Comparison of monolithic, Micro, and Hybrid kernel based on performance and architectural parameters

[1]Prof (Dr) Ashwinee Barbadekar, [2]Nikita S. Gawande, [3]Asmita Gitte

Department of Electronics and Telecommunication
VIT, Pune.

Abstract– **In an operating system, the kernel is the core component responsible for managing hardware resources and it provides an essential service to another system. kernel serves as a bridge between software applications and the computer's hardware, enabling programs to communicate with devices like the processor, memory, and peripherals. This paper contains information on the comparison of different types of kernels including Monolithic kernels, Microkernels, and Hybrid kernels. The kernel provides functions such as process management, memory management, device drivers, and system call handling. A monolithic kernel is based on Operating System. Monolithic kernels have OS service and kernel in different addresses whereas microkernels have the same address. The hybrid kernel is a combination of both monolithic (for additional code in kernel space) and microkernel (for enhancing the performance of kernel attributes).**

Keywords- **kernel, memory, processors, monolithic, microkernel, hybrid.**

I. INTRODUCTION

An operating system is a component of software that manages the execution of application programs. Also, It provides an interface to the (virtual machine) user and hides the complexity of the software.
The only program that runs continuously on a computer is the operating system, generally termed the kernel. It acts as a connection between the software and hardware components. It is a low-level program that controls the system's resources, including memory, CPU, I/O devices, and other hardware peripherals, providing a communication interface between them. The kernel is responsible for managing the system's memory by allocating and deallocating it to different user applications. In this paper, we discuss the monolithic kernel, microkernel, and hybrid kernel.
A monolithic kernel is an architecture in which the complete operating system runs as a single, unified program in kernel space. This approach contrasts with a microkernel architecture, in which device drivers and other services run as separate, user-level processes that communicate with the kernel through well-defined interfaces. Monolithic kernels are often used in traditional desktop and server operating systems, such as Linux, Windows, and macOS.

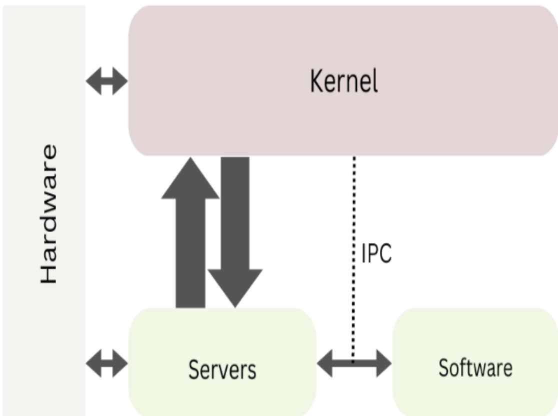

Fig.1 Monolithic kernel

A microkernel is a type of operating system kernel that provides a low-level, minimal set of functions and services to perform basic system operations, such as memory management, inter-process communication, and device driver support. The microkernel architecture differs from the traditional monolithic kernel architecture, which implements all system functions as a single, large code base. In a microkernel-based system, the majority of services, including file systems, network protocols, and user interfaces, are implemented as separate modules that run as user-level processes outside of the kernel.

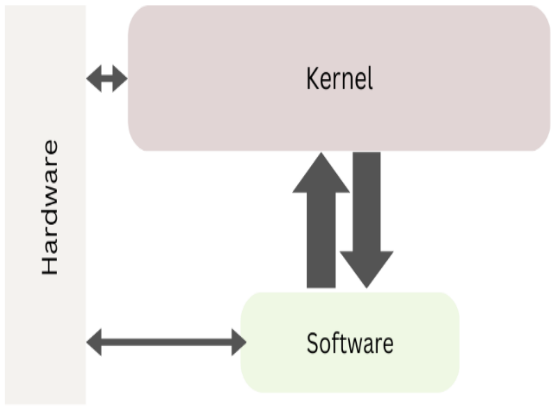

Fig. 2 Microkernel

A hybrid kernel is a type of operating system kernel that incorporates ideas from both the monolithic and microkernel computing architectures. Hybrid kernels aim to achieve a balance between the benefits of the two types of kernel structures while avoiding their potential drawbacks. Such kernels are considered a hybrid because they can switch between running certain tasks in kernel space and others in user space, offering greater flexibility and performance than either monolithic or microkernel designs alone.

