HW 12

Joshua Thompson

April 22, 2018

Questions

1. (20 points) For each of the following system calls, provide a breif, one sentence description:

   (a) `socket()`

   The first argument takes the domain of the socket, the second takes the type of socket which describes what transport protocol to use, the third argument describes further protocol information. The call returns a file descriptor if successful or -1 if failure.

   (b) `bind()`

   First argument takes an open socket from the socket() call, the second takes a reference to a socket address to be bound, and the third argument takes the size of the socket address. The call returns a 0 if successful or -1 if not.

   (c) `accept()`

   First argument takes server socket that has already been bound, second argument takes a reference to a socket address structure of the client, the third argument is a pointer to a size reference to the address. Returns an integer 0 for success or -1 for failure.

   (d) `listen()`

   The first argument is the server socket file descriptor and the second is the backlog argument which indicates to the OS how many incoming connections should be allowed to queue prior to the accept call. Returns 0 or -1 for success or failure.

   (e) `connect()`

   First argument takes the open socket file descriptor, the second takes the address of the socket, and the last takes the size of the socket. Returns integer for success of the call or failure.

   (f) `close()`

   Only arguement is the socket that needs to be closed.

2. (20 points) For each of the arguments and return value in the socket() call below, briefly explain their meaning for the socket being opened.

```
int sock;
sock = socket(AF_INET, SOCK_STREAM, 0);
```

socket() call returns a file descriptor to the sock variable, using the AF$_{I}NETdomain, SOCK_{S}TREAMtypeandnothin$

3. (10 points) For each of the arguments and return value in the connet() call below, briefly explain their meaning for the connection being established.

```
int res;
res = connect(sock, (struct sockaddr*) saddr_in, sizeof(saddr_in));
```

The connect() call is connecting a sock file descriptor to a server address.

4. (5 points) Explain why when accepting an incoming connection, a new socket is created and returned.

This is because the server may want to continue to use the socket to accept further incoming connections.

5. (5 points) Explain the second argument to listen(), the backlog. What does this argument refer to and what does it mean.

The backlog argument indicates to the OS how many incoming connections should be allowed to queue prior to the accept() before starting to reject connections.

6. (10 points) Look at the output from the hello$_{s}erverprograminthecoursenotes.Whydoestheportchangefromtheclienttot$

The port changes from the client to the server because it doesn't matter what port the client uses to communicate. It only matters that the client and the server agree on the port that they are using to communicate.

7. (10 points) Condiser the code loop for handling client sockets: Can this program handle multiple clients simultaneously? That is, if multiple clients are connected, will the server be able to service all the sockets when data is available? Explain.

```
char buf[BUF_SIZE];
int sockets[NUMSOCKS], i,n;
//iterate over all open sockets
for(i=0;i < NUMSOCKS; i++){
  if(i>0){
    //read from socket
    n = read(sockets[i], buf, BUF_SIZE);
    //socket closed
    if(n<0){
      close(sockets[i]);
      sockets[i] = -1;
    }

    //echo back
    write(sockets[i], buf, n);
  }
 }
```

No, it cannot handle multiple clients because while the server is waiting for something to happen with one client, all the others will just be waiting for something to happen with that client.