CPU Scheduling Algorthms:-

Frist come frist Serve:-

AIM: To write a C Program to simple the CPU Scheduling algrothm frist come frist serve (FCFS)

DESCRIPTION:

To calculate the average waiting thime Using the FCFS algrathem frist the waiting Time of the frist Process is kept zero and the wating time of the Second Process is the burst time of the frist and the Second Process and so on. After calculating all the waiting times the average waiting time is calculated as the average of all the waiting time. FCFS mainly Says first come frist serve the algroithm which come first will be served first.

ALGORITHM:-

STEP 1: Start the Process

STEP 2: Accept the number of Process in the ready Queue

STEP 3: for each Process in the ready Q, assign the Process name and the burst time step

STEP 4: Set the waiting of the frist Process as _0' and its burst time as its turnaround time step

STEP 5: for each Process in the Ready Q cal-culate

a) wating time (n) = watting time(n) + Burst time(n-1) b)

Turnaround time(n) = wating time (n) + Burst time(n)

STEP 6! calculate

a) Avrage waiting time = $\frac{\text{Total Turnaround Time}}{\text{Number of Process}}$

b) Average Turnaround time = $\frac{\text{Total Turnaround Time}}{\text{Number of Process}}$

STEP 7: STOP the Process

SOURCE CODE

```
# include <stdio.h>
# include <conio.h>
 main () {
  int bt[20], wt[20], tat[20], i, n;
   float wtavg, tatavg
   clrscr();
   Printf ("\n Enter the number of Process ---");
    Scanf ("%d", &n);
    for(i=0; i<n; i++){
     Printf("\n Enter Burst time of Process%d --", i);
        Scanf ("%d", &bt[i]);
    }
     wt[0] = wtavg = 0;
     tat[0] = tatavg = bt[0];
     for(i=1; i<n; i++){
      wt[i] = wt[i-1] + bi[i-1];
      tat[i] = tat(i-1) + bt[i];
      wtavg = wtavg + wt[i];
      tatavg = tatavg + tat[i];
     }
Printf ("\t PROCESS \t BurstTime\t WAitingtime\t Turarondtim
     for (i=0; i<n; i++) {
      Printf ("\n\t P%d \t %d\t %d \t %d", i, bt[i], wt[i], ta
      Printf ("\n Avg waiting time ---%f", wtavg/n );
     Printf ("\n Arg time --%f", tatavg/n);
     getch();
 }
```

INPUT:-

```
Enter the number of Processes -- 3
 Enter ~~the~~ Burst time Time.Processes 0 -- 24
  Enter Burst time for Processes 1 -- 3
   Enter Burst time for Processes 2 -- 3
```

Output:-

```
PROCESS        Burst time     waiting time    Turnaround
                                                 time
Po              24              0               24
P1              3               24              27
P2              3               27              30

Average wating time --17.000000
Average Turnaround time -- 27.00000
```