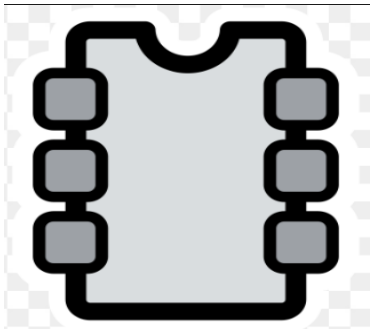
CT5052NP Network Operating System (NOS)

Submitted by		Submitted to	
Name:	Shusil Bk	Name:	Prasant Adhikari
London met id:	23048939	Deadline:	Sept-30,2024
Group:	C6 2nd year		

1.Kernel



1.1 Introduction

Kernal is the center part of Os (Operating system), which acts as a bridge between hardware and software. It helps to manage system resources such as memory, processor, and input/output devices securing the smooth functioning of different software. It plays a vital role in managing system resources, multiple works and creating a protected

environment for running the applications. It executes at a lower level, interacting directly with the hardware and controlling all the activities of the system.

1.2 Major functions and components of Kernal

1. Managing process: It manages procedures, creating them, planning them and completing them as required. It allocates CPU time to process, which allows their different operation (multitaksing) and securing proper execution through the transition.

2. Manages memory: It handles memory allocation and reallocation. It ensures that each applicant has sufficient memory without being overloaded by others. It also supports virtual memory, where the hard drive is used as an extension of RAM, making the system appear to have more memory than it has.

3. Device management: Kernel manages input/output devices through device drivers. This interface allows software to interact with hardware, allowing software to operate independently of the specific configuration.

4. File management: It manages the file system, allowing users and retrieving data in a structured manner. It also ensures secure access to files and preservation of data integrity in the system.

5. Security with access control: Kernels implement security measures to prevent unauthorized access to system resources. They also use access control mechanisms to restrict the actions of applications and users with specific system files, thus ensuring data protection and system stability.

6. Handling interruptions: The signal from computer devices that require immediate attention. The core handles these interruptions by interruptions by interrupting the current work, responding to the interruption, and then starting the work again.

1.3 Objectives of Kernel

The objectives of the kernel are:

1. Efficiency: It is designed to manage the system resources efficiently to ensure optimal performance of the system. It balances the CPU, memory and other resources to avoid deadlocks.

2. Security: The kernel is responsible for maintaining system security, ensuring that unauthorized users or procedures cannot access sensitive areas of the system.

3. Stability: One of the high main goals of the Kernel is to ensure that the system runs smoothly, even under high load conditions. It handles errors, preserves system integrity and efficiently distributes resources to avoid system failures.

4. Portability: Most kernels are designed to be portable, meaning they can run on different computer platforms without undergoing major changes. This allows the use of the same kernel on different devices, such as personal computers, servers, and mobile devices.

2. Types of Kernels:

Based on its architecture:

i. **Monolithic kernels**: All essential functions such as process management, memory management, and device drivers are integrated into a large codebase. This design facilitates fast execution by reducing transitions between the user and the kernel space. However, its maintenance can be difficult sometimes due to its size. For example, Linux, Unix etc.

Advantages:

1. Higher performance with minimal context transition.
2. Direct access to the hardware.

Disadvantages:

1. Difficult to debug and maintain.
2. More modular and flexible.

ii. **Microkernel:** A microkernel architecture keeps only essential components such as process and memory management, while other services run in user space as separate operations. This design enhances security. For example, QNX and Minix.

Advantages:

1. Better stability and security.
2. Easy to extend and modify.

Disadvantages:

1. Slower pperfomance due to switching between kernel and user.
2. More complex communication.

Based on its Modularity:

i. **Modular Kernel**: A modular Kernel allows dynamic loading of components such as device and system services. This flexibility means that the system is adapted to change the environment of hardware and software without any need to recompile. For example, Modern Linux, Kernels etc.

Advantages:

1. Flexibility, where new modules can be added without rebooting.
2. Helps to manage the system resource by replacing the necessary modules.

