**GOVERNMENTPOLYTECHNIC, AHMEDABAD**

**COMPUTER ENGINEERING DEPARTMENT**



**Affiliated To**

## **Gujarat Technological University, Ahmedabad**

**Micro project Report**

**D. E. Second Year (Semester–III)**

# **Sub: Basics of Operating System (4330703)**



**Government Polytechnic, Ahmedabad Computer Engineering Department**

**CERTIFICATE**

This is to certify that

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Enrollment No.** | **Name** |
| **1** | 226170307015 | Kavan Bhavsar |
| **2** | 226170307004 | Bhavya Bagadia |
| **3** | 226170307075 | Khush Panchal |

Of **Third** semester of Diploma in Computer Engineering of Government Polytechnic, Ahmedabad has completed the Micro-Project satisfactorily in Subject **Basics of Operating System (4330703)**for the academic year **2023-2024** as prescribed in the curriculum.

Lecturer, HOD

Computer Engg. Dept., Computer Engg. Dept., Government Polytechnic, Ahmedabad Government Polytechnic,

Ahmedabad

**RUBRICS FOR MICRO-PROJECT ASSESMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters** | **Allocated Marks** | **High** | **Medium** | **Low** |
| **Problem Analysis and Solution(R1)** | 8 | Problem is Properly Analyzed and Solved | Problem is Properly Analyzed but  Partially  Solved | Problem is Properly Analyzed but not  Solved. |
| 8Marks | 5Marks | 2Marks |
| **Viva Voce(R2)** | 2 | Student Answered All  The Viva Voce Questions | Student Answered Only A Few Viva Voce Questions | Student Did Not Answer Any Viva Voce  Questions |
| 2Marks | 1Marks | 0Marks |

# INDEX

1. Description of Problem/ Case Study
2. Solution of Problem/ Explanation/ Conclusion
3. References if any

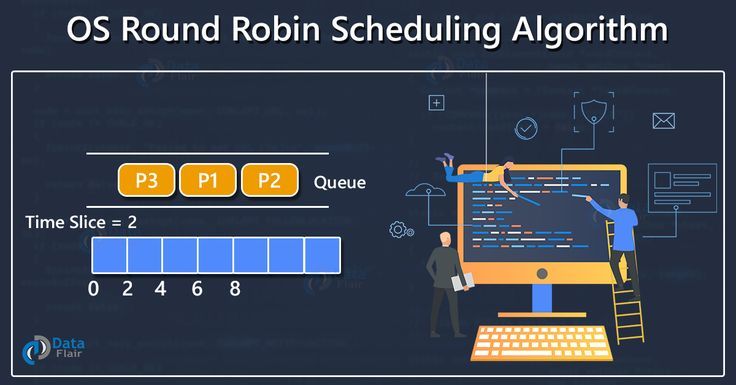
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Enrollment**  **Number** | **Student Name** | **Marks(R1)** | **Marks(R2)** | **Total**  **Marks** |
| 226170307015 | Kavan Bhavsar |  |  |  |
| 226170307004 | Bhavya Bagadia |  |  |  |
| 226170307075 | Khush Panchal |  |  |  |
| **Name and Sign of Faculty:** | | | | |

# Description:

In the realm of task scheduling for computing resources, the Round Robin (RR) algorithm assumes a pivotal role. This method, characterized by its cyclic time-slicing approach, endeavors to achieve an equitable distribution of CPU time among tasks. Notable for its balance between efficiency and fairness, the Round Robin algorithm plays a crucial part in the sophisticated orchestration of operating systems.

# Round robin:

**What is Round Robin (RR):** is a CPU scheduling algorithm primarily used in multitasking environments. It's a pre-emptive algorithm where each process is assigned a fixed time unit, often called a time quantum or time slice, and the CPU switches between processes once their allotted time expires.



Process:

**1. Initialization:**

- Define a time quantum (e.g., 10ms) for each process in the system.

- Set up a ready queue to hold incoming processes.

**2. Arrival of Processes:**

- As processes arrive or become ready for execution, they join the ready queue.

**3. Process Selection:**

- The scheduler chooses the first process from the ready queue for CPU execution.

**4. Execution:**

- Allocate the CPU to the selected process for the defined time quantum.

- If the process completes its task within the time quantum, it voluntarily releases the CPU and moves out.

- If the quantum expires before task completion, the scheduler pre-empts the process and places it at the queue's end.

