Name .... ID .... Section ...... Seat No ......

Sirindhorn International Institute of Technology
Thammasat University

Final Exam: Semester 2, 2010

Course Title: ITS332 Information Technology Laboratory II

Instructor: Steven Gordon

Date/Time: Friday 4 March 2011; 13:30–15:00

Instructions:

- This examination paper has 14 pages (including this page).
- Conditions of Examination: Closed book; No dictionary; Non-programmable calculator is allowed
- Students are not allowed to be out of the exam room during examination. Going to the restroom may result in score deduction.
- Students are not allowed to have communication devices (e.g. mobile phone) in their possession.
- Write your name, student ID, section, and seat number clearly on the front page of the exam, and on any separate sheets (if they exist).
- Assume the user in all questions has administrator privileges (that is, you can ignore the need for `sudo`).
- Reference material at the end of the exam may be used.

Question 1 [4 marks]

Consider an internet that has the following components and requirements:

- Subnet A is a switched Fast (100Mb/s) Ethernet LAN with four hosts (H1, H2, H3, H4) and one router (R1).
- Subnet B is a switched 10Mb/s Ethernet LAN with three hosts (H5, H6, H7) and one router (R2).
- Subnet C is a switched Fast (100Mb/s) Ethernet LAN with two hosts (H8, H9) and two routers (R2 and R3).
- Subnet D connects subnets A and C using a single Ethernet cable
- Subnet E contains only one host (H10) connected direct to a router.
- All 10 hosts can communicate with each other.
- There are no other subnets, routers or hosts in the internet.

(a) Draw a diagram illustrating the design of the internet. Use squares to represent hosts, circles to represent routers and rectangles to represent other devices if present. Label each device with a descriptive name (e.g. H1, R1). Use lines to represent cables. Label each cable with the type used: straight-through or cross-over. There may be multiple correct designs. [4 marks]

Question 2 [13 marks]

You have four computers (C1, C2, C3, C4) with current interface information as presented below. You want to connect the computers into a single internet with three subnets and two hosts, i.e. C1—C2—C3—C4. The subnet addresses you must use are: `111.111.0.0/16`, `222.222.222.0/24` and `3.0.0.0/8`. One host must be on the subnet `111.111.0.0/16` and the other host on subnet `222.222.222.0/24`. You have various Ethernet cables available, but no other devices.

```
C1:
eth0      Link encap:Ethernet  HWaddr 20:cf:30:a8:f7:be
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0

C2:
eth0      Link encap:Ethernet  HWaddr 00:17:31:5a:e7:e8
          inet addr:111.111.1.2  Bcast:111.111.255.255  Mask:255.255.0.0

eth1      Link encap:Ethernet  HWaddr 00:17:31:5a:e7:d3
          inet addr:3.0.0.1  Bcast:3.255.255.255  Mask:255.0.0.0

C3:
eth3      Link encap:Ethernet  HWaddr 48:14:66:23:a4:b5
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0

eth2      Link encap:Ethernet  HWaddr 48:14:66:23:a4:b6
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0

C4:
eth0      Link encap:Ethernet  HWaddr 00:17:31:5a:e5:89
          inet addr:222.222.222.2  Bcast:222.222.222.255  Mask:255.255.255.0
```

Explain how to setup the network by answering the following questions. When giving commands for the answers, use the lines provided and where necessary show them in order. You do not need to use all lines provided—you could add more lines if necessary.

(a) For computers C1 and C3, show the complete command(s), including any input arguments, to set the correct IP addresses. (IP address of C2 and C4 are already correctly set) [3 marks]

C1: ____

____

____

C3: ____

____

____

(b) Assuming after setting the IP addresses the routing tables for computers C1 and C2 are empty, show the complete command(s), including any input arguments, to set the correct routing tables. (Routing tables for C3 and C4 are correctly configured) [4 marks]

C1: ____

____

____

____

____

C2: ____

____

____

____

____

(c) Explain what else you need to do (including any commands/files, as well as which computer(s) changes need to be made on) to allow the two hosts to communicate. [2 marks]

Now that your internet is configured, you run some tests. Assume the delay to send a packet across a single Ethernet link is always 1ms (no matter what the size of the packet is). Delays inside computers (processing, queuing) are so small that they can be ignored. From one host you ping the other host (C4). The output of the ping is shown below. There are three variables: *IPADDRESS*, *TTL* and *RTT*.

```
$ ping IPADDRESS -c 5
PING IPADDRESS(IPADDRESS) 56(84) bytes of data.
64 bytes from IPADDRESS: icmp_seq=1 ttl=TTL time=RTT ms
64 bytes from IPADDRESS: icmp_seq=2 ttl=TTL time=RTT ms
64 bytes from IPADDRESS: icmp_seq=3 ttl=TTL time=RTT ms
64 bytes from IPADDRESS: icmp_seq=4 ttl=TTL time=RTT ms
64 bytes from IPADDRESS: icmp_seq=5 ttl=TTL time=RTT ms

--- IPADDRESS ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4001ms
rtt min/avg/max/mdev = RTT/RTT/RTT/0 ms
```

