Assignment 5

Q1. You will write two simple programs pipe_reader.c and pipe_writer.c that use a named pipe to communicate. The pipe reader program will set up a named pipe using mkfifo(), open it read only, and read strings from it until it recieves the string exit. The writer will open the named pipe file, read strings from the user and write them to the named pipe. When the user enters exit, the program will write the string to the pipe and then exit. Execution should look something like this (note that you must start the reader first):

reader:

```
$ ./pipe_reader
Creating named pipe: /tmp/mypipe
Waiting for input...Got it: ’Oh! God’
Waiting for input...Got it: ’OS lab trouble’
Waiting for input...Got it: ’exit’
Exiting
```

writer:

```
$ ./pipe_writer
Opening named pipe: /tmp/mypipe
Enter Input: Oh! God Writing buffer to pipe...done
Enter Input: OS lab trouble
Writing buffer to pipe...done
Enter Input: exit
Writing buffer to pipe...done
Exiting
```

Note : pipe_reader and pipe_writer need to be executed in separate shells at the same time. The reader stops at emphWaiting for input... until it recieves data from the pipe (the read completes).

Q2. Write two programs that use named pipe. The first program receives 10 numbers as input and write these numbers into the pipe. The second program reads these numbers and performs the sorting and write it into the pipe. The first program reads the output written by second program and displays it.

Q3. (i) Create a named pipe. Open second terminal. Attach a consumer process (for example less or cat) to that pipe. From the first terminal start piping some text here. What do you see on a less window?

(ii) do different sequence of read and write- i.e. read first or write first. What do you see?

(iii) Using single terminal message with the background job, send more than one message to the fifo. Run the jobs command. What do you see? After that, attach a read process on the same terminal. After you complete the read, do the jobs command again. What do you see?

(iv) Attach more than one consumer and start sending messages from the producer’s end. Which one of the consumer gets messages?

(v) Send some text to the fifo, then do cat “yourpipe” > some_file. Open your file, what do you see?