A.Y.: 2022-23   **Sub:** System Fundamentals

Experiment 2

(CPU Scheduling –Non Preemptive algorithm)

Aim: Implement CPU Scheduling –Non Preemptive Algorithm.

Theory: CPU scheduling is the task of selecting a waiting process from the ready queue and allocating the CPU to it. The CPU is allocated to the selected process by the dispatcher (It is the module that gives control of the CPU to the processes by short-term scheduler). Scheduling is a fundamental operating system function. In a single processor system, only one process can run at a time; any other process must wait until the CPU is free and can be rescheduled. The objective of multiprogramming is to have some process running at all times, to maximize CPU utilization.

CPU scheduling decisions may take place under the following four circumstances:

1. When a process switches from the running state to the waiting state

2. When a process switches from the running state to the ready state.

3. When a process switches from the waiting state to the ready state.

4. When a process terminates.

Depending on the above circumstances the two types of scheduling are:

1. Non-Preemptive

2. Preemptive

1. **Non-Preemptive:** Under this scheduling, once the CPU has been allocated to a process, the process keeps the CPU until it releases the CPU either by terminating or by switching to the waiting state.

2. **Preemptive:** Under this scheduling, once the CPU has been allocated to a process, the process does not keep the CPU but can be utilized by some other process. This incurs a cost associated with access to shared data. It also affects the design of the operating system kernel.

Scheduling Criteria:

1. **CPU utilization:** It can range from 0-100%. In a real system, it ranges should range from 40-90%.

2. **Throughput:** Number of processes that are completed per unit time.

3. **Turnaround time:** How long a process takes to execute. It is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU, and doing I/O

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

4. **Waiting time:** It is the sum of the periods spent waiting in the ready queue.

5. **Response time:** Time from the submission of a request until the first response is produced. It is desirable to maximize CPU utilization and Throughput and minimize Turnaround time, waiting time and Response time.

NON-PREEMPTIVE SCHEDULING ALGORITHMS:

1. FCFS

2. SJF

3. PRIORITY

1. FCFS (First-Come, First-Served):

- It is the simplest algorithm and NON-PREEMPTIVE.
- The process that requests the CPU first is allocated to the CPU first.
- The implementation is easily managed by queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue
- The average waiting time, however, is long. It is not minimal and may vary substantially if the process's CPU burst time varies greatly.
- This algorithm is particularly troublesome for time-sharing systems.

2. SJF (Shortest Job First):

- This algorithm associates with each process the length of the process's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS is used to break the tie.
- It is also called shortest next CPU burst algorithm or shortest remaining time first scheduling.
- It is provably optimal; in that it gives the minimum average waiting time for a given set of processes.
- The real difficulty with SJF knows the length of the next CPU request.
- It can be either PREEMPTIVE (SRTF- Shortest Remaining Time First) or NON-PREEMPTIVE.

3. PRIORITY SCHEDULING:

- The SJF is a special case of priority scheduling.
- In priority scheduling algorithm, a priority is associated with each process, and the CPU is allocated to the process with the highest priority.
- It can be either PREEMPTIVE or NON-PREEMPTIVE
- A major problem with priority scheduling algorithms is indefinite blocking, or

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

starvation.

- A solution to starvation is AGING. It is a technique of gradually increasing the priority of process that wait in the system for long time.

Lab Assignments to complete in this session

1. Consider the set of 6 processes arrived at zero instance and burst time are 7, 5, 3, 1, 2, 1. If the CPU scheduling policy is First Come First Serve, calculate the average waiting time.
2. Consider the set of 3 processes arrived at zero instance and burst time are 9, 4, 9 respectively and priority 2, 1 and 3 respectively. If the CPU scheduling policy is SJF, calculate the average waiting time and average turnaround time.
3. Consider the set of 3 processes arrived at zero instance and burst time are 9, 4, 9 respectively and priority 2, 1 and 3 respectively. If the CPU scheduling policy is Priority Scheduling, calculate the average waiting time and average turnaround time.

Expected Output:

Enter the no. of process: 5

Enter the arrival time for process P1: 0

Enter the arrival time for process P2: 1

Enter the arrival time for process P3: 3

Enter the arrival time for process P4: 9

Enter the arrival time for process P5: 12

Enter the burst time for process P1: 3

Enter the burst time for process P2: 5

Enter the burst time for process P3: 2

Enter the burst time for process P4: 5

Enter the burst time for process P5: 5

Inputs given by the user are:

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

==

Process	AT	BT
P1	0	3
P2	1	5
P3	3	2
P4	9	5
P5	12	5

==

Gant Chart is as follows:

0->P1->3->P2->8->P3->10->P4->15->P5->20

................................TABLE................................

