CSCI 360 - Lab 5 Report

Sami Al-Qusus

Oct 16, 2018

Assignment Goals:

- Producer-Consumer Problem with TCP/IP Message Passing

Assignment Instructions (Dr. LoPinto):

- Use TCP/IP networking for interprocess communication in the consumer-producer problem.
- Start from client.c, server.c, and makefile.
    - Improve the code by adding comments.
    - Handling return codes for all function calls and making other changes you want.
    - You should invent and implement a way to ensure that complete messages are received at both ends. In other words, make sure there is no possibility of reading a partial message. You should describe all functions used and the parameters used with them.

What I did:

1. Went through the example code, understood it and added comments.
2. Then I added more code to count the length of the message and send it to the server along with the message, so that the server loops until it receives the whole message.
3. Then I added code to handle return codes for all the function calls.
4. Then added more comments to describe all functions used and the parameters used with them.

Note: please refer to comments in client and server files for line by line details.

What I learnt:

- The most important thing I learnt is that with TCP we do not always read the complete message. Therefore we have to ensure that complete messages are received at both ends.

The socket() system call creates one end of the socket.
int socket(int <family>, int <type>, int <protocol>);

- The first parameter specifies the communication family, AF_UNIX or AF_INET.
- The second parameter specifies the socket type, SOCK_STREAM or SOCK_DGRAM.
- The third parameter is usually zero because communication families usually have only one protocol.

The socket() system call returns the socket descriptor

Set sock options are used for setting/getting various options for a socket.

- For example, if you are testing a server application that crashes, you don't wont to wait a certain number of minutes before the kernel let you reuse the port avoiding the "Address already in use" error messages. This can be avoided if you use the SO_REUSEADDR option, letting other sockets to bind to the same port unless there is an active listener bound already.
- You can also retrieve data about a socket, such as the number of lost packets/retransmissions etc by using the TCP_INFO on linux machines. Basically, you can configure all the fine settings.

- srv_addr.sin_family = AF_INET;
  - sets the address family
- srv_addr.sin_port = htons(PORT);
  - set destination port number
- srv_addr.sin_addr.s_addr = INADDR_ANY;
  - set destination IP number

- bind() system call associates an address with the socket descriptor.
- Function listen(), listens for incoming connections, allowing a queue of up to n pending connection.
- Function accept(),takes connections from the queue
  - If there is no connection waiting on the queue the program waits until a connection is received.

- Function read(), reads length bytes of input into the memory area indicated by given buffer.
  - Return length of bytes read.
- Function write(), Writes N bytes from buffer to the socket associated with accepted connection.
- Function close(),closes/frees file descriptor.

Advantages of TCP/IP: TCP/IP is non-proprietary and, as a result, is not controlled by any single company. Therefore, the Internet protocol suite can be modified easily. It is compatible with all operating systems, so it can communicate with any other system. The internet protocol suite is also compatible with all types of computer hardware and networks.

Note: please refer to comments in client and server files for more details.

Credits:

- Dr. LoPinto lab5 examples.
- https://searchnetworking.techtarget.com/definition/TCP-IP
- https://stackoverflow.com/questions/4233598/about-setsockopt-and-getsockopt-function
- https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.3.0/com.ibm.zos.v2r3.bpxbd00/rtrea.htm
- https://www.systutorials.com/docs/linux/man/2-setsockopt/
- https://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vmci.pg.doc%2FvsockAppendix.8.3.html
- https://en.wikibooks.org/wiki/C_Programming/Networking_in_UNIX