

IMPLEMENTATION OF ROUND ROBIN SCHEDULING FOR QUERIES

Appaji Sree Dharma Shasta Rao

Reg No:RA2211026010162

Guntoori Mrunal Varma

Reg No:RA2211026010156

Sesetti Sai Venkata Pavan

Reg No:RA2211026010144

Batch ID: `

Guide name and Designation : S. VIMAL



SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

Table of contents

- ABSTRACT
- INTRODUCTION
- PROBLEM STATEMENT
- LITERATURE SURVEY
- METHODS AND METHODOLOGIES
- RESULT AND DISCUSSION
- CONCLUSION
- REFERENCE

ABSTRACT

The brief abstract of implementing Round Robin Scheduling for Queries is to ensure fairness, responsiveness, and efficiency in managing multiple queries in a database system. It aims to distribute system resources evenly, prevent query starvation, adapt to varying workloads, and improve user satisfaction by delivering timely responses while minimizing system overhead.

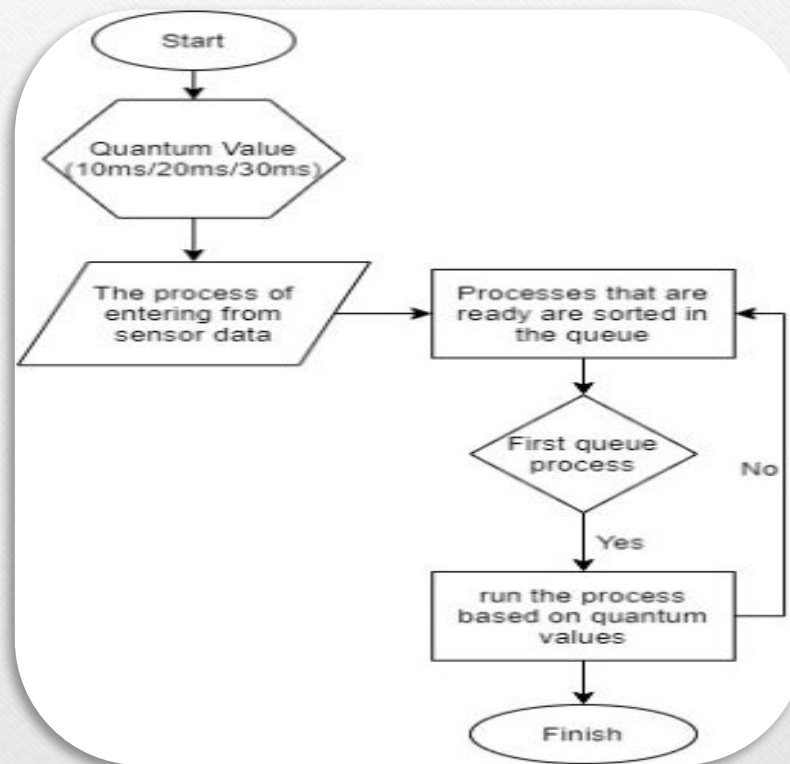
INTRODUCTION

- Welcome to the presentation on Project .
- We will be discussing the implementation of Round Robin Scheduling for queries.
- Topics will include: Problem Statement, Algorithm Design, Implementation, Results.
- Presented by team for the Operating Systems course.

PROBLEM STATEMENT

- Need to schedule multiple queries from students and faculty.
- Queries have arrival times and must be processed fairly.
- Faculty queries should get higher priority.
- Goal is to minimize average waiting and turnaround times.

Flowchart:



METHODS AND METHODOLOGIES

1. Used the Round Robin scheduling algorithm.
2. Sort queries by arrival time and merge with faculty queries at start.
3. Loop through queries and allocate CPU for time quantum.
4. After quantum expires, context switch to next query.
5. Repeat until all queries are processed.

