

CSci 414/553: Unix Network Prog & Software Development***Test 2, Spring 2008***

Answer the following questions in the space provided, or as instructed by editing and adding files to your student account. Please write clearly and make clear your final solution to a question if you make changes or corrections (pencils are recommended). This test is open book / open source material, you may use any source materials you like and find helpful for this class. However, this is an individual test, no cell phones, IM or other collaboration will be allowed during the test.

For official grading use only.																							Total: _____ / 100		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
/8	/2	/7	/2	/2	/2	/4	/2	/2	/2	/2	/2	/4+1	/4+1	/4	/2+1	/2	/2	/3	/2	/3+1	/2	/4	/10	/10+2	/10

1. (8 pts) The following is a list of attributes of the UDP and TCP protocols. Place each attribute with the protocol that most appropriately displays the attribute:
- reliable
 - slow
 - connectionless
 - unreliable
 - fast
 - connection oriented
 - streaming
 - packet oriented

UDP: **c (connectionless) d (unreliable), e (fast) h (packet oriented)** _____

TCP: **a (reliable) b (slow) f (connection oriented) g (streaming)** _____

2. (2 pts) One common problem of designing and developing a system with multiple parts that act concurrently (e.g. several things happening at once) is that deadlocks may occur. A deadlock is a situation where:
- the final result depends on whether A or B goes last
 - A gets selected to run more often, not allowing B to run
 - A is Big-Endian while B is Little-Endian
 - A is waiting for B while B is waiting for A

Answer: **D** _____

3. (7 pts) Match the following tools/commands with the description of their purpose (you can draw lines and/or match the number with the letter).

- | | | | |
|---------------|-------------|--------------------------------------------------------------------------|----------|
| D | 1) tcpdump | a) report a snapshot of the current processes and their status | 3 |
| | 2) ping | b) send a signal to a process | 7 |
| A or G | 3) ps | c) configure a network interface | 5 |
| E | 4) netstat | d) dump traffic on a network | 1 |
| C | 5) ifconfig | e) print network connections, routing tables, interface statistics, etc. | 4 |
| F | 6) nslookup | f) query internet name servers | 6 |
| B | 7) kill | g) report a snapshot of the current processes | 3 |

4. (2 pts) Another big problem of designing and developing a system with multiple parts that act concurrently is that a race condition may occur. A race condition is a situation where:

- a) A is waiting for B while B is waiting for A
- b) the final result depends on whether A or B goes last
- c) A is Big-Endian while B is Little-Endian
- d) A gets selected to run more often, not allowing B to run

Answer: **B**

5. (2 pts) Another difference of building a distributed systems is the need to deal with partial failures. Which of the following are types of partial failures that need to be dealt with in a distributed system:

- a) A message or packet is lost in transmission
- b) A node running a server goes down
- c) A receive buffer becomes full
- d) All of the above

Answer: **D**

6. (2 pts) A client/server architecture is one in which many clients communicate with a central server while a peer-to-peer architecture is one in which all processes exchange information equally. Which of the following examples of distributed services is NOT implemented as a client/server architecture:

- a) Mail protocols such as SMTP and POP
- b) Bit Torrent
- c) File Transfer Protocol (FTP)
- d) Relational database distributed access systems like Oracle and MySQL

Answer: **B**

7. (4 pts) The Hypertext Transfer Protocol (HTTP) specifies how programs exchange documents over the web, and is designed to be implemented using a client/server architecture. Mark each of the following applications as either a client or a server
- a) Firefox: ___ **Client** _____
 - b) Apache: ___ **Server** _____
 - c) Internet Explorer: ___ **Client** _____
 - d) Internet Information Services (IIS): ___ **Server** _____
8. (2 pts) The CGI protocol does *not* specify a particular programming language that must be used to write CGI applications:
True False
9. (2 pts) The Web Server passes information to a CGI process using:
- a) Environment Variables
 - b) Standard Input
 - c) Procedure Call Invocations
 - d) a and b**
 - e) All of the above
10. (2 pts) Which of the following mechanisms does the TCP protocol use in order to implement reliable service:
- a) Acknowledgments
 - b) Retransmissions
 - c) Flow Control
 - d) Timeouts
 - e) a and b
 - f) All of the above**
11. (2 pts) When writing unit tests, you should only write test cases that you expect to succeed
True **False**
12. (4 pts) What are the 4 typical steps that you would do to build and install a project from source on a Linux/Unix system that was developed using the make automated build tool:
- 1) **./configure**
 - 2) **make**
 - 3) **make test**
 - 4) **make install**

