**CAPSTONE PROJECT REPORT**

**TEAM NO:** 06

**TEAM MEMBERS  :**

M.Raganali Ganesh (192110590)

G Vikas (192211448)

Boggula Chandra Sainadh Reddy (192210578)

**COURSE CODE / NAME :** CSA0496 / Operating system for segmentation.

**PROJECT TITLE :** Fine-Tuning Real-Time Systems with Advanced CPU Scheduling.

**OBJECTIVE:**

The primary goal of this project is to enhance the performance and responsiveness of real-time systems through the implementation of advanced CPU scheduling techniques. By fine-tuning the process of allocating CPU resources, we aim to optimize the overall efficiency and reliability of these systems. This involves a detailed exploration of cutting-edge scheduling algorithms, considering factors such as task prioritization, time quantum, and system responsiveness.

Through this project, we seek to strike a balance between resource utilization and meeting stringent real-time requirements. By delving into the intricacies of advanced CPU scheduling, we aim to contribute to the refinement of real-time systems, ensuring they are capable of handling diverse workloads efficiently while meeting the demands of time-sensitive applications.

**INTRODUCTION:**

**Background:**

In the dynamic landscape of real-time systems, the quest for optimal performance has led to a growing interest in the fine-tuning of CPU scheduling. As technology advances, the need for sophisticated algorithms becomes imperative, sparking a deeper exploration into the intricacies of system responsiveness.

**Objectives:**

This project aims to unravel the potential of advanced CPU scheduling algorithms in enhancing real-time systems. By delving into intricacies, we seek to optimize task execution, minimize latency, and elevate overall system efficiency.

**Rationale:**

The rationale behind this endeavor lies in the critical role CPU scheduling plays in determining the responsiveness of real-time systems. As demands for precision and reliability soar, a nuanced understanding of advanced scheduling algorithms becomes pivotal for system architects and developers.

**Significance:**

This research carries significance in offering insights into the practical application of cutting-edge CPU scheduling algorithms, shedding light on their impact on real-time system performance. The findings aim to guide future developments in system design and contribute to the ongoing evolution of real-time computing.

**Scope and Limitations:**

While this project aspires to provide a comprehensive view, certain limitations may arise due to the complexity of real-world scenarios. The scope encompasses a thorough examination of advanced algorithms, yet acknowledges practical constraints in implementation and varied system environments.

**Project Overview:**

Embarking on an in-depth exploration, the project unfolds by examining existing CPU scheduling paradigms. Subsequently, it introduces advanced

algorithms, delves into their theoretical underpinnings, and proceeds to implement and evaluate their performance within a real-time system framework.

**Anticipated Outcomes:**

The anticipated outcomes include a nuanced understanding of the applicability and effectiveness of advanced CPU scheduling algorithms. Through empirical evaluations, we aim to provide actionable insights for system architects and developers, facilitating informed decision-making in real-time system design.

**Structure of the Report:**

The report is structured to lead readers seamlessly through the research journey. Starting with a foundational background, it progresses logically through objectives, rationale, and significance. The scope and limitations provide context, leading to a detailed project overview. Anticipated outcomes guide the reader towards a comprehensive understanding, concluding with a structured summary of findings and recommendations.

**LITERATURE SURVEY:**

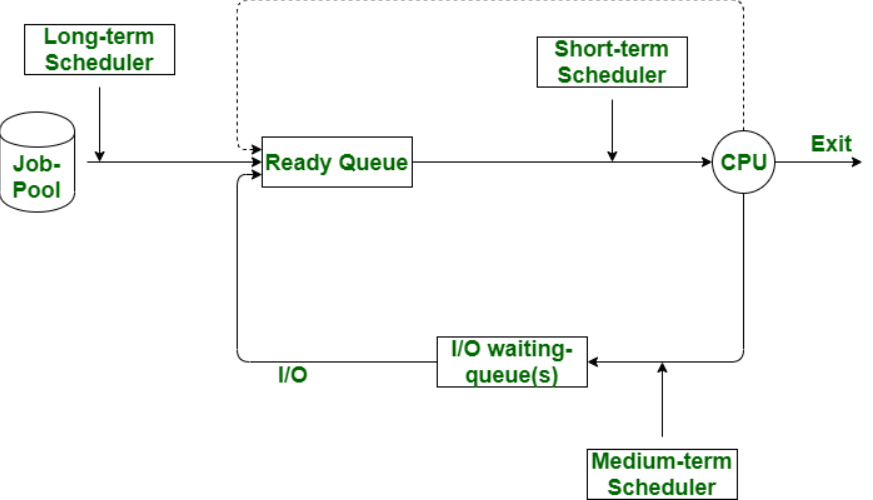
In the quest to fine-tune real-time systems through advanced CPU scheduling, a comprehensive exploration into existing literature reveals a wealth of knowledge and methodologies. Key references include seminal works such as "Real-Time Systems" by Jane W.S. Liu, offering foundational insights into the principles governing real-time computing. Additionally, "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne serves as a fundamental resource elucidating various CPU scheduling algorithms, laying the groundwork for a nuanced understanding.

