



Chat Application using Interprocess Communication

OPERATING SYSTEMS LAB PROJECT

Assigned TA: Shipra Shukla



Group Members

1. Rachit Rahul Mishra (17JE003017)
2. Samyak Singh (17JE003024)
3. Vipul Bandi (17JE003026)
4. Sandesh Sinha (17JE003038)
5. Sandeep Sheela (17JE003048)
6. Avi Sahney (17JE003050)
7. Navya Srivastava (17JE003052)
8. Vipin Prakash (17JE003061)
9. Thakur Ashutosh Suman (17JE003067)

PROBLEM STATEMENT

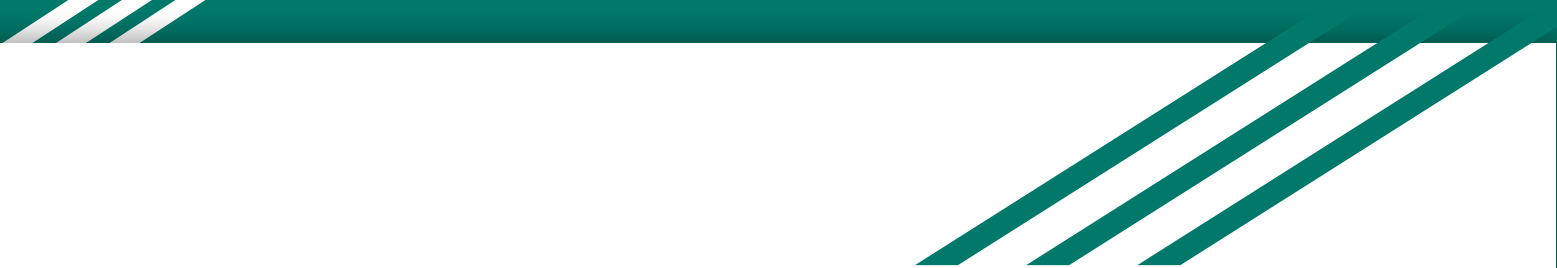
Title: Develop a chat application using Interprocess Communication.

Use any one of the methods to implement IPC and develop a chat application which can handle multiple clients.

Multiple clients should also simultaneously be able to chat with the server.

The client/user also must be able to see his/her message history.


Inter-Process Communication




A process can be of two types namely Independent processes and Cooperating processes. An independent process is not affected by the execution of other processes while a co-operating process can be affected by other executing processes.

Inter-process communication (IPC) is a mechanism which allows processes to communicate with each other and synchronize their actions. The communication between these processes can be seen as a method of co-operation between them. Processes can communicate with each other using these two ways:

1. Shared Memory
 2. Message Passing
- 

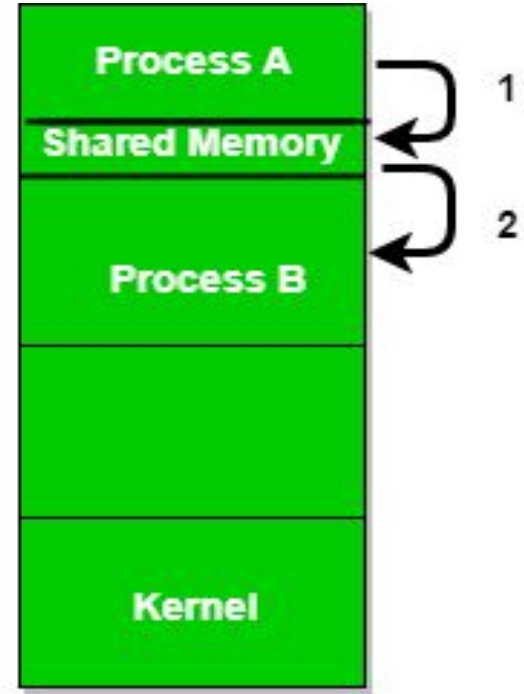


**We have used the concept of shared
memory for Inter Process
Communication in our project.**



Shared Memory Concept

Shared Memory is an efficient means of passing data between programs. One program will create a **memory** portion which other processes (if permitted) can access. Once created, a **shared** segment can be attached to a process address space using `shmat()`. It can be detached using `shmdt()`.

