Collaborative Whiteboard with chat system

Ruchi Chaurasiya

Department of Computer Science & Engineering

Amrita School of Engineering, Bengaluru

Amrita Vishwa Vidyapeetham, India

bl.en.u4cse22278@bl.students.amrita.edu

Swagata Malik

*Department of Computer Science & Engineering*

*Amrita School of Engineering, Bengaluru*

Amrita Vishwa Vidyapeetham, India

bl.en.u4cse22252@bl.students.amrita.edu

Shakshi Yadav

*Department of Computer Science & Engineering*

*Amrita School of Engineering, Bengaluru*

Amrita Vishwa Vidyapeetham, India

bl.en.u4cse22286@bl.students.amrita.edu

Anaswara Reji

Department of Computer Science & Engineering

Amrita School of Engineering, Bengaluru

Amrita Vishwa Vidyapeetham, India

bl.en.u4cse22273@bl.students.amrita.edu

*Abstract*— Inter Process Communication through shared memory is a concept where two or more process can access the common memory and communication is done via this shared memory where changes made by one process can be viewed by another process. Each process has its own address space; if any process wants to communicate with some information from its own address space to other processes, then it is only possible with inter-process communication techniques. Whereas ,shared memory is the fastest inter-process communication mechanism. The operating system maps a memory segment in the address space of several processes to read and write in that memory segment without calling operating system functions.  shared memory segment is a portion of physical memory that is shared by multiple processes. In this region, processes can set up structures, and others may read/write on them. When a shared memory region is established in two or more processes, there is no guarantee that the regions will be placed at the same base address. The goalof this collaborative system is to provide a convenient environment for participants to interact with each other and support collaborative applications such as instant messaging, distance learning and conferencing.

**Index Terms**—Inter Process Communication, Shared memory

# Introduction

The use of whiteboards has been found as a beneficial medium to communicate spontaneous handwritten ideas andconcepts. Several workplace field studies have shown the importance of physical whiteboards that are a locale for discussion and collaboration. However, physical whiteboards are only visible locally in one place. Their information not easily available for remote sites. Especially, strokes written on the whiteboard with ink cannot be erased or edited by remote users. Hence, real-time multi-user physical whiteboard collaboration is limited locally to people in the same room. When a group of people gets together in one room to work on an agenda, we refer to this scenario as an ideal collaboration. A collaborative system can thus be viewed as a collection of technologies enabling users at multiple locations to seamlessly communicate with each other and to work jointly on the object of interest.

In this project, we design and implement a shared whiteboard system that can be edited simultaneously, using a P2P structure with a centralized index server and shared memory concept. The system supports a range of features such as freehand drawing, drawing multiple shapes such as lines, circles, rectangles, and oval with specific colours. We also implement a function to save a file in png format. Besides, the system also provides a chat button upon clicking on which clients can start chat window to chat among clients or can send message privately. so that all the users in the same shared whiteboard can send messages to each other. Chat contents are encrypted when transmitted, which ensures users' privacy. Moreover, we provide a lobby system for our users to create and join for specific whiteboard rooms, which we think is a very user-friendly improvement.

# Related Works

In this paper,[1] the design issues in an ordinary thread system, such as thread management, load balancing, and synchronization, have been reconsidered with the memory manage ment provided by the DSM system. Several real applications have been used to evaluate the performance of the system.

[2] Collaborative System offers a convenient environment for participants to interact with each other and supports collaborative applications such as instant messaging, distance learning, and conferencing. The[3] research contributes to the understanding of remote collaborative design and the effectiveness of shared whiteboard tools, providing valuable insights into the dynamics of interactions, task completion, and the impact of group structure and work mode on collaborative design tasks

The paper[4] explores multithreading's evolution, addressing drawbacks of the traditional process model and advocating threads for efficient concurrent execution. It categorizes threads by implementation, programming interface, and discussing implementations like user-level, kernel-level, and hybrid models such as POSIX, Win32, and Java threads.

The paper[5] introduces User-Level Remote Procedure Call (URPC) as a solution, which eliminates kernel mediation for IPC on shared memory multiprocessors, improving performance by directly passing messages between address spaces and minimizing processor reallocation overhead. URPC achieves this by integrating thread management and message passing at the user level, leading to enhanced performance and flexibility, especially on shared memory multiprocessors.

