

EXPERIMENT NO. 1

AIM: Write a program to implement FCFS scheduling algorithm for process management.

RESOURCES REQUIRED:

H/W Requirements: P-IV and above, Ram 128 MB, Printer, Internet Connection.

S/W Requirements: Python Compiler.

THEORY:

FCFS (First Come First Serve):

First Come First Serve (FCFS) scheduling algorithm simply schedules the jobs according to their arrival time. The job which comes first in the ready queue will get the CPU first. The lesser the arrival time of the job, the sooner will the job get the CPU. FCFS scheduling may cause the problem of starvation if the burst time of the first process is the longest among all the jobs.

1. Completion Time (CT): Time at which process completes its execution.
2. Turn Around Time (TAT): Time difference between completion and arrival time. $\text{Turn Around Time} = \text{Completion Time} - \text{Arrival Time}$.
3. Waiting Time (WT): Time difference between turn around time and burst time. $\text{Waiting Time} = \text{Turn Around Time} - \text{Burst Time}$.

Problems with the FCFS scheduling algorithm:

1. It is Non Pre-emptive algorithm, which means the **process priority** doesn't matter.

If a process with very least priority is being executed, more like **daily routine backup** process, which takes more time, and all of a sudden some other high priority process arrives, like **interrupt to avoid system crash**, the high priority process will have to wait, and hence in this case, the system will crash, just because of improper scheduling.

2. Not optimal Average Waiting Time.
3. Resources utilization in parallel is not possible, which leads to **Convoy Effect**, and hence poor resource(CPU, I/O etc) utilization.

CONCLUSION: Hence, we have implemented a program on FCFS scheduling algorithm on process management.

CODE:

```
from prettytable import PrettyTable

def fcfs():

    pid,at,bt,ct,tt,wt = [],[],[],[],[],[]

    print()

    z = int(input("Enter number of Process: "))

    print()

    for i in range(0,z):

        pid.append(int(input("Enter Proccess id:")))

        print()

        at.append(int(input("Enter arrival time:")))

        print()

        bt.append(int(input("Enter burst time:")))

        print()

    for i in range(0,z):

        min = [pid[i],at[i],bt[i]]

        j = i-1

        while(j>=0 and at[j]>min[1]):

            at[j+1],pid[j+1],bt[j+1] = at[j],pid[j],bt[j]

            j = j-1

        pid[j+1],at[j+1],bt[j+1] = min[0],min[1],min[2]

    ct.append(at[0]+bt[0])

    tt.append(ct[0]-at[0])

    wt.append(tt[0]-bt[0])

    for i in range(1,z):
```

```

ct.append(at[i]+bt[i]) if (at[i] > ct[i-1]) else ct.append(ct[i-1]+bt[i])
tt.append(ct[i]-at[i])
wt.append(tt[i]-bt[i])

for i in range(0,z):
    min = [pid[i],at[i],bt[i],ct[i],tt[i],wt[i]]
    j = i-1
    while(j>=0 and pid[j]>min[0]):
        at[j+1],pid[j+1],bt[j+1],ct[j+1],tt[j+1],wt[j+1] =
at[j],pid[j],bt[j],ct[j],tt[j],wt[j]
        j = j-1
    pid[j+1],at[j+1],bt[j+1],ct[j+1],tt[j+1],wt[j+1] =
min[0],min[1],min[2],min[3],min[4],min[5]

x = PrettyTable()
x.field_names = ["Process id","Arrival Time","Burst Time","Completion
Time","Turnaround Time","Waiting Time"]
for a,b,c,d,e,f in zip(pid,at,bt,ct,tt,wt):
    x.add_row([a,b,c,d,e,f])
print(x)

print("Total turnaround time: "+str(sum(tt))+"\nTotal waiting time:
"+str(sum(wt)))

print("Average turnaround time: "+str(sum(tt)/z)+"\nAverage waiting time:
"+str(sum(wt)/z))

if __name__ == "__main__":
    print("55_Adnan_Shaikh")
    fcfs()
}

```

OUTPUT:

```

C:\Users\adnan\OneDrive\Desktop\College\Sem 4\OS\Scheduling algorithm>python fcfs.py
55_Adnan_Shaikh

Enter number of Process: 6

Enter Proccess id:1
Enter arrival time:6
Enter burst time:3

Enter Proccess id:2
Enter arrival time:3
Enter burst time:1

Enter Proccess id:3
Enter arrival time:4
Enter burst time:3

Enter Proccess id:4
Enter arrival time:2
Enter burst time:2

Enter Proccess id:5
Enter arrival time:1
Enter burst time:4

Enter Proccess id:6
Enter arrival time:5
Enter burst time:5

```

```

C:\Users\adnan\OneDrive\Desktop\College\Sem 4\OS\Scheduling algorithm>python fcfs.py
55_Adnan_Shaikh

+-----+-----+-----+-----+-----+-----+
| Process id | Arrival Time | Burst Time | Completion Time | Turnaround Time | Waiting Time |
+-----+-----+-----+-----+-----+-----+
| 1 | 6 | 3 | 19 | 13 | 10 |
| 2 | 3 | 1 | 8 | 5 | 4 |
| 3 | 4 | 3 | 11 | 7 | 4 |
| 4 | 2 | 2 | 7 | 5 | 3 |
| 5 | 1 | 4 | 5 | 4 | 0 |
| 6 | 5 | 5 | 16 | 11 | 6 |
+-----+-----+-----+-----+-----+-----+
Total turnaround time: 45
Total waiting time: 27
Average turnaround time: 7.5
Average waiting time: 4.5

```