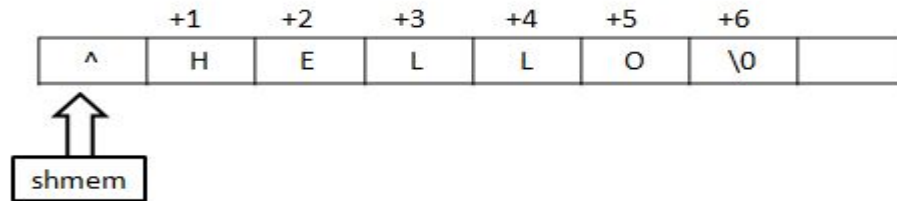




OUR APPROACH

Sending and receiving messages:

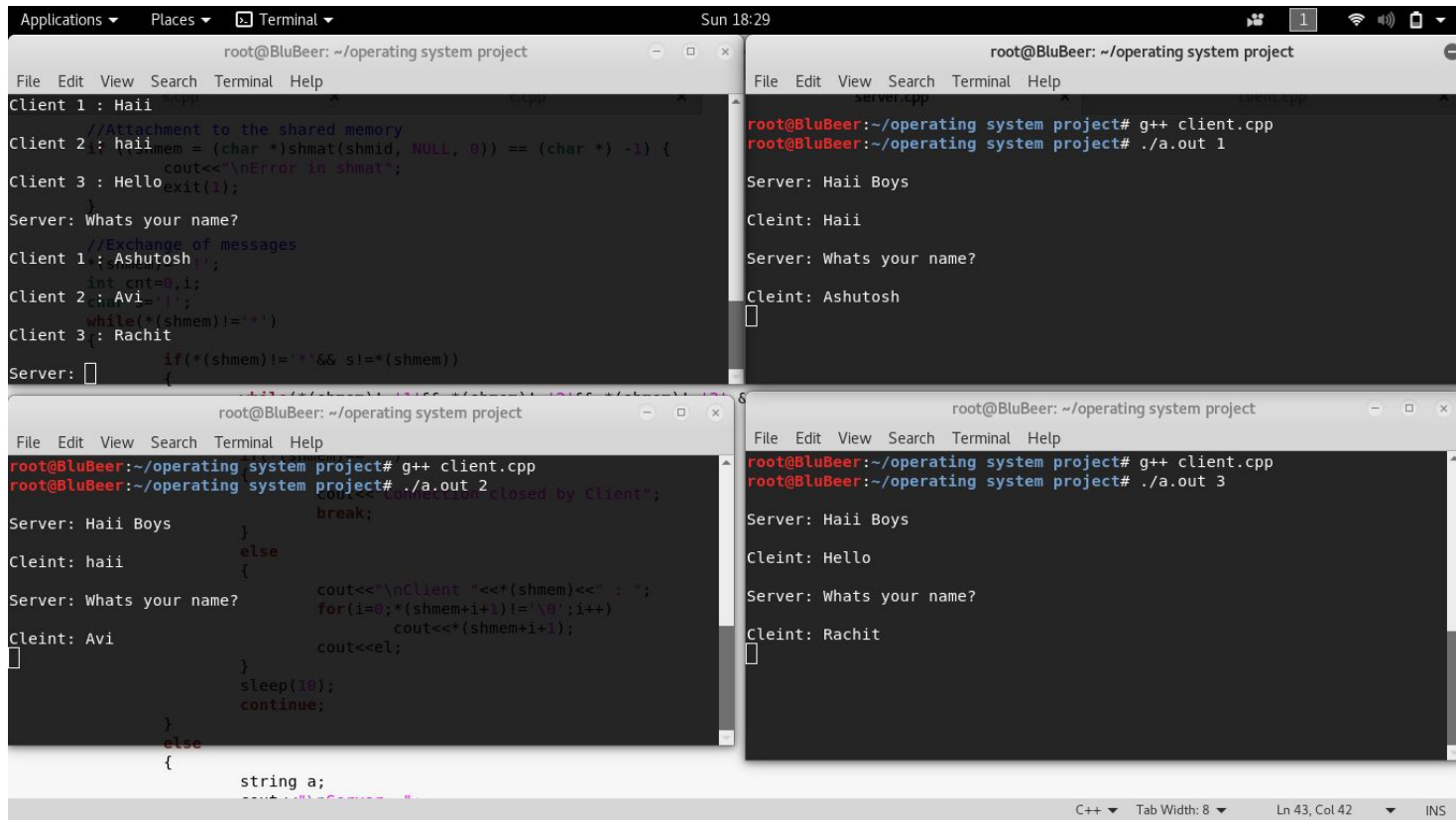
The problem statement requires the server to handle multiple clients messages and communicate with them. We have designed a system where messages of multiple clients are displayed at the server's end and the server can respond to them. In this system, the server reads the message once a client writes and then waits for another client to write. If no client writes to the shared memory location in a given amount of time the server again takes the control. The message written by the server is displayed to all the clients. The way message is written is given below.



