Programme Code: TU856/TU858

Module Code: CMPU3006

TECHNOLOGICAL UNIVERSITY DUBLIN

Grangegorman

_____

TU856 BSc. (Honours) Degree in Computer Science

TU858 BSc. (Honours) Degree in Computer Science (International)

Year 3

_____

SEMESTER 1 EXAMINATIONS 2023/24

_____

CMPU3006, Client Server Programming

Internal Examiner: Mr. Damian Bourke

Dr. Paul Doyle

External Examiner: Ms. Sanita Tifentale

Attempt Question 1 and three of the four remaining questions. All questions carry equal marks.

Duration: 2 hours.

1. Figure 1 (A snippet of code) reveals some code without error checking. Referring to this figure and to the following list of terms:

0 (the number zero), servAddr, inet_ntop, MAXPENDING, sockaddr, clntAddr, sin_port, sock, htons, inet_pton, AF_INET, atoi, sin_port, clntSock, sockaddr_in.

```
24  .............IGNORE ALL PREVIOUS LINES OF CODE...............
25
26      int XXXXX = socket(XXXXX, SOCK_STREAM, IPPROTO_TCP);
27
28      struct sockaddr_in servAddr;
29
30      memset(&servAddr, 0, sizeof(servAddr));
31
32      XXXXX.sin_family = AF_INET;
33
34      int rtnVal = XXXXX(AF_INET, servIP, &servAddr.sin_addr.s_addr);
35
36      servAddr.XXXXX = htons(servPort);
37
38      if (connect(sock, (struct XXXXX *) &servAddr, sizeof(servAddr)) < 0)
39          DieWithSystemMessage("connect() failed");
40
41  .............IGNORE ALL SUBSEQUENT LINES OF CODE...............
```

Figure 1: A snippet of code.

(a) Using line numbers, identify which XXXXX value can be replaced with which term from the above list.

(12 marks)

(b) State whether this is a client or server application justifying your answer.

(4 marks)

(c) Explain the purpose of the *struct* identified in line 28 and identify the required byte order of data stored in key members of this *struct*.

(9 marks)

2. The following string representing a correctly formatted HTTP/1.1 Request message for the default homepage is sent to the TU Dublin webserver:

GET / HTTP/1.1\r\nHost: www.tudublin.ie\r\n\r\n`

Question continues on next page.

(a) Explain the concept of *multi-homing* explaining the significance of the *Host:* header in this regard and, explain how a HTTP/1.1 server should respond if this header is omitted.

(7 marks)

(b) HTTP/1.0 and HTTP/1.1 differ in how they deal with connections. Use an example of downloading a HTML file containing five image tags to highlight the differences between the two protocols. In your answer, explain the concept of *persistent connections* stating which protocol supports this type of connection.

(11 marks)

(c) Identify the HTTP header and value that over-rules the default persistent connection state and, explain how a server should respond if this header is included in a HTTP Request.

(7 marks)

3. Consider the following host address configurations and referring to Figure 2 - Output from *netstat*.

Host A: Port 36408, IP address: 192.168.1.100 and MAC address: 08:00:27:15:9f:33

Host B: Port 25, IP address: 192.168.1.200 and MAC address: 08:00:27:9e:97:73

Figure 2 shows output from running the *netstat* command separately on each of the hosts, A and B.

Line	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
1	TCP	0	0	0.0.0.0:**A**	0.0.0.0:**B**	LISTEN
2	TCP	0	0	**C**	**D**	ESTABLISHED

Line	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
3	TCP	0	0	**E**	**F**	ESTABLISHED

Figure 2 – Output from *netstat.*

Question continues on next page.

(a) There are values missing from this output listing which are identifiable by the letters A through to F inclusively. Assuming the application on Host A has requested a connection to the application on Host B, use the appropriate addressing information provided for each host to replace each letter.

(12 marks)

(b) Identify which lines, 1, 2 and 3, relate to the client TCP entity and which relate to the server TCP entity. Justify your answer.

(6 marks)

(c) If a non-zero value appeared in column labelled Recv-Q on either lines 2 or 3, explain what this value would represent and, explain what would have to happen within either application for this value to return to zero. In your answer, identify any socket primitives called within the application.

(3 marks)

(d) Identify the name of the ranges from which the *client* and *server* port numbers are derived.

(4 marks)

4. Refer to Figure 3 – Using *fork()*.

```
 1. for ( ; ; ) {
 2.
 3. connfd = accept(listenfd, (SA *) NULL, NULL);
 4.
 5. if ( (childpid = fork()) == 0)
 6.       {
 7.
 8.       close(listenfd);
 9.
10.       OMITTED LINES OF CODE RELATING TO DATA EXCHANGE WITH CLIENT APPLICATION
11.
12.       close(connfd);
13.
14.       exit(0);
15.
16.       }
17.
18. close(connfd);
19.
20. }
21.
```

Figure 3 – Using *fork()*

Question continues on next page.

(a) Describe the purpose of the *fork()* function.

(5 marks)

(b) The following statement is often associated with the fork() function: "*It is called once but it returns twice*".

Explain what this means with reference to *child* and *parent* processes and identify which lines of code relate to the *child* process.

(8 marks)

(c) Socket descriptors are shared between *child* and *parent* processes. Explain what this means. In your answer explain why closing the listening socket on line 8 does not impact on the ability to accept new connections as per the call to *accept( )* on line 3.

(12 marks)

5. Refer to Figure 4 - Screenshot from Wireshark. This screenshot shows a sequence of segment exchanges between a http client and http server host machines. In particular, lines 1 to 3 inclusive show the sequence of segments used to **open** a TCP connection between a HTTP client (port 52903) and a HTTP server (port 80).

```
1 52903→80 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1440 W
2 80→52903 [SYN, ACK, ECN] Seq=0 Ack=1 Win=28560 Len=0 MSS=
3 52903→80 [ACK] Seq=1 Ack=1 Win=131360 Len=0 TSval=5613506
4 GET / HTTP/1.1
```

Figure 4 – Screenshot from Wireshark

(a) Identify the TCP messages used to open a connection and identify which line numbers in figure 4 relate to each of messages exchanged. In your answer, identify which application has initiated the opening sequence, *client* or server, justifying your answer with reference to the direction of the segment transmissions.

(8 marks)

(b) Explain the purpose of the values contained in the following fields: *Seq*, *Ack* and *Win*.

(6 marks)

(c) Do ACKs consume a sequence number? In consideration of this question, identify the value in the Seq field contained in the segment carrying the HTTP Request identified on line 4. Justify your answer with reference to previous segment exchanges.

(6 marks)

(d) If the HTTP Request on line 4 contains 100 bytes, identify the ACK value stored in a returning acknowledgment segment. Justify your answer.

(5 marks)