**Notes to look into, for the Sending and Receiving Messages application. (by Anind Duttaroy)**

**(A simple example of IPC (Inter-Process Communication)).**

**1.** There are two players, who can send and receive messages between them.

**2.** One of the players (The “Initiator”), is in one process, and sends a message to the second player (The “Receiver”), who is in another process.

**3.** Let’s say the Initiator creates a message (“Ping1”), containing the actual message (“Ping”), and the message number (“1”), and sends it to the Receiver. The Receiver receives the message, creates a new message (“Pong1”), containing the actual message (“Pong”), and the message number (“1”). The Receiver should send back the received message concatenated with the message it has created. So the Receiver should send back (“Ping1Pong1”).

**4.**  The Initiator receives “Ping1Pong1”, creates its own message (“Ping2”), concatenates with the received message, and sends back “Ping1Pong1Ping2”. The Receiver receives this message, creates its own message, concatenates with the received message and sends back to the Initiator, and so it goes on.

**5.** The program ends gracefully after the Initiator sent 10 messages and received back 10 messages.

**6.**  The program is in C++, with C system calls, and compiles and runs on Linux. It has been compiled with g++.

**To build and execute the program, on Linux:-**

1. Copy the files “Message\_Sender\_Receiver\_Player.h”, “Message\_Sender\_Receiver\_Player.cpp”,

“Initiator\_main.cpp”, and “Receiver\_main.cpp” to a directory in Linux.

1. Compile with “g++ -Wall -std=c++17 Initiator\_main.cpp Message\_Sender\_Receiver\_Player.cpp -o Message\_Initiator”
2. Compile with “g++ -Wall -std=c++17 Receiver\_main.cpp Message\_Sender\_Receiver\_Player.cpp -o Message\_Receiver”
3. First, run "Message\_Initiator" in one terminal.
4. Second, run "Message\_Receiver" in the other terminal.
5. The outputs will be printed in both terminals.