CS 484/504 Programming Assignment I: Emailing in C

Fall 2022
Graduate Level Course Code CS 504
Computer Science Department, New Mexico State University

Graduate Level class participants must meet the graduate extension requirements

1 Requirements

Programming with sockets in C will expose you to low-level Linux networking, and is a rite of passage for computer scientists. Write a program in C which uses network sockets to connect to an SMTP server and send an email.

./email_sender ⟨SMTP ADDR IPv4⟩ ⟨DEST Email ADDR⟩ ⟨Email Filename⟩

Your program must follow a consistent style, include explanatory comments, and include a Makefile that allows it to be compiled from your source code using the GCC compiler. We will test on Ubuntu 20.04, we strongly suggest you test your code on a virtual machine running that OS!

Graduate students, your email sender must properly close all connections, free all allocated memory, and have no memory leaks. We will check with Valgrind

2 Error Conditions

Your program must handle error conditions. Specifically, you will be okay if you handle all of the following:

- Incorrect number of command line arguments specified
- Invalid email address specified (try a regular expression for this. Don't worry about IP email addresses – hostname only is fine)
- Invalid IP Address Specified (Don't worry about supporting IPv6)
- Invalid or nonexistent email file specified
- Socket creation or connection to SMTP server failed
- SMTP server rejects a command
- SMTP server sends multi-line responses
- SMTP server blocks you from sending email too often

3 Networking Standards and SMTP

You will need to learn how the SMTP works to do this assignment. We will discuss this in class. In case you want to review or double-check something about SMTP, some great resources are:

- https://www.ietf.org/rfc/rfc2821.txt

- https://postmarkapp.com/guides/everything-you-need-to-know-about-smtp

Remember that for TCP over IP, the standard byte order is big endian. We will be testing on a little endian machine, so be sure to change the TCP Port to big-endian when connecting.

Email services HATE spam. If you look like a spammer, they will happily block you. For this assignment, only send email to our university SMTP server.

4 No Sender Spoofing

You will see that there is no requirement in SMTP to prove that you actually are authorized to send email on behalf of a particular address. You must not spoof the sender by pretending to be anyone else. Send emails as if they are coming from your school email. Set the destination as your own email address as well. If you get in trouble for sending other things, there may be criminal or school disciplinary consequences and we will not bail you out!

Also notice that Outlook marks your emails with the warning which usually only is added to external emails. NMSU recently turned on security features that allow them to check whether an email was sent from their actual email server.

5 Helpful Functions:

```
// Copies all received data from the TCP socket buffer to <buff>.
// This is a blocking call.
// Suggested buffer length for SMTP: 1024 or more bytes.
// Returns the number of bytes read
int socket_receive(int ssocket, char* buff, int buffsize) {
    return read(ssocket,buff, buffsize);
        //https://man7.org/linux/man-pages/man2/read.2.html
}
```

```
// Sends data out over a TCP socket.
// Warning: this function will work fine for sending SMTP messages line by line
    (because they are short), but in general you many need to handle the case were
    only part of your message is sent at a time.
// Returns the number of bytes sent
int socket_send(int ssocket, char* msg, int msglen) {
    return write(ssocket,msg,msglen);
        //https://man7.org/linux/man-pages/man2/write.2.html
}
```

5.1 Manual pages for other helpful library functions you may need:

- read(2) for reading from anything file-like
- write(2) for writing to anything file-like
- socket(2) to create a socket

- connect(2) to tell the OS to do a TCP handshake and attach the TCP conversation to a specific socket
- inet_pton(3) to convert an IP address string into binary form
- htons(2) to convert a little-endian port number to big endian (network standard order)
- fopen(3) to open a file
- fseek(3) to move the reading pointer in a file
- ftell(3p) to measure how far into a file you have moved the reading pointer
- getline(3) to get bytes from a file line-by-line

6 Allowed Libraries and Sources

If you want to use any libraries other than the following, you need to obtain permission from course staff. Everything else you must write, yourself.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <regex.h>
#include <string.h>
```

7 Sample Email

The following is a template email you could fill out and save as a plain text file and use to test your email sender. Those lines at the top are email headers. You can use those to set things like the subject line of your emails. Remember that SMTP expects CRLF line endings!

```
From: CS 484 Project <[[EMAIL]]@nmsu.edu>
To: [[NAME]] <[[EMAIL]]@nmsu.edu>
Subject: CS484 - I can program with sockets!

Hello [[NAME]],

Congratulations! If you are reading this in your email client, then something has
    gone right.

Yours most faithfully,
[[NAME]]
```

8 What to Submit

A zipped directory containing:

1. The C source file(s) for your email sender
2. A makefile where the default rule compiles your code from source

9 Acceptable collaboration

Please help each other be successful on this assignment. The rule you must follow to ensure you are not plagiarizing is never to show your code to anyone else nor read anyone else's code. You may look at resources like stack overflow to learn how functions work, etc. If you use any small code snippets from an online resource, cite them with a URL in a comment. You many not copy a pre-built SMTP sender from github because you will learn nothing.

If we are concerned you have plagiarized, we may ask you to come walk us how everything in your code works and explain the design decisions you made. If you don't know what is going on, this could obviously become very awkward...