Process	AT	BT	FT	TAT	WT
P1	0	3	3	3.000000	0.000000
P2	1	5	8	7.000000	2.000000
P3	3	2	10	7.000000	5.000000
P4	9	5	15	6.000000	1.000000
P5	12	5	20	8.000000	3.000000

Average Turn Around Time: 6.200000

Average Waiting Time: 2.200000

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

```
#60009210116
#Divyesh khunt
import numpy as np
from statistics import mean
```

FCFS

```
n = int(input("Enter number of elemets: "))
a=[]
b=[]
index=[]
for i in range(n):
    index.append(i+1)
    print(i+1,">>")
    x=int(input("Enter the arriving time: "))
    a.append(x)
    y=int(input("Enter the burst time  : "))
    b.append(y)

print("index\tArriving time\tBurst time\n ")

for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\n')
```

```
Enter number of elemets: 3
1 >>
Enter the arriving time: 4
Enter the burst time  : 24
2 >>
Enter the arriving time: 3
Enter the burst time  : 7
3 >>
Enter the arriving time: 4
Enter the burst time  : 9
index    Arriving time   Burst time

1         4                24

2         3                7

3         4                9
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

```python
#arranging
print("after arranging")
for i in range (n):
    for j in range(i + 1, n):
        if(a[i] > a[j]):
            temp = a[i]
            a[i] = a[j]
            a[j] = temp

            temp = b[i]
            b[i] = b[j]
            b[j] = temp

            temp = index[i]
            index[i] = index[j]
            index[j] = temp

print("index\tArriving time\tBurst time\n ")
for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\n')
arr=a
```

```
after arranging
index    Arriving time    Burst time

2           3                7

1           4                24

3           4                9
```

```python
[13] #TAT
tat=[]
sum=0
for i in range(n):
    a=int(sum+b[i])
    tat.append(a)
    sum=a
print("The trunaround time is:",tat)

#WT
wt=[0]
for i in range(1,n):
    a=b[i-1] + wt[i-1]
    wt.append(a)
print("waiting time is" ,wt)

The trunaround time is: [7, 31, 40]
waiting time is [0, 7, 31]
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

```
[36]  print("The respective waiting and turnaround time is shown in table :")
      print("index\tArriving time\tBurst time\tWaiting Time\tTurnaround Time")
      for i in range(n):
          print(index[i] ,'\t' ,arr[i], '\t\t', b[i],'\t\t',wt[i],'\t\t',tat[i],'\t')
      print("\n Average waiting time is ",mean(tat) )
      print(" Average waiting time is ",mean(wt) )
```

```
      The respective waiting and turnaround time is shown in table :
      index   Arriving time   Burst time      Waiting Time    Turnaround Time
      1       0               9               0               9
      3       0               6               9               15
      4       0               7               15              22
      2       0               4               22              26

       Average waiting time is  18
       Average waiting time is  11.5
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

SJF

```python
n = int(input("Enter number of elemets: "))
a=[]
b=[]
index=[]
for i in range(n):
    index.append(i+1)
    print(i+1,">>")
    x=int(input("Enter the arriving time: "))
    a.append(x)
    y=int(input("Enter the burst time  : "))
    b.append(y)

print("index\tArriving time\tBurst time\n ")
for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\n')
```

```
Enter number of elemets: 3
1 >>
Enter the arriving time: 4
Enter the burst time  : 24
2 >>
Enter the arriving time: 6
Enter the burst time  : 3
3 >>
Enter the arriving time: 0
Enter the burst time  : 9
index   Arriving time   Burst time

