**A Survey on Microkernel IPC Real-Time Signal and Shared Memory in Operating System**

**ABSTRACT**

All system resources are managed by the microkernel, but the user and kernel services are handled in different address spaces. The system's flexibility and security are provided by the microkernel operating system. Multiple Microkernel operating systems include **L4**, **L4FIFO**, **Minix3** Kernel. Microkernel has the software that is necessary to construct an operating system. These constructs include thread control, address control and inter process communication (**IPC**). A communication between process through **shared memory** and **address-mapping** method to achieve inter-process communication. Microkernel ensures the set of real-time scheduling policies and system calls reducing the complexity of the kernel also keeping the most used fundamental resources. Shared memory is a technique that is used to transfer data efficiently. As a result, using these techniques will decrease the copy times during the message passing procedure. This communication achieves very low latency and nearly very minimal consumption performance.

**Keywords:** IPC, inter-process communication, shared memory, Address mapping, Micro-kernel

**1. INTRODUCTION**

The microkernel is an important key element in operating systems. It controls all operating system resources including clock, interrupt, storage, process, device driver, primitive, and others [1]. The primary function has an operating system run on microkernel. And the secondary function has operating systems run on user space [2]. A communication between process through shared memory and address-mapping method to achieve inter-process communication. IPC has a curial part of the microkernel because it impacts on how data efficiently be transferred and how efficiently microkernel operating system being used. The microkernel depends on communication medium to execute system functions. The message medium utilized in the inter-process communication has been an old method for microkernel, this method has satisfied the system's minimal requirements. but it has low performance which leads to microkernel operating system being ineffective and expensive also far away from meeting the requirements for real-time communication. That is why in this survey we are explore new IPC technique to used system more effectively microkernel operating system and it has fulfilled the meeting requirement of real-time scheduling [3].

**1.1 INITIAL GENERATION MICRO-KERNEL OPERATING SYSTEM**

Due to concerns for safety. Initial microkernel OS are divided into two address spaces .one is user address and other is kernel address. Only the kernel can read kernel addresses. It is impossible for user processes to see another process's user address. When using this technique, the data must go through the kernel address through IPC. Figure 1

If process A to transfer data to process B, process A should copy the data into the kernel address, Data is copied by the kernel from kernel address to the process B user address, the successful completion of the message transfer process between processes A and B

This inter process communication (IPC) technique requires two copies to finish the connection between processes. But the two copies decrease the IPC's effectiveness. If the cache hits will fail, then this situation will be worse, and IPC's speed will be strongly affected.[4] so if the copy method technique will improve then the efficiency and execution of inter-process communication will also improve

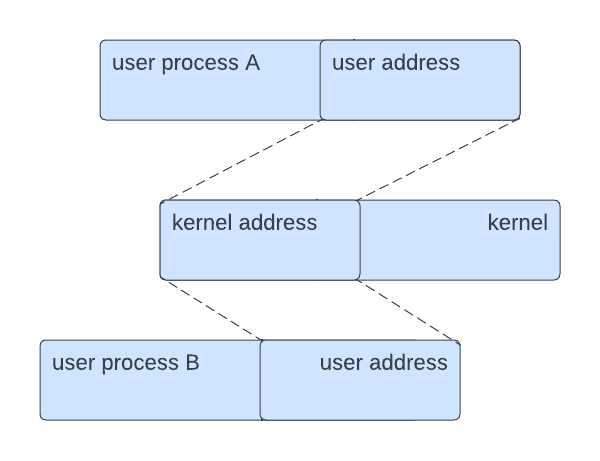


Figure 1. IPC of first-generation micro-kernel

**1.2 MICRO-KERNEL INTER-PROCESS COMMUNICATION MECHANISM**

The microkernel is differed from conventional macro-kernel operating system by using client-server network [5]. Because of security concerns, Microkernel must require IPC techniques to perform system functions, but macro-kernel could finish the communication between processes with a system call [5]. That is why this option will increase to message passing and frequent context switches which will make the system less effective. Because IPC is inefficient, early micro-kernel performs substantially worse than macro-kernel. Every micro-kernel employs a unique approach to enhance the IPC's performance. These processes do improve the IPC's effectiveness to some extent, but there are certain drawbacks as well.