**5. Context Switching:**

- Save the current state of the pre-empted process and switch to the next process in the queue.

- This context switch incurs overhead as the CPU transitions from one process to another.

**6. Cycle Continuation:**

- Repeat steps 3 to 5, cycling through the processes in the ready queue.

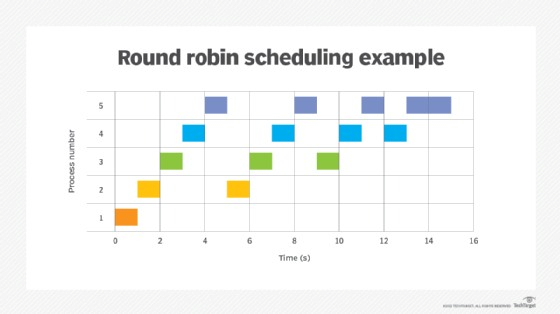
- The scheduler continues this process until all tasks finish or reach completion.

**7. Fairness and Equal Time Allocation:**

- RR ensures fairness by providing equal CPU time slices to each process in the queue, preventing any process from monopolizing resources.

**8. Completion:**

- As processes complete their tasks, they exit the system.



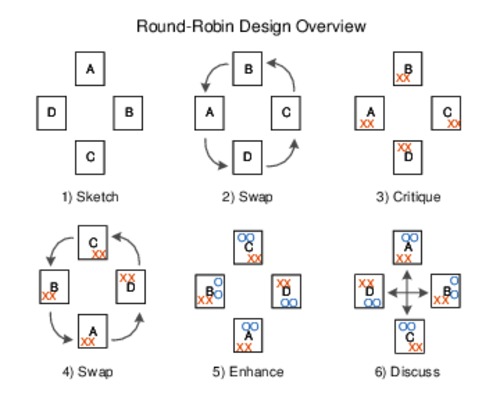
**Features of Round Robin:**

**1. Fairness:** Provides fair utilization of CPU time among processes by allocating equal time slices initially.

**2. Pre-emptive:** Allows the operating system to interrupt a process and allocate the CPU to another process when the time quantum expires.

**3. Simple Implementation:** Relatively easy to implement and understand compared to other scheduling algorithms.

**4. Supports Time-Sharing:** Suitable for time-sharing systems where multiple users require access to the CPU.



**Pros of Round Robin:**

**1. Fairness:** Ensures that all processes get a chance to execute, preventing starvation.

**2. Response Time:** Provides better response time for short jobs compared to some other algorithms.

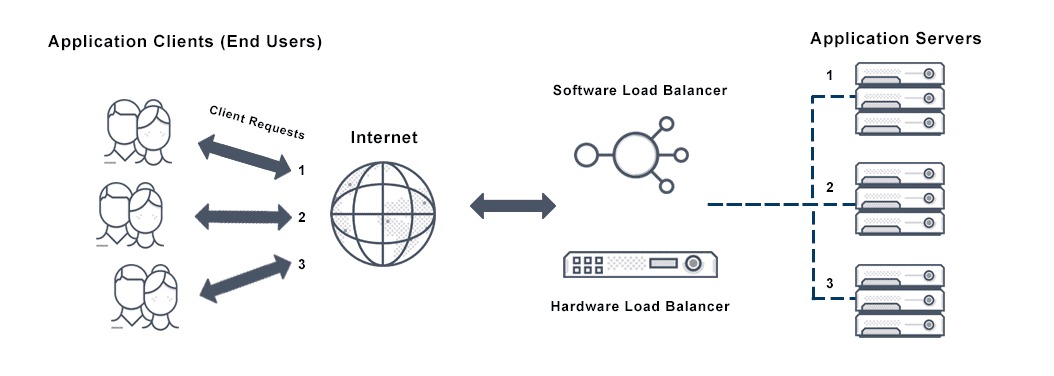
**3. Simple:** Has a simple implementation and is easy to understand.

**Cons of Round Robin:**

**1. High Context Switching Overhead:** Constantly switching between processes at regular intervals can lead to increased overhead due to context switching.

**2. Long Average Waiting Time:** May result in longer average waiting times, especially for CPU-bound tasks or when the time quantum is too large.

**3. Inefficiency with Varying Process Requirements:** Inefficient for processes with different CPU burst times as short processes may wait for their turn even if the CPU is available.



Round Robin is also used in load balancing for server