Text

Description automatically generated

**Activity based**

**Project Report on**

**PROGECT MANAGEMENT BTECCE22511**

**Project - I**

**Submitted to Vishwakarma University, Pune**

**Under the Initiative of**

**Contemporary Curriculum, Pedagogy, and Practice (C2P2)**

**By**

**MARIYAM PATEL**

**SRN No : 202201178**

**Roll No : 34**

**Div : C**

**Third Year Engineering**

**Department of Computer Engineering**

**Faculty of Science and Technology**

**Academic Year**

**2024-2025**

**1)Problem Statement-**

Design dynamic CPU scheduling algorithms that can adapt to variable workloads, ensuring optimal performance and resource utilization.

**2)Introduction-**

**a. Overview of the Topic:**

Dynamic CPU scheduling is an essential aspect of operating systems, allowing for the efficient management of process execution based on fluctuating workloads. Unlike static scheduling, which relies on predefined criteria, dynamic scheduling adapts in real-time to the current demands of the system. This adaptability is vital in environments where process loads can vary significantly, such as cloud computing, multi-user systems, and real-time applications. By intelligently allocating CPU time to processes based on priority, resource needs, and current workload, dynamic scheduling helps maintain system performance and responsiveness.

**b. Importance of the Topic:**

The significance of dynamic CPU scheduling cannot be overstated. As modern systems handle increasingly complex and diverse workloads, the ability to optimize CPU utilization while ensuring minimal latency for high-priority tasks becomes crucial. Efficient scheduling impacts overall system performance, including throughput, response time, and resource management. Furthermore, with the rise of cloud computing and virtualization, understanding and improving scheduling algorithms is essential for enhancing user experience and resource efficiency in shared environments.

**3)Project Scope-**

This project focuses on the design and evaluation of dynamic CPU scheduling algorithms within the realm of process management. The project will explore various existing algorithms, such as Shortest Job First (SJF), Round Robin (RR), and Priority Scheduling, and analyze their effectiveness in adapting to variable workloads. The scope will also encompass performance metrics like turnaround time, waiting time, and CPU utilization, aiming to propose improvements or new algorithms that can better handle the dynamic nature of modern workloads.

**4)Objectives-**

 **Literature Review**: Conduct a thorough review of existing dynamic scheduling algorithms and their performance metrics.

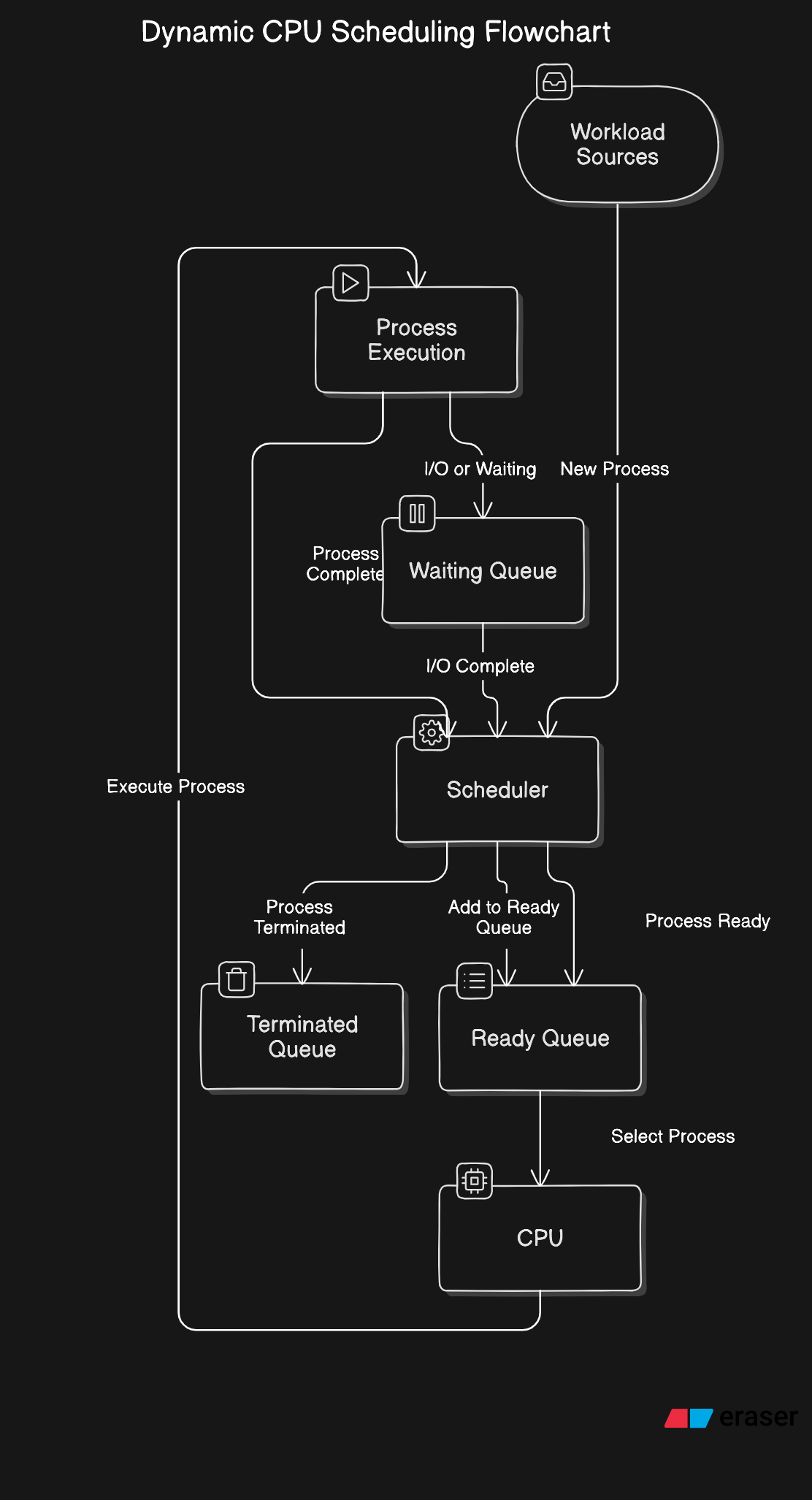
 **Algorithm Development**: Develop new or modified dynamic scheduling algorithms that can adapt to real-time workload changes.