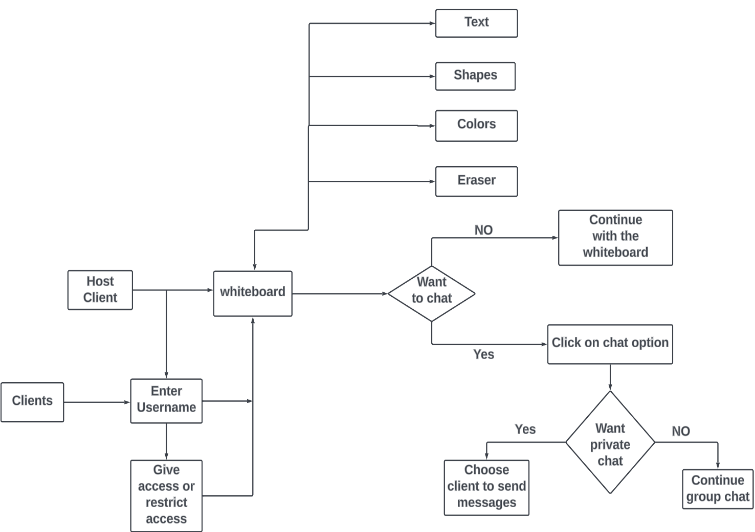
The efficiency of inter-process communication (IPC) mechanisms in micro-kernel operating systems and proposes a new approach based on shared memory and address mapping to reduce message passing overhead. The proposed[6] mechanism aims to optimize IPC by establishing shared memory in the kernel and mapping it to processes, reducing copying to enhance system efficiency and security, particularly in multi-processor architectures.

The author’s[7] approach divides the lecture hours into a theory track and a programming track. The former covers most materials that can be found in a typical operating systems textbook. The latter, with the help of seven programming assignments and one mini-project, covers the concept, merit, and skills of multithreaded programming. In this way, students will not only learn multithreaded programming, but also be familiarized in many topics that are not available in a typical operating systems course.

This paper[8] shows how at present many chat applications exist and are being used in many areas, but they are not education oriented. Besides, many e-learning systems do use their own applications for synchronous communication, although they do not always meet all the desirable requirements for their use in teaching

# Design

In this proposed model, concept of interprocess communication has been used to implement shared memory concept to make collaborative whiteboard system for easy accessible of all users connected via Local Area Network(LAN).



The model involves creating a new independent whiteboard and can give access to other users to access same whiteboard. Here, other users can join the exisitng whiteboard for the discussion or can create its own independent whiteboard.

# Implementation

This section describes the practical realization of the proposed model. To develop this model, study of java, it’s libraries, and concept of Operating System(OS) has been used. To realize the practical implementation, we have usedtwo major concepts of OS. They are

* Shared Memory
* Message Passing System

Firstly, all the essential java libraries were imported that is required for creating and terminating a process and implementing concept of shared memory and message passing system. In addition, java servlet has been used to create GUI based interface to make it more efficient for users. To make our model more relevant to the problem statement we had implemented multi-chat system along with collaborative whiteboard.

Secondly, user were given privilage of creating an independent whiteboard and can give access to other users or can restrict other users access. In addition, Multi-chat system were implemented inside whiteboard which can further enhance the application as it allows user to send message in a group i.e. all the clients who had joined the whiteboard or can send private messages to the particular client .

## Shared Memory

In this concept, memory may be simultaneously accessed by multiple programs with an intent to provide communication among them or avoid redundant copies. Shared memory is an efficient means of passing data between programs. The implemetation part had been divided into two parts. The first one is producer process and second one is consumer process.

Below given is the Pseudo code to design collaborative whiteboard using shared memory concept of Operating System

Start

* shared\_memory = create\_shared\_memory(size)
* process producer:
* while true:
* item = produce\_item()
* write\_item\_to\_shared\_memory(shared\_memory, item)
* process consumer:
* while true:
* item= read\_item\_from\_shared\_memory(shared\_memory)
* consume\_item(item) // Consume the item
* initialize\_shared\_memory(shared\_memory)
* destroy\_shared\_memory(shared\_memory)
* End

## Message Passing System

It is a method of Inter Process Communication in OS. It involves the exchange of messages between processes, where each process sends and receives messages to coordinate its activities and exchange data with other processes. The implementation part had been divided into sender and receiver’s mechanism where sender process sends the messages and receiver process receives the messages.

