Module Code & Module Title

Level 5 – CT5052, Network Operating Systems

Assessment Type

Logbook 1.

Semester

2023/24 Spring/Autumn

Student Name: Asim Sapkota

London Met ID: 23048768

College ID: E.g. NP04CP4A230225

Assignment Due Date: 30 septs 2024

Assignment Submission Date: 30 sept, 2024

Submitted to: Mr Prasant Adhikari

Word Count (Where Required):1457

I confirm that I understand my coursework needs to be submitted online via Google 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.

TABLE OF CONTENTS

TABLE OF FIGURES:

OPERATING SYSTEM-KERNEL

INTRODUCTION:

The operating system is the one program running at all times on the computer usually called the kernel. Along with the kernel, there are two other types of programs: system programs, which are associated with the operating system but are not necessarily part of the kernel, and application programs, which include all programs not associated with the operation of the system. (Abraham Silberschatz, Kernel, 2018)

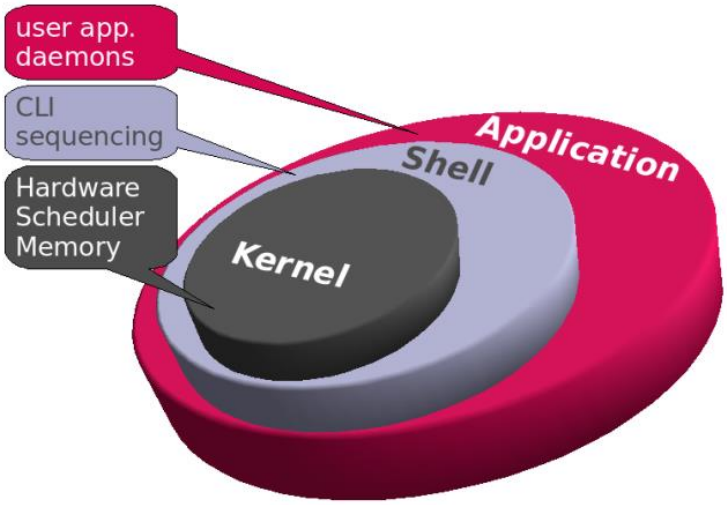

Figure 1: figure of Kernel

Major Components and Functions of a Kernel:

- Process Management: It controls the process creation, process scheduling, and process termination.
- Memory Management: The allocating and deallocating in memory.
- Device Management: This controls and communicates with the hardware devices comprising the communities through device drivers.
- File System Management: It deals with files on storage devices for maintenance and access.
- Security and Access control: Ensures that access to system resources is not given to unauthorized users.

OBJECTIVE:

The objective of this report is to provide a comprehensive understanding of the operating system kernel, its types, and its role in popular operating systems. Additionally, the report will explore the boot process and provide references from recognized textbooks and peer-reviewed papers.

TYPES OF KERNEL:

1. **Based on Functionality:**

   Monolithic Kernel:

By far the most common organization, in the monolithic approach the entire operating system runs as a single program in kernel mode. The operating system is written as a collection of procedures, linked together into a single large executable binary program. When this technique is used, each procedure in the system is free to call any other one, if the latter provides some useful computation that the former needs. Being able to call any procedure you want is very efficient, but having thousands of procedures that can call each other without restriction may also lead to a system that is unwieldy and difficult to understand. Also, a crash in any of these procedures will take down the entire operating system. (Tanenbaum, Monolithic Kernel , 2014)