 **Performance Evaluation**: Implement the proposed algorithms and evaluate their performance against traditional static algorithms using various workload scenarios.

 **Analysis of Results**: Analyze the results to determine the effectiveness of the proposed scheduling strategies in terms of resource utilization and system responsiveness.

**5)System Architecture-**

**a. Diagram:**



**b.Explanation:**

* **Workload Sources**: Represents where processes originate, including user inputs, system processes, and I/O devices.
* **Scheduler**: Manages how processes are selected for execution, including implementing scheduling policies and managing different queues.
* **CPU**: The core component that executes the processes, managing resources and context switching.
* **Processes**: Represents various states of processes (ready, waiting, terminated) that the scheduler interacts with.

**6)Project Environment to be Used-**

**Tools, Programming Languages, and Platforms:**

* **Operating System**: Linux, which provides a robust environment for process management and scheduling simulations.
* **Programming Languages**: C/C++, due to their efficiency and control over system resources, making them suitable for implementing scheduling algorithms.
* **Simulation Tools**: Tools like gdb for debugging and valgrind for performance profiling.
* **IDE**: An integrated development environment such as Code::Blocks or Eclipse for ease of coding and testing.

**7)Key Algorithms and Data Structures to be Used-**

**a. Explanation of Algorithms:-**

* **Dynamic Scheduling Algorithms:**
  + Shortest Job First (SJF): Chooses the process with the smallest execution time, which can lead to minimized average waiting time but may suffer from starvation.
  + Round Robin (RR): Allocates a fixed time slice to each process in the ready queue, ensuring fairness and responsiveness, particularly in time-sharing environments.
  + Priority Scheduling: Assigns priority levels to processes, executing higher-priority tasks first. However, it requires careful management to prevent starvation of lower-priority processes.

**b. Data Structures:-**

* Queues: Used to manage processes in different states (ready, waiting, terminated).
* Heaps/Priority Queues: Essential for implementing priority-based scheduling to quickly access the highest-priority process.
* Arrays: For tracking process information and metrics such as turnaround time and waiting time.

**8)Conclusion-**

This project aims to investigate and enhance dynamic CPU scheduling algorithms to better adapt to variable workloads. By examining existing strategies and proposing new approaches, the project seeks to improve CPU utilization and overall system performance. The findings will provide valuable insights into the impact of scheduling on resource management in modern computing environments.

**9)References-**

 **Operating Systems: Three Easy Pieces**: <https://pages.cs.wisc.edu/~remzi/OSTEP/>

 **GeeksforGeeks - CPU Scheduling**: GeeksforGeeks CPU Scheduling

 **TutorialsPoint - CPU Scheduling**: TutorialsPoint CPU Scheduling

 **Liu, C. L., & Layland, J. W.** (1973). "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment." *Journal of the ACM*, 20(1), 46-61.

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