.

KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

**(APPROVED BY AICTE, NEW DELHI)**



*Course Activity Report*

*on*

***OPERATING SYSTEMS (18CS42)***

*Submitted in the partial fulfilment for the academic requirement**of*

**4TH *Semester B.E.***

***In***

***Computer Science Engineering***

***Submitted by***

**Shambhavi Shirgurkar 2GI20CS142**

**Sanat S 2GI20CS113**

**Sanket Patil 2GI20CS189**

**Sahana Tavari 2GI20CS133**

**Under Guidance of,**

**Prof. Ranjana Battur**

**Asst. Prof., Dept. of CSE**

**Academic Year 2021-2022 (Even semester)**

**CERTIFICATE**



This is to certify that the Seminar entitled “” is a bonafide record of the Seminar work done by **Shambhavi Shirgurkar, Sanat K S, Sanket Patil, Sahana Tavari having USN 2GI20CS135, 2GI20CS128, 2GI20CS132, 2GI20CS117** under my supervision and guidance, in partial fulfilment of the requirements for the Outcome Based Education Paradigm in CSE from Gogte Institute of Technology for the academic year **2020-2021** .

Prof. Ranjana Battur

Dept of CSE

Dr.V S Rajpurohit

Professor and Head

Dept of CSE

Place: KLS Gogte Institute of Technology

Belgaum.

**Team Members Details:**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **USN** | **Student Name** |
| 1 | 2GI20CS135 | Shambhavi Shirgurkar |
| 2 | 2GI20CS128 | Sanat S |
| 3 | 2GI20CS132 | Sanket Patil |
| 4 | 2GI20CS117 | Sahana Tavari |

**Rubrics for evaluation of Course Project:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.No | Batch No. : | | | | | |
| 1. | Project Title: | Marks Range | USN | | | |
| 2GI20CS135 | 2GI20CS128 | 2GI20CS132 | 2GI20CS117 |
| 2. | Problem statement (PO2) | 0-1 |  |  |  |  |
| 3. | Need Analysis, Variables involved (PO1,PO2) | 0-2 |  |  |  |  |
| 4. | Alternate solutions to solve the problem(PO3) | 0-3 |  |  |  |  |
| 5. | Comparison between the solutions and reason for selecting the final solution(PO1,PO3,PO4) | 0-4 |  |  |  |  |
| 6. | Working model of the final solution (PO3,PO12) | 0-5 |  |  |  |  |
| 7. | Report and Oral presentation skill (PO9,PO10) | 0-5 |  |  |  |  |
|  | Total | 20 |  |  |  |  |

**Signature of Staff in Charge.**

**ABSTRACT**

The **OS Simulator** provides multithreaded processes and supports multiple CPU simulations. It works with the cpu simulator and manages multiple processes and virtual memory using different process scheduling mechanisms.

The OS Simulator is designed to support two main aspects of a computer system’s resource management: process management and memory management. Once a compiled code is loaded in CPU Simulator’s memory, its image is also available to the OS Simulator. It is then possible to create multiple instances of the program images as separate processes. The OS Simulator displays the running processes, the ready processes and the waiting processes. Each process is assigned a separate (PCB) that contains information on the process state. This information is displayed in a separate window.

The process scheduler includes various selectable scheduling policies that include priority-based, pre-emptive and round-robin scheduling with variable time quanta. The OS is able to carry out context switching which can be visually enhanced by slowing down or suspending the progress at some key stage to enable the students to study the states of CPU registers, stack, cache, pipeline and the PCB contents.

The simulator incorporates an input output console device, incorporating a virtual keyboard, and is used to display text and accept input data.

The OS simulator supports dynamic library simulation which is supported by the appropriate language constructs in the teaching language. The benefits of sharing code between multiple processes are visually demonstrated.

**Table Of Contents:**

**1.problem statement**

**2.solution of problem statement**

**3.Screenshots of the solution**

**4.conclusion**

**OSSIM ARCHITECTURE OVERVIEW:**

**OSSIM (Open Source Security Information Management)** is an open source security event and management system, integrating a selection of tools designed to aid [network administrators](https://en.wikipedia.org/wiki/Network_administrator) in [computer security](https://en.wikipedia.org/wiki/Computer_security), [intrusion detection](https://en.wikipedia.org/wiki/Intrusion_detection) and [prevention](https://en.wikipedia.org/wiki/Intrusion_prevention).

