National University of Computer & Emerging Sciences

Department of Computer Science

CL 205 Operating System Lab

Lab # 7

Inter-Process Communication: Pipes

Objective:

Now that we know how to create processes, let us turn our attention to make the processes communicate among themselves. There are many mechanisms through which the processes communicate and in this lab we will discuss such a mechanism: Pipes. A pipe *is used for one-way communication of a stream of bytes*. In this lab we will learn how to create pipes and how processes communicate by reading or writing to the pipe and also how to have a two-way communication between processes.

The system calls covered in this lab are:

- Pipe
- dup / dup2
- mkfifo

Exercise:

1. Write a program for inter-process communication among child and parent process with following needs;
    a. A parent and two childs
    b. Parent take data from the user
    c. Send to a child for addition and subtraction.
    d. Again take input from user and send to another child for multiplication and division.
    e. First child process creates another child **C3**.
    e. Both the childs send result to the child process **C3** to show output.
    f. Parent process after completion of each child process task kill the process.
2. Write a program that creates a child process using fork (). Parent sends a message using Pipe () to the Child. The Child receives and displays the message.
   Hint: To allow one way communication each process should close one end of the pipe. The file Descriptors associated with a pipe can be closed with the close (fd) system call
   Output should be of this form:
   Parent: I am Parent I have sent message to the Child
   Child: I have received this message from parent: .....
3. Write two different programs and make them communicate with each other using ***Named Pipes (FIFO)***. The program 1 writes first, then reads. The program 2 reads first, then writes. They both keep doing it until terminated.

Sample Output:

Program 1	Program 2
Write a message : Hello Program2 says: hi Write a message:	Program1 says: Hello Write a message: hi . . . . . .