Ex. No.: 7 a
Date: 26.3.24

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:

```
# fcfs.py

print("FIRST COME FIRST SERVE SCHEDULING")
n= int(input("Enter no. of processes: "))
d = dict()
for i in range(n):
    a= int(input("Enter AT for P"+str(i+1)+": "))
    b= int(input("Enter BT for P"+str(i+1)+": "))
    l=[]
    l.append(a)
    l.append(b)
    d[key] = 1

d= sorted(d.items(), key= lambda item: item[1][0])

CT =[]
```

```
for i in range(len(d)):
    #first process
    if(i==0):
        CT.append(d[i][1][1])

    # get previous CT + new BT
    else:
        CT.append(CT[i-1]+d[i][1][1])

TAT=[]
for i in range(len(d)):
    TAT.append(CT[i]-d[i][1][0])

avg_TAT=0
    for i in TAT:                    #to find avg. TAT
        avg_TAT += i
avg_TAT/=n

WT=[]
for i in range(len(d)):
    WT.append(TAT[i]-d[i][1][1]

avg_WT=0
    for i in WT:
        avg_WT += i
avg_WT/=n
print("Process  AT  BT  CT  TAT  WT")
for i in range(n):
    print(d[i][0]," ",d[i][1][0]," ",d[i][1][1]," ",CT[i]," ",TAT[i],
                                                    WT[i])
print("Average TAT : ", avg_TAT, "ms.")
print("Average WT : ", avg_WT, "ms.")
```

Output: FIRST COME FIRST SERVE SCHEDULING

Enter NO. of processes : 5

Enter AT of P1 : 0
Enter BT of P1 : 10
Enter AT of P2 : 1
Enter BT of P2 : 1
Enter AT of P3 : 2
Enter BT of P3 : 2
Enter AT of P4 : 1
Enter BT of P4 : 1
Enter AT of P5 : 2
Enter BT of P5 : 5

P	AT	BT	CT	TAT	WT
P_1	0	10	10	10	0
P_2	1	1	11	10	9
P_3	2	2	12	11	10
P_4	1	1	14	12	10
P_5	2	5	19	17	12

Average TAT : 12.0 Ms.

Average WT : 8.2 Ms.

RESULT:

The program has been executed and output has been verified successfully.