

EE450 Socket Programming Project 1, Fall 2017
Due Date: Thursday October 5th, 2017 11:59 PM (Midnight)
Hard deadline (Strictly enforced)

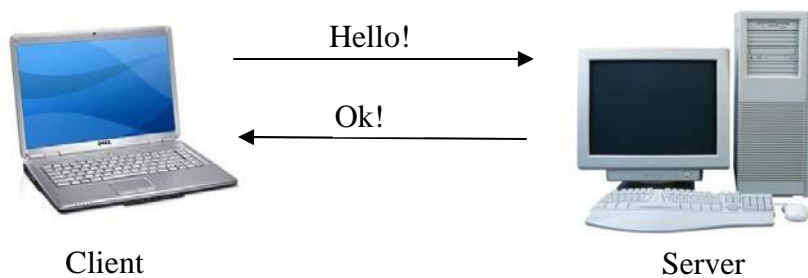
The objective of this assignment is to familiarize you with UNIX socket programming. This assignment is worth **5%** of your overall grade in this course.

It is an individual assignment and no collaborations are allowed. Any cheating will result in an automatic F in the course (not just in the assignment).

If you have any doubts/questions please feel free to contact TA.

The Problem:

In this mini project you will be simulating client-server socket programming using TCP as follows,



Requirements:

1. The client should say your "**Name**" to the server. And the server responds with "**Ok!**" information.
2. Submit your code to Digital Dropbox in blackboard by the deadline.
3. The code should be written with C++, and your code should work on ubuntu14.04.
4. TCP port number should be your 2+last four digits of student id.
ex: if your id is xxxxx-1234, then the port number is 21234
5. You must use the name for client code: **client.cpp** (all small letters).
6. You must use the name for server code: **server.cpp** (all small letters).
7. You are not allowed to pass any parameter or value or string or character as a command-line argument. No user interaction must be required (except for when the user runs the code obviously).
8. Please do remember to close the socket and tear down the connection once you are done using that socket.
9. README.txt file is required, which describes how your code works. And make file is welcome.
10. To run on the local machine please set your sockets as follows.

```
hints.ai_family = AF_UNSPEC;// family type of socket
hints.ai_socktype = SOCK_STREAM;// socket type
hints.ai_flags = AI_PASSIVE;//fill in IP
getaddrinfo(NULL, TCPPORT, &hints, &hs))
socket(hs->ai_family, hs->ai_socktype, hs->ai_protocol)
```

You don't need to follow exactly the same way to run your socket program on the local machine. Just make sure both the ip of server , and client are the ip of local.

Programming platform and environment:

1. All your codes must run on Linux system. (Unix (Mac) or Windows system is not allowed)
2. We will run your code on Ubuntu 14.04.
3. No MS-Windows programs will be accepted.
4. If you are using your own computer at home or at the office, you must download, install VMware, or Virtual box for Ubuntu. Download Ubuntu 14.04 from <http://releases.ubuntu.com/14.04/>
5. Install it on VMware or Virtualbox.

Programming languages and compilers:

You must use **only C++** on Linux as well as Linux Socket programming commands and functions. Here are the pointers for Beej's Guide to C Programming and Network Programming (socket programming):

https://beej.us/guide/bgnet/output/print/bgnet_A4.pdf

<https://beej.us/guide/bgnet/>

(If you are new to socket programming please do study this tutorial carefully as soon as possible and before starting the project)

You must use the following commands and switches to compile yourfile.c or yourfile.cpp. It will make an executable by the name of "yourfileoutput".

For Unix system:

```
g++ -o yourfileoutput yourfile.cpp -lsocket -lnsl -lresolv
```

For Linux system:

```
g++ -o yourfileoutput yourfile.cpp -lnsl -lresolv
```

Note we will test your code only on Ubuntu 14.04 (linux system), but you can still debug on your local machine. Just make sure your program is ok on Ubuntu 14.04.

