CAPSTONE PROJECT
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

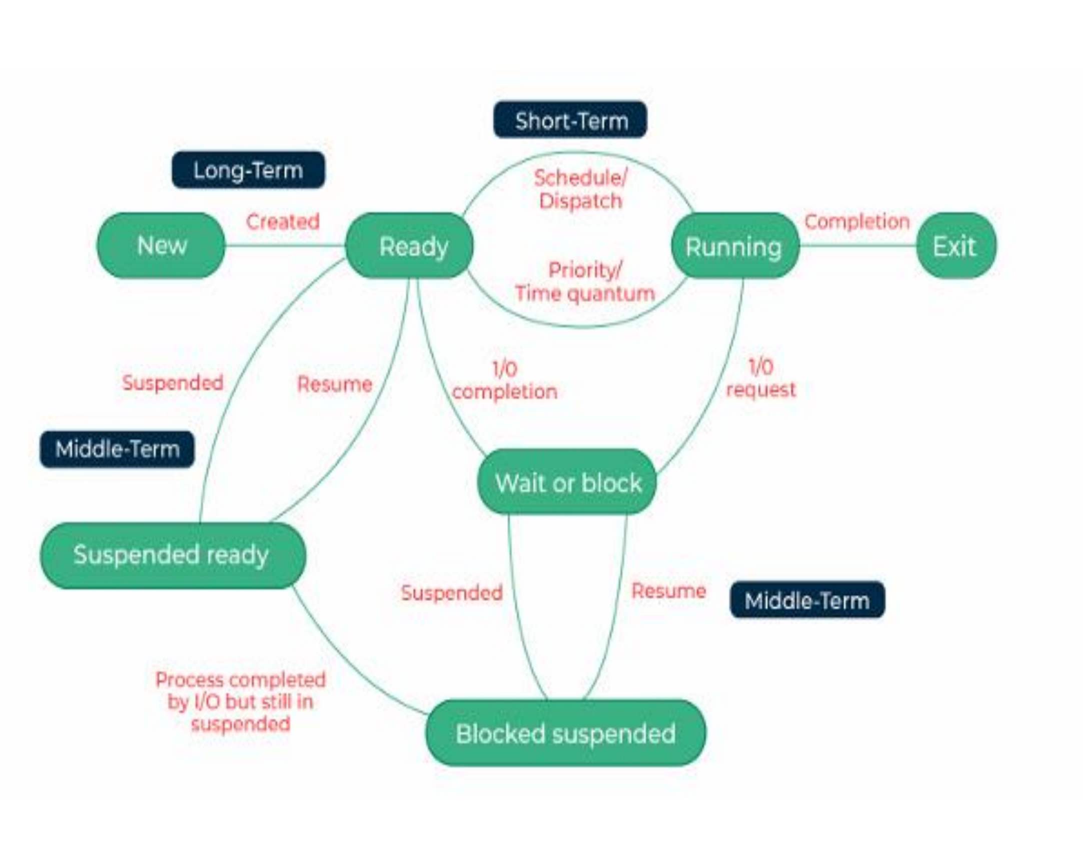
- In the realm of operating systems, process scheduling stands as a cornerstone, orchestrating the allocation of system resources to numerous concurrent processes. The efficacy of this allocation profoundly influences system performance, responsiveness, and overall efficiency.
- As the demands on modern computing systems continue to escalate, understanding and optimizing process scheduling algorithms become increasingly imperative. The intricacies of process scheduling algorithms necessitate thorough examination under controlled conditions,. This paper introduces the Process Scheduler Simulator (PSS), a comprehensive simulation platform tailored for exploring the nuances of diverse scheduling policies.

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. 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 synchronization between processes. Processes can be categorized into several types based on their behavior and requirements
- 1.Foreground Processes
- 2.Background Processes
- .System Processes
- .User Processes