Reading the message history:

When the server reads the messages of different clients, it also stores these messages to a text file which clients can access to see their message history. Each client has its own message history text file and can access only that file. The message history can only be displayed after the chatting connection has been closed.

OUTPUT



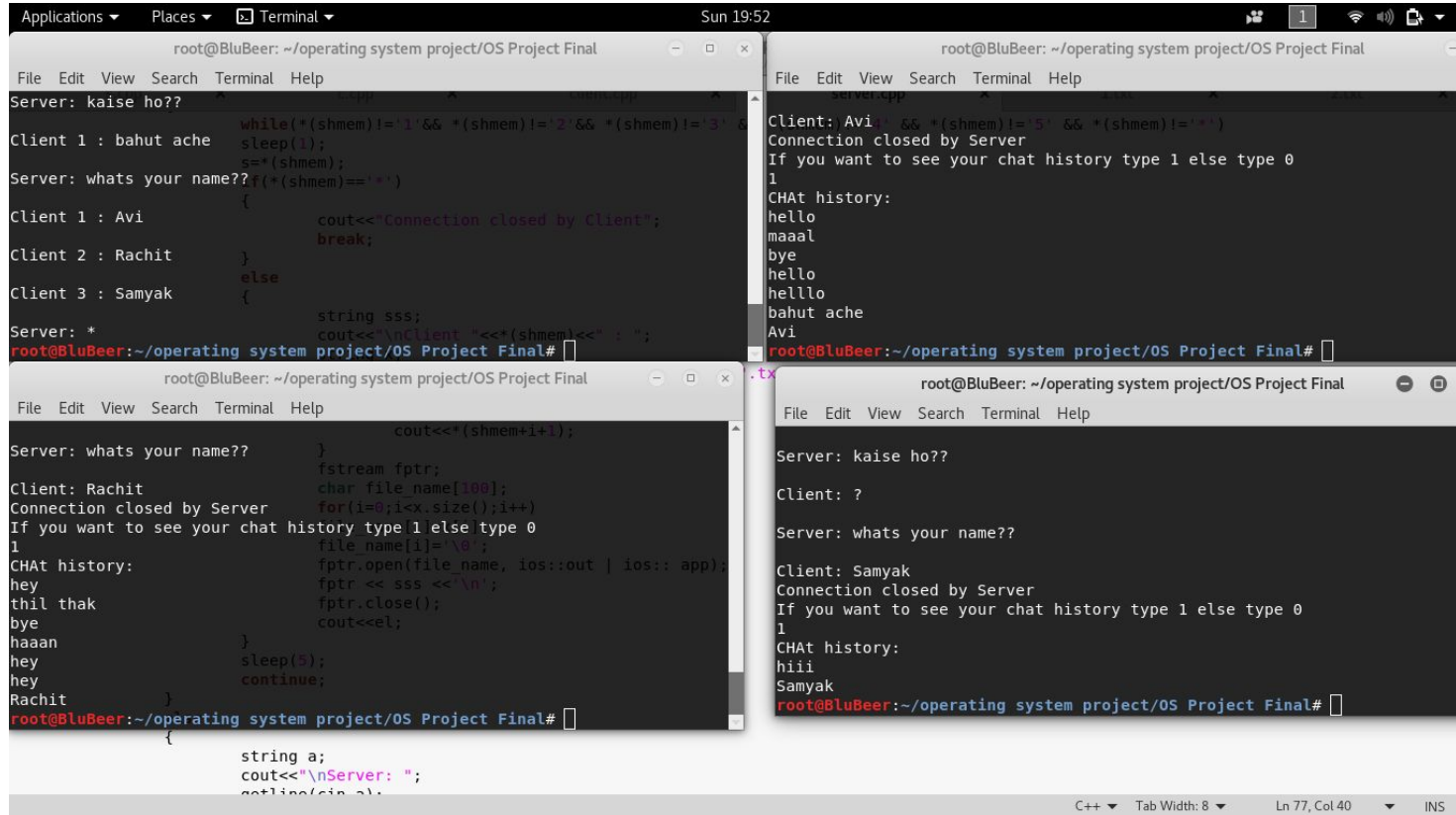
```
root@BluBeer: ~/operating system project
File Edit View Search Terminal Help
Client 1 : Haii
Client 2 : haii
Client 3 : Hello
Server: Whats your name?
Client 1 : Ashutosh
Client 2 : Avi
Client 3 : Rachit
Server:

root@BluBeer: ~/operating system project
File Edit View Search Terminal Help
root@BluBeer:~/operating system project# g++ client.cpp
root@BluBeer:~/operating system project# ./a.out 1
Server: Haii Boys
Cleint: Haii
Server: Whats your name?
Cleint: Ashutosh

root@BluBeer: ~/operating system project
File Edit View Search Terminal Help
root@BluBeer:~/operating system project# g++ client.cpp
root@BluBeer:~/operating system project# ./a.out 2
Server: Haii Boys
Cleint: haii
Server: Whats your name?
Cleint: Avi
Server:

root@BluBeer: ~/operating system project
File Edit View Search Terminal Help
root@BluBeer:~/operating system project# g++ client.cpp
root@BluBeer:~/operating system project# ./a.out 3
Server: Haii Boys
Cleint: Hello
