Department of Computer Science
Computer Networks
Due: Sunday 30th August (23.59)

Student: Ýmir órleifsson
TA Name: Hópur 5
Time Taken: 3+3+6+2+2
Estimated Time: 20 hours

This is an individual assignment and should be submitted as a pdf, with accompanying code, using Canvas.

For those who like to dabble in the dark arts, the latex version is also available. You may submit in any legible form you wish, but please use tar to bundle files.

Practical programming exercises may be done on your local laptop, or using your account on skel.ru.is. Part of this assigment is getting your programming environment setup for the rest of this course. We strongly recommend that you create a suitable environment on your laptop or other machine which you can use to run client and monitoring software such as tcpdump. If you have any issues at all in getting setup, please contact us **immediately**.

Marks are awarded for question difficulty. While there is typically a relationship between difficulty and length of answer, it may not be a strong one. Always justify your answer if necessary, especially with somewhat open ended design questions.

All submissions must be bundled up using the *tar* command. If you submit an assignment using anything else (zip say), and persuade Canvas to accept the upload by renaming the file, you will receive an automatic 0.

Question:	1	2	3	4	Total
Points:	10	15	15	10	50
Score:					

Useful Network Commands (Linux)

man	Online manual for command	man -k	keyword search on manual
whois	Domain registry information	htop	Enhanced top, show processes
iftop	Show top traffic/network	iptraf-ng	IP traffic monitor
route	show local routing	arp -v	Show address resolution cache(root)
ncat < *ip* >< *port* >	Connect to remote system	nmap	Scan remote host
netstat	Show network statistics	netstat -antup	Process path info
netstat -t	Show only tcp connections	netstat -u	show only udp connections
ss	Show detailed network info	ss -s	Summary of network info
ifconfig	Network Interface	ip	Enhanced ipconfig/network info.
nmap -A -T4 scanme.nmap.org		scan host ports - os, version and traceroute	
dig	DNS record lookup	dig < *domain* >	Get full record for < *domain* >
nslookup	interactively query DNS	set type=A	Specify A records
		server=	Specify server to query

Introduction

The commands above are a summary list of command line tools that can be used for networking purposes. In particular, nmap and ncat (note, another version of ncat called nc exists, but is not always as reliable), will be useful for this assignment. Some of these commands may need to be installed using the package management for your machine, and should be available via the bash console on Windows.

Network Connectivity

The goal of this exercise is to first introduce you to some useful network command line tools, to help you explore and debug network programs, and then get you to write your first very simple network program.

1 .... *10 points*

Download the server.cpp file attached to this assignment. This is a very simple TCP/IP server, which listens for incoming TCP connections on port 5000. Compile and run it using the commands:

```
Linux and Windows/Bash:
    g++ -Wall -std=c++11 server.cpp -o server
OSX
    g++ -std=c++11 server.cpp -o server

./server
```

Note, you may need to install g++ (On OSX use brew).

Now connect to it from the same machine you ran the server on using the standard loopback network interface "127.0.0.1" by performing the following steps.

From a separate terminal, verify that you can connect to the server which is listening on port 5000, using the ncat command:

```
ncat 127.0.0.1 5000
```

The server should tell you a client has connected. Now on a third terminal, run the tcpdump command (which may require sudo privilege) to monitor the traffic between the client and the server, using the following incantation:

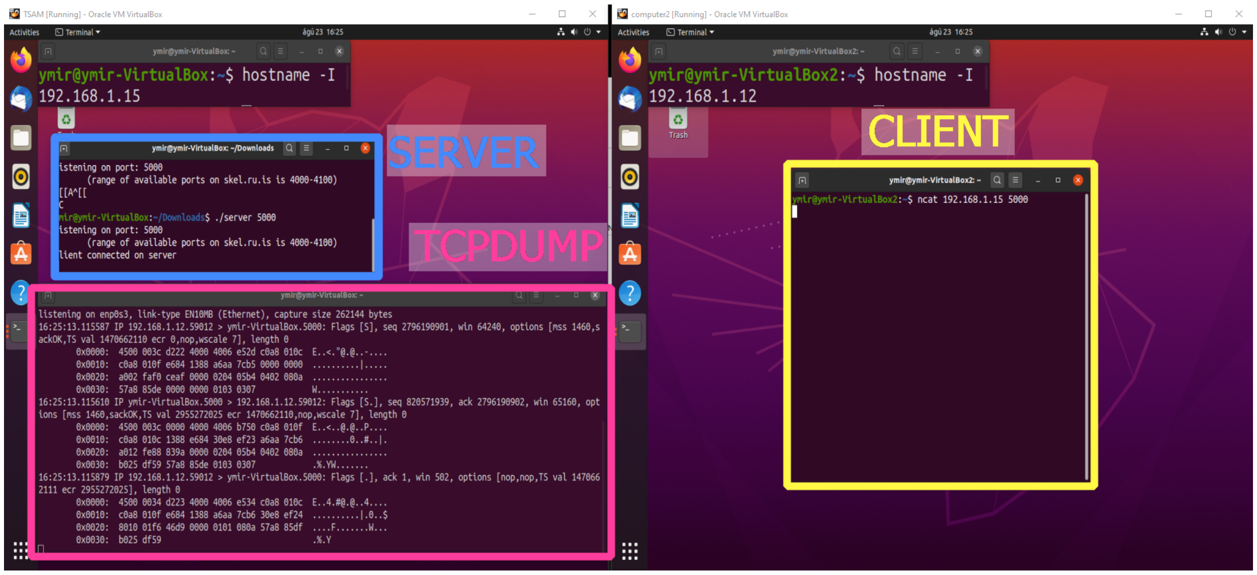
```
tcpdump –X –i lo host 127.0.0.10 and port 5000
```

The server supports one command, SYS < *command* > which will run the one word command specified on the server. Enter a command of your choice (eg. who, ls, w, etc), and observe the results on your terminal.

(a) (10 points) For full marks on this question, repeat the above series of steps, connecting to the server from a **different** machine than the one the server is running from, and submit a screen capture of each terminal, ncat, ./server and tcpdump (three in total).

If for **any** reason you are not able to connect from a different machine, submit the screen captures from the same machine as above, and a convincing explanation, including the name of the TA you spoke to to get help, of the issue you ran into.

Answer


TSAM [Running] - Oracle VM VirtualBox
Activities
Terminal
ágú 23 16:25
SERVER
TCPDUMP
computer2 [Running] - Oracle VM VirtualBox
Trash
CLIENT

```
ymir@ymir-VirtualBox:~$ hostname -I
192.168.1.15
```

```
istening on port: 5000
      (range of available ports on skel.ru.is is 4000-4100)
[[A^[[
C
mir@ymir-VirtualBox:~/Downloads$ ./server 5000
istening on port: 5000
      (range of available ports on skel.ru.is is 4000-4100)
lient connected on server
```

```
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
16:25:13.115587 IP 192.168.1.12.59012 > ymir-VirtualBox.5000: Flags [S], seq 2796190901, win 64240, options [mss 1460,s
ackOK,TS val 1470662110 ecr 0,nop,wscale 7], length 0
        0x0000:  4500 003c d222 4000 4006 e52d c0a8 010c  E..<."@.@..-....
        0x0010:  c0a8 010f e684 1388 a6aa 7cb5 0000 0000  ..........|.....
        0x0020:  a002 faf0 ceaf 0000 0204 05b4 0402 080a  ................
        0x0030:  57a8 85de 0000 0000 0103 0307            W...........
16:25:13.115610 IP ymir-VirtualBox.5000 > 192.168.1.12.59012: Flags [S.], seq 820571939, ack 2796190902, win 65160, opt
ions [mss 1460,sackOK,TS val 2955272025 ecr 1470662110,nop,wscale 7], length 0
        0x0000:  4500 003c 0000 4000 4006 b750 c0a8 010f  E..<..@.@..P....
        0x0010:  c0a8 010c 1388 e684 30e8 ef23 a6aa 7cb6  ........0..#..|.
        0x0020:  a012 fe88 839a 0000 0204 05b4 0402 080a  ................
        0x0030:  b025 df59 57a8 85de 0103 0307            .%.YW.......
16:25:13.115879 IP 192.168.1.12.59012 > ymir-VirtualBox.5000: Flags [.], ack 1, win 502, options [nop,nop,TS val 147066
2111 ecr 2955272025], length 0
        0x0000:  4500 0034 d223 4000 4006 e534 c0a8 010c  E..4.#@.@..4....
        0x0010:  c0a8 010f e684 1388 a6aa 7cb6 30e8 ef24  ..........|.0..$
        0x0020:  8010 01f6 46d9 0000 0101 080a 57a8 85df  ....F.......W...
        0x0030:  b025 df59                                .%.Y
```

```
ymir@ymir-VirtualBox2:~$ hostname -I
192.168.1.12
```

```
ymir@ymir-VirtualBox2:~$ ncat 192.168.1.15 5000
```