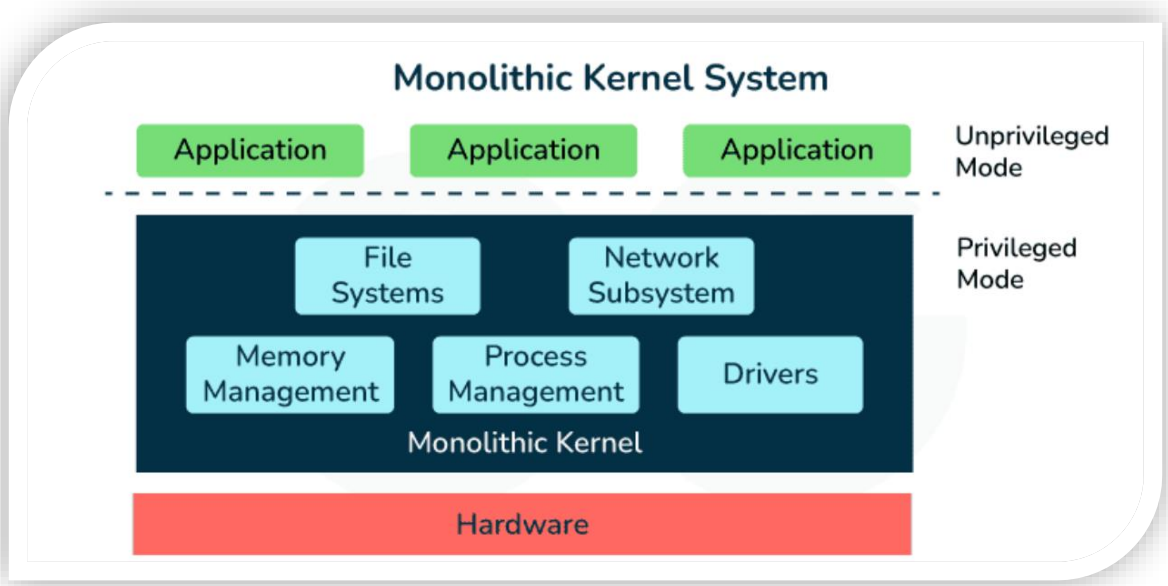

Figure 2 : figure of monolithic kernel system

Advantages of Monolithic Kernel:

- A monolithic kernel is fast because services like memory management, file management, etc. are implemented in the same address space.
- A process runs entirely in a single address space in monolithic kernels.
- A monolithic kernel is a single static file.
- Few bugs and security problems are also less.

Disadvantages of Monolithic Kernel:

- If any service fails, then the entire system fails too.
- If any new features are added, then the issue is to modify the complete system.
- Coding and debugging in the kernel space is difficult.

Micro Kernel:

The basic idea behind the microkernel design is to achieve high reliability by splitting the operating system up into small, well-defined modules, only one of which—the microkernel—runs in kernel mode and the rest run as relatively powerless ordinary user processes. In particular, by running each device driver and file system as a separate user process, a bug in one of these can crash that component, but cannot crash the entire system. Thus a bug in the audio driver will cause the sound to be garbled or stop, but will not crash the computer. (Tanenbaum, Micro Kernel, 2014)

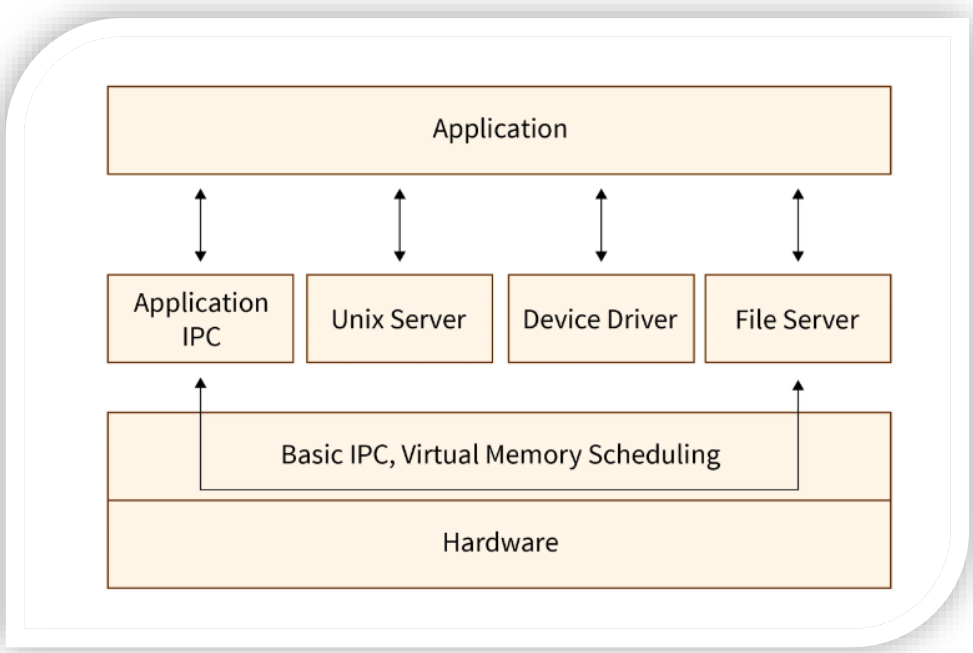

Figure 3: figure of micro kernel

Advantages of Micro kernel:

- Microkernels are secure as only those parts which might influence the system's functionality are added.

