Ex. No: 7a
Date: 20-03-2024

FIRST COME FIRST SERVE

Aim:

To implement First-come First- serve(FCFS) scheduling technique

Algorithm:

1. Get the number of processes from the user.
2. Read the process name and burst time.
3. Calculate the total process time.
4. Calculate the total waiting time and total turnaround time for each process
5. Display the process name & burst time for each process.
6. Display the total waiting time, average waiting time, turnaround time

Program Code:

```
#include<stdio.h>
#include<string.h>

void findWaitingTime(int n, int * bt, int *at, int *wt){
   int service_time[n];
   memset(service_time, 0, n);
   service_time[0] = 0;
   wt[0] = 0;
   for(int i = 1; i < n; i++){
      service_time[i] = (service_time[i - 1] + bt[i - 1]);
      wt[i] = service_time[i] - at[i] ;
      if (wt[i] < 0){
         wt[i] = 0;
      }
   }
}

void findTurnAroundTime(int n, int *bt, int *wt, int *tat){
   for(int i = 0; i < n; i++){
      tat[i] = bt[i] + wt[i];
   }
}

void findAvgTime(int n, int *bt, int *at){
   int waiting_time[n], turn_around_time[n];
   memset(waiting_time, 0, n);
   memset(turn_around_time, 0, n);
   findWaitingTime(n, bt, at, waiting_time);
   findTurnAroundTime(n, bt, waiting_time, turn_around_time);
   printf("Process\tA.T\tB.T\tC.T\tW.T\tT.A.T");
   float total_wait_time = 0, total_tat = 0;
   for(int i = 0; i < n; i++){
      total_wait_time += waiting_time[i];
```

```
        total_tat += turn_around_time[i];
        int completion_time = turn_around_time[i] + at[i];
        printf("\nP%d\t%d\t%d\t%d\t%d\t%d", i, at[i], bt[i], completion_time,
    waiting_time[i], turn_around_time[i]);
    }
    printf("Average Wait Time: %f", total_wait_time / n);
    printf("Average Turn Around Time: %f", total_tat / n);
}
int main(){
    int n;
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int burst[n], arrival[n];
    for(int i = 0; i < n; i++){
        printf("Enter burst time of P%d: ", i);
        scanf("%d", &burst[i]);
        printf("Enter arrival time of P%d: ", i);
        scanf("%d", &arrival[i]);
    }
    findAvgTime(n, burst, arrival);
}
```

Output:

```
Enter number of processes: 5
Enter burst time of P0: 4
Enter arrival time of P0: 1
Enter burst time of P1: 3
Enter arrival time of P1: 2
Enter burst time of P2: 6
Enter arrival time of P2: 4
Enter burst time of P3: 1
Enter arrival time of P3: 5
Enter burst time of P4: 5
Enter arrival time of P4: 7
Process A.T     B.T     C.T     W.T     T.A.T
P0      1       4       5       0       4
P1      2       3       7       2       5
P2      4       6       13      3       9
P3      5       1       14      8       9
P4      7       5       19      7       12Average Wait Time: 4.000000Average Turn Around Time: 7.800000
```

Result:

Hence the C program to implement the First-come First-serve scheduling technique has been successfully completed and executed.