The project began in 2003 as a collaboration between Dominique Karg, Julio Casal  and later Alberto Román. In 2008 it became the basis for their company [AlienVault](https://en.wikipedia.org/wiki/AlienVault).[]](https://en.wikipedia.org/wiki/OSSIM#cite_note-4) Following the acquisition of the [Eureka](https://en.wikipedia.org/wiki/Eureka_(organisation)) project label and completion of [R&D](https://en.wikipedia.org/wiki/R%26D), AlienVault began selling a commercial derivative of OSSIM ('AlienVault Unified Security Management'). AlienVault was acquired by [AT&T Communications](https://en.wikipedia.org/wiki/AT%26T_Communications) and renamed AT&T Cybersecurity in 2019.

OSSIM has had four major-version releases[]](https://en.wikipedia.org/wiki/OSSIM#cite_note-6) since its creation and is on a 5.x.x version numbering. An [information visualization](https://en.wikipedia.org/wiki/Information_visualization) of the contributions to the source code for OSSIM was published at [8 years of OSSIM](https://www.alienvault.com/blogs/labs-research/8-years-of-ossim). The project has approximately 7.4 million lines of code. The current version of OSSIM is 5.7.5 and was released on September 16, 2019. Information about this release and past versions can be found [here](https://success.alienvault.com/s/topic/0TO0Z000000oRSEWA2/appliance-product-announcements)

As a [SIEM](https://en.wikipedia.org/wiki/SIEM) system, OSSIM is intended to give security analysts and administrators a more complete view of all the security-related aspects of their system, by combining [log management](https://en.wikipedia.org/wiki/Log_management_and_intelligence) which can be extended with plugins and [asset management and discovery](https://en.wikipedia.org/wiki/IT_asset_management) with information from dedicated information security controls and detection systems. This information is then correlated together to create contexts to the information not visible from one piece alone. Alarm and availability views along with reporting capabilities are provided to enhance the capabilities of the tool and its utility to the security and systems engineers.

OSSIM performs these functions using other well-known[open-source software](https://en.wikipedia.org/wiki/Open-source_software) security components, unifying them under a single browser-based user interface. The interface provides graphical analysis tools for information collected from the underlying open source.

**Scheduling Algorithms**

There are various algorithms which are used by the Operating System to schedule the processes on the processor in an efficient way.

The Purpose of a Scheduling algorithm

1. Maximum CPU utilization
2. Fare allocation of CPU
3. Maximum throughput
4. Minimum turnaround time
5. Minimum waiting time
6. Minimum response time

There are the following algorithms which can be used to schedule the jobs.

1. First Come First Serve

It is the simplest algorithm to implement. The process with the minimal arrival time will get the CPU first. The lesser the arrival time, the sooner will the process gets the CPU. It is the non-preemtive type of scheduling.

2. Round Robin

In the Round Robin scheduling algorithm, the OS defines a time quantum (slice). All the processes will get executed in the cyclic way. Each of the process will get the CPU for a small amount of time (called time quantum) and then get back to the ready queue to wait for its next turn. It is a preemptive type of scheduling.

3. Shortest Job First

The job with the shortest burst time will get the CPU first. The lesser the burst time, the sooner will the process get the CPU. It is the non-preemptive type of scheduling.

4. Shortest remaining time first

It is the preemptive form of SJF. In this algorithm, the OS schedules the Job according to the remaining time of the execution.