- The various modules of a microkernel can be swapped, reloaded, and modified without affecting the kernel.
- A microkernel architecture is compact and isolated, so it performs better.

Disadvantages of Micro kernel:

- When the drivers are implemented as procedures, a context switch or a function call is needed.
- In a microkernel-based OS, providing services is costlier in comparison to a monolithic system.
- The performance of a microkernel system might stay uniform and cause issues.

2. Based on Structure:

Hybrid Kernel:

Hybrid kernels blend monolithic and microkernel features, extending the microkernel idea with added code in kernel space and some monolithic kernel attributes for enhanced performance.

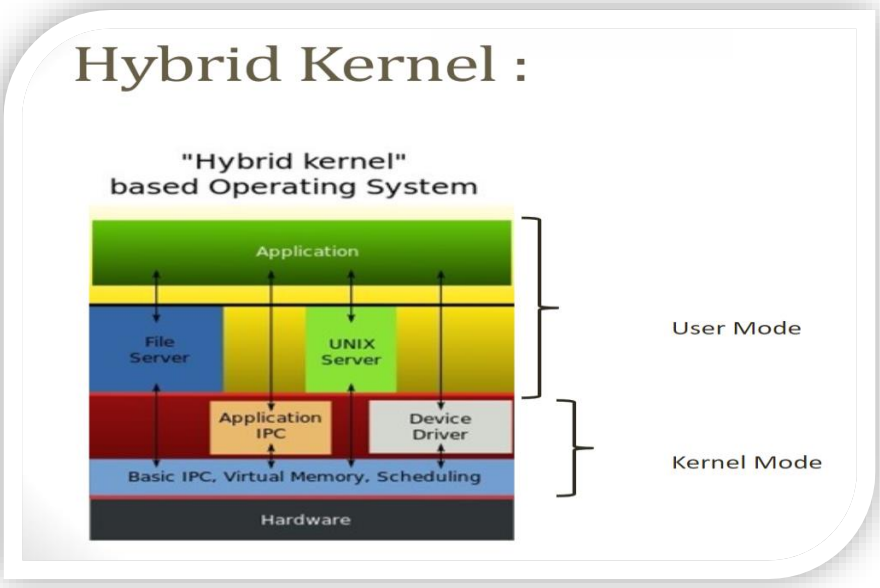

Figure 4: figure of Hybrid kernel

Advantages of hybrid kernel:

- A hybrid kernel is easy to manage due to its layered approach.
- Number of layers is not very high generally.
- Kernel is small in size and isolated.
- Provides better security and protection.

Disadvantages of hybrid kernel:

- Since a hybrid kernel strikes a balance between the gains in monolithic kernel and microkernel, there are certain benefits of both that are lost out or compromised.

Exo Kernel:

At the bottom layer, running in kernel mode, is a program called the exokernel (Engler et al., 1995). Its job is to allocate resources to virtual machines and then check attempts to use them to make sure no machine is trying to use somebody else's resources. Each user-level virtual machine can run its own operating system. (Tanenbaum, Exo kernel, 2014)

