EXPERIMENT 10
Named Pipes

National University of Computer and Emerging Sciences

Laboratory Manual

for

Operating Systems Lab

Course Instructor	Mr. Ibrahim Naddir
Lab Instructor(s)	Mohammad Amir Waseem Zumirrah Khalid
Section	CS-4E
Semester	Spring 2021

Department of Computer Science

FAST-NU, Lahore, Pakistan

OBJECTIVE:

· Learn and execute InterProcess Communication using implementation of named

pipes

Named Pipes:

- It is an extension to the traditional pipe concept on Unix. A traditional pipe is "unnamed" and lasts only as long as the process.
- A named pipe, however, can last as long as the system is up, beyond the life of the process. It can be deleted if no longer used.
- Usually a named pipe appears as a file, and generally processes attach to it for inter-process communication. A FIFO file is a special kind of file on the local storage which allows two or more processes to communicate with each other by reading/writing to/from this file.
- A FIFO special file is entered into the filesystem by calling mkfifo() in C. Once we have created a FIFO special file in this way, any process can open it for reading or writing, in the same way as an ordinary file. However, it has to be open at both ends simultaneously before you can proceed to do any input or output operations on it.
- Reading from or writing to a named pipe occurs just like traditional file reading and writing; except that the data for named pipe is never written to or read from a file in hard disk but memory.

Question 1:

Create 2 independent programs that perform communication using named pipes. One program will be the server program that will wait for client to send some data via a named pipe. The data sent by the client is as follows:

Operator operand1 operand2

The operands can be +, -, *, /. The server will then apply the operator on the operands and return the result to client via named pipe. The client will then print the result on the screen

For example, if the client passed the following to the server: + 4 10, then the server will calculate 4+10 and return 14 to client via the pipe. The client will then print it.