Server: Whats your name?
Cleint: Rachit
```

OUTPUT



```
root@BluBeer: ~/operating system project/OS Project Final
File Edit View Search Terminal Help
Server: kaise ho??

Client 1 : bahut ache
Server: whats your name??
Client 1 : Avi
Client 2 : Rachit
Client 3 : Samyak
Server: *
root@BluBeer:~/operating system project/OS Project Final#

root@BluBeer:~/operating system project/OS Project Final
File Edit View Search Terminal Help
Client: Avi
Connection closed by Server
If you want to see your chat history type 1 else type 0
1
CHAT history:
hello
maaal
bye
hello
helllo
bahut ache
Avi
root@BluBeer:~/operating system project/OS Project Final#

root@BluBeer:~/operating system project/OS Project Final
File Edit View Search Terminal Help
Server: whats your name??
Client: Rachit
Connection closed by Server
If you want to see your chat history type 1 else type 0
1
CHAT history:
hey
thil thak
bye
haaan
hey
hey
Rachit
root@BluBeer:~/operating system project/OS Project Final#

root@BluBeer:~/operating system project/OS Project Final
File Edit View Search Terminal Help
Server: kaise ho??
Client: ?
Server: whats your name??
Client: Samyak
Connection closed by Server
If you want to see your chat history type 1 else type 0
1
CHAT history:
hihi
Samyak
root@BluBeer:~/operating system project/OS Project Final#
```

C++ Tab Width: 8 Ln 77, Col 40 INS



Thank You