CSE 333 13su Exercise 13

out: Wednesday, Aug. 7, 2013
due: Monday, Aug. 12, 2013 by **9:00 am**.

Write a C++ program that accepts three command line arguments:

- the hostname of a server
- the port number of that server
- the name of a local file

The program should connect (via TCP) to the server on the supplied hostname and port. Once connected, the program should read in the bytes from the local file, and it should write those bytes over the TCP connection. Once all of the bytes have been written, the program should close the TCP connection and exit.

To test your program, you can run a server using the "nc" program. For example, to run the server on port 5555, and to have the server redirect the incoming bytes to file "output.bin", run the following command:

```
nc -l 5555 > output.bin
```

Note that nc will exit once it has processed a single connection, so you'll need to rerun nc each time you test your client.

There are a few requirements on your code:

- you should modularize it nicely; consider splitting it into multiple source files if that makes sense, for example by isolating all of the network-specific code in a module.
- you should use read/write both to read from the input file and write over the socket to the server; note this means you need to pay attention to the possibility that read/write might return EINTR, and they might read/write less than you ask for. We recommend writing some utility functions to deal with this.
- write a Makefile so that we can compile your code by typing "make". Your makefile should produce a binary called "ex13".

In addition, your code must:

- compile without errors or warnings on CSE Linux machines (lab workstations, attu, or CSE home VM)
- have no crashes, memory leaks, or memory errors on CSE linux machines
- follow the style guidelines we've been using in class, including naming conventions for classes, methods, and instance variables. If in doubt, consult the Google C++ style guide (Google it to find it if you missed the link on the course web page). Cpplint is also useful for suggesting things to check.
- be pretty: the formatting, modularization, variable and function names, and so on must make us smile rather than cry.
- have a comment at the top of the source file with your name, student number, and CSE or UW email address.

You should submit your exercise using the assignment dropbox linked on the main course web page.