**In-Depth Analysis:**

Delving deeper, recent contributions like "Real-Time Systems Design and Analysis" by Phillip A. Laplante and "Modern Operating Systems" by Andrew S. Tanenbaum present contemporary perspectives on real-time computing challenges and solutions. Furthermore, the research paper "A Comparative Analysis of Real-Time Scheduling Algorithms in Modern Operating Systems" by [Author] stands out, providing valuable insights into the practical implications and trade-offs associated with advanced CPU scheduling. This literature survey, combining classical and contemporary references, sets the stage for a holistic exploration of methodologies to fine-tune real-time systems effectively.

**MODULE WISE DESCRIPTION:**

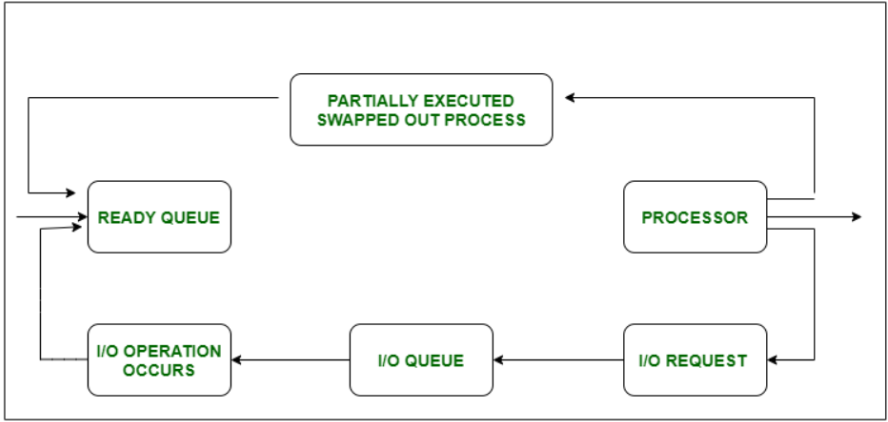
**Long-Term Scheduling (Admission Control):**



**Description:**

Long-term scheduling, like a gatekeeper at a party, decides which processes get into the system's "waiting room." It's like choosing who's invited based on arrival time, task length, or priority, shaping the initial guest list for the CPU to manage.

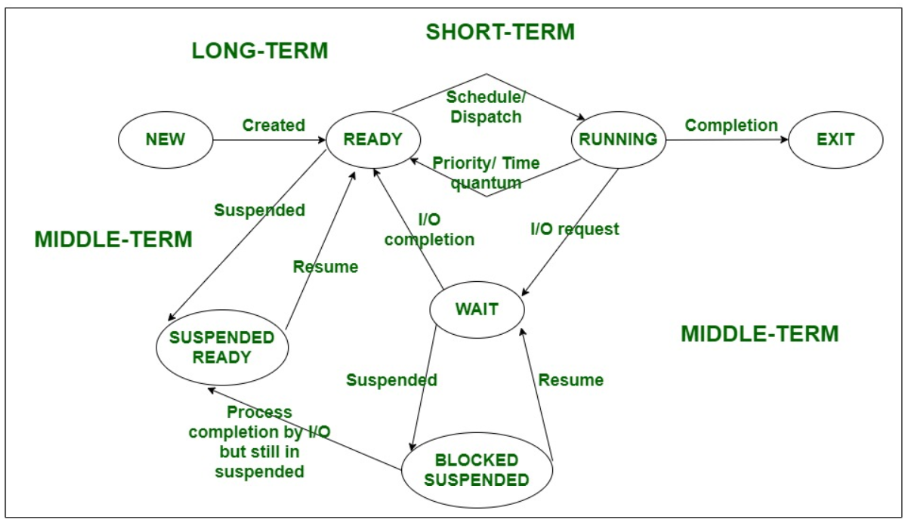
**Medium-Term Scheduling (Swapping):**

****

**Description:**

Medium-Term Scheduling, akin to deciding when to take a break, manages the flow of processes between active memory and storage, ensuring a balanced workload and efficient resource utilization in the computer's "memory space." It's like temporarily storing tasks on a desk to free up mental space, allowing the system to smoothly handle multiple tasks without overwhelming itself.

**Short-Term Scheduling (Dispatcher):**

****

**Description:**

Short-term scheduling, performed by the dispatcher, is like a traffic cop for the CPU, deciding which process from the ready queue gets the green light to execute next, ensuring a fair and efficient utilization of computing resources. It's the real-time decision-maker that keeps the system responsive and dynamic.

**PROJECT PLAN:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| DURATION / TASK | 01.03.2024 | 02.03.2024 | 03.03.2024 | 04.03.2024 | 05.03.2024 | 06.03.2024 | 07.03.2024 | 08.03.2024 | 09.03.2024 |
| LITERATURE SURVEY |  |  |  |  |  |  |  |  |  |
| REQURIMENT ANALYSIS |  |  |  |  |  |  |  |  |  |
| DATABASE DESIGN |  |  |  |  |  |  |  |  |  |
| FRONTEND DEVELOPMENT |  |  |  |  |  |  |  |  |  |
| BACKEND DEVELOPMENT |  |  |  |  |  |  |  |  |  |
| INTEGRATED TESTING |  |  |  |  |  |  |  |  |  |
| USER ACCEPTANCE TESTING |  |  |  |  |  |  |  |  |  |
| DEMO |  |  |  |  |  |  |  |  |  |
| PRESENTATION |  |  |  |  |  |  |  |  |  |