

# Process Scheduling System XAVIER INSTITUTE OF ENGINEERING

TEAM MEMBERS: ROHAN TALELE & VARUN SINGH



# Abstract

- We propose a java based application with features designed to calculate the different parameters related to process scheduling.
- This application will provide the user with the choice to select the different algorithms used to schedule processes to the CPU.
- At the end the results will be shown in a tabular manner for easier comparison between the distinct algorithms.
- The results will be accompanied by message highlighting the most efficient algorithm to be used for that particular set of inputs.

# Introduction

### Why do we need scheduling?

- A typical process involves both I/O time and CPU time. In a Uniprogramming system like MS-DOS, time spent waiting for I/O is wasted and CPU is free during this time. In multiprogramming systems, one process can use CPU while another is waiting for I/O. This is possible only with process scheduling.
- This application is designed to schedule the different processes according to the scheduling techniques.
- The main objective of this application is to calculate and provide the parameters like waiting time and turnaround time for each individual process to the user.
- The purpose of this mini-project is to provide the user with some knowledge about how the CPU and the

# Methodology

- This GUI java application is named as ProSio. The Pro stands for process and the Sio comes from a popular company manufacturing calculators known as Casio.
- ProsSio will provide the user with the different parameters calculated for each process along with a message emphasizing the most efficient algorithms to be used for that particular set of inputs.
- The comparison for the most efficient algorithm will be done on the basis of average waiting time.
- However ProSio will not simulate and display a Gnatt chart for each algorithm dynamically.



- ProSio is entirely created on the NetBeans IDE for Java. NetBeans features of Drag and Drop has been used extensively for the creation of the GUI.
- ProSio is made up of four classes: Main, App, Cal, Result.
- Main is used as an initializer for the application.
- App provides the Main Screen.
- Cal enables the user to select the different algorithms and provide their inputs.
- Result will display the results in a clear tabular manner for easier comparison accompanied with a message showcasing the most efficient algorithms to be used.

# Results

Accessing through the Main Screen.

|   | - D X   |  |  |  |  |  |  |  |  |
|---|---|--|--|--|--|--|--|--|--|
| ProSio  |   |  |  |  |  |  |  |  |  |
| Have you ever felt annoyed, betrayed, frustrated or just plain stupid when you were being taught the a seemed you were the only one who just couldn't get the right answers among all your friends.  Well those days are behind of us | lifferent process scheduling techniques and it  |  |  |  |  |  |  |  |  |
| We present to you ProSio  ProSio is exclusively created for the purpose of making your second year of engineering just a little bit Matters.  | more comfortable, and as we know every bit  |  |  |  |  |  |  |  |  |
| ProSio will calculate all of the parameters required during process scheduling and display it in a clear a different algorithms along with a message highlighting the most effecient  |   |  |  |  |  |  |  |  |  |
| Click here to start   | For more information regarding ProSio or its workings, Contact<br>Varun Singh: 99691 76748 Rohan Talele:73030 12567<br>Batch of 2021, Department of Computer Engineering<br>Xavier Institute of Engineering |  |  |  |  |  |  |  |  |

Selecting the scheduling algorithms

|                 | Choose your algo | ⊕ FCFS  ⊕ SJF(NP)  Start | ● SJF(PRE) ● PRORITY  Clear | ⊕ ROUND ROBIN  Exit |  |
|-----------------|------------------|--------------------------|-----------------------------|---------------------|--|
| Process Details | Time Quar        | turn is 5                |                             |                     |  |
| Process No      |                  | Process No Arrival Time  |                             | Priority            |  |
| Arrival Time    | 1<br>2<br>3<br>4 | 2<br> 3<br> 1<br> 5      | 3 4<br>4 3<br>6 2<br>1 1    |                     |  |
| Burst Time      |                  |                          |                             |                     |  |
| Priority        |                  |                          |                             |                     |  |
|                 |                  |                          |                             |                     |  |
| Add Process     | Calculate        |                          |                             |                     |  |
| Reset Table     |                  |                          |                             |                     |  |
|                 |                  |                          |                             |                     |  |