**A. L4 mapping**

The inefficient use of first-generation microkernels has negatively impacted on performance. As a result of the IPC mechanism, the system cost overhead or increased and then researchers design a L4 microkernel. This is a second-generation micro-kernel. In order to speed up IPC and cut down on message copy time [6]. L4 uses the window mechanism method and its a plan to ensure direct communication between processes. Communication window is a area of address that is Placed on memory address of micro-kernel. The data that would be sent would also be copied into the communication window by the message-sending procedure. communication window show at. Figure 2

The L4 communication method only requires a single copy for communication between processes, significantly reducing the IPC system's energy requirements and enhancing system performance. This approach makes the kernel to access the user address of any process, which has an impact on system security.

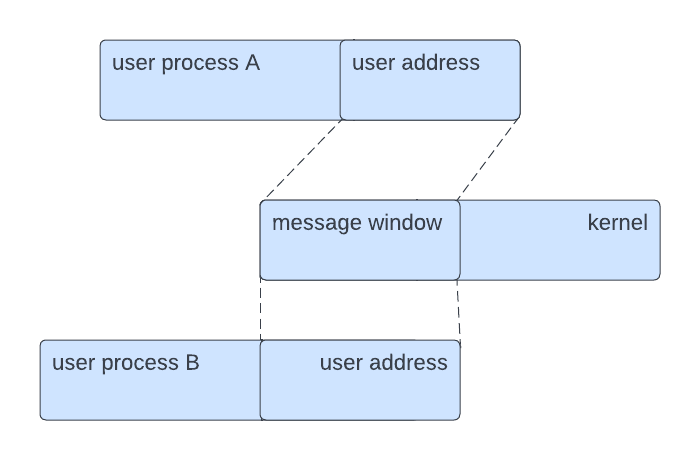


Figure 2. IPC of L4

**B. L4FIFO Server**

According to the L4 micro-kernel design, L4FIFO is a type of architecture in which a FIFO server that is in user mode is positioned on top of the micro-kernel. L4 micro-kernel is the base on L4FIFO architecture. Communication among processes is handled by the FIFO server, which uses its shared memory feature to finish messages between numerous processes [7]. Figure 3

Diagram

Description automatically generated

Figure 3. L4FIFO

If process A and process B need to interact with each other, process A, which is the process of sending a message, should send a request to the FIFO server. If the request is successful, process A will copy the message to the shared memory that is contained in the FIFO server module. When its request is approved, process B will then make a read request to the FIFO server and, once the message-transmission process is complete, then process can read the data in shared memory directly.

Inter-process interaction is accomplished by L4FIFO via the shared memory method in user mode, but this method requires the addition of a module to the L4 micro-kernel architecture. that will modify the microkernel module. Additionally, no communication passes through the kernel, that would compromise the safety of microkernel. Figure 4

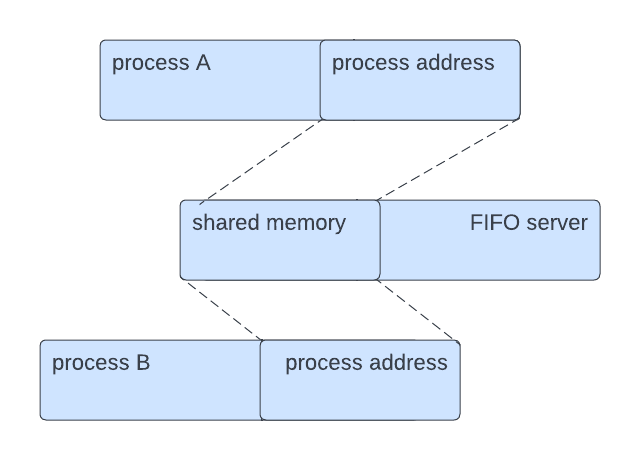
****

Figure 4. IPC of L4FIFO

**C. Minix3**

Another microkernel that is called minix3. Minix3 utilizing **Segmentation** technique. Every process in Minix3 is placed into the appropriate place in physical memory, and the kernel can access the system's physical memory. Since every process is installed in physical memory, the kernel can see every process's memory address [6]. All the processes are placed in the physical memory; therefore, you can say the kernel can see every address in the process’s memory.

