# Assignment no 5

**Problem (A):**

Write a C program called Program1 which is responsible for creating a named PIPE or FIFO with necessary permission for all users by using the mknod( ) system call. Now, you keep the facility to read the information from the FIFO if any other process write some information into the FIFO and as well as keep the facility to write some information into the FIFO, so that Program2 can read it.

Again write another C program called Program2 which is responsible for taking the user input and write it back to the FIFO so that Program1 can read it. Moreover, you have to provide the facility to read the FIFO in this program.

(N.B. - First run the Program1 and keep it in running state in the background. Then run the Program2 which will take user input string and write the input string into the FIFO. Then Program1 will read it from the FIFO and display it and after that Program1 will write something into the FIFO as ACKNOWLEDGEMENT and program2 will read the ACKNOWLEDGEMENT string from the FIFO. In Program2, you may take the input string as a command line argument)

Hints:

- Learn about mknod() system call, write () system call, read () system call.
- To keep synchronize the Program1 execution with Program2 execution and vice versa, you may need to use the sleep () system call.
- For the above mentioned system call, follow the man pages.

**Problem (B):**

Write a two C programs to be one client program and one server program so that you can demonstrate the IPC mechanism using Message Queue functionality.

* Print the message queue ID in the stderr file.
* Put attention in the command macros used in msgctl( ).

.

Hints:

* Try for System V Message Queues.
* Learn the system call msgget(), msgsnd(), msgrcv(), msgctl(), etc.
* Go through google search for better understanding and UNIX NETWORK PROGRAMMING by W.RICHARD STEVENS.