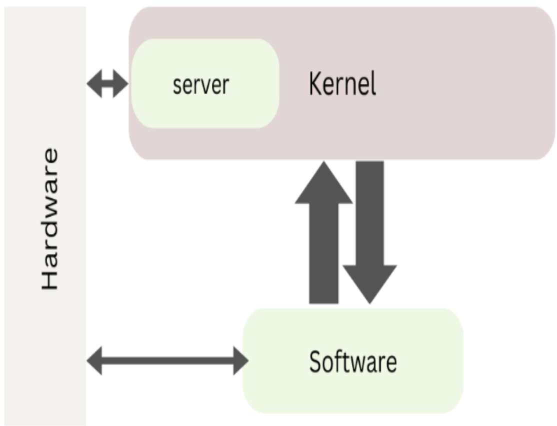

Fig.3 Hybrid Kernel

II. LITERATURE SURVEY

In Less is More – A Secure Microkernel-Based Operating System [1] author discusses the security mechanism of microkernels and also discusses the High-Level software and Low-level software. this paper is just an overview of the operating system developed at TU Dresden. The core functionality of this system is explained in this paper such as Kernel, L4 Runtime Environment, and Platform Management. The paper [1] and [8] discuss the security of microkernels. So, In the operating system, the kernel is the main component that is dependent on the services being affected.

So, in the paper “the Microkernel concept based Dependable system architecture” [2] author describes the dependable system architecture. They decouple an operating system so that fault errors in this system can not affect all dependent applications. Because of this proposed system enable more reliability and dependability. This system will help in constructing a reliable system architecture. Now, we can see the research about the hybrid kernel. The hybrid kernel is a combination of a monolithic kernel and a microkernel.

In [3] “Architecture of Building Hybrid kernel-User Space Virtual Network Functions “the author explains Network Functions Virtualization which is an important aspect of modern network architecture. NFV decouples the decouples NFs
from hardware. In the paper [5] “Design and Implementation of a Novel Embedded Real-Time Kernel Based on Hybrid Architecture” the author Weihua XU, Qiang Huang discusses the real-time kernel-based hybrid architecture. Time-trace and task-overrun handling methods are used in this system.

In the [5] “Enhanced Security of Building Automation Systems Through Microkernel-Based Controller Platforms” Xiaolong Wang, Richard Habeeb, Xinming Ou explain about the security enhancement in microkernels. How to enhance the security of microkernel which is one type of Operating system. Basically, they describe in three ways 1) using microkernel MINIX 3. 2) using sel4 3) using Linux. [6].

In [9] paper "Evaluation of Processing Distribution for Application Program and OS in Microkernel OS" Yuya Kobayashi, Masaya Sato, and Hideo Taniguchi discuss the two different operating systems as monolithic kernel and microkernel in terms of the Application program(AP) and Operating system(OS). They say that ISPC is an overhead of approximately 5 ms or less than Ant [9].

In many research papers authors discuss the architecture and performance parameters of a microkernel, monolithic kernel, and hybrid kernel. The hybrid kernel is a combination of the micro and monolithic kernel. Its corporates ideas of both the kernel.

III. COMPARISION PARAMETER

1. Architectural Parameter:

In this section, we are discussing architectural parameters for the monolithic kernel, microkernel, and hybrid kernel. This is an just overview of the different kernels of architecture based on research papers.

*A. **Monolithic Kernel Architecture:***

In the monolithic kernel architecture, as shown in(fig. 4) all operating system services are included in the kernel, such as device drivers, file systems, and network protocols. Since all services are tightly integrated, the monolithic kernel provides better performance than the microkernel. However, this results in a larger kernel size and decreased flexibility.as we can see in fig. 4 all the methods are in kernel mode.