Here kernel is involved in process communication. The kernel is coping the data of sender process address and give it to receiver process address to finish the whole procedure of communication. Figure 5, This system is built on the Minix3 **segmentation mechanism**, although it is not practical to use the **paging method** for inter-process communication.

Diagram

Description automatically generated

Figure 5. IPC Minix3

**D. Minix3 based on shared memory**

The stability and durability of the micro-kernel operating system were significantly improved by Minix3 [5]. Based on that, Shared memory can also accomplish the inter process communication. This approach utilizes a shared memory that is missing from the Minix3 micro-kernel address

whenever communication happen It Just take this chunk of shared memory that is allocated to the process address. The process sent a message and that message copy into shared memory after then receiver process should receive the message through the mapping mechanism. figure 6

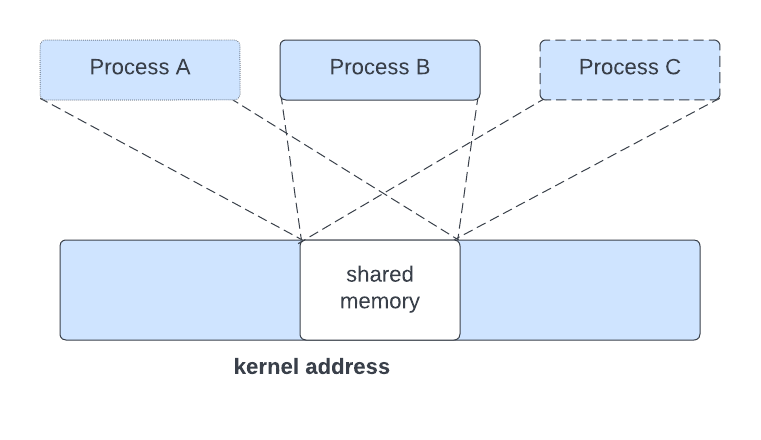
****

Figure 6. IPC of Minix3 based on shared memory

**2 MECHANISMS BASED ON SHARED MEMORY AND ADDRESS MAPPING**

**2.1.1 Theoretical basis**

We are aware that using shared memory the process effectiveness of inter process communication can be increased. Shared memory can enhance the effectiveness of inter process communication and improve the effectiveness and performance of the microkernel operating system [8]. To some level, shared memory in user mode can be used in efficient of inter process communication but in some shared memory mechanisms, processes have to finish the tough procedure of the operation during the using of shared memory, which greatly decrease the system's efficiency and performance and security on microkernel operating systems to apply shared memory on user mode shared Memory in the user mode may be the reason the process can easily obtain information from other processes that could compromise system security [8].

The initial phase of microkernel makes use of two copies to communicate between more than one processes. This is based on the page table data of every program is present in the address space of the kernel, hence the kernel gets the relevant program address by using examine the data [6]. Hence, to transfer All communication between addresses must go through the kernel address. This survey provides a shared memory and address-mapping-based technique to inter-process communication is achieve and usage of shared memory and an address mapping technique allow us to set a compact region of memory in the system kernel address as shared memory

**B. Fundamentals**

When a communication is taking place, the sending process simply transfers the data to the shared memory in the kernel. The data can be used by other processes through a Communications technique show in figure 7

When process A has interacted with process B, process C and some other processes, process A has to transfer the sending a message to the kernel's shared memory. Next, the process B's page table data is examined by kernel, Processes C, N, and others are used to obtain then transferring the matching process's address to these processes' shared memory address. Hence The processes can receive the message in the shared memory, completing the message transmission process

