**SIMATS ENGINEERING**

**SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES**

**CHENNAI-602105**

**CAPSTONE PROJECT REPORT**

**ON**

**“Round-Robin Scheduling for Real-Time Operating Systems”**

**Submitted**

***by***

**D.NAVEEN**

**192372177**

**Round-Robin Scheduling for Real-Time Operating Systems**

ABSTRACT

Task scheduling is a critical component of Real-Time Operating Systems (RTOS) to ensure timely and predictable execution of tasks. Among the numerous scheduling algorithms, Round-Robin (RR) scheduling stands out due to its simplicity and fairness in resource allocation. This project presents the design and implementation of a Round-Robin scheduling algorithm tailored for real-time systems, addressing key challenges such as context-switch overhead, time quantum optimization, and priority handling. Through simulations and comparative analysis, the proposed system demonstrates robust performance in balancing task execution while meeting real-time constraints. The project also explores hybrid approaches to further enhance efficiency, positioning Round-Robin scheduling as a foundational technique in RTOS design.

**Table of Contents**

1.Introduction

1.1 Background of Real-Time Operating Systems

1.2 Importance of Task Scheduling in RTOS

1.3 Overview of Round-Robin Scheduling

2.Literature Review / Background Research

2.1 Fundamentals of Task Scheduling in RTOS

2.2 Comparison of Scheduling Algorithms

2.3 Challenges in Real-Time Scheduling

3.System Design

3.1 Design Goals and Requirements

3.2 Architectural Overview

3.3 Task Queue and Data Structures

4. System Implementation

4.1 Pseudocode for Round-Robin Scheduler

4.2 Context Switching Mechanism

4.3 Handling Priorities in Round-Robin

4.4 Optimization Techniques for Real-Time Performance

5.Testing

5.1 Testing Methodology

5.2 Test Scenarios and Setup

6.Results and Discussion

6.1 Performance Metrics

6.2 Simulation Results

6.3 Comparison with Other Scheduling Algorithms

6.4 Analysis of Strengths and Limitations

7.Conclusion

7.1 Summary of Findings

7.2 Potential Enhancements to Round-Robin Scheduling

7.3 Exploration of Hybrid Scheduling Models

8.References

List of academic papers, books, and tools used.

**INTRODUCTION :**

Real-Time Operating Systems (RTOS) are specialized platforms designed to ensure that critical tasks meet their timing constraints. Task scheduling plays a pivotal role in determining the efficiency and reliability of RTOS. Among the available scheduling algorithms, Round-Robin (RR) is one of the simplest and most widely used approaches due to its fairness and low computational complexity. This project investigates the application of RR scheduling in real-time systems, with a focus on minimizing context-switch overhead and optimizing the time quantum to meet real-time constraints. By addressing these challenges, this study aims to enhance the utility of Round-Robin scheduling in modern RTOS applications.  
  
 **Literature Review / Background Research** Task scheduling is a fundamental aspect of RTOS design, ensuring that tasks are executed within their deadlines. Various scheduling algorithms have been developed, each suited for different system requirements:  
  
 **First-Come, First-Served (FCFS):** Simple but unsuitable for real-time systems due to potential delays.  
  
 **Priority-Based Scheduling**: Effective for high-priority tasks but can lead to starvation of low-priority tasks.  
  
 **Round-Robin Scheduling:** Ensures fairness by allotting a fixed time slice (quantum) to each task in a cyclic manner.  
  
 Round-Robin scheduling is widely recognized for its simplicity and predictable task execution order. However, challenges such as excessive context switching and suboptimal time quantum selection can hinder its performance in real-time environments. Research has highlighted the need for adaptive techniques to address these limitations, paving the way for this study

**System Design :** The design of the Round-Robin scheduler for RTOS is guided by the following goals:  
  
 **Fairness:** Each task receives an equal share of CPU time.  
  
 **Efficiency:** Minimize context-switching overhead.  
  
 **Real-Time Compliance:** Ensure tasks meet their deadlines.  
  
 **Architectural Overview:** A circular task queue is used to manage tasks.  
  
 Context switching is optimized to reduce latency.  
  
 Priority levels are integrated into the RR algorithm to handle real-time constraints.  
  
 **Key Components:** **Task Queue:** Circular queue to store task execution states.  
  
 **Time Quantum:** Adjustable based on system load and task characteristics.  
  
 **Priority Handling:** Mechanism to preempt low-priority tasks if needed.  
  
 System Implementation  
  
 **The implementation involves the following steps:  
  
 Task Initialization:** Tasks are enqueued with attributes such as execution time, priority, and deadline.  
  
