

COSC 3360-Operating System Fundamentals

Assignment #2: Predicting your Future

➔ Due Monday, April 9 2018 at 11:59:59 pm ➔

1. OBJECTIVE

You will learn to use stream sockets.

2. YOUR PROGRAMS

You are to write *two* programs:

1. A client program that will connect with your server and send it requests for the average early career and mid-career pays for a specific college major, say “Hospitality management.”
2. A server program that will wait for connection requests from your client and average early career and mid-career pay for the requested college major.

3. THE SERVER PROGRAM

Your server must start by prompting for the name of the file that contains the average early career and mid-career pay for popular college major:

```
Aerospace Engineering 66,300 113,300
Applied Mathematics 57,600 113,200
Engineering Physics 60,400 112,600
```

read it in and store it in a table. It should then prompt for a port to listen to as in

```
Enter server port number: 2468
```

It will then create a stream socket, bind it to the specified port number, do a `listen()` to specify a maximum number of queued connection requests and loop through `accept()` calls that will let it wait for connection requests.

Whenever the server accepts a connection request, it will receive a college major and reply with two numbers respectively representing the average early career and mid-career pay for the requested college major.

Should the college major *not* be in the table, it should reply with two *negative* values. (It is easier that way.)

4. THE CLIENT PROGRAM

Your client should start by prompting the user for a server host name and a server port number as in:

```
Enter server host name: localhost
Enter server port number: 2468
```

It should then create a stream socket and through a loop in which it will prompt the user for a college major, `connect()` the socket to the server port, send a request to the server by doing a `write()` on the socket, `read()` the server reply and print it out on the screen as in:

```
Enter a college major: Statistics
The average early career pay for a Statistics
major is $60000
The corresponding mid-career pay is $104600
```

```
Enter a college major: Electrical Sciences
That major is not in the table
```

```
...
```

Your client should end the loop when the user enters an *empty string*.

HINTS

1. Please refer to the two online socket tutorials at: <http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html> or <http://www.cs.uh.edu/~paris/3360/Sockets.html> through the course Piazza page. It contains a general introduction to sockets. You can include any code from these documents in your submissions.
2. You might also want to look at the client and the server demo programs that will be posted on Piazza. They were tested under Bash for Ubuntu for Windows.
3. Keep in mind that server and client processes read the messages byte by byte and have no way to know how many bytes they should read. The easiest way to do it is to put your messages into fixed size buffers. Both `sprintf()` and `sscanf()` could come handy.
4. Specifying the name of the machine on which you will run your client server/pair as `localhost` instead of, say, `program.cs.uh.edu` will work even if your laptop does not have a valid host name.
5. There will never be more than 512 lines in an input file. Individual entries on each line will be separated by TABS (“\t”).
6. You can safely assume that your input files will always be in the proper format.
7. Use a *single-threaded server* to keep things simple. You will not have to not worry about zombies and can safely ignore the `fireman()` call in the primer.
8. Your client *should* request a new connection for each request as doing otherwise would either require a multithreaded server or prevent the server from handling multiple clients.
9. Yes, you will have to turn in two different programs, namely a client program and a server program.

This document was updated last on Wednesday, February 28, 2018.