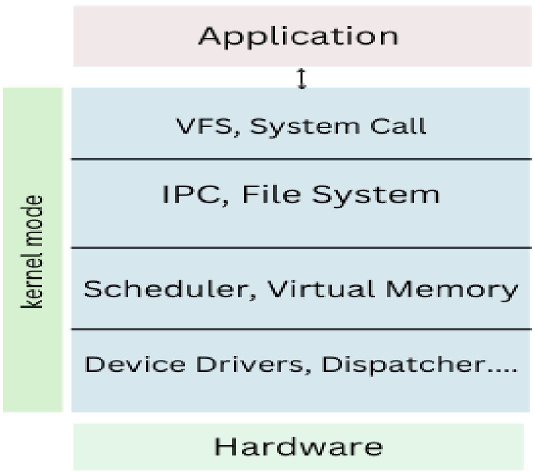

Fig. 4 Architecture of monolithic kernel

*B. **Microkernel Architecture**:*

The microkernel architecture follows the principle of minimalism, (as shown in fig.5) in which only the most essential functions are included in the kernel. The microkernel provides basic functionalities, such as scheduling, memory management, and inter-process communication (IPC), and all other services are implemented as user-level processes. This design allows for greater flexibility and easier maintenance as it separates the kernel from the operating system services.

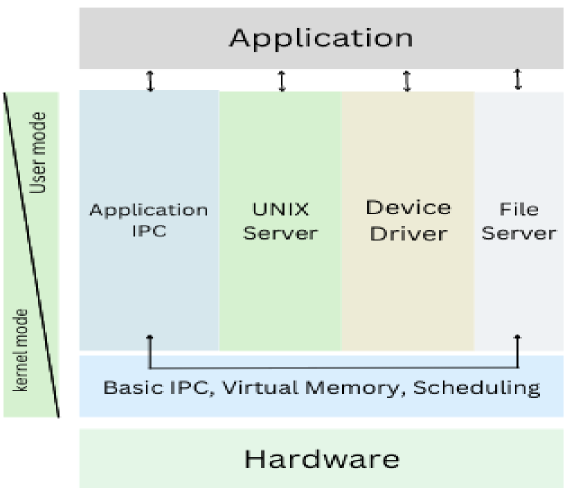

Fig. 5 Architecture of microkernel

C. ***Hybrid Kernel Architecture:***

(In fig. 6)The hybrid kernel architecture combines the features of both the microkernel and monolithic kernel architectures. The hybrid kernel provides the key benefits of the microkernel, such as separation of services and easier maintenance, while also providing the performance advantages of the monolithic kernel. In the hybrid kernel architecture, some operating system services are implemented as user-level processes, while others remain in the kernel. This results in a smaller kernel size, better modularity, and improved performance.

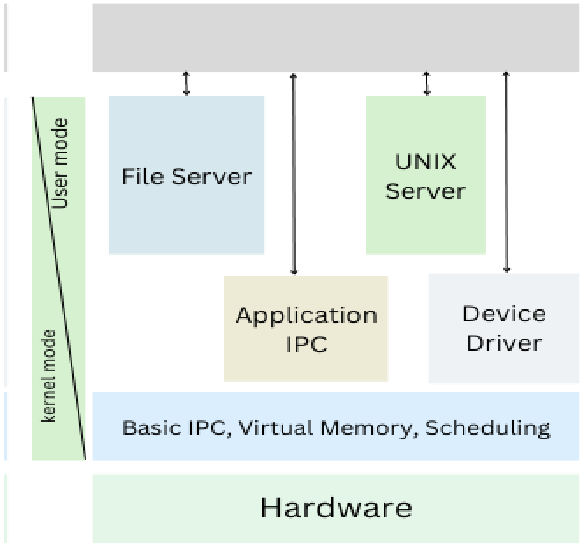

Fig. 6 Architecture of Hybrid Kernel

As a result, each kernel architecture has its advantages and disadvantages. The microkernel architecture provides greater flexibility and easier maintenance, while the monolithic kernel architecture offers better performance at the expense of larger kernel size. The hybrid kernel architecture provides the best of both worlds by combining the benefits of the microkernel and monolithic kernel architectures.