One big complication I had with this assignment is that when I was simulating the machines, Oracle VM VirtualBox would attach the virtual OS' to NAT.
I had to change this setting to Bridged Adapter. With this changed I was able to get a working IP address.
I had to modify the given tcpdump command to:

```
sudo tcpdump –X –i enp0s3 host 192.168.1.15 and port 5000
```

2 . . . . *15 points*

For the second part of this assignment, you need to write a client program that connects to the server remotely, in the same way you used ncat to do. All the client needs to do is to connect to the server, and send a SYS command to the server. For full marks for this question submit:

(a) (10 points) Client code connecting as described above

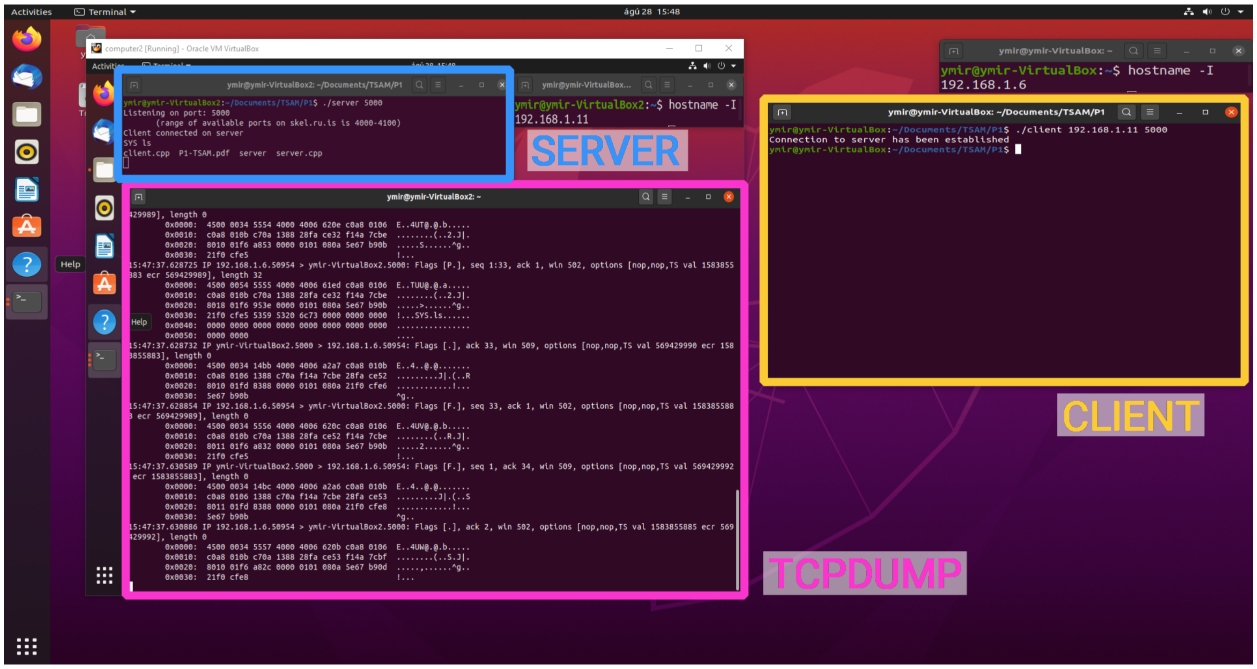
(b) (2 points) Screen shot of server receiving command and executing it

(c) (3 points) tcpdump of command being sent to server

Answer

```

// Simple client for TSAM-409 Assignment 1
//
// Compile: g++ -Wall -std=c++11 client.cpp -o client
//
// Command line: ./client <ip address> <ip port>
//
// Author: Ymir Thorleifsson (ymir19@ru.is)

#include <sys/socket.h>
#include <sys/types.h>
#include <arpa/inet.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netinet/in.h>

#include <iostream>

int sock;
char buffer[4096];

int establish_connection(char ip_addr[], char port[]) {

    if ((sock = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0) {
        printf("%n",&sock);
        perror("Failed to make socket");
        return -1;
    }

    struct sockaddr_in serv_addr;
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_port = htons(atoi(port));

    if (inet_pton(AF_INET, ip_addr, &serv_addr.sin_addr) <= 0) {
        perror("Failed to set socket address");
        return -1;
    }

    if(connect(sock, (struct sockaddr *) &serv_addr, sizeof(serv_addr)) < 0) {
        perror("Failed to connect to server");
        return -1;
    }

    printf("Connection to server has been established\n");

    return 0;
}