(e) What is the value of the variable *IPADDRESS*? [1 mark]

(f) What is the value of the variable *TTL*? Explain your answer. [1.5 marks]

(g) What is the value of the variable *RTT*? Explain your answer. [1.5 marks]

Question 3 [11 marks]

Consider the C source code in files `file1.c` and `file2.c` in Listings 1 and 2, respectively. They provide similar (but not the same) functionality as the sockets code used in the lab. Note that some error checking code is removed (but they still compile and execute). Answer the questions based on this code.

Listing 1: Source code for file1.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define MAX_CHAR 1000
#define CONST_A 20
#define CONST_B 50
int main(int argc, char *argv[])
{
        int s, p, n;
        struct sockaddr_in dst_addr;
        struct hostent *dst;
        char d[MAX_CHAR];

        s = socket(AF_INET, SOCK_STREAM, 0);

        dst = gethostbyname(argv[1]);
        p = atoi(argv[2]);

        bzero((char *) &dst_addr, sizeof(dst_addr));
        dst_addr.sin_family = AF_INET;
        bcopy((char *)dst->h_addr,
                (char *)&dst_addr.sin_addr.s_addr,
                dst->h_length);
        dst_addr.sin_port = htons(p);

        connect(s,(struct sockaddr *) &dst_addr,sizeof(dst_addr));

        bzero(d,MAX_CHAR);
        /* fgets also reads the single Enter character '\n' if typed */
        fgets(d,CONST_A,stdin);
        n = write(s,d,strlen(d));

        bzero(d,MAX_CHAR);
        n = read(s,d,CONST_B);

        printf("%s\n",d);
        return 0;
}
```

Listing 2: Source code for file2.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#define MAX_CHAR 1000
#define CONST_C 10
int main(int argc, char *argv[])
{
        int s, news, p, pid, n;
        struct sockaddr_in my_addr, src_addr;
        size_t srcaddrlen;
        char b[MAX_CHAR];

        s = socket(AF_INET, SOCK_STREAM, 0);

        bzero((char *) &my_addr, sizeof(my_addr));
        p = atoi(argv[1]);
        my_addr.sin_family = AF_INET;
        my_addr.sin_addr.s_addr = INADDR_ANY;
        my_addr.sin_port = htons(p);

        bind(s, (struct sockaddr *) &my_addr, sizeof(my_addr));
        listen(s,5);
        srcaddrlen = sizeof(src_addr);

        while (1) {
                news = accept(s, (struct sockaddr *) &src_addr, &srcaddrlen);

                pid = fork();
                if (pid == 0)  {
                        close(s);
                        bzero(b,MAX_CHAR);
                        n = read(news,b,CONST_C);
                        printf("%s\n",b);

                        n = write(news,b,strlen(b));
                        exit(0);
                }
                else  {
                        close(news);
                }
        }
    return 0;
}
```

(a) What command, including arguments, should be used to compile `file1.c` into an executable called `program1`? [1 mark]

Assume files `file1.c` and `file2.c` are compiled to produce `program1` and `program2`, respectively. `program2` is started on computer with IP address `1.1.1.1` and using port `44444`. `program1` will be run on a computer with IP address `2.2.2.2`.

(b) What command, including arguments, is used to run `program1`? [1 mark]

Consider now after `program1` is started, the user of the program types in to the terminal: *steven*

(c) What is printed at the computer running `program2`? [1 mark]

(d) What is printed at the computer running `program1`? [1 mark]

(e) How many bytes of data did `program1` send to `program2`? [1 mark]

Now consider that `program1` is started again, but this time the user types in to the terminal: *stevenismyname*

(f) What is printed at the computer running `program2`? [2 marks]

Now consider that the value of `CONST_B` in `file1.c` is changed from 50 to 5. After recompiling, the user starts `program1` and types into the terminal: *steven*

(g) What is printed at the computer running `program1`? [2 marks]

Assume `program2` always uses port 44444. Therefore you want to modify `program1` so that you do not need to type in the port—instead it is hardcoded into the source. In addition, rather than the user typing text in to the terminal after `program1` starts, you want to change it so that the text is entered as a command line argument. (You may assume the user does not type spaces in the text entered in the terminal).

