# INFA723 Cryptography and Network Security

# Lab3 Send/Receive a File through a Socket Connection

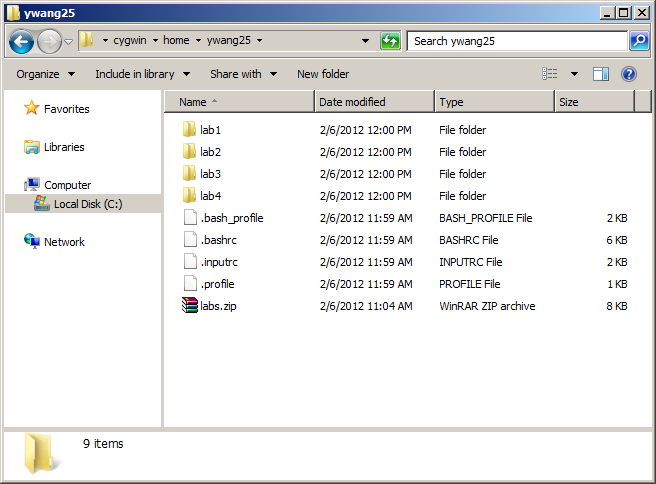
# 1 Introduction

We can send/receive a file through a socket connection. In this exercise, we will send/receive a file through a socket connection. A lab package has been posted on the class’ website. If you have not installed the package on your computer, follow Section 2 to install the package. If you have already installed the package on your computer, go to Section 3.

The lab procedure assumes the Cygwin is used. If you have a Linux OpenSSL environment, open a terminal and the lab procedure is similar.

# 2 Setup

A Lab package (labs1-4.zip) has been posted on the class’s website. Go to class’s website and download the package to your local computer. Assume you use the default Cygwin installation directory “C:\cygwin”, copy labs.zip to “C:\cygwin\home” directory. Unzip the file to your home directory and you will see four Labs listed here: lab1, lab2, lab3, and lab4.



# Two execution files will be created in the Lab, client.exe and server.exe. The formation of these two programs is listed below.

client server\_ip\_address port\_num plain\_text\_file

server\_ip\_address: server’s IP address.

port\_num: server’s port number.

plain\_text\_file: a text file you want to send to server

server host\_ip\_address port\_num

host\_ip\_address: host IP address

port\_num: server’s port number

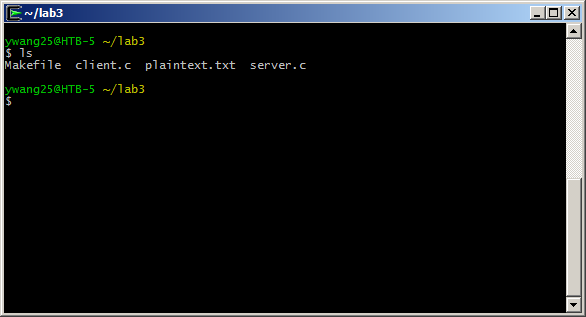
# 3 Send/Receive a File through a Socket Connection

Double click Cygwin icon on your desktop and open a new Cygwin terminal. Follow the steps below to finish the lab.

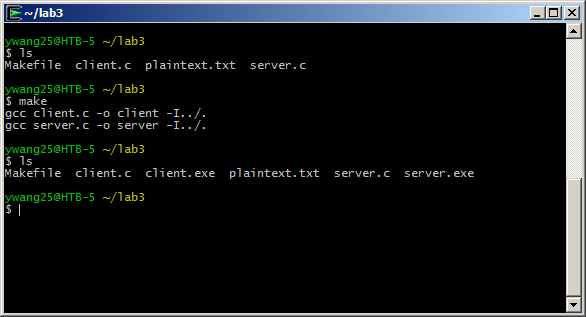
1. Go to “lab3” directory.