1       4               24

2       6               3

3       0               9
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

```python
#arranging according to burst time
print("after arranging")
for i in range (n):
    for j in range(i + 1, n):
        if(b[i] > b[j]):
            temp = a[i]
            a[i] = a[j]
            a[j] = temp

            temp = b[i]
            b[i] = b[j]
            b[j] = temp

            temp = index[i]
            index[i] = index[j]
            index[j] = temp

print("index\tArriving time\tBurst time\n ")
for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\n')
arr=a
```

```
after arranging
index    Arriving time    Burst time

2        6                3

3        0                9

1        4                24
```

```python
#TAT
tat=[]
sum=0
for i in range(n):
    a=int(sum+b[i])
    tat.append(a)
    sum=a
print("The trunaround time is:",tat)

#WT
wt=[0]
for i in range(1,n):
    a=b[i-1] + wt[i-1]
    wt.append(a)
print("waiting time is" ,wt)
```

```
The trunaround time is: [3, 12, 36]
waiting time is [0, 3, 12]
```

```
print("The respective waiting and turnaround time is shown in table :")
print("index\tArriving time\tBurst time\tWaiting Time\tTurnaround Time")
for i in range(n):
    print(index[i] ,'\t' ,arr[i], '\t\t', b[i],'\t\t',wt[i],'\t\t',tat[i],'\t')
print("average waiting time is ",mean(tat) )
print("average waiting time is ",mean(wt) )
```

```
The respective waiting and turnaround time is shown in table :
index    Arriving time    Burst time      Waiting Time     Turnaround Time
2        6                3               0                3
3        0                9               3                12
1        4                24              12               36
average waiting time is  17
average waiting time is  5
```

Priority Scheduling

```
n = int(input("Enter number of elemets: "))
a=[]
b=[]
index=[]
pri=[]
for i in range(n):
    index.append(i+1)
    print(i+1,">>")
    x=int(input("Enter the arriving time: "))
    a.append(x)
    y=int(input("Enter the burst time  : "))
    b.append(y)
    p=int(input("Enter the priority : "))
    pri.append(p)
```

```
Enter number of elemets: 4
1 >>
Enter the arriving time: 0
Enter the burst time  : 9
Enter the priority : 2
2 >>
Enter the arriving time: 0
Enter the burst time  : 4
Enter the priority : 1
3 >>
Enter the arriving time: 0
Enter the burst time  : 3
Enter the priority : 4
4 >>
Enter the arriving time: 0
Enter the burst time  : 7
Enter the priority : 7
```

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

A.Y.: 2022-23   **Sub:** System Fundamentals

```
print("index\tArriving time\tBurst time\tPriority ")
for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\t\t',pri[i],'\n')
```

```
index    Arriving time    Burst time      Priority
1        0                9               2

2        0                4               1

3        0                3               4

4        0                7               7
```

```
#arranging
print("after arranging")
for i in range (n):
    for j in range(i + 1, n):
        if(pri[i] > pri[j]):
            temp = pri[i]
            pri[i] = pri[j]
            pri[j] = temp

            temp = a[i]
            a[i] = a[j]
            a[j] = temp

            temp = b[i]
            b[i] = b[j]
            b[j] = temp

            temp = index[i]
            index[i] = index[j]
            index[j] = temp

print("index\tArriving time\tBurst time\tPriority ")
for i in range(n):
    print(index[i] ,'\t' ,a[i], '\t\t', b[i],'\t\t',pri[i],'\n')
arr=a
```

```
after arranging
index    Arriving time    Burst time      Priority
2        0                4               1

1        0                9               2

3        0                3               4

4        0                7               7
```

A.Y.: 2022-23

SHRI VILEPARLE KELAVANI MANDAL'S
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC ACCREDITED with "A" GRADE (CGPA : 3.18)

Sub: System Fundamentals

```
[43]  #TAT
      tat=[]
      sum=0
      for i in range(n):
          a=int(sum+b[i])
          tat.append(a)
          sum=a
      print("The trunaround time is:",tat)


      #WT
      wt=[0]
      for i in range(1,n):
          a=b[i-1] + wt[i-1]
          wt.append(a)
      print("waiting time is" ,wt)
```

```
The trunaround time is: [4, 13, 16, 23]
waiting time is [0, 4, 13, 16]
```

```
print("The respective waiting and turnaround time is shown in table :")
print("index\tArriving time\tBurst time\tWaiting Time\tTurnaround Time\tPriority")
for i in range(n):
    print(index[i] ,'\t' ,arr[i], '\t\t', b[i],'\t\t',wt[i],'\t\t',tat[i],'\t\t',pri[i],'\n')
print("\n Average waiting time is ",mean(tat) )
print(" Average waiting time is ",mean(wt) )
```

```
The respective waiting and turnaround time is shown in table :
index   Arriving time   Burst time      Waiting Time    Turnaround Time Priority
2       0               4               0               4               1

1       0               9               4               13              2

3       0               3               13              16              4

4       0               7               16              23              7


 Average waiting time is  14
 Average waiting time is  8.25
```