Fatima
Business School
SALIM HABIB UNIVERSITY

# FINAL PROJECT REPORT

## OPERATING SYSTEM

**SUPERVISED BY:**

MISS SUMRA KHAN

SIR SHOAIB AHMED

# PROJECT TITLE

# MULTITHREADED SYSTEM HEALTH MONITORING TOOL

(This project is about creating a System Health Monitoring Tool that runs using multithreading.)

## GROUP MEMBERS:

**Sadia Tehreem (F23CSC043)**

**Syed Noor Ali Shah (F23CSC047)**

## Table of Contents

# MULTITHREADED SYSTEM HEALTH MONITORING TOOL

This project focuses on the development of a Multithreaded System Health Monitoring Tool that observes and displays important system performance parameters in real time. The tool monitors CPU usage, memory usage, disk usage, and running processes by using multithreading, where each monitoring task runs in a separate thread. This approach improves performance, responsiveness, and efficiency compared to single-threaded monitoring systems. The project helps in understanding core Operating System concepts such as threads, process management, concurrency, and resource utilization.

## INTRODUCTION

An operating system is responsible for managing computer hardware resources and providing services to programs. One of the important responsibilities of an OS is to monitor system resources such as CPU, memory, disk, and running processes. System health monitoring tools help users and administrators understand how system resources are being used.

In this project, a **Multithreaded System Health Monitoring Tool** is developed to display system performance information in real time. The tool uses multithreading so that each system parameter is monitored independently. This makes the application faster, smoother, and more responsive. The project provides a practical understanding of how multithreading works in operating systems.

## OBJECTIVES

The main objectives of this project are:

- To design a tool that continuously monitors system health parameters such as CPU usage, memory usage, disk usage, and running processes.
- To implement multithreading so that each monitoring task runs independently.
- To display real-time system information in a simple and user-friendly manner.
- To understand and demonstrate the role of threads in improving system performance and responsiveness.
- To gain practical experience with Operating System concepts like concurrency and resource management.

## PROBLEM STATEMENT

Many existing system monitoring tools are either heavy, slow, or not open-source. Some tools use a single-threaded approach, which can cause the application to freeze or respond slowly when multiple monitoring tasks are performed at the same time. This creates performance and usability issues.

There is a need for a **lightweight and efficient system health monitoring tool** that can handle multiple monitoring tasks simultaneously without slowing down the system. Using multithreading can solve this problem by allowing different system resources to be monitored in parallel.

## SCOPE OF THE PROJECT

The scope of this project is limited to monitoring basic system performance parameters on a single machine. The tool is designed for educational purposes and focuses on demonstrating multithreading concepts rather than replacing professional system monitoring software.

The project can be extended in the future by adding features such as graphical user interfaces, alerts for high resource usage, logging system data, or network monitoring.

## METHODOLOGY

The project is developed using Python programming language with multithreading techniques. Each system parameter is monitored using a separate thread so that all information is updated simultaneously.

The following steps are followed:

Study the concept of multithreading and system resource monitoring.

Set up the development environment and required libraries.

Create separate threads for CPU, memory, disk, and process monitoring.

Collect system information using built-in libraries.

Display the collected data in real time.

Test and debug the application to ensure smooth performance.

This approach ensures parallel execution and efficient utilization of system resources.

## SYSTEM MODULES

### CPU Monitoring Module

This module monitors CPU usage percentage and load on each core. It helps users understand how much processing power is being used at any given time.

### Memory Monitoring Module

This module displays total memory, used memory, and available memory. It provides insight into RAM utilization and helps identify memory-intensive processes.

### Disk Usage Monitoring Module

This module tracks disk space usage and basic read/write statistics. It helps users monitor storage availability and disk performance.

### Process Monitoring Module

This module lists currently running processes along with their resource usage. It helps identify processes that consume high CPU or memory.

### Display Module

The display module combines data from all monitoring threads and updates it live on the screen. It ensures that users can easily view system health information in one place.

## TOOLS AND TECHNOLOGIES

**Programming Language:** Python (or C)

**Operating System:** Windows / Linux

**Libraries:**

- threading or multiprocessing (for multithreading)
- psutil (for system information)

- time, os, sys (for process handling)

**IDE:** VS Code or PyCharm

## WORK PLAN

The project is completed according to the following work plan:

1. Research and planning, and study of multithreading concepts.
2. Setting up the development environment and installing required libraries.
3. Implementation of CPU and memory monitoring modules.
4. Addition of disk usage and process monitoring modules.
5. Designing the display interface.
6. Testing and debugging the application.
7. Documentation and preparation of the final report.

## EXPECTED OUTCOMES

A working multithreaded system health monitoring tool.

Real-time display of CPU, memory, disk, and process information.

Improved performance compared to single-threaded monitoring tools.

Better understanding of multithreading and OS resource management.

A simple and easy-to-use monitoring interface.

## CONCLUSION

The **Multithreaded System Health Monitoring Tool** demonstrates how multithreading can improve system performance and responsiveness in operating system applications. By running multiple monitoring tasks in parallel, the tool provides real-time system information efficiently.

This project successfully combines theoretical OS concepts with practical implementation. It serves as a strong learning example of process management, concurrency, and system resource monitoring, making it a valuable academic project.

# REFERENCES

1. Silberschatz, A., Galvin, P. B., & Gagne, G. *Operating System Concepts*.
2. Python Documentation – Multithreading and psutil library.