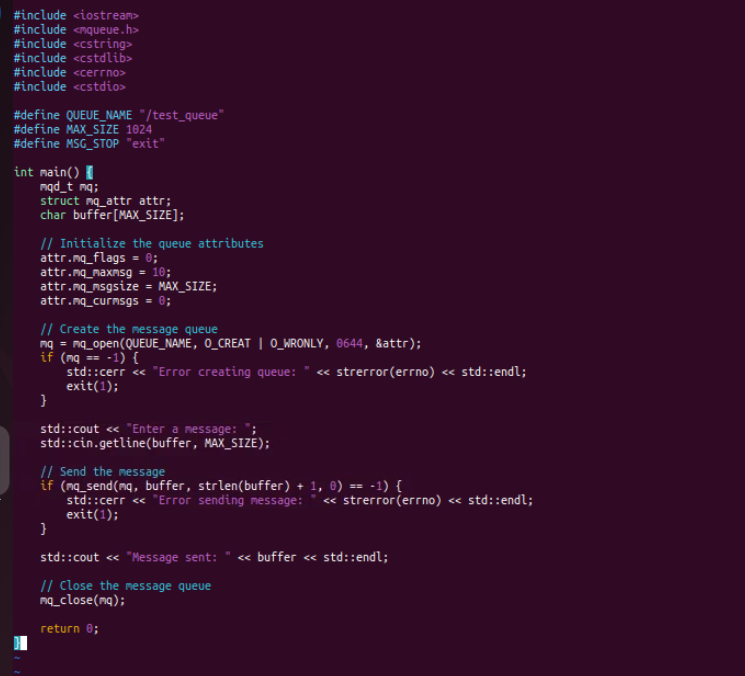
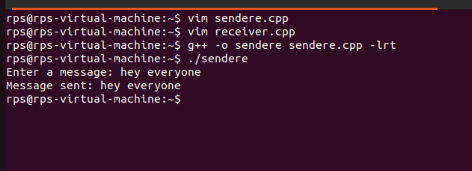
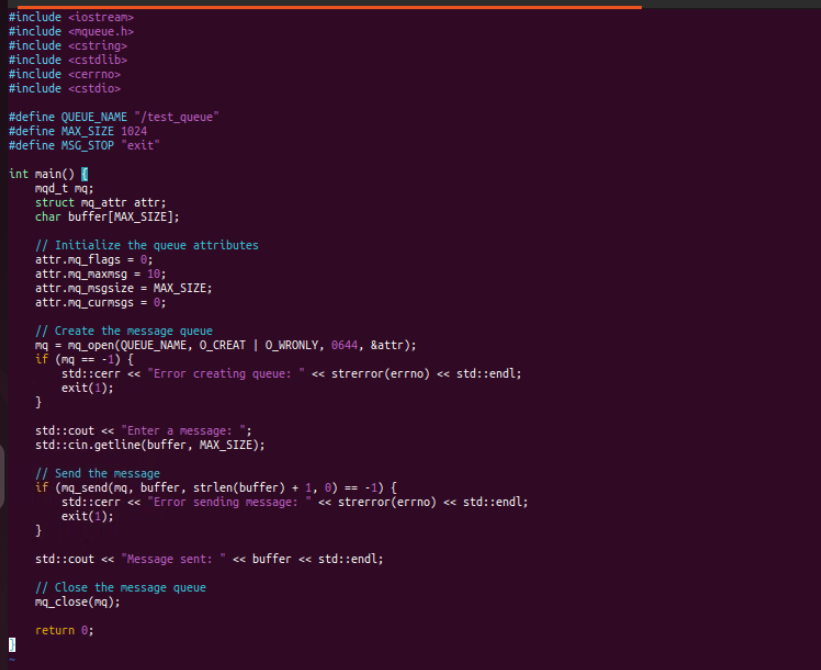
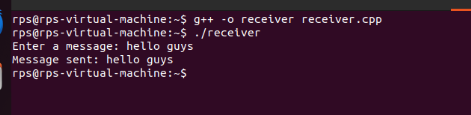
Sender.cpp





Receiver.cpp





Problem Statement: Socket Programming in C

Design and implement a reliable and efficient network communication system using socket programming in C to enable data exchange between two or more processes running on different machines over a network.

Specific Requirements:

Socket creation: Create appropriate socket descriptors for the desired communication protocol (TCP, UDP, etc.).