(h) Referring to the line numbers, explain what you would change to implement the above functionality. (E.g. what code would you add/modify/delete) [2 marks]

Question 4 [10 marks]

Referring to the code in Listings 1 and 2, give the line number that implements the following functionality. Each part is worth 1 mark. (Give only one line number for each part, although there may be multiple correct answers.)

(a) Converts a string into an integer. Line:

(b) Converts a domain name to an IP address. Line:

(c) Blocks until a new TCP connection is established. Line:

(d) Initiates establishment of a TCP connection. Line:

(e) Waits until TCP data is received from another host. Line:

(f) Associates an IP address with a socket. Line:

(g) Returns the number of bytes successfuly sent. Line:

(h) Returns value 0 when a child process is created. Line:

(i) Reads data from standard input. Line:

(j) Obtains the address of the host that connected to the server. Line:

Question 5 [4 marks]

You are running Ubuntu Linux on your home computer. Installed is Apache Web Server, which you use only for testing. That is, you only access the web server on your computer from localhost (`127.0.0.1`). You want to prevent everyone on the Internet from accessing your web server. Of course, you still want to allow your own computer to access web servers and other servers (email, instant messaging, ssh, . . .) on the Internet. You use `iptables` as the firewall on your computer (there are no other firewalls or network address translation in your network). Assume the default firewall policy is ACCEPT.

(a) Give the exact `iptables` command(s) to create the desired firewall. Explain any assumptions. [2.5 marks]

Now assume you want to prevent everyone on the Internet from accessing any server running on your computer (instead of just the web server).

(b) Give the exact `iptables` command(s) to create the desired firewall. Explain any assumptions. [1.5 marks]

Question 6 [8 marks]

Consider an `iptables` firewall running on a router in SIIT Bangkadi. The router has two interfaces, `eth0` and `eth1`. Inside the SIIT Bangkadi network (accessible via `eth0`) are three subnets: students (`1.1.1.0/24`), faculty (`2.2.2.0/24`) and public (`3.3.3.0/24`). Interface `eth1` connects to an Internet Service Provider, and then the rest of the Internet.

Give the exact `iptables` command(s) to implement the desired conditions below. Assume each part is independent of each other and the firewall table has been flushed (all rules deleted) before each part (that is, the rule you add in part (a) is deleted before part (b)). Assume the default firewall policy is ACCEPT.

(a) Prevent all ICMP ping packets from leaving or entering SIIT Bangkadi network. [2 marks]

(b) Prevent all computers on the students subnet from accessing the Facebook web server (which has IP `66.220.149.18`). [3 marks]

(c) Prevent all hosts on the Internet from accessing the SSH server on Steve's computer (`2.2.2.100`). [3 marks]

Reference Material

Below is the syntax of commonly used commands. The values that the user must choose are given enclosed in < and >. Optional fields are enclosed in [and]. You may use this information in your answers.

```
ifconfig [<interface>] [up | down]
ifconfig <interface> <ipaddress> netmask <subnetmask>
ping [-c <count>] [-s <packetsize>] [-i <interval>] <destination>
tracepath <destination>
nslookup <domain> [<dnsserver>]
route [-n]
route add -net <netaddress> netmask <subnet> [gw <gateway>] [dev <interface>]
route del -net <netaddress> netmask <subnet> [gw <gateway>] [dev <interface>]
route add default gw <gateway>
arp [-n]
dhclient [<interface>]
apache2ctl [start | stop | restart]
htpasswd <passwordfile> <username> [-b <password>]
iptables -A <chain> [<options>]
  where <options> include:
   [-s <sourceip>] [-d <destip>] [-i <ininterface>] [-o <outinterface>]
   [-p <protocol>] [--sport <sourceport>] [--dport <destport>]
   [-j <action>]
iptables -D <chain> [<options>]
iptables -L <chain>
iptables -F <chain>
  where <chain> may be: INPUT | OUTPUT | FORWARD
```

Commonly used files and directories are listed below. You may use this information in your answers.

```
/etc/hosts
/etc/resolv.conf
/etc/network/interfaces
/etc/services
/var/lib/dhcp3/dhclient.leases
/proc/sys/net/ipv4/ip_forward
/var/www/
/etc/apache2/sites-available/default
```

Port numbers used by common applications include:

20 FTP data transfer

21 FTP connection control

22 SSH, secure remote login

23 TELNET, (unsecure) remote login

25 SMTP, email transfer between servers

53 DNS, domain name lookups

67 DHCP server

80 HTTP, web servers

110 POP3, client access to email

123 NTP, network time

443 HTTPS, web servers with secure access

520 RIP, routing protocol

631 IPP, Internet printing

Protocol numbers for commonly used transport protocols include:

1 ICMP

2 IGMP

6 TCP

17 UDP

33 DCCP

41 IPv6 encapsulation

47 GRE

89 OSPF