

Ex. No.: 6a)

Date: 19/02/25

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.
6. Display the process name & burst time for each process. 6. Display the total waiting time, average waiting time, turnaround time

Program Code:

```
n = int(input("Enter the number of processes: "))
processes = []
for i in range(0, n):
    x = int(input("Enter the processes: "))
    processes.append(x)

bt = []
for i in range(0, n):
    x = int(input("Enter the burst time: "))
    bt.append(x)

at = 0
ct = []
for i in range(0, n):
    if (i == 0):
        ct.append(bt[i])
    else:
        ct.append(bt[i] + ct[i-1])

print("completion time:", ct)
```

tat = ct

print ("turn around time:", tat)

wt = []

for i in range (0, n):

wt.append (tat[i] - bt[i])

print ("waiting time:", wt)

sumt = 0

for i in range (0, n):

sumt = sumt + tat[i]

ant = (sumt)/n

sumw = 0

for i in range (0, n):

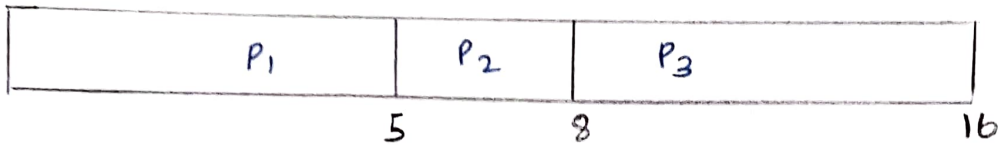
sumw = sumw + wt[i]

avw = (sumw)/n

print ("average TAT:", ant)

print ("average WT:", avw)

gantt chart :



Tabulation :

Process	BT (ms)	CT (ms)	AT (ms)	TAT = CT - AT (ms)	WT = TAT - BT (ms)
1	5	5	0	5	0
2	3	8	0	8	5
3	8	16	0	16	8

Sample Output:

Enter the number of process:

3

Enter the burst time of the processes:

24 3 3

Process	Burst Time	Waiting Time	Turn Around Time
0	24	0	24
1	3	24	27
2	3	27	30

Average waiting time is: 17.0

Average Turn around Time is: 19.0

Vi fcts.py

python3 fcts.py

Enter the number of processes: 3

Enter the processes: 1

Enter the ^{Pr}ocesses: 2

Enter the processes: 3

Enter the burst time: 5

Enter the burst time: 3

Enter the burst time: 8

completion time: [5, 8, 16]

turn around time: [5, 8, 16]

waiting time: [0, 5, 8]

average TAT: 9.6666

average WT: 4.3333

Result:

Thus the FCFS (first-come-First serve) scheduling technique was implemented using python, executed successfully and gave the expected output.