Practical No 4

Aim : Develop, debug and Execute a C program to simulate the FCFS CPU scheduling algorithms to find turnaround time and waiting time.

Apparatus: Computer system with windows installed in it.

Mingw compiler for C++, and a text editor for developing C code file.

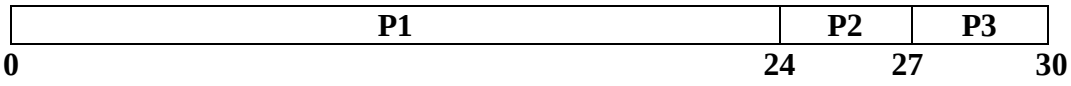
Theory :

What is FCFS scheduling?

- FCFS is short for 'First Come First Serve' Scheduling algorithm.
- FCFS is a nonpreemptive algorithm.
- In this scheduling scheme, the process that requests the CPU first is executed first.
- The implementation of this algorithm is easily managed with a first-in-first-out 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 downside of the FCFS scheduling is that its average waiting time under an FCFS policy is generally not minimal and may vary substantially if the processes' CPU burst times vary greatly.
- If a process with higher burst time starts its execution, the other processes have to wait for the big process to get off the CPU. This is known as convoy effect.

Example:

Process	Duration	Order	Arrival time
P1	24	1	0
P2	3	2	0
P3	4	3	0



P1 waiting time: 0ms

P2 waiting time: 24ms

P3 waiting time: 27ms

Average waiting time = (0 + 24 + 27) / 3

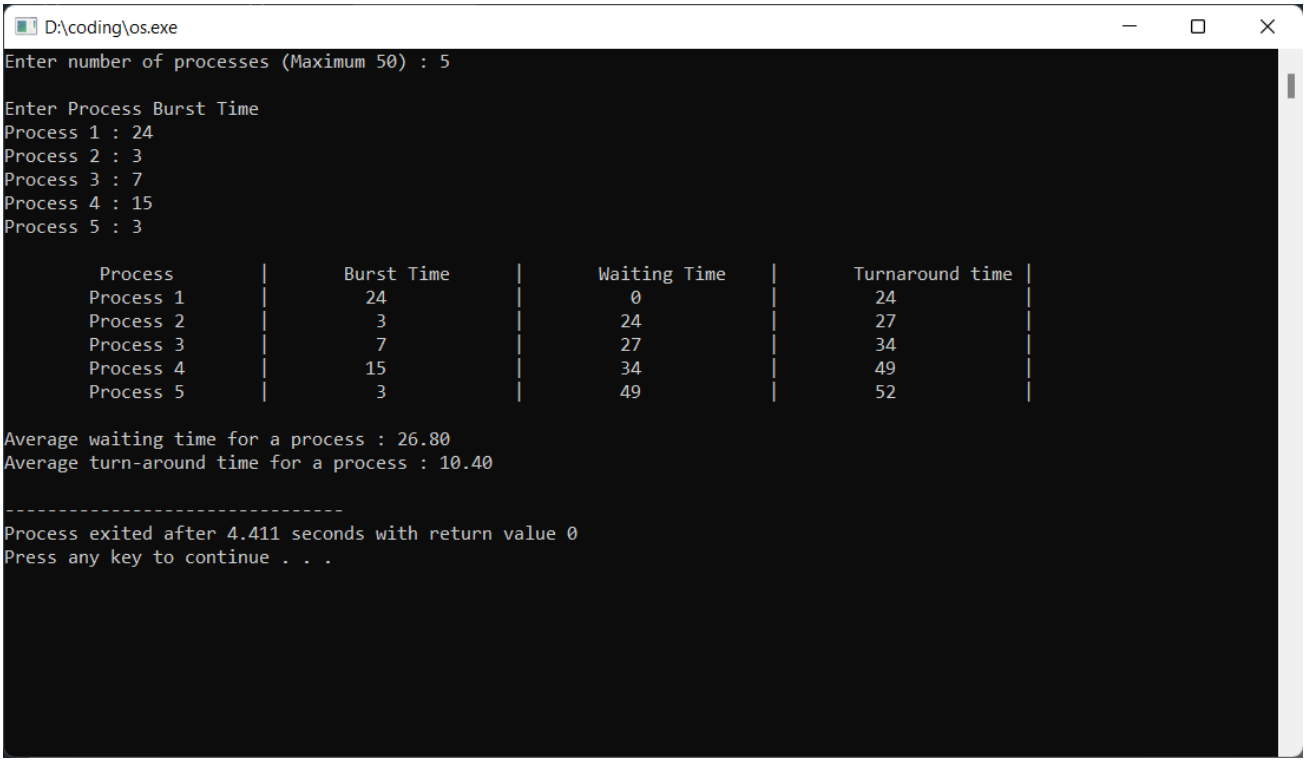
= 51/3
= 17ms

Code :

```c
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main(){
    int n, process[50], burstTime[50], waitingTime[50], turnAroundTime[50];
    float averageWaitingTime = 0, averageTurnAroundTime = 0;
    printf("Enter number of processes (Maximum 50) : ");
    scanf("%d",&n);
    printf("\nEnter Process Burst Time \n");

    for(int i = 0; i< n; i++){
            printf("Process %d : ", (i + 1));
            scanf("%d", &burstTime[i]);
    }
```

```
  waitingTime[0] = 0; //first process gets executed instantly without any wait time
   for(int i = 0; i < n; i++){
       waitingTime[i] = 0; // initializing wait time to avoid garbage values
       for(int j = 0; j < i; j++){
           waitingTime[i] += burstTime[j];
       }
   }

   printf("\n\t Process\t|\tBurst Time\t|\tWaiting Time\t|\tTurnaround time\t|\t");

   for(int i = 0; i<n; i++){
       turnAroundTime[i] = waitingTime[i] + burstTime[i];
       averageWaitingTime += waitingTime[i];
       averageTurnAroundTime = turnAroundTime[i];
       printf("\n\tProcess %d\t|\t%4d\t\t|\t%4d\t\t|\t%4d\t\t|\t"
      ,i+1,burstTime[i],waitingTime[i],turnAroundTime[i]);
   }

   averageTurnAroundTime = averageTurnAroundTime / (float)n;
   averageWaitingTime = averageWaitingTime / (float)n;

   printf("\nAverage waiting time for a process : %0.2f \n",averageWaitingTime);
printf("Average turn-around time for a process : %0.2f \n",averageTurnAroundTime );

}
```

Output:



Conclusion:

Hence, by performing this practical I got to know about the concept of First Come First Serve scheduler, its advantages, disadvantages, its use, and its implementation. I also developed, debugged and executed a c program to simulate the FCFS CPU scheduling algorithms to find turnaround time and waiting time.