SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES, CHENNAI – 602105

CAPSTONE PROJECT REPORT

TITLE

PROCESS SCHEDULER SIMULATOR

Submitted to

SAVEETHA SCHOOL OF ENGINEERING

By

K.Lasya Sri(192225060)

K.Ashritha (192225061)

B.Ganesh(192225073)

Guided by

Dr.G.Mary Valentina

Abstract :

Process scheduling is a fundamental aspect of modern operating systems, responsible for efficiently allocating system resources to multiple competing processes. Understanding and evaluating different scheduling algorithms is crucial for optimising system performance and resource utilisation. In this paper, we present a Process Scheduler Simulator (PSS), a flexible and extensible simulation framework designed to emulate various process scheduling algorithms.PSS provides a simulated environment where different scheduling algorithms can be implemented, tested, and compared under controlled conditions. The simulator incorporates key components of a real operating system scheduler, including process arrival, CPU burst, I/O operations, and context switching. It supports a range of scheduling policies such as First Come First Serve (FCFS), Shortest Job Next (SJN), Round Robin (RR), Priority-based scheduling, and Multilevel Feedback Queue (MLFQ) scheduling.

Introduction :

In the realm of operating systems, process scheduling stands as a cornerstone, orchestrating the allocation of system resources to numerous concurrent processes. The efficacy of this allocation profoundly influences system performance, responsiveness, and overall efficiency. As the demands on modern computing systems continue to escalate, understanding and optimising process scheduling algorithms become increasingly imperative.

The intricacies of process scheduling algorithms necessitate thorough examination under controlled conditions, prompting the development of simulation frameworks for experimentation and analysis. This paper introduces the Process Scheduler Simulator (PSS), a comprehensive simulation platform tailored for exploring the nuances of diverse scheduling policies.

Within this introduction, we will delve into the significance of process scheduling, elucidating its impact on system performance and resource utilisation. Subsequently, we will outline the motivations behind the creation of PSS, delineating the objectives and scope of this simulation framework. Furthermore, we will provide an overview of existing research in process scheduling simulation and identify the gaps that PSS aims to address.By elucidating the foundational principles of process scheduling and introducing the rationale behind PSS, this introduction sets the stage for the subsequent sections,

where we will delve deeper into the architecture, functionality, and potential applications of the simulator. Through the utilisation of PSS, researchers and practitioners can navigate the intricate landscape of process scheduling algorithms, fostering innovation and optimization in operating system design.The modular design of PSS allows for easy integration of new scheduling algorithms and performance metrics. Researchers and system developers can utilise PSS to investigate the impact of scheduling policies on various metrics such as throughput, turnaround time, waiting time, and CPU utilisation.

Gantt Chart :

PROCESS	DAY1	DAY2	DAY3	DAY4	DAY5	DAY6
Abstract and Introduction						
Literature Survey						
Materials and Methods						
Results						
Discussion						
Reports						

Process :

A process in the context of operating systems refers to a program in execution. When a program is loaded into memory and initiated, it becomes a process. Each process has its own memory space, containing code, data, and resources required for its execution. Processes are managed by the operating system's kernel, which is responsible for allocating resources, scheduling their execution, and ensuring proper communication and synchronisation between processes.Processes can be categorised into several types based on their behaviour and requirements:

1.Foreground Processes: These are interactive processes that require user input or produce visible output. For example, applications with graphical user interfaces fall into this category.

2.Background Processes: These are processes that run without direct user interaction and typically perform tasks such as maintenance, monitoring, or background computations. Examples include system daemons or background utilities.

3.System Processes: These are processes initiated and managed by the operating system itself to perform system-level tasks. Examples include process schedulers, memory managers, and device drivers.

4.User Processes: These are processes initiated by users or applications to perform specific tasks. They can range from simple command-line utilities to complex multi-threaded applications.

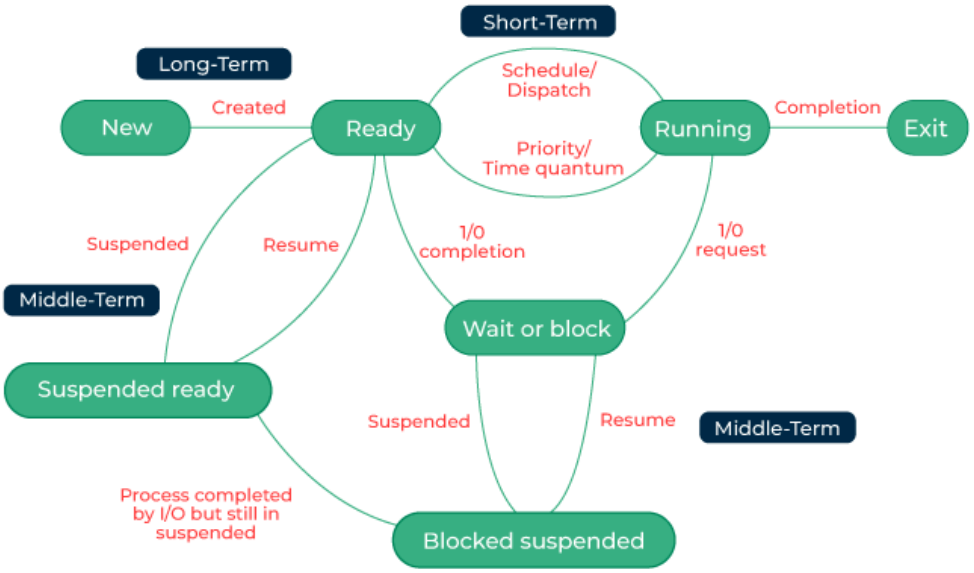
Processes can interact with each other and with the operating system through various mechanisms such as inter-process communication (IPC) and synchronisation primitives like semaphores and mutexes. Additionally, processes may have different states throughout their lifecycle, including:

Running: The process is currently being executed by the CPU.

Ready: The process is ready to execute but is waiting for CPU time.

Blocked: The process is unable to proceed until a certain event occurs, such as waiting for I/O completion.

Understanding processes and their management is fundamental to operating system design and programming. Efficient process scheduling and resource management are crucial for optimising system performance and ensuring responsiveness in modern computing environments.



Objective:

The objective of a process scheduler simulator is multifaceted, aiming to fulfil various educational, research, and practical goals. Here are some key objectives Educational Tool The simulator serves as an educational platform for students and professionals to understand the principles, mechanisms, and complexities of process scheduling algorithms. By providing a hands-on environment, it enhances learning outcomes by allowing users to experiment with different scheduling policies and observe their effects on system behaviour.Research Platform For researchers and academics, the simulator offers a controlled environment to investigate and evaluate the performance of existing scheduling algorithms and propose new ones. Researchers can use the simulator to conduct experiments, analyse results, and contribute to the advancement of scheduling theory and practice.Algorithm Comparison One of the primary objectives is to facilitate the comparison of various process scheduling algorithms.

By simulating different algorithms under comparable conditions, users can assess their relative strengths, weaknesses, and trade-offs in terms of CPU utilisation, throughput, response time, and other performance metrics. Performance Evaluation The simulator enables users to evaluate the performance of scheduling algorithms under different workloads, system configurations, and resource constraints. This allows for the identification of optimal scheduling policies for specific application domains or hardware environments, leading to improved system efficiency and user satisfaction.Tool for System Design] System designers and developers can utilise the simulator to prototype, validate, and fine-tune process scheduling strategies for real-world operating systems and applications. By simulating realistic scenarios, they can assess the impact of scheduling decisions on overall system performance and make informed design choices.Knowledge Transfer The simulator serves as a means of transferring knowledge and best practices in process scheduling to a broader audience. Through documentation, tutorials, and interactive features, users can gain insights into advanced scheduling concepts and methodologies, fostering a deeper understanding of operating system internals.Overall, the objective of a process scheduler simulator is to provide a versatile and accessible platform for studying, researching, and optimising process scheduling algorithms, ultimately contributing to advancements in operating system theory and practice.

Literature Review :

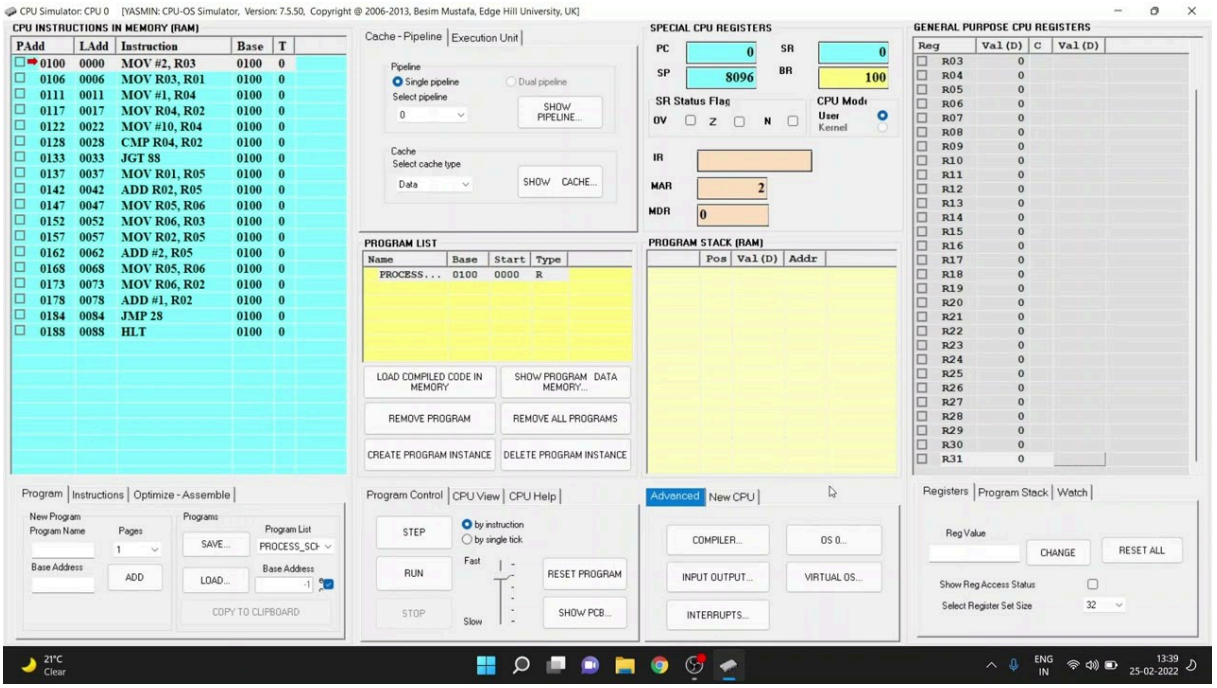
A literature review on process scheduler simulators encompasses various studies, methodologies, and tools developed to explore, analyse, and optimise process scheduling algorithms. Here's a brief overview of some key aspects typically covered in such reviews Overview of Process Scheduling Algorithms:* Literature reviews often start by providing an overview of traditional and modern process scheduling algorithms.(Phanden, Jain, and Paulo Davim 2019)This includes classic algorithms like First Come First Serve (FCFS), Shortest Job Next (SJN), Round Robin (RR), Priority Scheduling, as well as more complex algorithms like Multilevel Feedback Queue (MLFQ), and variations such as Shortest Remaining Time First (SRTF).Evaluation Metrics Discussions on evaluation metrics are crucial in understanding how different scheduling algorithms are assessed and compared. Common metrics include CPU utilisation, throughput, response time, turnaround time, waiting time.