METHODS AND METHODOLOGIES

Hardware Components:

1. **Central Processing Unit (CPU):** The CPU is the brain of the computer. It executes instructions from software programs and manages data processing.
2. **Memory (RAM):** Random Access Memory (RAM) is used to temporarily store data that the CPU is actively using. It provides fast, volatile storage for running applications.
3. **Storage Devices:** These include hard disk drives (HDDs) and solid-state drives (SSDs) for long-term data storage. Storage devices store the operating system, applications, and user data.
4. **Motherboard:** The motherboard connects and houses various hardware components, including the CPU, RAM, and expansion slots for other hardware like graphics cards and network adapters.
5. **Graphics Processing Unit (GPU):** The GPU is responsible for rendering graphics and accelerating specific calculations. It is particularly crucial for tasks like gaming, video editing, and scientific computing.
6. **Input/Output (I/O) Devices:** These include peripherals such as keyboards, mice, monitors, printers, and external storage devices.
7. **Network Interface Cards (NICs):** NICs enable network connectivity, allowing the computer to communicate with other devices over networks, including the internet.
8. **Power Supply Unit (PSU):** The PSU provides electrical power to the computer components.
9. **Cooling System:** To prevent overheating, computers often have cooling systems like fans and heat sinks.
10. **Expansion Cards:** These are additional cards that can be added to the motherboard to enhance functionality, such as graphics cards, sound cards, or network cards.

• **Software Components:**

1. **Operating System (OS):** The OS is system software that manages hardware resources and provides services to software applications. Popular operating systems include Windows, macOS, and Linux.
2. **Application Software:** These are the programs and software applications that users run on their computers. Examples include web browsers, word processors, and video editing software.
3. **Utilities:** These are software tools that help manage and maintain the computer system. Examples include antivirus software, disk cleanup tools, and backup utilities.
4. **Device Drivers:** Drivers are software components that facilitate communication between the OS and specific hardware devices, ensuring they work correctly.
5. **Firmware:** Firmware is low-level software that resides on hardware components like the motherboard and peripheral devices. It provides basic control and functionality for these components.
6. **Programming Languages and Libraries:** Developers use programming languages and software libraries to write applications and software.
7. **Web Browsers:** These applications allow users to access and interact with websites on the internet.

RESULT AND DISCUSSION

```
Output Clear
/tmp/3-JAGoHrT5s.o
WELCOME TO THE OS PROJECT MADE BY DHARMA,MUNNA,PAWAN.

Please follow these instructions to execute the program:
1. Enter number of queries between 0 & 120
2. Make sure to keep value of TimeQuantum minimum for convinience
3. Enter Query Arrival Time in the format of HHMM
   Example: 10:25 should be entered as 1025
4. Next Query's ArrivalTime must be less than previous Query's CompletionTime (ArrivalTime + BurstTime)
5. BurstTime must be entered such that (ArrivalTime + BurstTime) < 120

Enter total number of Queries: 1
Enter Time Quantum for each query: 10
Type of Query (1 for Faculty, 2 for Student): 1
Enter Query ID: 2
Enter Query Arrival Time: Enter Burst Time: 22
==> Time is in minutes for all calculations

Query ID    ArrivalTime BurstTime    WaitingTime TurnAroundTime  CompletionTime
2           1000 hh:mm  22 minutes  0 minutes   22 minutes  1022 hh:mm

Summary of Execution:

Total Time Spent on handling Queries: 22 minutes
Average TurnAround Time : 22.00 minutes
Average Waiting Time : 0.00 minutes

Program Execution Completed!
```


Implementation of code:

- Implemented in C.
- Used vectors and queues for data structures.
- Sorted vectors using comparator function.
- Used Round Robin logic in main simulation loop.
- Computed waiting and turnaround times.

REFERENCE

- Silberschatz et al. Operating System Concepts. Wiley.
- Tanenbaum, Modern Operating Systems. Prentice Hall.
- [https://en.wikipedia.org/wiki/Scheduling_\(computing\)](https://en.wikipedia.org/wiki/Scheduling_(computing)).
- <https://www.geeksforgeeks.org/round-robin-scheduling-algorithm/>.

CONCLUSION

- Implemented Round Robin scheduling in C.
- Algorithm was fair and efficient.
- Faculty queries handled with priority.
- Achieved reasonable turnaround and waiting times.
- Met the goals and requirements of the project.



SRM

INSTITUTE OF SCIENCE & TECHNOLOGY
Deemed to be University u/s 3 of UGC Act, 1956

THANK YOU ALL!
