24 milliseconds for process P2, and 27 milliseconds for process P3.
* Thus, the average waiting time is (0+ 24 + 27)/3 = 17 milliseconds.
* There is a **convoy effect** as all the other processes wait for the one big process to get off the CPU.

Burst Time: The process needs time to complete its execution is called Burst Time.

Arrival Time: The time at which the process arrived is called Arrival time. Generally, starts from 0.

Waiting Time: The amount of time the process waited for its execution.

Finish Time: The time at which the process completes its execution.

**PROGRAM:**

#include<stdio.h>

struct process

{

char name[10];

int bt,wt,ft,tat,at;

}p[10],t;

void main()

{

int i,j,n,ttat=0,twt=0;

float atat,awt;

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

scanf("%d",&n);

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

{

printf("\nEnter the name of the process:");

scanf("%s",p[i].name);

printf("Enter the burst time of the %s process:",p[i].name);

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

printf("Enter the arrival time of the %s process:",p[i].name);

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

printf("\n");

}

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

{

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

{

if(p[i].at>p[j].at)

{

t=p[i];

p[i]=p[j];

p[j]=t;

}

}

}

printf("Sorted order is:");

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

{

printf("\t%s",p[i].name);

}

printf("\n");

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

{

if(i==0)

{

p[i].ft=p[i].bt;

p[i].wt=0;

p[i].tat=p[i].bt+p[i].wt;

}

else

{

p[i].ft=p[i-1].ft+p[i].bt;

p[i].wt=p[i-1].ft-p[i].at;

p[i].tat=p[i].wt+p[i].bt;

}

}

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

{

ttat=ttat+p[i].tat;

twt=twt+p[i].wt;

}

atat=(float)ttat/n;

awt=(float)twt/n;

printf("----------------------------------------------------------");

printf("\n| pro-name\tbt\tat\t ft\t wt\t tat\t|\n");

printf("----------------------------------------------------------\n");

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

{

printf("| %s\t\t%d\t %d\t %d\t %d\t %d\t|",p[i].name,p[i].bt,p[i].at,p[i].ft,p[i].wt,p[i].tat);

printf("\n----------------------------------------------------------");

printf("\n");

}

printf("\n\nTotal turn around time is %d",ttat);

printf("\nTotal waiting time is %d",twt);

printf("\nAverage turn around time is %.2f",atat);

printf("\nAverage waiting time is %.2f",awt);