Address binding: Bind the created socket to a specific network address and port number for both client and server applications.

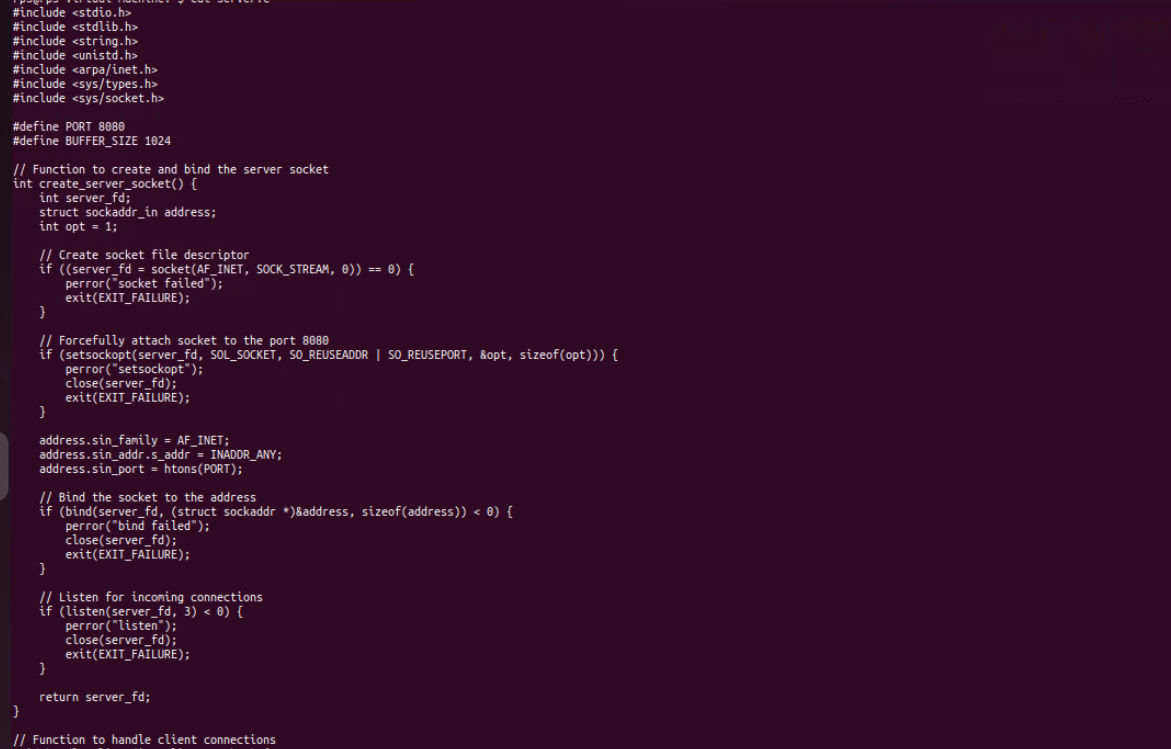
Connection establishment: Implement connection setup mechanisms (connect, accept) for TCP-based communication.

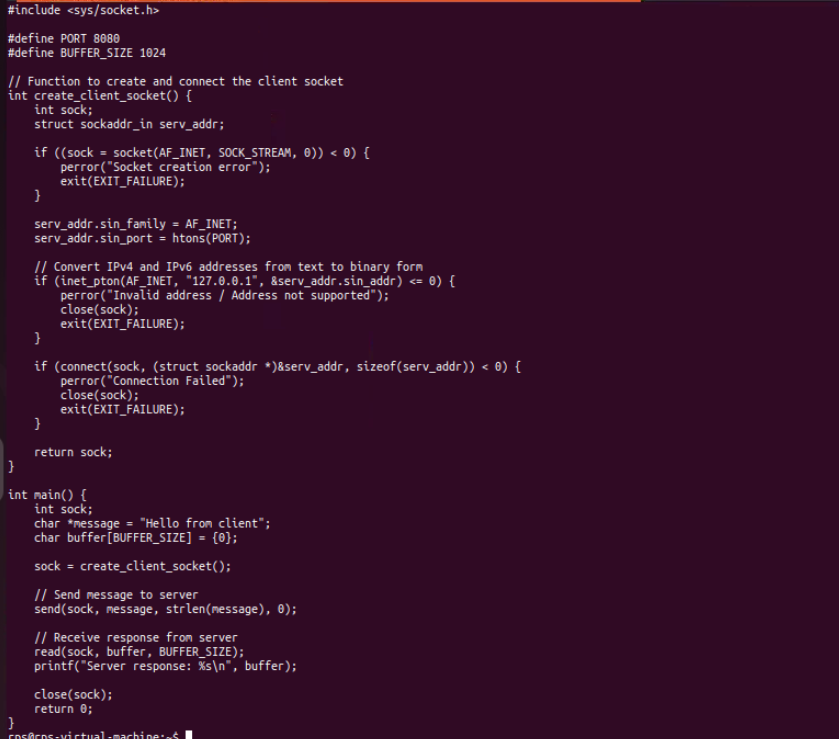
Data transfer: Develop functions for sending and receiving data over the established socket connection.

