

# OS Project

## **Modules :**

- Implementation of 7 scheduling algorithms :
  - FCFS
  - SPN
  - SRTF
  - RR
  - Lottery
  - HRRN
  - MLFQ
- Handling simple Memory management
- Handling some I/O queues :
  - Network
  - Disk
  - Memory
  - Interrupt
- Calculation of different evaluation values :
  - Utilization
  - Waiting time
  - Turn around
- Implementation of mid-term scheduler

## ***Input File :***

There are two options for input file :

- Raw Input :

```
# Memory: TotalMemory
```

```
Memory: 1024
```

```
Algorithms:
```

```
# AlgorithmName TimeQuantum Start
```

```
SPN - 0
```

```
RR 2 12
```

```
Processes:
```

```
# ProcessName Start Memory NumOfEvents(including CPU & IO)
```

```
process1: 4.35 65 2
```

```
# CPU time
```

```
CPU 5.1
```

```
# IO type priority time
```

```
IO RAM 5 3.76
```

```
process2: 7.25 32 3
```

```
CPU 12.2
```

```
IO DISK 2 4.76
```

```
CPU 2.01
```

- XML Input :

```
<scheduler memory='1024'>
  <algorithms>
    <algorithm start='2'>SPN</algorithm>
    <algorithm start='12' quantum='2'>RR</algorithm>
  </algorithms>
  <processes>
    <process start='4.35' memory='65'>
      <cpu>5.1</cpu>
      <io name='RAM' priority='5'>3.76</io>
    </process>
    <process start='7.25' memory='32'>
      <cpu>12.2</cpu>
      <io name='DISK' priority='2'>4.76</io>
      <cpu>2.01</cpu>
    </process>
  </processes>
</scheduler>
```

Algorithms :

- In the implementation of the algorithms you are not supposed to use the future data, so in SPN or HRRN you must use estimation algorithms according to past.

Memory management :

- You will have a fixed amount memory and each process will use a fixed amount, so if the memory is full either the mid-term scheduler must throw out some of the processes.

IO Implementation :

- In IO implementation you must use priority queues.

Evaluation values :

- When a process ends you must print all its evaluation values; At the end of the program you must print all average evaluation values of the system.

Mid-term scheduler :

- The algorithm & frequency of mid-term scheduler is according to your own decision.

Sample Output :

- A simple output with no interactive value should be similar to the sample below :

```
000.000: System starting up
002.000: Algorithm FCFS is in use
004.450: Process 001 was created
004.450: Process 001 entered ready queue
004.450: Process 001 started using CPU
007.670: Process 002 was created
007.670: Process 002 entered ready queue
012.567: Process 001 finished using CPU
012.567: Process 001 entered RAM queue
012.567: Process 002 started using CPU
012.567: Process 001 started using RAM
015.274: Algorithm RR is in use
017.160: Process 003 was created
017.160: Process 003 entered ready queue
017.160: Process 002 slot ended
017.160: Process 002 entered ready queue
017.160: Process 003 started using CPU
021.563: Process 001 finished using RAM
021.563: Process 001 ended : W=xxx T=xxx U=xxx
024.632: Process 003 finished using CPU
024.632: Process 003 ended : W=xxx T=xxx U=xxx
024.632: Process 002 started using CPU
027.141: Process 002 finished using CPU
027.141: Process 002 ended : W=xxx T=xxx U=xxx
027.141: System shutting down
```