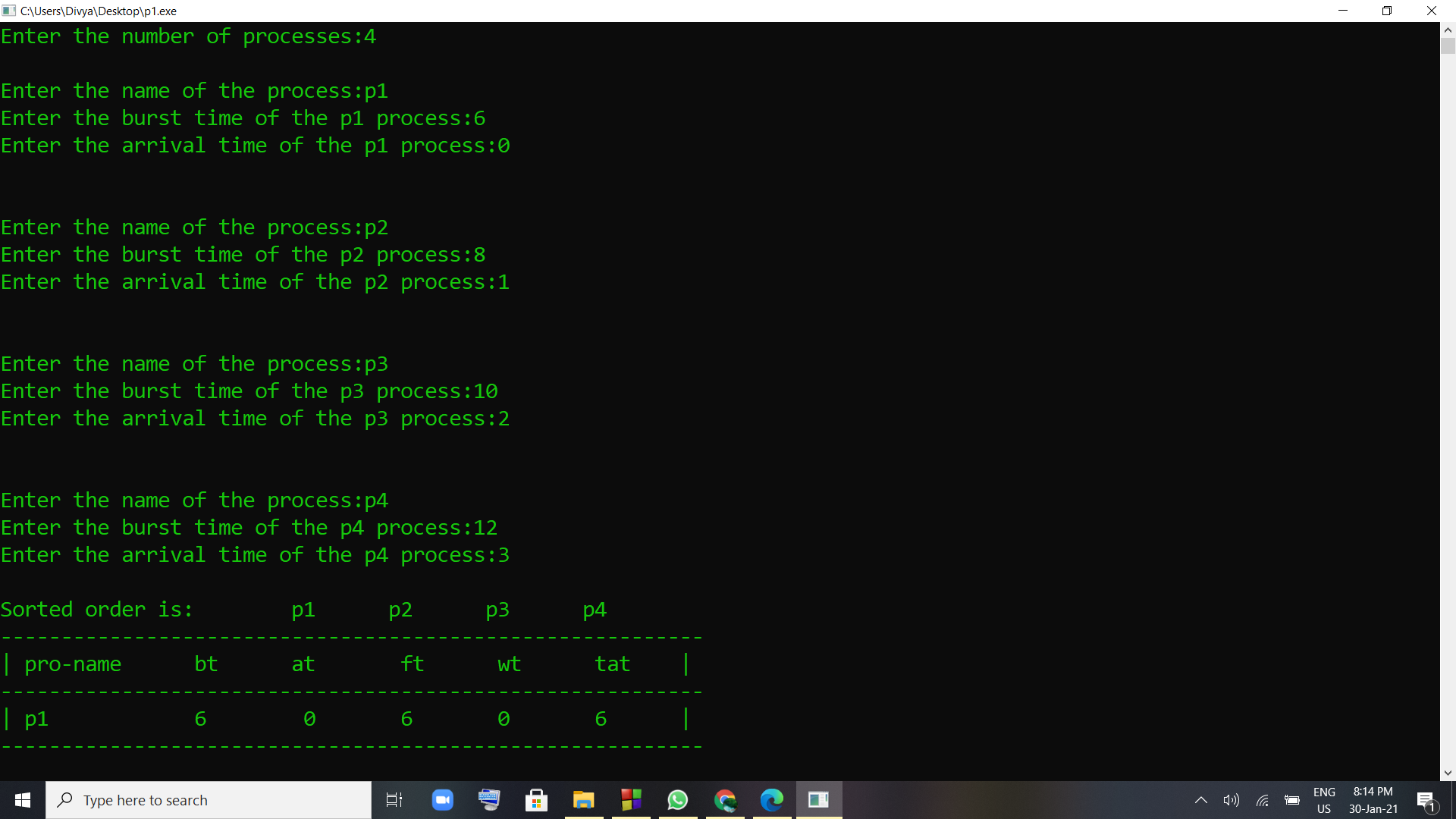
}

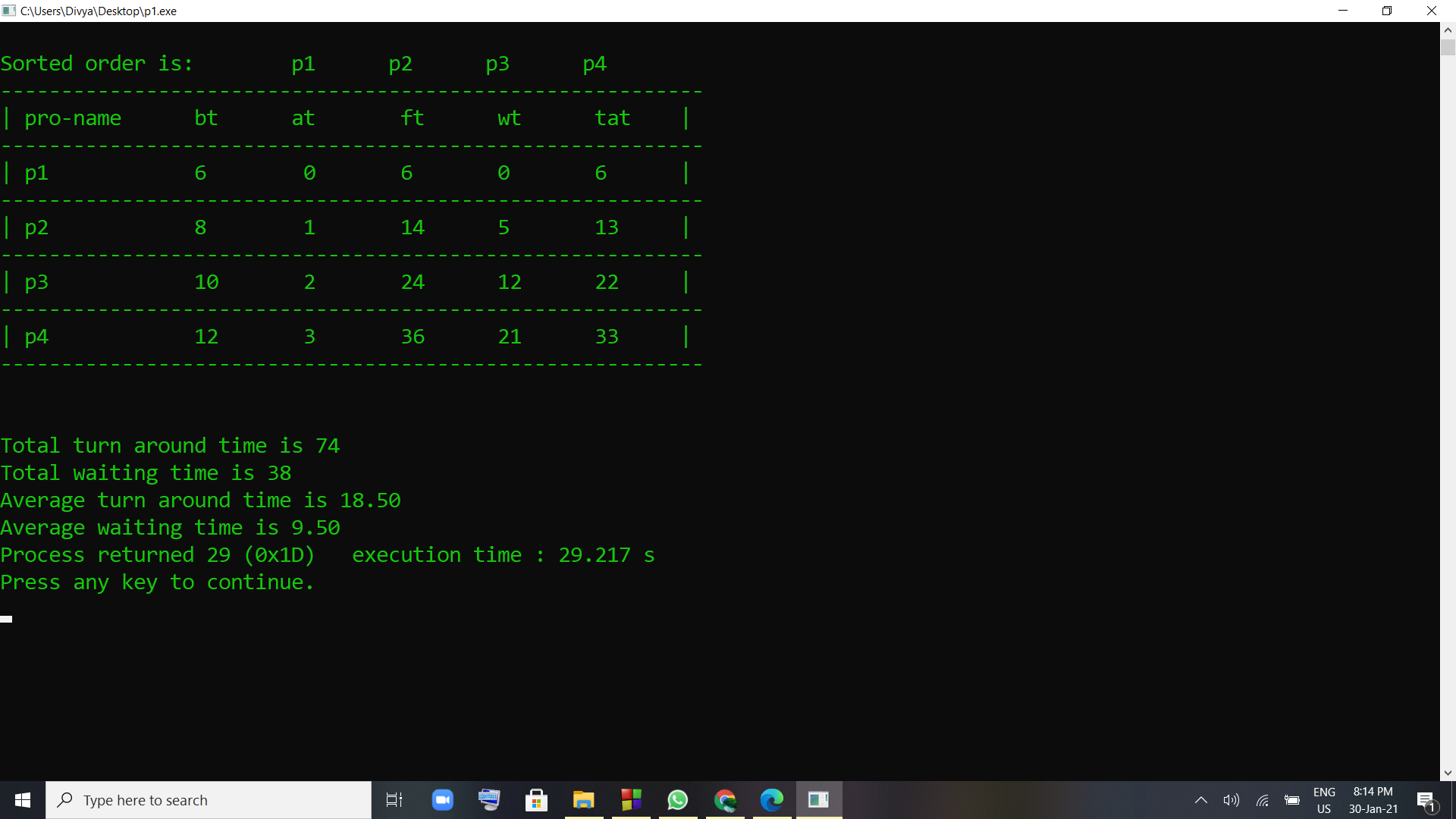
**INPUT:**

Initially, we will be entering no.of processes. And the for each process we need to enter process name, process burst time and process arrival time.

**OUTPUT:**

We output the values of finish time, waiting time and turn-around time for each process using FCFS CPU Scheduling Algorithm. And then we output the values of average waiting time and average turn-around time.

****

****