```

```

int main(int argc, char* argv[]) {

    if(argc != 3) {
        printf("Usage: client <ip address> <ip port>\n");
        exit(0);
    }

    establish_connection(argv[1], argv[2]);


    char cmd[32] = "SYS ls";
    send(sock, cmd, sizeof(cmd), 0);

}
```



```
ymir@ymir-VirtualBox2:~/Documents/TSAM/P1$ ./server 5000
Listening on port: 5000
        (range of available ports on skel.ru.is is 4000-4100)
Client connected on server
SYS ls
client.cpp  P1-TSAM.pdf  server  server.cpp
```

```
ymir@ymir-VirtualBox2:~$ hostname -I
192.168.1.11
```

```
ymir@ymir-VirtualBox:~$ hostname -I
192.168.1.6
```

```
ymir@ymir-VirtualBox:~/Documents/TSAM/P1$ ./client 192.168.1.11 5000
Connection to server has been established
ymir@ymir-VirtualBox:~/Documents/TSAM/P1$
```

```
429989], length 0
        0x0000:  4500 0034 5554 4000 4006 620e c0a8 0106  E..4UT@.@.b.....
        0x0010:  c0a8 010b c70a 1388 28fa ce32 f14a 7cbe  ........(..2.J|.
        0x0020:  8010 01f6 a853 0000 0101 080a 5e67 b90b  .....S......^g..
        0x0030:  21f0 cfe5                                !...
15:47:37.628725 IP 192.168.1.6.50954 > ymir-VirtualBox2.5000: Flags [P.], seq 1:33, ack 1, win 502, options [nop,nop,TS val 1583855
883 ecr 569429989], length 32
        0x0000:  4500 0054 5555 4000 4006 61ed c0a8 0106  E..TUU@.@.a.....
        0x0010:  c0a8 010b c70a 1388 28fa ce32 f14a 7cbe  ........(..2.J|.
        0x0020:  8018 01f6 953e 0000 0101 080a 5e67 b90b  .....>......^g..
        0x0030:  21f0 cfe5 5359 5320 6c73 0000 0000 0000  !...SYS.ls......
        0x0040:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0050:  0000 0000                                ....
15:47:37.628732 IP ymir-VirtualBox2.5000 > 192.168.1.6.50954: Flags [.], ack 33, win 509, options [nop,nop,TS val 569429990 ecr 158
3855883], length 0
        0x0000:  4500 0034 14bb 4000 4006 a2a7 c0a8 010b  E..4..@.@.......
        0x0010:  c0a8 0106 1388 c70a f14a 7cbe 28fa ce52  .........J|.(..R
        0x0020:  8010 01fd 8388 0000 0101 080a 21f0 cfe6  ............!...
        0x0030:  5e67 b90b                                ^g..
15:47:37.628854 IP 192.168.1.6.50954 > ymir-VirtualBox2.5000: Flags [F.], seq 33, ack 1, win 502, options [nop,nop,TS val 158385588
3 ecr 569429989], length 0
        0x0000:  4500 0034 5556 4000 4006 620c c0a8 0106  E..4UV@.@.b.....
        0x0010:  c0a8 010b c70a 1388 28fa ce52 f14a 7cbe  ........(..R.J|.
        0x0020:  8011 01f6 a832 0000 0101 080a 5e67 b90b  .....2......^g..
        0x0030:  21f0 cfe5                                !...
15:47:37.630589 IP ymir-VirtualBox2.5000 > 192.168.1.6.50954: Flags [F.], seq 1, ack 34, win 509, options [nop,nop,TS val 569429992
 ecr 1583855883], length 0
        0x0000:  4500 0034 14bc 4000 4006 a2a6 c0a8 010b  E..4..@.@.......
        0x0010:  c0a8 0106 1388 c70a f14a 7cbe 28fa ce53  .........J|.(..S
        0x0020:  8011 01fd 8388 0000 0101 080a 21f0 cfe8  ............!...
        0x0030:  5e67 b90b                                ^g..
15:47:37.630886 IP 192.168.1.6.50954 > ymir-VirtualBox2.5000: Flags [.], ack 2, win 502, options [nop,nop,TS val 1583855885 ecr 569
429992], length 0
        0x0000:  4500 0034 5557 4000 4006 620b c0a8 0106  E..4UW@.@.b.....
        0x0010:  c0a8 010b c70a 1388 28fa ce53 f14a 7cbf  ........(..S.J|.
        0x0020:  8010 01f6 a82c 0000 0101 080a 5e67 b90d  .....,......^g..
        0x0030:  21f0 cfe8                                !...
```

The server received the SYS ls command from the client that is on another machine.