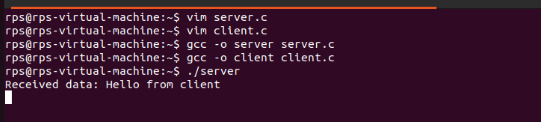
Error handling: Incorporate robust error handling mechanisms to address potential network issues and unexpected exceptions.

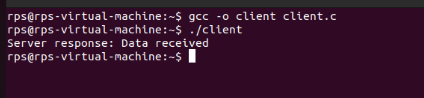
Concurrency: For server-side applications, consider handling multiple client connections concurrently using appropriate techniques (e.g., threading, forking).

Security: Implement appropriate security measures to protect data integrity and confidentiality (e.g., encryption, authentication).



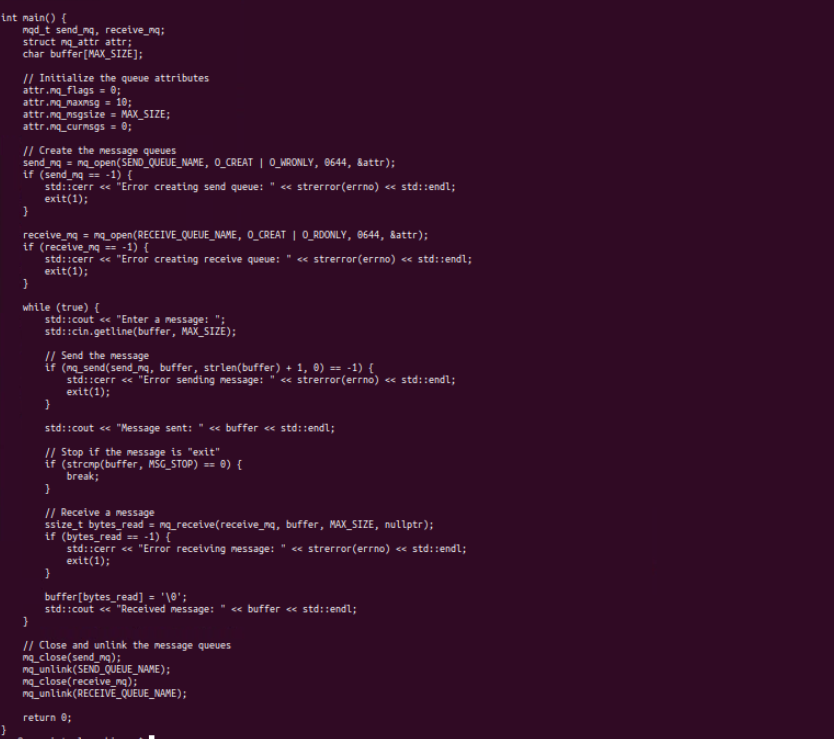




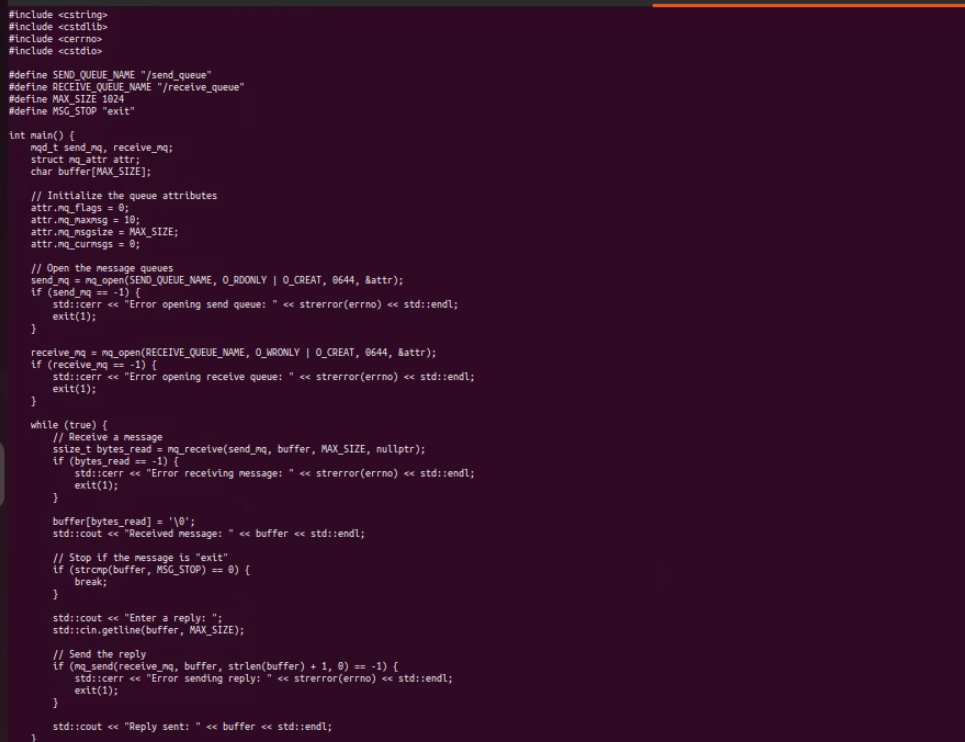


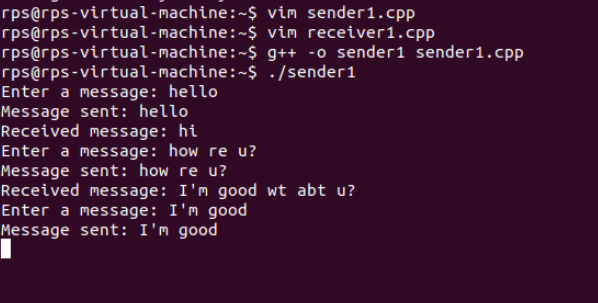
Chat with both side sender and receiver

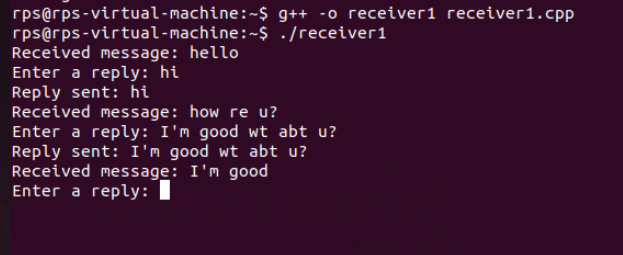
Sender1