13. (4 pts) Fill in the following blanks
- An IPv4 address consists of a total of how many bits: 32
 - The standard dotted IPv4 address notation consists of how many parts: 4
 - Each part of the dotted IPv4 address consists of how many bits (hint a,b & c should be consistent with one another): 8
 - An IPv6 address consists of a total of how many bits: 128
 - (Extra credit 1pt) I misled people in class, but a hexadecimal number in the range from 0000-FFFF actually represents a 16 bit number (each hexadecimal digit represents 4 bits). This implies that a full IPv6 address must consist of how many :: separated hexadecimal groups: 8
14. (4 pts) List two of the three main mechanisms we studied in this class for implementing a concurrent server (1 extra credit points for getting all 3 correct).
- threads**
 - fork child processes**
 - I/O multiplexing using select**
15. (4 pts) Match the protocol with the socket type constant that is used in the socket() function call to create a socket using the desired protocol. Draw a line from the socket type constant on the left to the corresponding protocol on the right:
- | <u>Socket Type</u> | <u>Protocol</u> |
|-------------------------------|-----------------|
| SOCK_STREAM -> TCP | IP |
| SOCK_SEQPACKET -> SCTP | UDP |
| SOCK_RAW -> IP | SCTP |
| SOCK_DGRAM -> UDP | TCP |
16. (2 pts) What is the IPv4 address associated with the loopback network interface? 127.0.0.1
- (Extra Credit 1pt) What is the IPv6 loopback address: ::1
17. (2 pts) When you open up a socket connection to the loopback interface, what will be the destination of your connection? the local machine / localhost / etc.
18. (2 pts) Which end of a TCP connection goes into the TIME_WAIT state (circle one):
- The end performing the active close**
 - The end performing the passive close
19. (3 pts) Port numbers are divided into three ranges by the IANA. Briefly what are the names and purpose of these 3 ranges:
- 0 – 1023: well-known services
 - 1024-49151: IANA registered ports
 - 49152-65535: dynamic or private ports

20. The SIGCHLD signal is sent by the kernel when a child process writes data to its standard output for the parent to read:

True **False**

21. (3 pts) When we execute the following command:

```
[dharter@corvuslap ~]$ ps uwx
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
dharter   2964  0.0  0.0    2296  1100 ?        S    09:31   0:00 /var/tmp/lm_TMW.ld -z -c /var/tmp/lm_TMW.dat
dharter   3507  0.0  0.0    4596   308 ?        Sl   09:32   0:00 /usr/bin/gpg-agent -s --daemon
dharter   3526  0.0  0.0    1584   132 ?        S    09:32   0:00 start_kdeinit --new-startup +kcm_init_startup
dharter   3527  0.0  0.1   28640  3772 ?        Ss   09:32   0:00 kdeinit Running...
dharter   3530  0.0  0.1   17604  3196 ?        S    09:32   0:01 dcopserver --nosid
dharter   3532  0.0  0.3   30384  6172 ?        S    09:32   0:00 klauncher [kdeinit] --new-startup
dharter  10114  0.0  0.8   35608 16508 ?        R    12:03   0:01 konsole
dharter  10117  0.0  0.0    4824   1560 pts/3    Ss   12:03   0:00 /bin/bash
dharter  10978  4.0  0.0    4524    948 pts/3    Z    12:38   0:00 ./socketserver.py
dharter  10979  4.0  0.0    4524    948 pts/3    R+   12:38   0:00 ps uwx
```

S status means that the process is sleep

R status means that the process is running

Z status means that the process is zombie

(1 Extra Credit) what do Ss and Sl indicate? How about R+? **s: session lead l: multithread +: foreground**

22. (2 pts) A client/server architecture is one in which many clients communicate with a central server while a peer-to-peer architecture is one in which all processes exchange information equally. which of the following examples of distributed services is NOT implemented as a client/server architecture:

- a) Mail protocols such as SMTP and POP
- b) Bit Torrent
- c) File Transfer Protocol (FTP)
- d) Relational database distributed access systems like Oracle and MySQL

Answer: **B**

23. (4 pts) A socket pair for a TCP connection contains 4 parts or a 4-tuple as we have referred to it in class (as seen when using the netstat command). What are the 4 parts designating a TCP socket connection:

- 1. server ip
- 2. server port
- 3. client ip
- 4. client port

24. (10 pts) The following are a list of function names (in the C socket api) that are used to creating clients and servers to communicate with each other using sockets:

accept()
 bind()
 close()
 connect()
 listen()
 read()
 socket()
 write()

Write the typical sequence of elementary socket functions, in the correct order, that would be called to:

(Show the full set of steps that would be needed using the C language socket interface, some language's socket libraries, like Java, combine some of the above basic operations into single operations.)

A) create a TCP client:

B) create a TCP server:
TCP Server

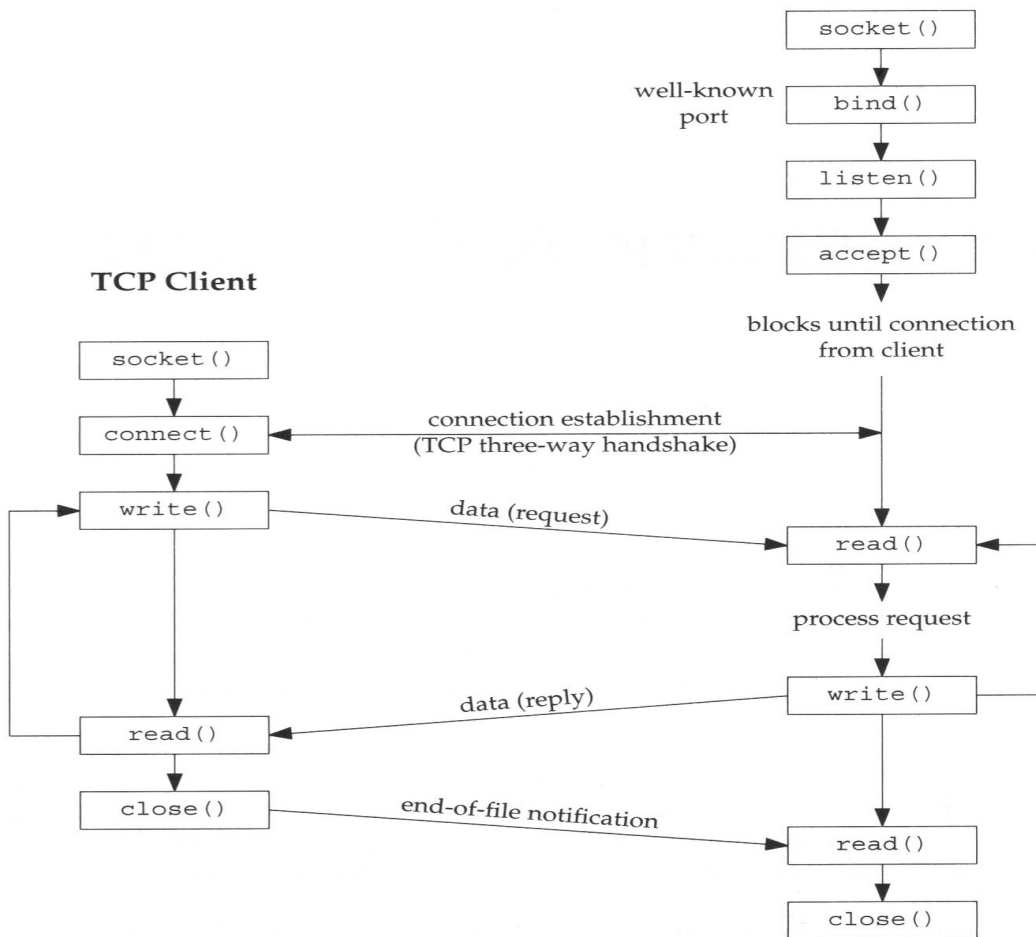


Figure 4.1 Socket functions for elementary TCP client/server.

25. (10 pts) There is currently a tcp client running from the host corvus.tamu-commerce.edu that is interacting with a server on the nisl server.
- a) Create a tcpdump filter (tcpdump command is located at /usr/sbin/tcpdump) to determine the port number of the server that the client on corvus is communicating with. What is the port number of the server?

Server port number on nisl: 56187

- b) (Extra Credit 2 pts) Further, what message is the client sending the server? What message, if any, is the server sending back to the client?

Client: Secret Phrase

Server: Linux is Fun

- c) There is a flag that can be given to the netstat command in order to display the process id and the name of the program that has opened a given network connection. What is this flag? Use this flag to determine the process id (pid) and name of the command that is running the tcp server on nisl.

-p pid of server: 1225 name python ./SuperSecretTestServer.py

- d) Use the ps command to determine who is the user who owns the running server in question. Which user is running the indicated server?

user: ssuh more full name: python ./SuperSecretTestServer.py

- e) If you were the root/administrator of nisl, and you wanted to stop this running server, what command would you use to terminate the running process?

kill -9 1225

26. (10 pts) There is another server also running on the nisl server currently, not so secret this time, that can be found at port 54321. The server is a TCP server. Write a client (you may use whichever language you feel most comfortable with) to communicate with this server. You should create a new directory in your repository named test2. This program should be named TestClient.x (using the appropriate extension for the language you are using). Make sure you add and commit your new directory and the TextClient.x program to your repository when you are done.

The server implements the following (pseudo) protocol:

- When the server receives a valid student user name, it will send back a keyword. Have your client fetch this keyword from the server and display it to standard output, along with the username you sent. For example:

```
[harry@nisl test2]$ ./TestClient.py  
harry: TheHarryKey
```
- The keyword is unique to each student. Upon receiving your keyword, you can use it to acquire your secret phrase from the server (again each phrase is unique to each username). Send your keyword as your second message from your client, and display the secret phrase you receive back from the server to standard output, for example:

Secret Phrase: "Harry's Secret Phrase!"

What was your secret phrase? _____ **Answer is unique to each student, example answer next page_**

```

#!/usr/bin/env python
import sys, socket

buffer_size = 1024      # bytes
host = 'nisl.tamu-commerce.edu'      # local machine (loopback address)
port = 54321            # hope nobody else is using it...

# AF_INET means 'Internet socket'.
# SOCK_STREAM means 'TCP'.
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
sock.connect((host, port))

# Send the message.
user = "harry"
sock.send(user)

# Receive and display the reply.
keyword = sock.recv(buffer_size)
print '%s: %s' % (user, keyword)

# Send the message.
sock.send(keyword)

# Receive and display the reply.
phrase = sock.recv(buffer_size)
print 'Secret Phrase: <%s>' % phrase

# Tidy up.
sock.close()

```