3 . . . . *15 points*

For the third part of this assignment, you need to modify both the server, and the client you have just written.

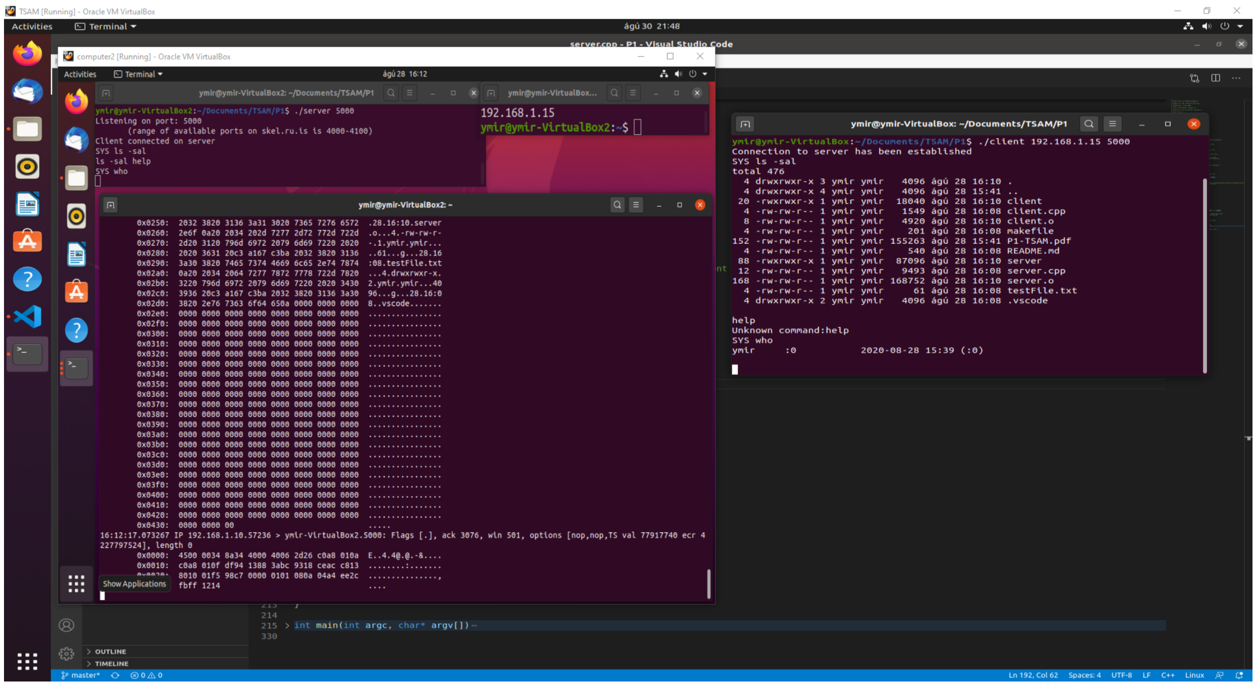
The server should be modified to send the client the output from the command executed as a response and also to handle parameters on the command being sent from the client. For example, ”ls -sal”. The client should be modifed to receive the output of the commend from the server, and to print it out on the comand line, and then be ready to accept and send another command. That is, the client operates in a loop, send command, receive results, print, send command, etc., as network clients often do.

(a) (5 points) Modified server code meeting above specification

(b) (5 points) Modified client code meeting above specification

(c) (5 points) tcpdump of at least two commands being sent to server and response being received.