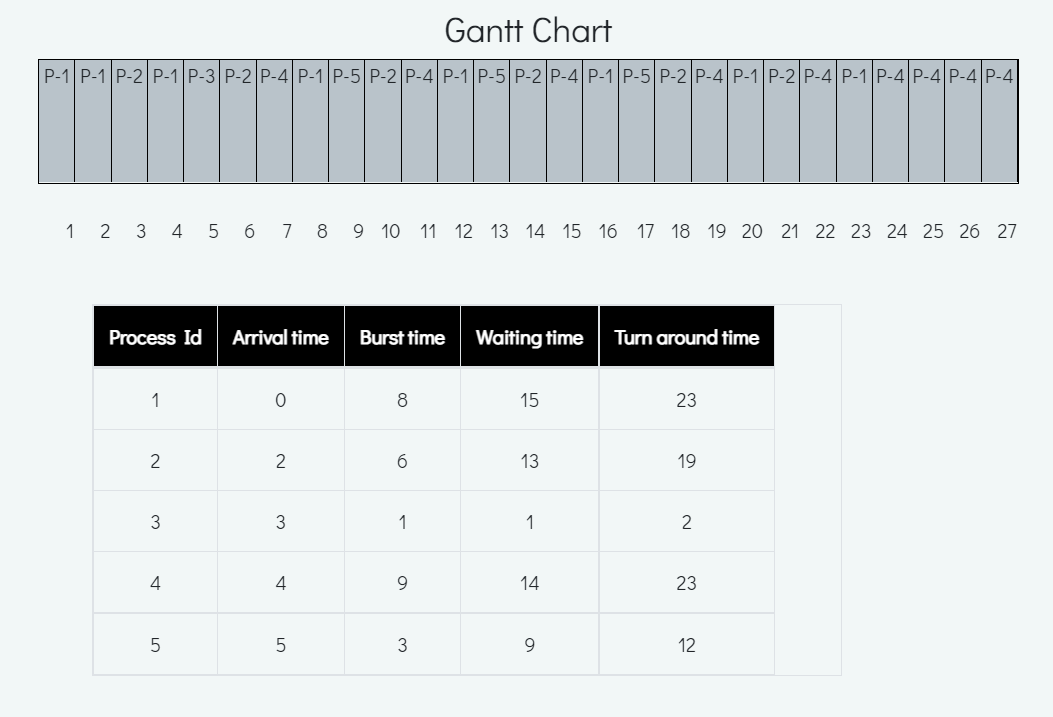
5. Priority based scheduling

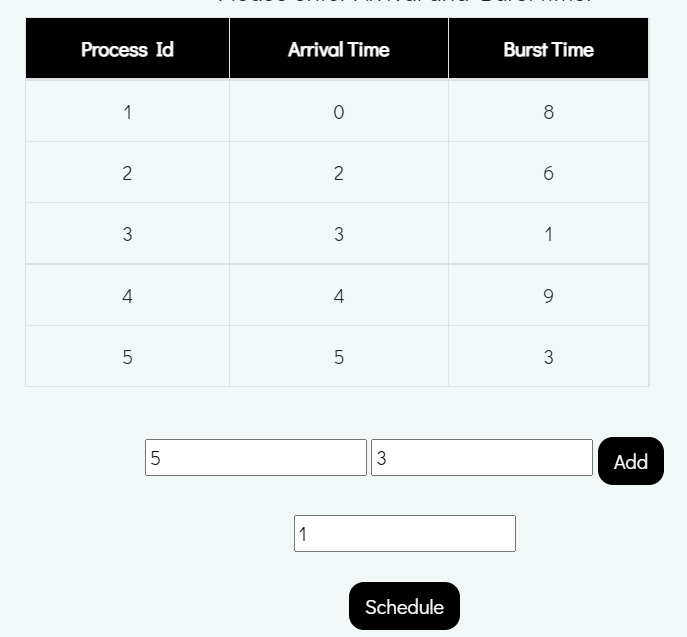
In this algorithm, the priority will be assigned to each of the processes. The higher the priority, the sooner will the process get the CPU. If the priority of the two processes is same then they will be scheduled according to their arrival time.