$ cd lab3

1. Check the files under the directory and make sure you have all the files as listed below:



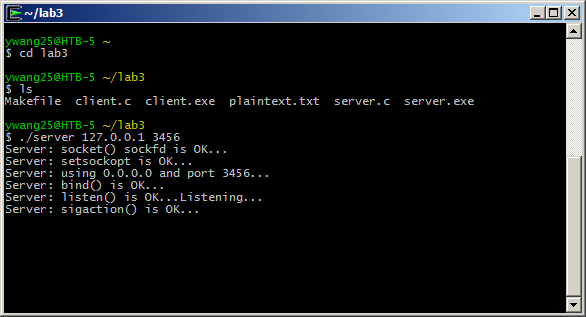
1. Enter and run “make” command in the prompt. Two new files client.exe and server.exe will be created.



1. Open another Cygwin window (for server program, we will use previous windows for client program) and go to “lab3” directory.
2. In the server Cygwin window, enter and run the command

$./server 127.0.0.1 3456

127.0.0.1 is your host IP address and 3456 is an unused port number.



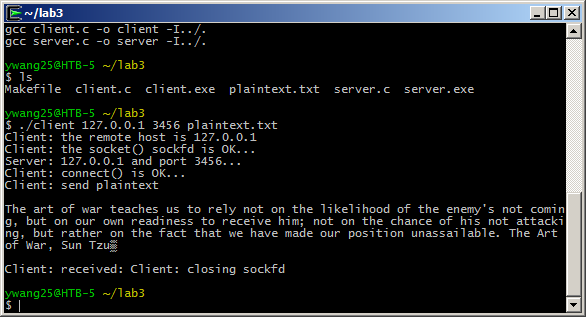
Windows will pop up a window to ask if you allow the access. Click the button “Allow access” to temporarily authorize the access.



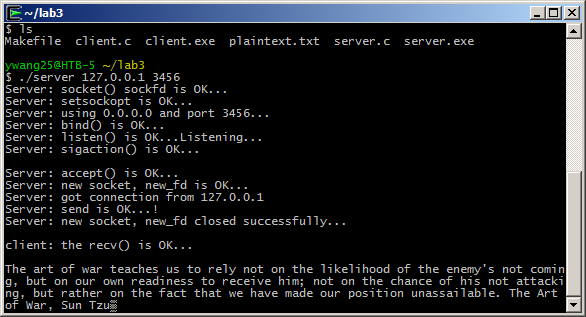
1. In the client Cygwin window, enter and run the command

$ ./client 127.0.0.1 3456 plaintext.txt

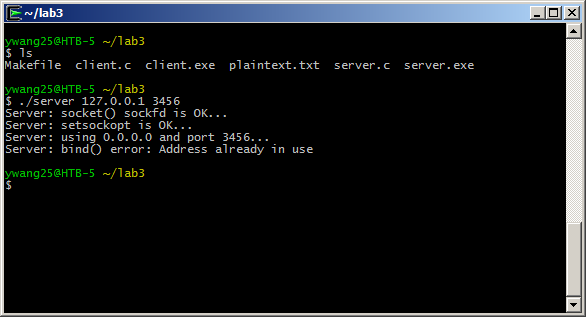
127.0.0.1 is you host IP address. 3456 is the port number and plaintext.txt is a clear text.



1. Check both client and server Cygwin windows and you will see a file is sent from the client to the server. Client Cygwin output is shown above and server Cygwin output is shown below.



1. You can enter and run the command “./client 127.0.0.1 3456 plaintext.txt” again and you will see the server window keeps receiving the file.
2. Use a text editor to open Client.c and server.c to see how the source code is structured.
3. If you stop the server program (ctrl-d) and run the server program again. You may see a bind error message like below:



This is because the port number is not released after you terminated the server program. You can force kill the process (using kill) or simply use another port number to start the server program. Make sure the client and the server program use the same IP address and port number every time.

# 4 Questions

1. (5 pts) Check the source code, client.c and server.c. Describe the differences of the implementation on the client side and the server side.
2. (5pts) What is a port in an operating system? What are the differences between a TCP port and UDP port?
3. (5pts) Which applications or protocols use ports 21, 22, 23, 80, 443, 3306, 3389?