Below given is the Pseudo code to design myocarditis prediction model using logistic regression

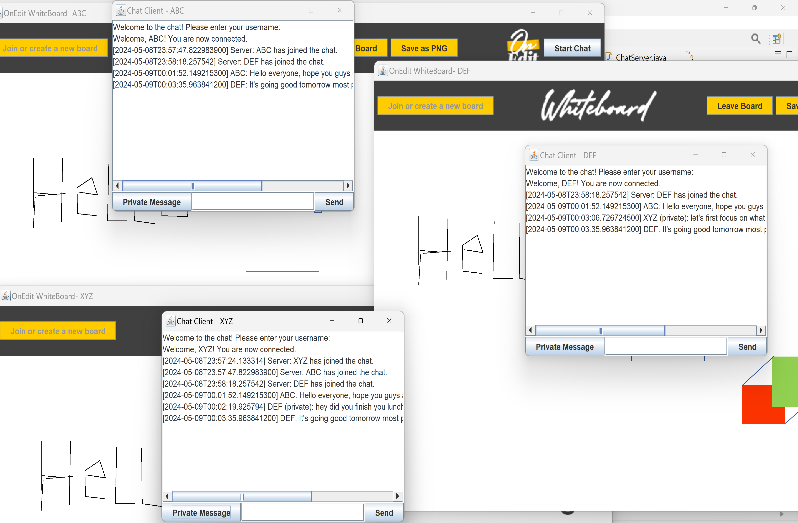
Start

* process sender:
* while true:
* message = produce\_message()
* send\_message(receiver\_process\_id, message)
* process receiver:
* while true:
* message = receive\_message()
* process\_message(message)

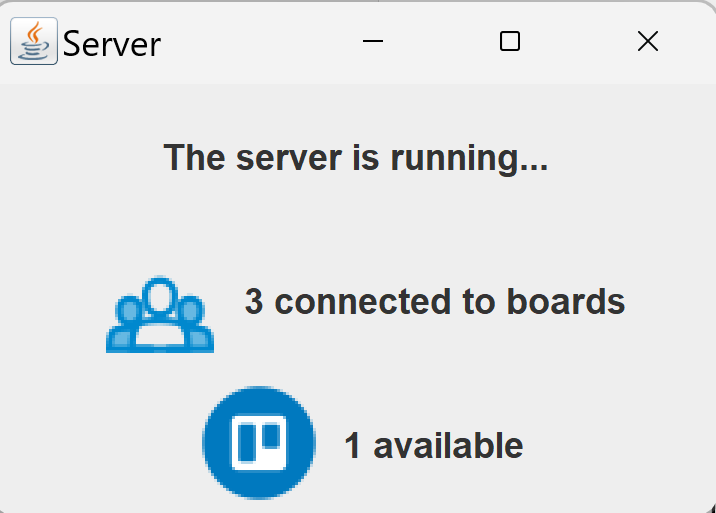
End

# Results

To analyze our collaborative whiteboard with chat system model, we have practically realized its implementation taking 3 sample clients with following results.



The above implementation shows how the same whiteboard has been accessed among three clients and also realizes the group as well as private chat mechanism that has been implemented.



This image depicts that server is running in the background tracks the count of how many independent whiteboard has been created and how many have joined the existing system.

# Conclusion

By using shared memory and message passing system concept of Inter Process communication of Operating System , the predicted model has been designed. Also, using the concept of data structures overall time complexity has been reduced.

# References

1. Jyh-Chang UENG, Ce-Kuen SHIEH, Su-Cheong MAC, An-Chow LAI, and Tyng-Yeu LIANG, “multi-threaded design for a software distributed shared memory system”, IEICE trans. inf. & syst., vol.e82–d, no.12 december 1999
2. Zon-Yin Shae, Belle Tseng, and Wing Ho Leung, “Immersive Whiteboard Collaborative System”, Annals of Software Engineering 12, 193–212, 2001 2002 Kluwer Academic Publishers. Manufactured in The Netherlands.
3. Stephanie Mailles Viard Metz, Philippe Marin, and Emilie Vayre, “The shared online whiteboard: An assistance tool to synchronous collaborative design”, Revue européenne de psychologie appliquée 65 (2015) 253–265
4. Krishna Kavi, “Multithreading Implementation”, The University of Texas at Arlington NSF, MIP 9622593
5. Brian N. Bershad, Thomas E. Anderson, Edward D. Lazowska,“User-Level Interprocess Communication for shared memory multiprocessors ”, ACM Transactions on Computer Systems, Vol 9, No. 2, May 1991, Pages 175-198
6. Xiao-hui Cheng, and Liang Zhang “A Research of Inter-process Communication Based on Shared Memory and Address-mapping”, 0 II International Conference on Computer Science and Network Technology 978-1-4577-1587-7/111$26.00 ©201I IEEE
7. Ching-Kuang Shene, “ Multithreaded Programming Can Strengthen an Operating Systems Course”, Michigan Technological University Houghton, MI 49931–1295, USA
8. Ana Belén Cara Carmona, Carlos Moreno Muñoz, Antonio Cañas Vargas, “virtual whiteboard and chat for a learning management system”, Dpt. Computer Architecture and Technology, University of Granada C/ Periodista Daniel Saucedo Aranda, s/n. E-18071 Granada (Spain)