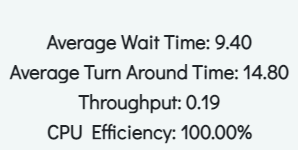
**PROBLEM STATEMENT:**

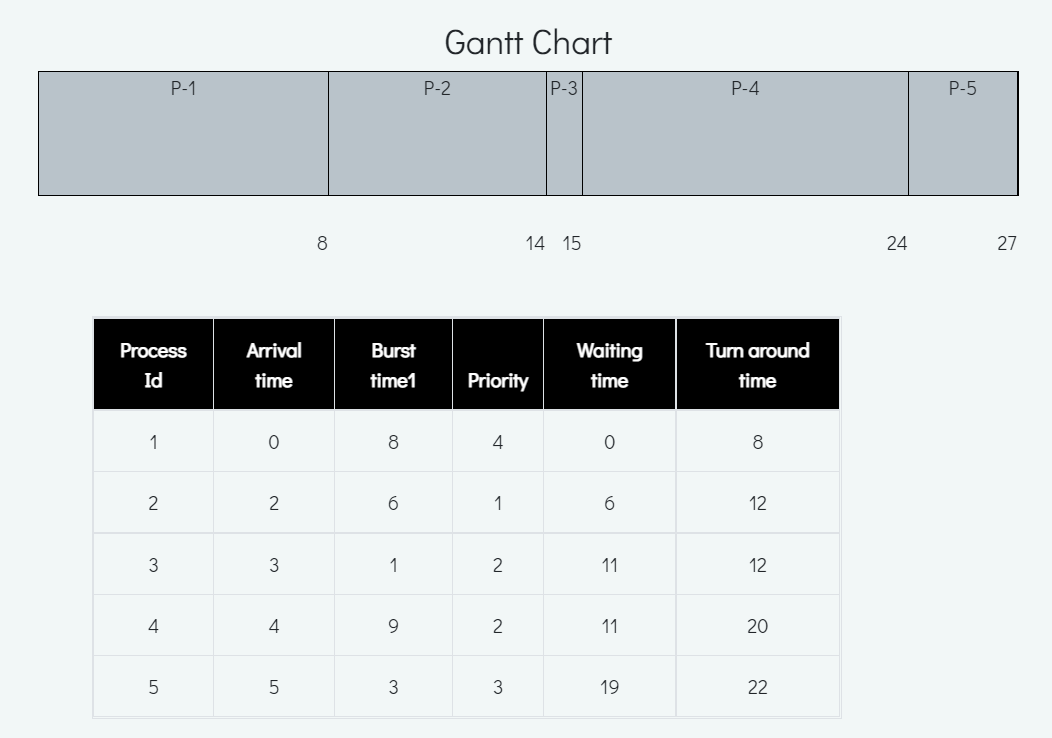
**SIMULATE THE NON-PRE-EMPTIVE PRIORITY AND ROUND ROBIN (1ms QUANTUM) SCHEDULING ALGORITHM USING OSSIM SIMULATOR FOR THE DETAILS GIVEN BELOW:**

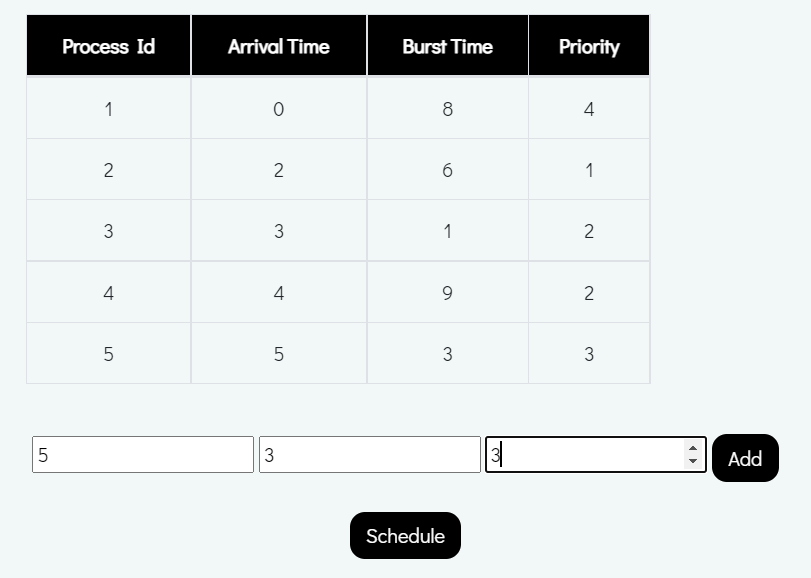
|  |  |  |  |
| --- | --- | --- | --- |
| PROCESS | ARRIVAL TIME (in ms) | BURST TIME (in ms) | PRIORITY (LOWER NO. LOWER PRIORITY) |
| P0 | 0 | 8 | 4 |
| P1 | 2 | 6 | 1 |
| P2 | 3 | 1 | 2 |
| P3 | 4 | 9 | 2 |
| P4 | 5 | 3 | 3 |

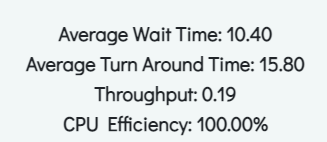












**APPLICATIONS OF OSSIM:**

1.OSSIM provides the capability to process images from satellite and aerial cameras and transform them into accurate image maps associated with three-dimensional positions on the Earth.

2. The library is capable of concurrently handling many large image files of arbitrary pixel depth. It has been designed from the ground up to support high performance parallel processing, map projections, three-dimensional models and scientific applications.

3. OSSIM has been designed to support dynamic linking of image chains. Images can be opened in their native format and sampled on demand to fill a viewport on the screen or to generate an output product.

**CONCLUSION:**