Disadvantages:

1. Slightly lower performance due to dynamic loading of modules.
2. Security risks from poorly managed modules.

ii. **Hybrid Kernel**: A hybrid Kernel combines elements of monolithic and microkernel. It retains the performance benefits of a monolithic kernel while incorporating microkernel elements for better security and modularity. For example, Windows NT, macOS (XNU).

Advantages:

1. There is a balance between flexibility and modularity.
2. Increased isolation of system components enhances security.

Disadvantages:

1. More complex than pure monolithic or microkernel design.
2. Lower performance than of a fully monolithic Kernel.

3. Popular kernels and their history

Different operating systems use different types of kernels depending on the requirements of design and architecture. We examine three well known kernels and used by popular OS:

1. Windows Kernel (NT Kernel): The NT kernel is used by Microsoft windows, which was introduced in 1993 with Windows NT 3.1. It is a Hybrid Kernel that combines elements of monolithic and microkernel designs, resulting in improved security. The NT kernel is essential in all current versions of windows, including Windows 10 and Windows 11. It has evolved to support new technologies and security while maintaining backwards compatibility.

2. IOS (XNU Kernel): Apple's iOS is based on the XNU (X is Not Unix) Kernel, originally designed for the NeXTSTEP OS. After Apple acquired NeXT in 1997, XNU became the foundation for macOS and later iOS. The XNU core is a combination of elements from march microcore and components from BSD (a variant of Unix). This architecture allows for felicity, making it an ideal choice for flexibility, making it an ideal choice for both home and mobile platforms.

3. Ubuntu (Linux Kernel): Ubuntu, a Linux based operating system, uses the Linux core, originally developed by Linus Torvalds in 1991. The Linux kernel adopts a monolithic design, with all essential functions running in the same memory space. Due to its open-source nature, it has been widely adopted and constantly

developed by the global community. Ubuntu, like other Linux distributions, benifits from the flexibility and customizability of Linux Core.

4. Boot process

1. Power on: When you press the power button, the computer gets electricity to start up.

2. POST (Power-ON Self-Test): The computer starts to check its components like storage and memory to make sure that everything is working properly.

3. Finding Bootloader: Here it looks for a special program called bootloader (like G|RUB for Linux or Windows Boot Manager) on the hard drive.

4. Loading kernel: The bootloader finds and then loads the computer's Kernel (the heart of the computer) into system memory.

5. Hardware Initialization: The Kernel starts running and configures the hardware (CPU, memory, etc.) to prepare everything for use.

6. Start User Space: Finally, the operating system's graphical interface or command line begins, allowing the use of the Operating System.

This is the way the boot process looks like.

Conclusion

In conclusion, the kernel is the heart of the computer which allows communication between software and hardware. It has a vital role in management in maintaining security, managing system resources and ensuring efficient operations of hardware and software. There are so many types of kernels which have already been discussed in our project.

It was a great experience doing this project which enhances my researching skills on different platforms like google scholar, ellicit.com, YouTube etc. To be honest sorry to say, but I have not gone through any books, but I ensure you in the next project i will also do research from the books. There were several problems while doing the project, I have gone

through much research to find the answer. It was a little difficult to finish but I enjoyed it very much.

References:

i. Peterson, J.L. and Silberschatz, A., 1985. Operating system concepts. Addison-Wesley Longman Publishing Co., Inc.. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Operating+System+Concepts+by+Silberschatz&btnG=

ii. Tanenbaum, A., 2009. Modern operating systems. Pearson Education, Inc.,. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Modern+Operating+Systems+by+Tanenbaum&btnG=

iii. Stallings, W., 2008. Operating systems: internals and design principles. Prentice Hall Press. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Operating+Systems+Internals+and+Design+Principles+by+Stallings&btnG=

iv. Love, R., 2010. Linux kernel development. Pearson Education. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Linux+Kernel+Development+by+Robert+Love&btnG=

v. Mauerer, W., 2010. Professional Linux kernel architecture. John Wiley & Sons. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Professional+Linux+Kernel+Architecture+by+Mauerer&btnG=