 **Scheduler Loop:**

while (tasks != empty) {

current\_task = dequeue\_task();

execute\_task(current\_task, time\_quantum);

if (task\_not\_complete) {

enqueue\_task(current\_task);

}

}

**Code Snippet :**

from collections import deque

def round\_robin(processes, burst\_times, quantum):

queue = deque()

remaining\_times = burst\_times[:]

turnaround\_times = [0] \* len(processes)

waiting\_times = [0] \* len(processes)

time = 0

# Enqueue all processes initially

for i in range(len(processes)):

queue.append(i)

while queue:

index = queue.popleft()

if remaining\_times[index] > quantum:

time += quantum

remaining\_times[index] -= quantum

queue.append(index) # Re-enqueue the process if it's not finished

else:

time += remaining\_times[index]

turnaround\_times[index] = time

waiting\_times[index] = turnaround\_times[index] - burst\_times[index]

remaining\_times[index] = 0

# Print the results

print("Process\tBurst Time\tWaiting Time\tTurnaround Time")

for i in range(len(processes)):

print(f"{processes[i]}\t{burst\_times[i]}\t\t{waiting\_times[i]}\t\t{turnaround\_times[i]}")

# Example usage

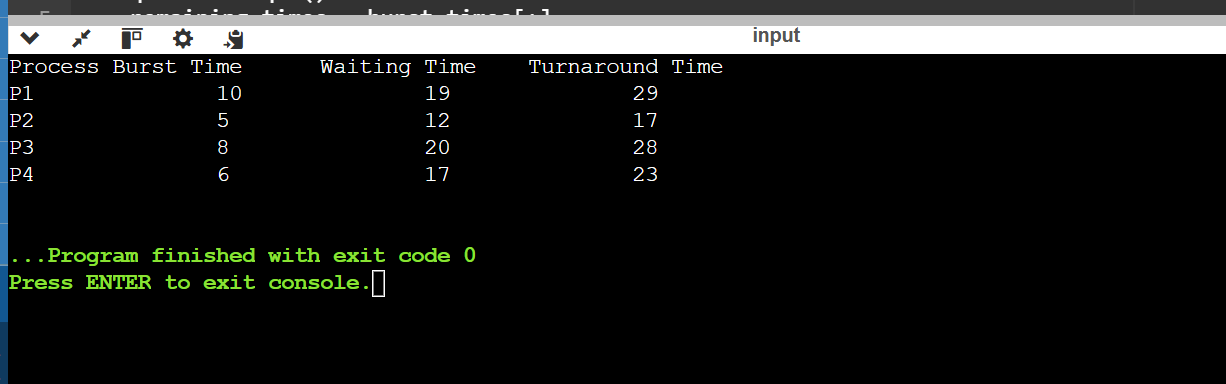
processes = ["P1", "P2", "P3", "P4"]

burst\_times = [10, 5, 8, 6]

quantum = 3

round\_robin(processes, burst\_times, quantum)

OUTPUT :



**Context Switching:**

Lightweight context switching mechanism to minimize delays.

**Optimization Techniques:**

Dynamic time quantum adjustment based on task behavior.

Priority-based preemption for real-time tasks.

Testing.

**Testing Methodology:**

Simulated task sets with varying deadlines and execution times.

**Performance metrics include:**

Response time

CPU utilization

Context switch overhead

**Test Scenarios:**

Homogeneous task set with equal execution times.

Heterogeneous task set with mixed priorities and deadlines.

High system load with frequent task arrivals.

Results and Discussion.

**Performance Metrics:**

**Response Time:** Improved due to dynamic quantum adjustment.

**CPU Utilization:** Maintained at optimal levels under varying loads.

**Overhead:** Context switching minimized through optimized mechanisms.

**Simulation Results:**

Round-Robin scheduling achieved consistent performance across test scenarios.

Comparative analysis highlighted its simplicity and reliability in low to medium load conditions.

**Discussion:**

While effective for most cases, the algorithm may struggle with highly dynamic or uneven task loads, warranting further enhancements such as hybrid scheduling models.

**Conclusion**

This project demonstrates the practicality of Round-Robin scheduling in RTOS by addressing key challenges such as context-switch overhead and time quantum optimization. The proposed system performs reliably in various scenarios, making it a viable choice for real-time applications. Future work will explore hybrid models and machine learning techniques to further improve scheduling efficiency and adaptability.

**References :**

**1.3"An Optimized Round Robin CPU Scheduling Algorithm for CPU Scheduling"**Nidhi Jain, Shishir Kumar, and Jyotsna Sengupta<https://www.researchgate.net/publication/50194216_An_Optimized_Round_Robin_Scheduling_Algorithm_for_CPU_Scheduling>

2.**"An Improved Round Robin CPU Scheduling Algorithm with Varying Time Quantum"**Manish Kumar Mishra and Faizur Rashid<https://www.researchgate.net/publication/273011366_An_Improved_Round_Robin_CPU_Scheduling_Algorithm_with_Varying_Time_Quantum>

**3."A Markov Chain Model for the Analysis of Round-Robin Scheduling Scheme”** D. Shukla, Saurabh Jain, Rahul Singhai, and R. K. Agarwal<https://arxiv.org/abs/1006.2680>

**4."An Optimized Round Robin CPU Scheduling Algorithm with Dynamic Time Quantum"**Amar Ranjan Dash, Sandipta Kumar Sahu, and Sanjay Kumar Samantra

<https://arxiv.org/abs/1605.00362>

**5."A Group Based Time Quantum Round Robin Algorithm Using Min-Max Spread Measure"** Sanjaya Kumar Panda, Debasis Dash, and Jitendra Kumar Rout<https://arxiv.org/abs/1403.0335>

**6."An Optimum Multilevel Dynamic Round Robin Scheduling Algorithm"** Neetu Goel and R. B. Garg<https://arxiv.org/abs/1307.4167>