Displaying the calculated parameters

|                     |              |                 |                 |                  |                   |                                     |                   |              |            |                 |                  | -            |  |
|---------------------|--------------|-----------------|-----------------|------------------|-------------------|-------------------------------------|-------------------|--------------|------------|-----------------|------------------|--------------|--|
| FCFS                |              |                 |                 |                  |                   | PRIORITY                            |                   |              |            |                 |                  |              |  |
| Process No          | Arrival Time | Burst Time      | Completion Time | Turn Around Time | Waiting Time      | Process No                          | Priority          | Arrival Time | Burst Time | Completion Time | Turn around Time | Waiting Time |  |
| 3                   | 1            | 6               | 7<br>10         | 6<br>8           | 0 5               | 1 2                                 | 4 3               | 2            | 3 4        | 5               | 3<br>6           | 0 2          |  |
| 2                   | 3            | 4               | 14<br>15        | 11<br>10         | 7                 | 3                                   | 2                 | ĭ            | 6          | 14<br>15        | 13<br>10         | 7            |  |
| 4                   | 5            | 1               | 15              | 10               | 9                 | 4                                   | 1                 | 5            | 1          | 15              | 10               | 9            |  |
| werage Turn around  |              |                 |                 |                  |                   | Average Turn ard<br>Average Waiting |                   |              |            |                 |                  |              |  |
| SJF(PRE)            |              |                 |                 |                  |                   | SJF(NP)                             |                   |              |            |                 |                  |              |  |
| Process No          | Arrival Time | Burst Time<br>6 | Completion Time | Turn around Time | Waiting Time<br>8 | Process No<br>3                     | Arrival Time      | Burst Time   | Compl      | etion Time Tun  | n around Time    | Waiting Time |  |
| 1                   | 2            | 3               | 15<br>5         | 3                | Ô                 | 1                                   | 2                 | 3            | 11         | 9               |                  | 6            |  |
| 2                   | 3            | 4               | 10<br>6         | 7                | 3                 | 2                                   | 3                 | 4            | 15<br>8    | 12              |                  | 8            |  |
| Average Turn around |              |                 |                 |                  |                   | Average Turn an                     |                   |              |            |                 |                  |              |  |
|                     |              |                 |                 |                  |                   | ROUND ROBIN                         |                   |              |            |                 |                  |              |  |
|                     |              |                 |                 |                  |                   | Process No                          | Arrival Time      | Burst Time   |            |                 | n around Time    | Waiting Time |  |
| гие м               | OCT EF       | FICIENT         | ALGORI          | THM TO           | HCE IC            | 3                                   | 1 2               | 6            | 15<br>9    | 14<br>7         |                  | 8            |  |
| THE M               | IUSI EF      | PICIENI         | ALGUM           | IIIM IU          | USE IS            | 2                                   | 3                 | 4            | 13<br>14   | 10              |                  | 6            |  |
|                     |              |                 |                 |                  |                   | 4                                   | 5                 | i            | 14         | 9               |                  | 8            |  |
| SJF PR              | REEMPT       | IVE             |                 |                  |                   |                                     | ound Time is 10.0 |              |            |                 |                  |              |  |
| Back                | k            |                 | Exit            |                  |                   | Average Waiting                     | Time is 6.5       |              |            |                 |                  |              |  |

# Conclusion

- Process scheduling is an integral part of any Operating system. We can't afford to keep our CPU idle indefinitely in the hopes of a process to be executed in any order.
- With these algorithms we get the most efficiency, our throughput increases, latency of the CPU decreases and overall we get a much better performance.
- The industry implementation of schedulers have employed variations based on flavors and hybrids of the standard algorithms. As with most NP-hard problems, in designing the standard algorithms, scholars did not concern themselves with security matters.

# References

- https://www.geeksforgeeks.org/
- https://www.thejavaprogrammer.com
- http://www.ijesi.org/papers/Vol(6)11/Version
   -1/L0611016771.pdf
- https://www.cs.uic.edu/~jbell/CourseNotes/ OperatingSystems/5\_CPU\_Scheduling.html

# Contact

Rohan Talele: 73030 12567 rntalele@gmail.com

Varun Singh: 9969176748 varunsingh3000@gmail.com