2. Performance Parameter

2.1 Inter-Process communication

Inter-process communication (IPC) refers to the way in which different processes in an operating system communicate with each other. In microkernel, monolithic kernel, and hybrid kernel architectures, IPC can be implemented differently. The choice of the IPC method depends on the specific requirements of the operating system and the trade-offs between security, reliability, and performance.

In a microkernel architecture, IPC is generally implemented through message passing, where messages are sent between processes through a separate messaging service provided by the microkernel. This approach is often more secure and reliable than other methods, as processes are isolated from each other and message passing requires explicit permission to communicate.[1]

In a monolithic kernel architecture, IPC is generally implemented through shared memory, where processes can access the same regions of memory to communicate. This
approach is generally faster than message passing but can be more prone to security vulnerabilities as processes have greater access to each other's memory.

Performance Parameter	Monolithic Kernel	Microkernel	Hybrid kernel
Efficiency	More efficient operation as all the components	It is low overhead and high efficiency.	It is known for their efficient operation.
Complexity	More complex	Less Complex	less complex
Stability	less stable	less stability, reliability	more stable

Security	Less secure	more secure than traditional monolithic kernels.	More Secure
Speed	faster in terms of system performance as all the components	faster performance	slower than a pure monolithic kernel but faster than a pure microkernel

In a hybrid kernel architecture, IPC can be implemented using a combination of both message passing and shared memory, depending on the needs of the system. For example, message passing might be used for secure communication between processes, while shared memory might be used for high-speed communication.

2.2 Memory Management

Memory management is a crucial aspect of modern operating systems. Several kernel designs such as monolithic kernel, microkernel, and hybrid kernel have been proposed and implemented. the monolithic kernel design has the highest performance in memory management, while the microkernel design has the lowest performance. The hybrid kernel design offers a balance between the two designs. This paper examines how memory management is implemented in these three types of kernels and compare their performance.

In a *monolithic kernel* design, the kernel is a single large executable file that runs in kernel mode. The monolithic kernel is responsible for everything from process scheduling to device drivers. In a monolithic kernel, memory management is tightly integrated with other kernel functions.
The monolithic kernel uses virtual memory to manage memory allocation, the page fault mechanism to manage swapping and paging. The monolithic kernel design has
several advantages over other designs because it offers better performance over other kernel designs.

A *microkerne*l design separates the kernel into small, distinct components. A microkernel design moves several services, including device drivers and file systems, from the kernel to user space. Memory management in a microkernel design is simplified because most of the operating system services run in user space. The microkernel design uses interposes communication (IPC) to communicate between the user-level and kernel-level components. The microkernel design also uses virtual memory and page faults to manage memory allocation and paging.

The *hybrid kernel* design is a combination of both monolithic and microkernel designs. The hybrid kernel design is based on the microkernel design, where the kernel is separated into small components. However, unlike in the microkernel design, some services such as device drivers and file systems are implemented in kernel space. The hybrid kernel design uses IPC to communicate between user and kernel-level services. Memory management in the hybrid kernel design is more complex than in the microkernel design but simpler than in the monolithic kernel design.

2.3 Security and Stability

In monolithic kernel has all the functionality of the operating system in a single address space, and all the services run in kernel mode with full privileges. This design makes monolithic kernels less secure and less stable than other architectures because any error in the kernel can bring down the entire system. Additionally, monolithic kernels have a large code base, which increases the likelihood of vulnerabilities and exploits. However, monolithic kernels offer good performance because system calls,
and kernel services are all handled in a single layer.

