



Module Code & Module Title
Level 5 – Operating System

Assessment Type
Logbook 1
Semester 3
2024 Autumn

Student Name: Evani Raut

London Met ID: 23047473

College ID: np01nt4a230151

Assignment Due Date: Tuesday, September 24, 2024

Assignment Submission Date: Monday, September 23, 2024

Submitted To: Umesh Nepal

Word Count (Where Required):2041

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Contents

1.Research on Kernel.....	1
1.1 Aim:.....	1
1.2 Objectives:	2
1.3 Required tools and concepts.....	2
1.4 Steps Required for Lab	6
1.5 Conclusion	7
2.References.....	9

Figure 1:MS Word	2
Figure 2:Google	3

1. Research on Kernel

Introduction

The kernel is the central part of an operating system that directly interacts with the hardware to manage essential system functions. It acts as a bridge between applications and the physical hardware of a computer. It operates at a foundational level, acting as a mediator between software applications and the hardware components of a computer, such as the CPU, memory, and input/output devices. The kernel handles key tasks such as memory management, executing system calls, controlling hardware devices, and facilitating multitasking by managing processes. Kernels are categorized mainly into **monolithic** and **microkernels**. **Monolithic kernels** integrate all essential system services such as memory management, process scheduling, and device drivers within a single, large codebase. This design boosts performance but risks system stability, as a failure in one component can crash the entire system. **Microkernels**, by contrast, keep only essential functions within the kernel and run other services in user space, providing greater security and stability. However, this design can result in slower performance due to the overhead of communication between the kernel and external services (geeksforgeeks, 2024).

1.1 Aim:

This lab aims to explore the kernel, which serves as the foundational component of an operating system, focusing on its functionalities, various types, and its role in managing system resources. The research will cover various kernel architectures, such as monolithic and microkernels, and assess their pros and cons. In addition to studying kernel structures, the lab will examine how these different designs impact the execution of system processes, memory management, and hardware communication. The outcome of this research will ultimately enhance understanding of the kernel's influence on system efficiency, security, and overall performance while also providing insights into real-world applications, such as how different

operating systems implement their kernels for optimized functionality across various platforms and devices.

1.2 Objectives:

Understand the Kernel's Critical Role

This objective focuses on understanding the kernel's fundamental role in the system which is acting as the interface between hardware and software and ensuring the smooth operation of applications.

Discover Various Kernel Models

To investigate different kernel architectures, including monolithic and microkernels, and explore how each structure impacts system performance, security, and adaptability.

Understand Key Kernel Mechanics

To dive into the core operational mechanisms of the kernel, including how it manages system processes, handles memory, and interfaces with devices.

Evaluate the Pros and Cons of Kernel Architectures

Evaluate the pros and cons of monolithic and microkernel designs, comparing how each structure fares in terms of speed, security, and complexity.

Apply Kernel Knowledge to Modern OS Design

To connect theoretical knowledge with real-world applications, exploring how different operating systems like Linux, Windows, and macOS use distinct kernel architectures for optimal performance and user experience.

1.3 Required tools and concepts

Microsoft Word:



Figure 1:MS Word

For this assignment, I used Microsoft Word to effectively organize and present my research. I started by setting up the document with a proper layout, including sections like the cover page, table of contents, and headings for each key topic. Word's formatting tools helped me organize text with consistent fonts, paragraph spacing, and heading styles to make the document visually appealing and readable. I made use of tables and figure tools to insert diagrams and organize information clearly. Overall, Microsoft Word proved to be an efficient platform for compiling and presenting my assignment in a professional manner.

Google:



Figure 2:Google

Google was a key tool in helping me find valuable information and articles about the kernel. It provided access to a wide range of resources, allowing me to easily research different aspects of kernel architecture, functions, and types. By quickly connecting me to relevant materials, Google made the process of gathering data and understanding key concepts much more efficient as it's vast database allowed me to gather a wide range of insights in a short amount of time, streamlining the entire research process.

Concepts

Definition and critical role

A kernel is defined as the central part of an operating system which is responsible for managing communication between software and hardware components, it is the first layer of the operating system that loads into memory and remains there to control essential system tasks like resource allocation, process scheduling, and memory management.

The **critical role** of the kernel is to ensure the efficient and secure functioning of the computer. It acts as a manager for the system's CPU, memory, and I/O devices, deciding how and when to allocate these resources to running programs. It also ensures that multiple programs can run simultaneously without interference by managing resources and enforcing security. Additionally, it enforces system security by controlling user permissions and restricting access to sensitive areas of the system. Overall, the kernel is essential for maintaining stability, performance, and security within an operating system (redhat, 2019).

Types of Kernels

The two primary types of kernel architectures **monolithic kernels** and **microkernels** each differ in structure and functionality. Monolithic kernels integrate all core functions into a single large block of code which often provides faster performance due to direct access to system resources but may cause security risks which as a failure in any component can compromise the entire system. Whereas, microkernels keep only essential functions within the kernel and run other services in user space which enhance security and stability by isolating system processes but may experience performance overhead due to increased communication between components (wikipedia, 2024).

Core Operational Mechanisms of the Kernel

The kernel operates as the central component of an operating system which manages critical tasks that ensure efficient interaction between hardware and software. One of its primary functions is **process management** where it oversees the creation, scheduling, and termination of processes. The kernel uses scheduling algorithms to allocate CPU time to various processes ensuring fair and efficient execution which is essential for multitasking environments (wikiwand, 2024).

In terms of **memory management** the kernel is responsible for managing memory allocation for processes and controlling access to that memory to prevent conflicts. It uses methods like paging and segmentation to make efficient

use of RAM and to maintain system stability, which allows multiple applications to run at the same time without interfering with each other (Sercan Sari, 2024).

The kernel also plays an essential role in interfacing with devices through device drivers, which are accessed via **system calls**. These drivers act as intermediaries between the kernel and hardware components, allowing the kernel to send commands to and receive data from devices like printers, hard drives, and network cards. When an application issues a system call, it requests the kernel to execute specific operations related to these devices. The kernel simplifies device interactions, enabling applications to communicate with hardware without needing to understand the complexities involved. Together, these mechanisms ensure the kernel supports system performance, security, and responsiveness (Alexander S. Gillis, 2024).

Security and Stability

The kernel plays a critical role in maintaining the security and stability of an operating system.

Process Isolation: The kernel ensures that processes are kept separate from one another. Each process operates in its own space, preventing one process from accessing the data or memory of another. This isolation is key in safeguarding the system from malfunctioning or malicious software, preserving system integrity.

Memory Protection: It assigns and protects memory spaces for different processes, ensuring that no process can read or write to memory locations allocated to another process. This protection is crucial for preventing data corruption and maintaining system reliability.

Handling Privileged Operations: Certain operations such as direct hardware access or altering system settings are restricted to the kernel. Applications interact with hardware indirectly through system calls, and the kernel ensures that these privileged operations are executed securely and appropriately. This

control helps to prevent unauthorized programs from making harmful changes to the system.

User Space vs. Kernel Space Security: The kernel operates in a protected area known as kernel space, while applications run in user space. This separation prevents user-level applications from directly accessing critical system resources or hardware, adding an extra layer of security. The kernel manages transitions between user and kernel space, ensuring that only safe and verified processes can make use of system resources (phoenixnap, 2024).

Ensuring System Stability: The kernel enhances system stability by managing process execution and handling errors such as memory access violations or infinite loops. If an issue arises, the kernel can terminate problematic processes or reallocate resources to prevent system slowdowns or crashes, ensuring the operating system remains stable and responsive.

Real-World Kernel Implementations:

Theoretical knowledge about kernel architectures can be vividly illustrated through real-world examples of popular operating systems. For example, Linux, which uses a monolithic kernel showcases how tightly integrated services can enhance performance and efficiency in server environments. Its ability to handle numerous processes simultaneously makes it a preferred choice for high-performance computing. On the other hand, macOS employs a hybrid kernel, combining elements of both monolithic and microkernel designs. This allows it to maintain robust performance while also benefiting from the stability and security features typically associated with microkernels. By examining these real-world applications, we can appreciate how theoretical concepts of kernel design directly influence the performance, security, and user experience of operating systems in everyday use (cyberinsight, 2023).

1.4 Steps Required for Lab

Defining the Scope: The first thing I did was outline the key areas of research that I wanted to focus on, including kernel architecture, functions, and types. I

decided to concentrate on how the kernel handles memory management, process control, and system calls, as these are fundamental for understanding the kernel's role within an operating system. This helped me structure the research and set clear objectives for the lab. By defining these focal points, I was able to narrow down the most critical aspects of kernel operation that I needed to study.

Gathering Resources and Tools: I utilized both online resources and lecture materials to gather the necessary tools and knowledge for this lab. I made extensive use of the lecture slides provided in MST. These slides helped simplify complex concepts. I collected relevant academic papers, textbooks, and online resources on kernel architecture and functionality, which provided the theoretical background needed to fully understand kernel design and performance.

Documentation of Findings: Throughout the lab, I kept a detailed log of every step, observation, and finding. I documented my experiments and noted any differences between the kernel types, including performance metrics and security features. This documentation process was key to ensuring that all observations were well-supported and presented systematically.

Conclusion and Reflections: Finally, after completing the research, I took time to reflect on what I had learned about kernel design and its importance in modern operating systems. I summarized my findings, focusing on how each architecture impacts system performance, security, and adaptability. This research gave me a deeper understanding of the critical role that the kernel plays in system operation, and how its design choices can influence the efficiency and stability of an operating system.

1.5 Conclusion

This research provided a detailed exploration of the kernel's integral role in the operation of an operating system, highlighting its responsibilities in process control, memory management, and device interaction. The investigation into monolithic and microkernel structures helped me understand the key differences

between these architectures and their impact on system efficiency, security, and operational complexity.

Through this research, I've gained a clearer understanding of how the kernel manages resources, processes, and communication with devices. This knowledge has shown me how different operating systems, like Linux and Windows, use these architectures to deliver optimal performance.

Looking ahead, ongoing advancements in kernel design may focus on enhancing security measures, such as improved isolation techniques and better resource management to counteract emerging threats. Additionally, developments in microkernel architectures could lead to greater efficiency and adaptability in handling modern computing demands. Ultimately, the kernel is more than just a component; it is a crucial part of what makes modern computing work effectively, inspiring further interest in how we can improve operating system design.

2. References

- (2019, February 27). Retrieved from redhat:
<https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>
- (2023, June 17). Retrieved from cyberinsight: <https://cyberinsight.co/what-is-kernel-with-example/>
- (2024, July 31). Retrieved from geeksforgeeks: <https://www.geeksforgeeks.org/kernel-in-operating-system/>
- (2024, August 19). Retrieved from wikipedia:
https://en.wikipedia.org/wiki/Kernel_%28operating_system%29
- (2024, March 28). Retrieved from phoenixnap: <https://phoenixnap.com/glossary/what-is-a-kernel>
- (2024). Retrieved from wikiwand:
[https://www.wikiwand.com/en/articles/Kernel_\(operating_system\)](https://www.wikiwand.com/en/articles/Kernel_(operating_system))
- Alexander S. Gillis. (2024). Retrieved from techtarget:
<https://www.techtarget.com/searchdatacenter/definition/kernel>
- Sercan Sari. (2024, June 10). Retrieved from baeldung:
<https://www.baeldung.com/cs/os-kernel>