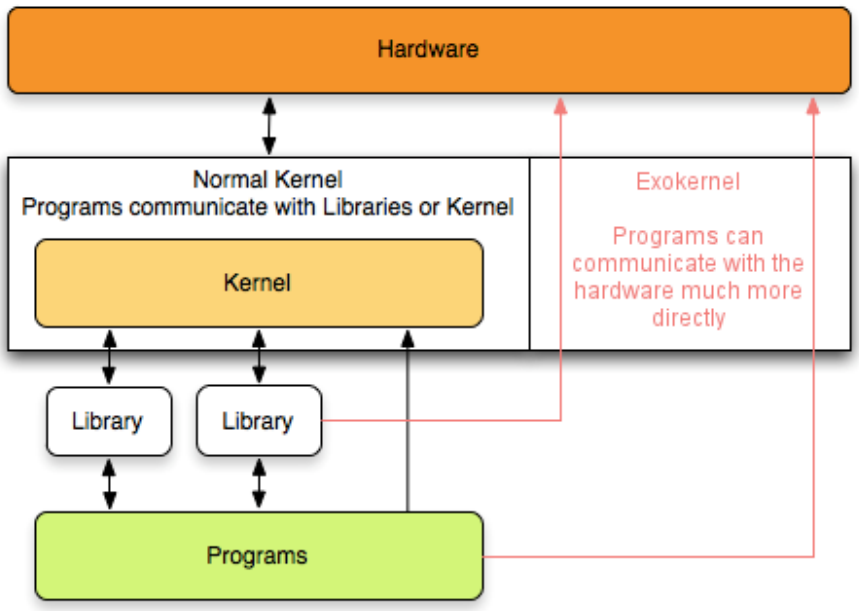

Figure 5: figure of Exo kernel

Advantages of Exo kernel:

- Better Support for Application Control.
- Separates Security from Management.
- Abstractions are moved securely to an untrusted "Library Operating System".
- Provides a low-level interface.
- Library operating systems offer Portability and Compatibility.

Disadvantages of Exo Kernel:

- When it comes to application/OS development, each developer will develop them the way they see fit. Due to this reason, the level of consistency will be a bit reduced.
- Design of the interface in an exokernel is complicated

POPULAR KERNELS AND THEIR HISTORY

1. iOS:

   Popular Kernel: XNU (X is not Unix)

   History: The XNU kernel was originally developed by NeXT for the NeXTSTEP operating system. After Apple acquired NeXT in 1996, XNU was adapted for use in macOS (formerly Mac OS X) and later iOS, iPadOS, watchOS, and tvOS .

2. Windows

   Popular Kernel: NT kernel

   History: Designed at Microsoft by the man generally regarded as the father of Windows NT, Dave Cutler, it was first released in 1993 with Windows NT 3.1. Being

a hybrid kernel, it has been designed to be portable across different hardware architectures; besides, it should provide support for multiprocessing and multi-user systems.

3. Ubuntu

   Popular Kernel: Linux Kernel

   History: At the core of the Ubuntu Operating System is called the Linux kernel. In 1991, the driving head was Linus Torvalds, whereas Ubuntu is one of the more famous distributions of Linux first released in 2004 by a company named Canonical Ltd. The Linux kernel has been in heavy development over the years by literally thousands of developers.

BOOT PROCESS:

Some computer systems use a multistage boot process: When the computer is first powered on, a small boot loader located in non-volatile firmware known as BIOS is run. This initial boot loader usually does nothing more than load a second boot loader, which is located at a fixed disk location called the boot block. The program stored in the boot block may be sophisticated enough to load the entire operating system into memory and begin its execution. More typically, it is simple code (as it must fit in a single disk block) and knows only the address on disk and the length of the remainder of the bootstrap program. Many recent computer systems have replaced the BIOS-based boot process with UEFI (Unified Extensible Firmware Interface). UEFI has several advantages over BIOS, including better support for 64-bit systems and larger disks. Perhaps the greatest advantage is that UEFI is a single, complete boot manager and therefore is faster than the multistage BIOS boot process. Whether booting from BIOS or UEFI, the

bootstrap program can perform a variety of tasks. In addition to loading the file containing the kernel program into memory, it also runs diagnostics to determine the state of the machine for example, inspecting memory and the CPU and discovering devices. If the diagnostics pass, the program can continue with the booting steps. The bootstrap can also initialize all aspects of the system, from CPU registers to device controllers and the contents of main memory. Sooner or later, it starts the operating system and mounts the root file system. It is only at this point is the system said to be running. (Abraham Silberschatz, Boot Process, 2018)