In contrast to monolithic kernels, a microkernel design keeps only the most basic functionality in kernel mode, while user-level processes handle the rest of the operating system services. This design reduces the attack surface and provides a more secure and stable system. In addition, the small size of the kernel code base reduces the likelihood of vulnerabilities. However, the performance of a microkernel can be slower than a monolithic kernel due to the overhead of communication between user-level processes.

A hybrid kernel combines the best of both worlds by incorporating both monolithic and microkernel designs. In a hybrid kernel, some services run in kernel mode, while others run in user mode. This allows for better performance than a microkernel without compromising security and stability.

2.4 Extensibility and Stability

Monolithic Kernel:

It is difficult to extend the monolithic kernel as all the modules are tightly integrated and depend on each other. Adding a new feature or module requires recompiling the entire kernel which may cause compatibility issues.

Monolithic kernels are less portable as they are designed for specific hardware architectures and require hardware-specific device driver implementations.

Microkernel:

Microkernel architecture allows easy extensibility as all the services are provided via separate user-level processes. Adding new features or services is as simple as adding a new process without changing the kernel code.

Microkernels are more portable than monolithic kernels as they are designed to be hardware independent. They can be easily ported to different hardware architectures with changes only required for the device drivers.

Hybrid Kernel:

Hybrid kernels provide a balance between extensibility and performance. The core services of the kernel are kept in the kernel space for better performance, while the less critical services are provided via user-level processes for easy extensibility.

Hybrid kernels are generally less portable than microkernels but more portable than monolithic kernels. They require hardware-specific device drivers but can be easily ported to different hardware architectures with minimal changes**.**

IV. CONCLUSION

In this paper, the comparison between the monolithic kernel, micro kernel, and Hybrid kernel is based on architecture and performance parameters. These three operating system architectures have many benefits and limitations. So, from that, we can't Define which is better and not. In the architecture parameters, the monolithic kernel runs in kernel and user services at the same address space, and in the microkernel run user and kernel service in different address spaces.

In Microkernels IPC is used for the messaging queues. On the other hand, monolithic kernels use sockets and signals to achieve IPC. The monolithic kernel design has the highest performance in memory management, less secure and stable, while the microkernel design has the lowest performance. The hybrid kernel design offers a balance between the two designs.

REFERENCES:

1. Adam Lackorzynski, Alexander Warg "Less is More – A Secure Microkernel-Based Operating System", 2011 IEEE.
2. Tae-You Lee, Yong-Ki Ku, Choon-Sung Nam, Dong-Ryeol Shin , "The microkernel concept based dependable system architecture" 2009 IEEE
3. Nguyen Van Tu, Kyungchan Ko, and James Won-Ki Hong, " Architecture for Building Hybrid Kernel-User Space Virtual Network Functions", 2017
4. Yan Liu and Ted Wong, "Component Architecture and Modeling for Microkernel-based Embedded System Development ", 2008 IEEE
5. Weihua Xu , Qiang Huang , Xiaofeng Liang , "Design and Implementation of a Novel Embedded Real-Time Kernel Based on Hybrid Architecture"
6. Xiaolong Wang∗, Richard Habeeb∗, Xinming Ou, "Enhanced Security of Building Automation Systems Through Microkernel-Based Controller Platform", 2017 IEEE
7. Susmit Bagchi, "Distributed IPC using Virtual Device Driver in Monolithic Kernel", 2012 IEEE
8. hosein marzi1 , yuntao yang, "microkernel-based real-time embedded operating systems efficiency improvement", 0 ©2012 IEEE
9. Yuya Kobayashi, Masaya Sato, Hideo Taniguchi " Evaluation of Processing Distribution for Application Program and OS in Microkernel OS ", 2019 IEEE
10. Dr. Jordan Shropshire " Analysis of Monolithic and Microkernel Architectures: Towards Secure Hypervisor Design ", 2014 IEEE

!