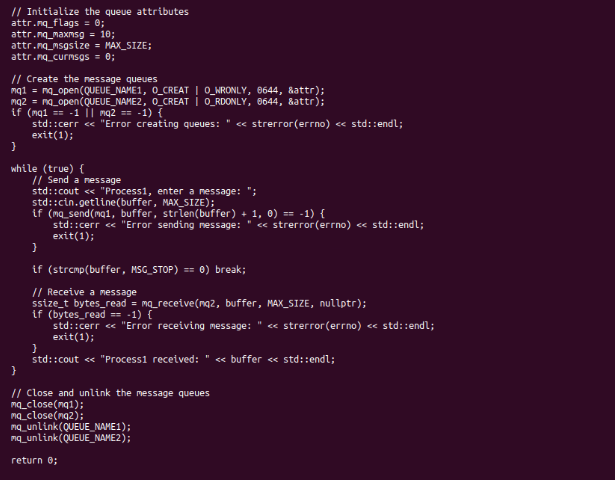
receiver



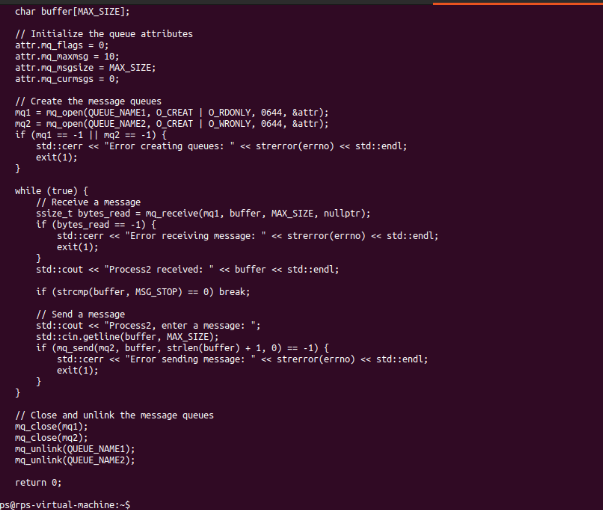


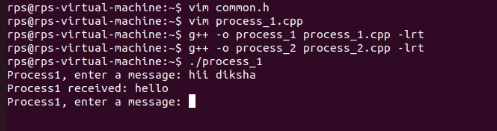


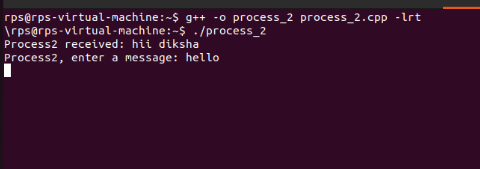
Process 1



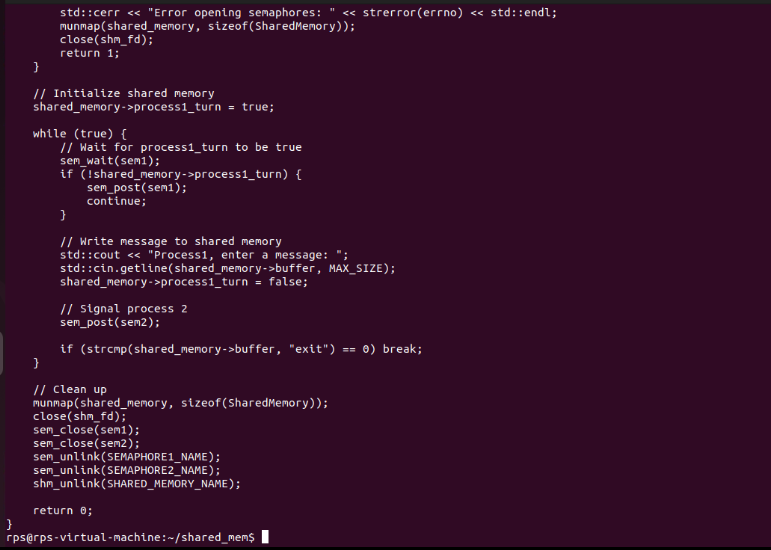
Process2



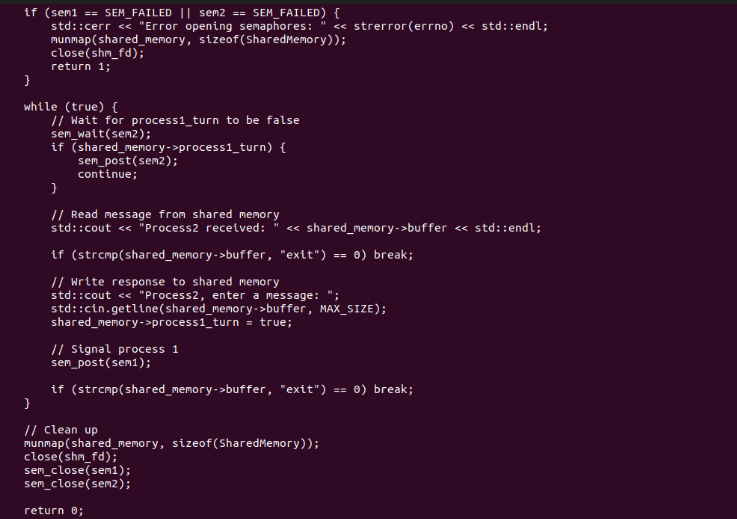


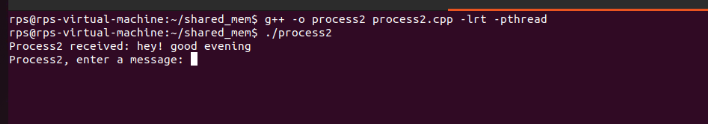


Thread process1



Thread process 2





Child process:

