

Ex. No.: 7 a
Date: 02/04/24

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:

```
def fcfs(processes):  
    processes.sort(key = lambda x: x[1])  
    execution_order = []  
    total_waiting_time = 0  
    total_turnaround_time = 0  
    current_time = 0  
    for process_id, arrival_time, burst_time  
        in processes:  
            if arrival_time > current_time:  
                current_time = arrival_time
```

$\text{waiting-time} = \max(0, \text{current_time} - \text{arrival_time})$

$\text{total_waiting_time} += \text{waiting_time}$

$\text{execution_order.append}((\text{process_id}, \text{current_time}, \text{current_time} + \text{burst_time}, \text{waiting_time}))$

$\text{current_time} += \text{burst_time}$

$\text{total_turnaround_time} += \text{current_time} - \text{arrival_time}$

$\text{average_waiting_time} = \frac{\text{total_waiting_time}}{\text{len(processes)}}$

return execution_order, total_waiting_time,

average_waiting_time, total_turnaround_time

if --name-- == "--main--":

processes = [(1, 0, 5), (2, 2, 3), (3, 4, 7), (4, 6, 2)]

execution_order, total_waiting_time,
average_waiting_time, total_turnaround_time = fcfs(processes)

Output:

Process	AT	BT	WT	T.T
1	0	5	0	5
2	5	8	3	3
3	8	15	4	7
4	15	17	9	2

Total Waiting Time : 16

Avg Waiting Time : 4.0

Total Turnaround Time : 33

Result: The above commands are executed
successfully