Answer


TSAM [Running] - Oracle VM VirtualBox
Activities
Terminal
ágú 30 21:48
server.cpp - P1 - Visual Studio Code
computer2 [Running] - Oracle VM VirtualBox
ágú 28 16:12
ymir@ymir-VirtualBox2: ~/Documents/TSAM/P1
ymir@ymir-VirtualBox...
ymir@ymir-VirtualBox2: ~
ymir@ymir-VirtualBox: ~/Documents/TSAM/P1
Show Applications
OUTLINE
TIMELINE
master*
Ln 192, Col 62
Spaces: 4
UTF-8
LF
C++
Linux

```
ymir@ymir-VirtualBox2:~/Documents/TSAM/P1$ ./server 5000
Listening on port: 5000
      (range of available ports on skel.ru.is is 4000-4100)
Client connected on server
SYS ls -sal
ls -sal help
SYS who
```

```
192.168.1.15
ymir@ymir-VirtualBox2:~$
```

```
        0x0250:  2032 3820 3136 3a31 3020 7365 7276 6572  .28.16:10.server
        0x0260:  2e6f 0a20 2034 202d 7277 2d72 772d 722d  .o...4.-rw-rw-r-
        0x0270:  2d20 3120 796d 6972 2079 6d69 7220 2020  -.1.ymir.ymir...
        0x0280:  2020 3631 20c3 a167 c3ba 2032 3820 3136  ..61...g...28.16
        0x0290:  3a30 3820 7465 7374 4669 6c65 2e74 7874  :08.testFile.txt
        0x02a0:  0a20 2034 2064 7277 7872 7778 722d 7820  ...4.drwxrwxr-x.
        0x02b0:  3220 796d 6972 2079 6d69 7220 2020 3430  2.ymir.ymir...40
        0x02c0:  3936 20c3 a167 c3ba 2032 3820 3136 3a30  96...g...28.16:0
        0x02d0:  3820 2e76 7363 6f64 650a 0000 0000 0000  8..vscode.......
        0x02e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x02f0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0300:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0310:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0320:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0330:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0340:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0350:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0360:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0370:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0380:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0390:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03a0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03b0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03c0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03d0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03e0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x03f0:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0400:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0410:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0420:  0000 0000 0000 0000 0000 0000 0000 0000  ................
        0x0430:  0000 0000 00                             .....
16:12:17.073267 IP 192.168.1.10.57236 > ymir-VirtualBox2.5000: Flags [.], ack 3076, win 501, options [nop,nop,TS val 77917740 ecr 4
227797524], length 0
        0x0000:  4500 0034 8a34 4000 4006 2d26 c0a8 010a  E..4.4@.@.-&....
        0x0010:  c0a8 010f df94 1388 3abc 9318 ceac c813  ........:.......
        0x0020:  8010 01f5 98c7 0000 0101 080a 04a4 ee2c  ...............,
                 fbff 1214                                ....
```

```
ymir@ymir-VirtualBox:~/Documents/TSAM/P1$ ./client 192.168.1.15 5000
Connection to server has been established
SYS ls -sal
total 476
  4 drwxrwxr-x 3 ymir ymir   4096 ágú 28 16:10 .
  4 drwxrwxr-x 4 ymir ymir   4096 ágú 28 15:41 ..
 20 -rwxrwxr-x 1 ymir ymir  18040 ágú 28 16:10 client
  4 -rw-rw-r-- 1 ymir ymir   1549 ágú 28 16:08 client.cpp
  8 -rw-rw-r-- 1 ymir ymir   4920 ágú 28 16:10 client.o
  4 -rw-rw-r-- 1 ymir ymir    201 ágú 28 16:08 makefile
152 -rw-rw-r-- 1 ymir ymir 155263 ágú 28 15:41 P1-TSAM.pdf
  4 -rw-rw-r-- 1 ymir ymir    540 ágú 28 16:08 README.md
 88 -rwxrwxr-x 1 ymir ymir  87096 ágú 28 16:10 server
 12 -rw-rw-r-- 1 ymir ymir   9493 ágú 28 16:08 server.cpp
168 -rw-rw-r-- 1 ymir ymir 168752 ágú 28 16:10 server.o
  4 -rw-rw-r-- 1 ymir ymir     61 ágú 28 16:08 testFile.txt
  4 drwxrwxr-x 2 ymir ymir   4096 ágú 28 16:08 .vscode

help
Unknown command:help
SYS who
ymir     :0           2020-08-28 15:39 (:0)
```

```
int main(int argc, char* argv[])…
```

4 . . . . *10 points*

In addition to the marks for the individual questions, the following marks will be awarded:

(a) (2 points) A Makefile is included which compiles the code submitted

(b) (2 points) README file is provided explaining how to compile and run submitted code. File should include command line commands used to compile (IDE's will not be accepted), and instructions on how to run the programs.

(c) (2 points) Code compiles using the Makefile, and runs following instructions in README file

(d) (4 points) Code is clearly structured (no 200 line functions), and well commented. Variable names are informative, and each function/class has a header describing its purpose.