The TA will use following commands for compilation:

```
g++ -o client client.cpp -lnsl -lresolv
```

```
g++ -o server server.cpp -lnsl -lresolv
```

to test:

```
./server &
```

```
./client &
```

The terminal should show the following message

```
"your name"
```

```
OK!
```

What to submit:

server.cpp, client.cpp, readme, makefile

server.cpp: cpp file to run server program

client.cpp: cpp file to run client program

readme: describe what are your programs doing, your name,

email.id

makefile:

```
g++ -o client client.cpp -lnsl -lresolv
```

```
g++ -o server server.cpp -lnsl -lresolv
```

Submission Rules:

1. Compress all your files (cpp(s), makefile, readme) into a single “tar ball” and call it:
ee450_yourFirstName_yourLastName.tar.gz (all small letters) e.g. my file name would be **ee450_HsiaoLun_Wang.tar.gz**.
Use the command:
tar zcvf ee450_yourFirstName_yourLastName.tar.gz DirName
DirName is the directory which includes cpp, makefile,readme
2. Upload “**ee450_yourFirstName_yourLastName.tar.gz**” to the **Digital Dropbox** (available under Tools) on the blackboard website. After the file is uploaded to the dropbox, you must click on the “**send**” button to actually submit it. If you do not click on send, the file will not be submitted.
3. Please take into account all kinds of possible technical issues and do expect a huge traffic on the blackboard website very close to the deadline which may render your submission or even access to blackboard unsuccessful.
4. Please do not wait till the last 5 minutes to upload and submit your project because you will not have enough time to email the TA and receive a confirmation email from him/her before the deadline.
5. **You have plenty of time to work on this project and submit it in time hence there is absolutely zero tolerance for late submissions! Do NOT assume that there will be a late submission penalty or a grace period. If you submit your project late (no matter for what reason or excuse or even technical issues), you simply receive a zero for the project.**

Grading Criteria:

Your project grade will depend on the following:

1. Correct functionality, i.e. how well your programs fulfill the requirements of the assignment.
2. Inline comments in your code. This is important as this will help in understanding what you have done.
3. Whether your programs work as you say they would in the README file.
4. Whether your programs print out the appropriate error messages and results.
5. If your submitted codes, do not even compile, you will receive 20 out of 100 for the project.
6. If your submitted codes, compile but when executed, produce runtime errors without performing any tasks of the project, you will receive 20 out of 100.
7. If your codes compile but when executed only perform one of two messages correctly, you will receive 50 out of 100.

8. If your codes compile and perform two messages correctly, you will receive 100 out of 100.
9. If you forget to include any of the code files or the README file in the project tar-ball that you submitted, you will lose 5 points for each missing file.
10. You will lose 5 points for each error or a task that is not done correctly.
11. The minimum grade for an on-time submitted project is 20 out of 100.
12. Your code will not be altered in any ways for grading purposes and no special test cases are used to grade your project. The TAs run your project as is, according to the project description and your README file and then check whether it works correctly or not.

Cautionary Words:

1. Start on this project early!!!
2. You may create zombie processes while testing your codes, please make sure you kill them every time you want to run your code. To see a list of all zombie processes even from your past logins to nunki, try this command: `ps -aux`
3. Identify the zombie processes and their process number and kill them by typing at the command-line:
4. Kill -9 processnumber
5. *Please do remember to terminate all zombie or background processes, otherwise they hold the assigned port numbers and sockets busy and can't be used for other testing.*

Academic Integrity:

All students are expected to write all their code on their own.

Copying code from friends is called **plagiarism** not **collaboration** and will result in an F for the entire course. Any libraries or pieces of code that you use and you did not write, must be listed in your README file. All programs will be compared with automated tools to detect similarities; examples of code copying will get an F for the course. **IF YOU HAVE ANY QUESTIONS ABOUT WHAT IS OR ISN'T ALLOWED ABOUT PLAGIARISM, TALK TO THE TA.** "I didn't know" is not an excuse.