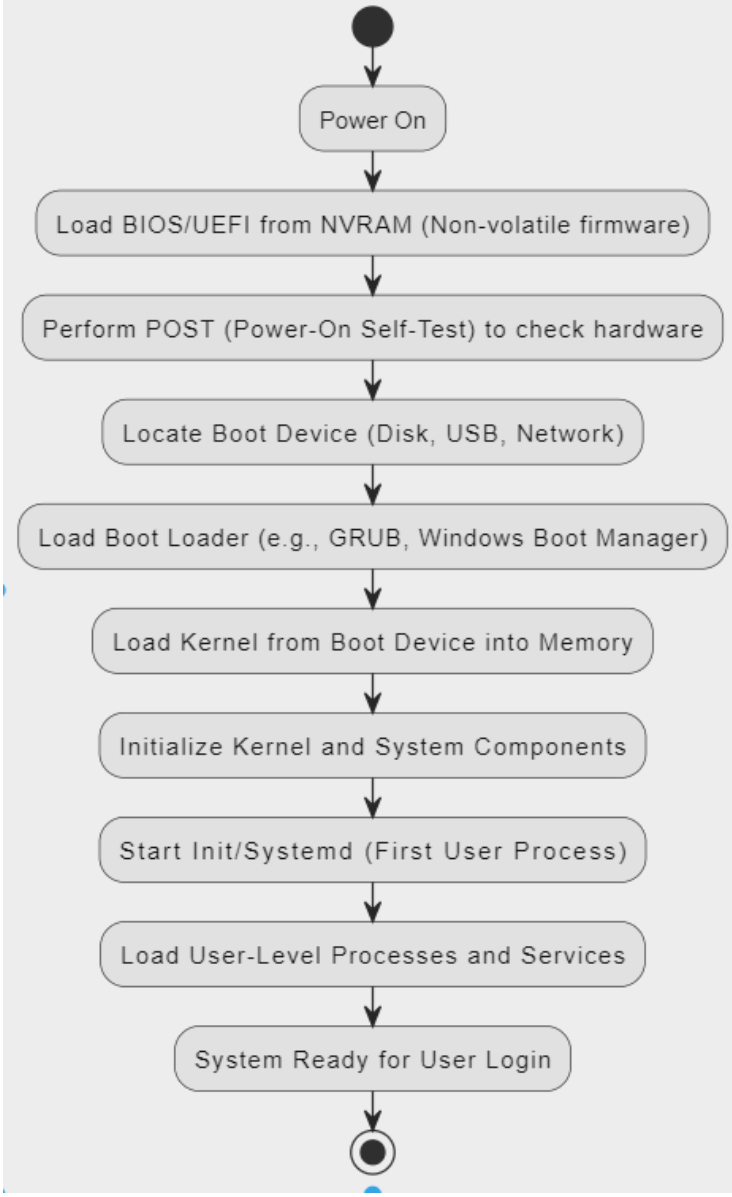

Figure 6: Boot Process

CONCLUSION:

Every operating system primarily originates from its kernel, which integrates both the hardware and software to provide that the system undergoes productive and safe wear. The kernel is responsible for functions such as process scheduling, memory allocation, communication with devices, management of file systems, etc.; in short it is the heart and soul of processing in a computer in terms of both performance and stability. Knowing the different types of kernels, regardless of the fact, monolithic or microkernel, as well as the pros and cons, is what characterizes the philosophy of various operating systems. The kernel, inspired by the historical background of iOS, Windows, and Ubuntu should be made a kernel with the capability to offer more functionalities, safety and user experience. As a consequence, the introduction of the kernel is given by the moment of the starting of computer through the initialization process which is another name for boot process. In short, the detailed process of the kernel and advancement shows its involvement in the computer age we now have.

REFERENCES

Abraham Silberschatz, P. B. (2018). Boot Process. In *Operating System Concepts.* Wiley.

Abraham Silberschatz, P. B. (2018). Kernel. In *Operating System Concepts.* Wiley.

Tanenbaum, A. S. (2014). Exo kernel. In *Modern Operating System.* Pearson.

Tanenbaum, A. S. (2014). Micro Kernel. In *Modern Operating System.* Pearson.

Tanenbaum, A. S. (2014). Monolithic Kernel . In *Modern Operating System.* Pearson.