1. Implement two-process communication to generate Echo server application using pipes: An echo server is usually an application which is used to test if the connection between a client and a server is successful. The server sends back whatever text the client sends.

1. Create two pipes pipe1 and pipe2

2. Create a child process

3. Child closes read end of pipe1

4. Child closes write end of pipe2

5. Child writes to pipe1 and waits for response from parent

6. Parent closes write end of pipe1

7. Parent closes read end of pipe2

8. Parent reads data from pipe1 and writes to pipe2 for child to read

9. Child reads data written by parent to pipe2 through read end of pipe2 and displays

on.

1. Implement a program for producer consumer problem using message passing :The problem describes two processes, the producer and the consumer, who share a common, fixed-size buffer. The producer's job is to generate a piece of data (A to Z) put it into the buffer and start again. At the same time the consumer is consuming the data (i.e. removing it from the buffer) one piece at a time. A producer process produces information that is consumed by a consumer process.

1. Create a semaphore.

2. Create a shared buffer- Message queue.

3. Initialize value of the semaphore to 1.

4. Create a child process.

5. Child process (Producer) writes to the Message queue.

6. Producer performs an up operation on the semaphore for the consumer to consume.

7. Parent process (Consumer) performs a down operation on the semaphore and reads or

consumes the data from the Message queue.

8. Consumer then performs an up operation.