LONDON METROPOLITAN UNIVERSITY

informatics college • pokhara

Module Code & Module Title

CT5052NI Network operating system

Assessment Type

Individual report writing

Year and Semester

2023-24 Autumn

Student Name: Aman Bahadur Paudel

Group: C6

London Met ID: 23048978

College ID: NP04CP4A230136

Assignment Submission Date: 30/09/2024

Declaration: I understand that I am required to submit my assignment through GitHub, under the appropriate module page prior to the specified deadline, in order for it to be considered for marking. I acknowledge that any assignment submitted after the deadline will be deemed as a non-submission and will not be marked, resulting in a score of zero.

Table of Contents

1. Introduction

Kernel is the core component of any operating system as it has control over everything in the system. It manages the system resource and acts as a bridge between applications and hardware. It controls the execution of applications by handling, managing and controlling the functions like process scheduling, memory management and input/output operations. Whenever the system starts, kernel is the first program that is loaded after the bootloader since it has to handle the rest of the thing of the system for the operating system and remains in the memory until the operating system is shut down.

The major components and functions of Kernel are as follows:

a. System initialization: It is responsible for starting up the operating system after the computer is powered on. This involves a sequence of loading the BIOS, loading bootloader, loading kernel and initializing the device drivers.
b. Device management: It is responsible for managing the interaction between operating system and the hardware devices by loading or unloading the device drivers as per the need for the specific processes.
c. Process management: It is responsible for allocating memory and resources to new processes. It also schedules them to run on the CPU and ensures that the shared resources are accessed in synchronized way.
d. Memory management: It is responsible for allocating or deallocating memory to processes and allows the multiple processes to share the same physical memory.
e. Security and protection: It implements the security features like encryption, firewalls etc. and protects the system resources by granting or denying the access based on the user or process privileges.

2. Objective

The objective of this report are as follows:

a. To explore the historical evolution of kernel.
b. To know the major components and functions of kernel.
c. To understand about the boot process and its relevance to kernel's operation.

3. Types of Kernels

3.1 On the basis of architecture

Monolithic kernel: In a monolithic kernel, user services and kernel services are implemented in the same memory space due to which the size of kernel is increased. By the use of bigger size kernel, the size of operating system also increases. Monolithic kernels achieve better performance due to reduced overhead but are more prone to system crashes if a component fails. (Dr. Barbadekar, et al., 2020)

Microkernel: In microkernel, user services and kernel services are implemented in different spaces. So, the size is relatively smaller, reducing the size of operating systems also. microkernels offer a cleaner separation of concerns, better security, and the ability to scale across multiple hardware platforms. (Manoel, et al., 2017)

3.2 On the basis of Performance and Functionality

Hybrid kernel: The hybrid kernel has aspects of both monolithic and microkernel design, aiming to provide the performance of a monolithic kernel while retaining the modularity of a micro kernel. Core services like memory management and process control remain in kernel space, whereas services like device drivers may run in user space.

Real-time kernel: A real-time kernel is designed to handle operations with strict timing constraints, ensuring that tasks are completed within specific deadlines. They are frequently used in embedded systems and critical applications where predictable and timely responses are essential. (Kumar, et al., 2024)

4. Popular kernels and their history

a. iOS (XNU Kernel)

The kernel used by iOS is called XNU (X is Not Unix), a hybrid kernel that merges components from the Mach microkernel and the FreeBSD monolithic kernel. It was originally developed by NeXT for the NeXTSTEP operating system, and later acquired by Apple when they purchased NeXT in the late 1990s. The XNU kernel is used in both macOS and iOS systems. (Apple, Inc., 2023)

b. Windows (Windows NT Kernel)

The Windows NT kernel is a hybrid kernel, combining microkernel-like modularity with monolithic kernel-like performance. Initially released in 1993, the kernel was designed with portability and extensibility in mind. It introduced new security features and the ability to run across multiple architectures (x86, PowerPC, and others). Windows NT has become the basis for modern Windows systems, from Windows 2000 to Windows 11. (Tanenbaum, 2022)

c. Ubuntu (Linux Kernel)

Ubuntu, a popular Linux distribution, uses the Linux kernel, which is monolithic in nature. Developed by Linus Torvalds in 1991, the Linux kernel is open-source and used across multiple distributions and devices, from servers to embedded systems. It powers a wide range of systems, including Ubuntu, Red Hat, and Android devices. (The Linux Foundation, 2023)

5. Boot process

The boot process is the sequence of events that occur when a computer is powered on, leading to the loading and execution of the operating system. It involves a series of critical steps mentioned below, which ensures the system is ready for use. (Tanenbaum, 2022)

a. Power-On Self Test (POST): When the computer is turned on, BIOS (Basic Input/Output System) performs a POST to ensure that the computer's hardware is functioning correctly. This includes testing the CPU, memory, storage devices, and other peripherals.

b. BIOS Configuration: The BIOS loads its configuration settings, which include boot device priority, time and date, and hardware options. The BIOS then determines the boot device based on the configured priority.

c. Bootloader Loading: The BIOS loads the bootloader, a small program that initiates the loading of the operating system. The bootloader is typically stored on the boot device, such as a hard drive or USB drive.

d. Kernel Loading: The bootloader loads the operating system kernel into memory. The kernel is the core component responsible for managing system resources and providing services to applications.

e. System Initialization: The kernel initializes the system, which includes loading device drivers, mounting file systems, and starting system services. This step prepares the system for user interaction.

f. Login Prompt: The operating system presents a login prompt, allowing a user to enter their credentials and gain access to the system. Once the user is authenticated, the operating system loads the user's desktop environment and applications.

6. Conclusion

The kernel is very essential part of an operating system, acting as the intermediary between hardware and software. Its primary responsibilities include process management, memory management, device control, and ensuring system security. Different types of kernels are available in market that offers distinct approaches to performance, security, and scalability based on their architecture and design. Furthermore, understanding the historical evolution of kernels, from the XNU kernel provides insight into how these systems have shaped modern computing. Additionally, the boot process, which involves the kernel from its initial loading to the initialization of device drivers and system services, emphasizes the kernel's fundamental importance in system startup and operation.

As technology gets better, the way kernels are designed will also change, balancing performance, security, and modularity to meet the demands of various computing environments, from personal devices to large-scale server systems. This report shows how important kernels are in modern operating systems.

7. References

Apple, Inc., 2023. *XNU Kernel Architecture Overview,* s.l.: s.n.

Dr. Barbadekar, A., Gawande, N. & Gitte, A., 2020. Comparison of Monolithic, Micro, and Hybrid Kernel based on Performance and Architectural Parameters. *International Journal,* 9(1).

Kumar, A., Singh, A. K. & Mehta, A., 2024. AI based Operating System. *International Journal of Advanced Research in Science, Communication and Technology,* pp. 291-298.

Manoel, P., R. P., F., Ana Carolina, B. M. & Rangel, A., 2017. Development of a Kernel: A Deeper Look at the Architecture of an Operating System. *International Conference on Embedded Systems and Applications.*

Tanenbaum, A., 2022. *Modern Operating Systems.* 5th ed. s.l.:Prentice Hall.

The Linux Foundation, 2023. *Linux Kernel Documentation,* s.l.: s.n.