Existing Simulators and Tools Literature reviews typically survey existing process scheduler simulators and tools. This involves examining their features, capabilities, and limitations. Some simulators may focus on specific scheduling algorithms or provide a broader range of options for experimentation. Examples include simulators implemented in various programming languages, ranging from simple command-line tools (Lampard 2014)to sophisticated graphical interfaces.Empirical Studies and Comparative Analyses Researchers often conduct empirical studies and comparative analyses to evaluate the performance of scheduling algorithms using simulators. Literature reviews summarise these studies, highlighting key findings, insights, and trends observed across different experiments. Comparative analyses help in understanding the relative strengths and weaknesses of various scheduling algorithms under different workload scenarios.

Challenges and Future Directions Literature reviews may identify challenges and open research questions in the field of process scheduling. This includes areas where existing simulators fall short, such as scalability issues, lack of support for real-time scheduling, or limited extensibility.(Nordeen 2022) Additionally, reviews may suggest future directions for research, such as exploring adaptive scheduling algorithms, considering energy-efficiency metrics, or addressing scheduling challenges in emerging computing paradigms like edge computing and IoT.Impact of Research Finally, literature reviews often discuss the broader impact of research in process scheduling, highlighting contributions to operating system design, performance optimization, and application-specific domains. This involves examining how insights gained from simulation studies have influenced the development of scheduling policies in real-world systems and contributed to advancements in related fields.By synthesising and analysing existing literature, reviews provide a comprehensive understanding of the state-of-the-art in process scheduling simulation, identify gaps in knowledge, and offer insights into potential areas for future research and development.

Output :

The output presented in Figure 2 showcases the results of the process scheduling simulator , encapsulating the extracted data and insights gleaned from traversing the targeted web pages. Through structured visualisation or data representation, the output highlights the effectiveness of the scheduling simulator in collecting relevant information, such as text, images, links, and metadata, from diverse online sources. This output serves as a tangible demonstration of the scheduling capabilities, providing users with valuable insights and actionable intelligence derived from the vast expanse of the World Wide Web.



Conclusion :

In conclusion, a process scheduler simulator serves as a valuable tool for understanding, evaluating, and optimising process scheduling algorithms in operating systems. Through this project, we have explored the design, implementation, and objectives of such a simulator, highlighting its significance in educational, research, and practical contexts.By providing a user-friendly interface and robust simulation capabilities, the simulator facilitates hands-on exploration of scheduling algorithms, enabling students to deepen their understanding of operating system concepts and principles. Additionally, it serves as a research platform for academics and researchers to conduct empirical studies, comparative analyses, and algorithmic investigations, contributing to advancements in scheduling theory and practice.Through comprehensive output including execution timelines, performance metrics, and comparative analyses, the simulator empowers users to evaluate the effectiveness of scheduling algorithms under various scenarios and workloads. This enables informed decision-making in system design and optimization, leading to improved resource utilisation, responsiveness, and user satisfaction.Future enhancements may focus on supporting real-time scheduling, accommodating heterogeneous computing environments, and exploring adaptive scheduling strategies to meet the evolving needs of modern computing systems.

References:

[1]Lampard, Bernard. 2014. *Program Scheduling and Simulation in an Operating System Environment*. GRIN Verlag.

[2]Nordeen, Alex. 2022. *Learn Operating Systems in 24 Hours*. Guru99.
Phanden, Rakesh Kumar, Ajai Jain, and J. Paulo Davim. 2019. *Integration of Process Planning and Scheduling: Approaches and Algorithms*. CRC Press.

[3]Tanenbaum, A. S., & Bos, H. (2014). Modern Operating Systems (4th ed.). Pearson Education.

[4]Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.