COSC 3360/6310 — FUNDAMENTALS OF OPERATING SYSTEMS
COSC 6360 — OPERATING SYSTEMS
ASSIGNMENT #2: GETTING YOUR COVID TEST RESULTS

Now due on Monday November 15, 2021 at 11:59:59 PM

OBJECTIVE

You will learn to use stream sockets.

OVERVIEW

You are to write two programs:

1. A client program that will connect with your server and request your COVID test results.
2. A server program that will wait for connection requests from its clients and reply with their test results (Negative, Positive, No test on record).

THE SERVER PROGRAM

Your server will start by reading in a file named **`testResults20.txt`** that will contain triples consisting of a nine-digit test code, a birthdate in **`MMDDYYYY`** format and a test result code (**`N`** or **`P`**) as in:

```
123456789 03142002 N
987654321 05131996 N
125792468 01022003 N
975318642 02032004 P
…
```

Your server should then prompt for a port to listen to as in

```
Enter the 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 get into an infinite loop that will let it **`accept()`** connection requests.

Whenever the server accepts a connection request, it will receive short ***fixed-size*** messages containing a nine-digit test code and a birthdate in **`MMDDYYYY`** format. For debugging purposes, your server should then print these two inputs along with the associated test result as in:

```
Test code: 987654321
Birthdate: 05131996
Test result: N
```

Use the letter **`X`** to indicate that your database does not have any test result corresponding to the test ID and the birthday your server has just received.

Your server should then send the test results to the client and wait for the next client request.

THE CLIENT PROGRAM

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

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

The only correct answers to the first prompt are the name of the local host as provided by **`gethostname()`** or the keyword **`localhost`**. Your client should ***reject*** any other entry and prompt for a new entry.

It should then create a stream **socket**, do a **`connect()`** request to the specified server, and prompt the user for a nine-digit test ID and a birthdate:

```
Enter your test code:
Enter your birthday in MMDDYYYY format:
```

Your client should then send to the server a ***single message*** containing both values and wait for the reply.

Depending that reply, your client should print one of the three following messages:

```
Your test result was NEGATIVE.
Your test result was POSITIVE
The test result you requested is not in
our database.
```

before terminating.

HINTS

1. Please refer to the two online socket tutorials at:

   http://www.linuxhowtos.org/C_C++/socket.htm
   http://www.cs.uh.edu/~paris/3360/Sockets.html

   or through the course Team. You can include any code from these two documents in your submissions.
2. Be sure that all messages sent by the client have the same length and all replies sent by the server are exactly one-byte long.
3. The library call **`sscanf()`** offers an easy way to split a null-terminated string containing multiple fields.
4. 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.
5. Unlike the server, each client will only handle a single request.
6. Use control-C to terminate your server.

Updated last on Friday, November 12, 2021.