**C. Performance advantages**

Using address mapping and shared memory, the creation of a shared memory with a continuous kernel address. the two copies of inter-process communication decrease to single copy while converting the process's address to the shared memory address. The benefit of using this method is not clear given the brief messages' duration, but it will substantially increase the effectiveness of communication over a longer period of time a message. To build a shared memory in the kernel instead compared to the user's address, which effectively increases security, such a technique can restrict a process's interface to other Addressing processes using a mapping mechanism message transmission system, decreased the amount of copy and increased the effectiveness of microkernel architecture. Because the address of shared memory this method can be related to a variety of processes, achieved more instances of process communication, to enhance the functionality of the IPC mechanism, particularly in the several procedures that require communication, and performs substantially better. This system a very strong foundation for micro-kernel systems based on Inter-process communication in multi-processor architecture

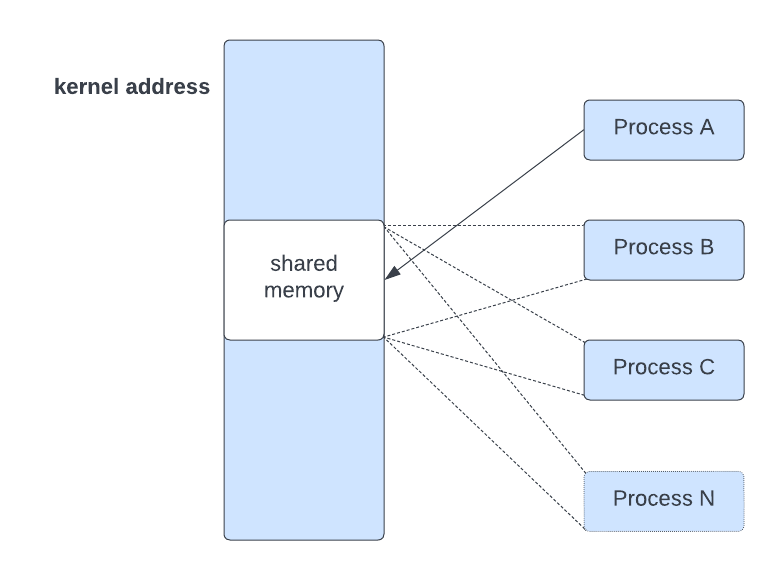
****

Figure 7. Mechanism based on shared memory and address-mapping

**CONCLUSION**

Executing a microkernel method for inter process communication inside the system based on these improved methodologies, a new upgraded Inter-process communication approach in microkernel systems shared memory and address mapping-based technique The effectiveness of these kinds of processes would increase. the effectiveness of inter process communication among several processes, which forms the foundation of the micro-kernel operating system inter-process a system for communication using several processors.

**REFERENCES**

[1] Tanenbaum A Sand Bos H 2015 Modern operating systems. Essex: Pearson Education Limited.

[2] Jorrit N.Herder. 2005 Towards a true microkernel operating system. Amsterdam: Vrije Universiteit.

[3] Xiao-hui Cheng and Liang Zhang, "A research of inter-process communication based on shared memory and address-mapping," Proceedings of 2011 International Conference on Computer Science and Network Technology, 2011

[4] Fu Yao. Huazhong University of Science, 2008. Embedded real-time operating system inter-process communication [D]. Wuhan

[5]Xiaohui Cheng, Zeqiu Li,Fuyuan Xiao. An advanced inter-process communication model of micro-kemel[1]. In ICCDA'II: Proceedings of IEEE 3rd International Conference on Computer Design and Applications. 2011.

[6] Tang Weijie. Hangzhou: Zhejiang University, 2008.Microkemel interprocess communication optimization and implementation [D]..

[7] Fang Zhengshu. The study and implementation of FIFO communication mechanism based on L4 [D]. Lanzhou: Lanzhou University, 2010

[8] Wu Wenjiang, Zhang Jie, Peng Zhengsen Based on RT-Linux Computer Systems, 2007 interprocess communication CNC systems research.