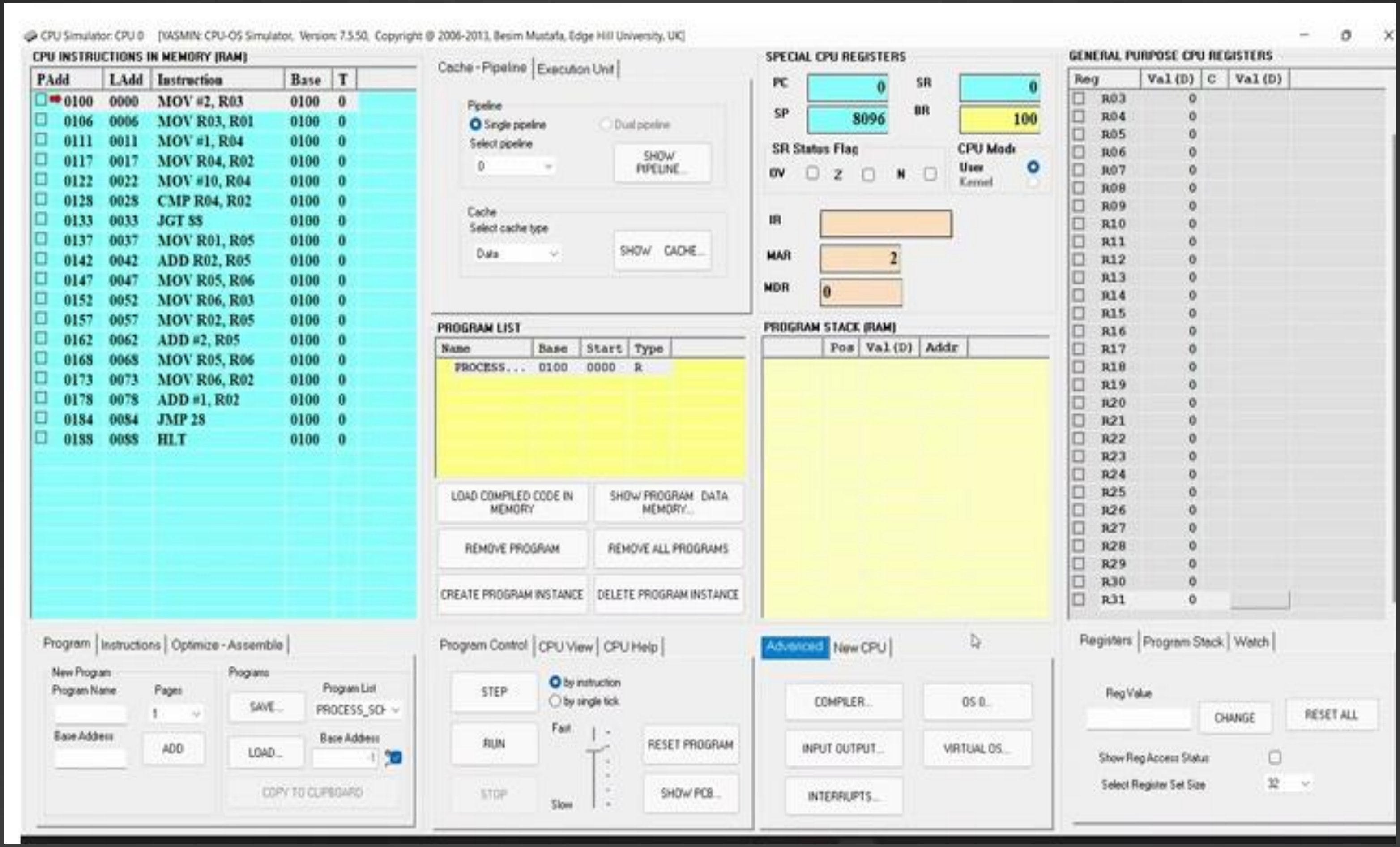
OBJECTIVE

- 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.
- Researchers can use the simulator to conduct experiments, analyze 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.
- 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.
- 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 Challenges and Future Directions Literature reviews may identify challenges and open research questions in the field of process scheduling.

OUTPUT



CPU Simulator: CPU 0 [YASMIN: CPU-OS Simulator, Version: 7.5.50, Copyright @ 2006-2013, Besim Mustafa, Edge Hill University, UK]

CPU INSTRUCTIONS IN MEMORY (RAM)

PAdd	LAdd	Instruction	Base	T
□ 0100	0000	MOV #2, R03	0100	0
□ 0106	0006	MOV R03, R01	0100	0
□ 0111	0011	MOV #1, R04	0100	0
□ 0117	0017	MOV R04, R02	0100	0
□ 0122	0022	MOV #10, R04	0100	0
□ 0128	0028	CMP R04, R02	0100	0
□ 0133	0033	JGT 88	0100	0
□ 0137	0037	MOV R01, R05	0100	0
□ 0142	0042	ADD R02, R05	0100	0
□ 0147	0047	MOV R05, R06	0100	0
□ 0152	0052	MOV R06, R03	0100	0
□ 0157	0057	MOV R02, R05	0100	0
□ 0162	0062	ADD #2, R05	0100	0
□ 0168	0068	MOV R05, R06	0100	0
□ 0173	0073	MOV R06, R02	0100	0
□ 0178	0078	ADD #1, R02	0100	0
□ 0184	0084	JMP 28	0100	0
□ 0188	0088	HLT	0100	0

Cache - Pipeline | Execution Unit

Pipeline: ● Single pipeline ○ Dual pipeline

Select pipeline: 0

Cache: Select cache type: Data

SPECIAL CPU REGISTERS

PC: 0

SR: 0

SP: 8096

BR: 100

SR Status Flag: OV □ Z □ N □

CPU Mode: User ● Kernel ○

IR:

MAR: 2

MDR: 0

PROGRAM LIST

Name	Base	Start	Type
PROCESS...	0100	0000	R

PROGRAM STACK (RAM)

Pos	Val (D)	Addr

GENERAL PURPOSE CPU REGISTERS

Reg	Val (D)
□ R03	0
□ R04	0
□ R05	0
□ R06	0
□ R07	0
□ R08	0
□ R09	0
□ R10	0
□ R11	0
□ R12	0
□ R13	0
□ R14	0
□ R15	0
□ R16	0
□ R17	0
□ R18	0
□ R19	0
□ R20	0
□ R21	0
□ R22	0
□ R23	0
□ R24	0
□ R25	0
□ R26	0
□ R27	0
□ R28	0
□ R29	0
□ R30	0
□ R31	0

Program Control: ● by instruction ○ by single tick

Program List: PROCESS_SCH

Pages: 1

Select Register Set Size: 32

Show Reg Access Status □

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.

REFERENCES

- [1]Lampard, Bernard. 2014. Program Scheduling and Simulation in an Operating System Environment. GRIN Verlag.
- [2]Norden, Alex. 2022. Learn Operating Systems in 24 Hours. Guru99. Phan den, Rakesh Kumar, Ajia Jain, and J. Paulo Daim. 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]Silber Schatz, A., Galvin, P. B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley