EXPERIMENT NO:6

SCHEDULING ALGORITHMS

AIM

To implement different scheduling algorithms.

A. FIRST COME FIRST SERVED (FCFS)

ALGORITHM

Step 1: Start the program.
Step 2: Input the number of processes (n).
Step 3: Input the **Burst Time (BT)** for each process.
Step 4: Initialize **Turnaround Time (TAT) and Waiting Time (WT)** arrays.
Step 5: Calculate **Turnaround Time (TAT) for each process**

- TAT[0] = BT[0]
- For each subsequent process i: TAT[i]=TAT[i−1]+BT[i]
- Compute **Total Turnaround Time**: Total_TAT=∑TAT[i]
- Compute **Average Turnaround Time**: Avg_TAT=Total_TAT/n

Step 6: Calculate **Waiting Time (WT) for each process**

- WT[0] = 0 (First process has no waiting time)
- For each subsequent process i: WT[i]=WT[i−1]+BT[i−1]
- Compute **Total Waiting Time**: Total_WT=∑WT[i]
- Compute **Average Waiting Time**: Avg_WT=Total_WT/n

Step 7: Display the **Process ID, Burst Time, Turnaround Time, and Waiting Time** for each process.
Step 8: Display **Total and Average Turnaround Time and Waiting Time**.
Step 9: Stop the program.

PROGRAM

```
#include<stdio.h>

void main()                              // Main function
{
   // Declare variables
   int p[30], bt[30], tot_tat = 0, wt[30], n, tot_wt = 0, tat[30];
   float awt, avg_tat, avg_wt;           // Variables for average calculations
```

```
int i;

                                        // Prompt the user to enter the number of processes
printf("\nEnter the No. of Processes: \n");
scanf("%d", &n);                        // Read the number of processes from the user

                                        // Prompt the user to enter burst times for each process
printf("Enter Burst time for each process:\n");
for(i = 0; i < n; i++)                  // Loop through each process
{
   scanf("%d", &bt[i]);                 // Read burst time
   p[i] = i;                            // Assign process ID (0, 1, 2,..., n-1)
}

                                        // Display that the FCFS algorithm is being used
printf("\n FCFS Algorithm \n");

                                        // Calculate Turnaround Time (TAT)
for(i = 0; i < n; i++)
{
   if(i == 0)
      tat[i] = bt[i];                   // First process TAT is equal to its Burst Time
   else
      tat[i] = tat[i - 1] + bt[i];      // Next process TAT = Previous TAT + Current BT

   tot_tat = tot_tat + tat[i];          // Sum up Turnaround Time
}

                                        // Calculate Waiting Time (WT)
wt[0] = 0;                              // First process has no waiting time
for(i = 1; i < n; i++)                  // Loop from the second process
{
   wt[i] = wt[i - 1] + bt[i - 1];       // WT = Previous WT + Previous BT
   tot_wt = tot_wt + wt[i];             // Sum up Waiting Time
}

                                        // Display results
printf("\nPROCESS\t\tBURST TIME\tTURN AROUND TIME\tWAITING TIME");
for(i = 0; i < n; i++)
   printf("\nprocess[%d]\t\t%d\t\t%d\t\t%d", p[i], bt[i], tat[i], wt[i]);

                                        // Print total turnaround time and average turnaround time
printf("\n\nTotal Turnaround Time: %d", tot_tat);
printf("\nAverage Turnaround Time: %d", tot_tat / n);
```

```
                                        // Print total waiting time and average waiting time
    printf("\nTotal Waiting Time: %d", tot_wt);
    printf("\nAverage Waiting Time: %d\n", tot_wt / n);
}
```

OUTPUT

Enter the No. of processes
4
Enter Burst time for each process
8
12
11
4

FCFS Algorithm

PROCESS	BURST TIME	TURN AROUND TIME	WAITING TIME
process[0]	8	8	0
process[1]	12	20	8
process[2]	11	31	20
process[3]	4	35	31

Total Turn around Time:94
Average Turn around Time :23
Total Waiting Time:59
Total avg. Waiting Time:14