DUBLIN INSTITUTE OF TECHNOLOGY

DT228 BSc. (Honours) Degree in Computer Science

Year 3

DT282 BSc. (Honours) Degree in Computer Science (International)

Year 3

WINTER EXAMINATIONS 2017/2018

CLIENT SERVER PROGRAMMING [CMPU3006]

MR. DAMIAN BOURKE

DR. DEIRDRE LILLIS

DR. MARTIN CRANE – DT228/DT282

THURSDAY 11TH JANUARY 9.30 A.M. – 11.30 A.M.

INSTRUCTIONS TO CANDIDATES

Attempt all questions. NOT all questions carry the same mark.

1. Consider the TalkTalk 100-102FM Radio Station. The producers want to run daily competitions across a number of their shows. Listeners will be able to submit *Answers* via a Mobile App that automatically connects to a server. The listeners are told that the *Answers* have to follow a particular format as follows: "**The Answer**" followed by "**The Programme**" followed by their "**Name**". For example, the presenter on "The Morning Show" might ask, "Who is the current President of America? Please send **Your Answer** followed by the **The Morning Show** followed by **Your Name**".

   You are asked to design the server application that receives and stores the *Answers* and, sends a response back to the listener's Mobile App indicating that their *Answer* to the show in question has been received. For example if a listener responded to the above question they would receive an *Acknowledgement* message as follows: "Your answer to The Morning Show quiz was successfully received".

   You are required to:

   (i) Develop a Protocol (using a Box diagram) for incoming *Answers* that would enable the server to filter *Answers* to quizzes from <u>different</u> shows. The diagram should identify how the fields are separated and how the *Answer* is terminated. (10 marks)

   (ii) Develop a Protocol (using a Box diagram) for outgoing *Acknowledgements* to the listener Mobile App. The diagram should identify how the *Acknowledgement* is terminated. (5 marks)

   (iii) Outline the **<u>high-level</u>** steps required (no arguments required) for the server application to perform each of the following functions from the point where the <u>connected</u> socket is returned:

   - Correctly receive incoming *Answers*. Describe how this might be achieved based on the Protocol you developed in (i) above and explain the role of the associated termination string. In your answer, identify the Primitive to be used and any built-in C functions that might be useful,
   - Filter and Store *Answers* for each show in separate files. Describe how this might be achieved and identify any built-in C functions that might be useful,
   - Return a standard *Acknowledgement* message (identifying the name of The Show) to the listener's Mobile App. Describe how this might be achieved based on the Protocol you developed in (ii) above and identify the Primitive to be used. (15 marks)

2. Refer to Figure 1 – *A snippet of code* and the following list of terms:

len, AF_INET, SOCK_STREAM, servaddr, INADDR_ANY, htonl, MAXLINE, htons, listenfd, LISTENQ, connfd, Inet_ntop, ntohs, sin_port;

You are required to replace the values XXXX with the correct term from the above list. Use the line numbers to identify where your corrections are made. Note, in relation to lines 23 and 25 there are multiple terms missing; identify the line number followed by the correct terms in sequence.

(20 marks)

```
1        socklen_t                  len;
2        int                        n, listenfd, connfd;
3        struct sockaddr_in         servaddr, cliaddr;
4        char                       wbuff[MAXLINE+1], rbuff[MAXLINE+1];
5
6
7        if (argc != 2)
8              err_quit("usage: a.out <Port>");
9
10       listenfd = Socket(AF_INET, XXXX, 0);
11
12       bzero(&servaddr, sizeof(servaddr));
13       servaddr.sin_family      = XXXX;
14       servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
15       servaddr.sin_port = XXXX(atoi(argv[1]));
16
17       Bind(listenfd, (SA *) &servaddr, sizeof(servaddr));
18       Listen(listenfd, XXXX);
19
20       for ( ; ; )
21       {
22         len = sizeof(cliaddr);
23         XXXX = Accept(listenfd, (SA *) &XXXX, &XXXX);
24         printf("\nConnection from %s, port %d\n", XXXX(AF_INET,
25   &cliaddr.sin_addr, buff, sizeof(buff)), XXXX(cliaddr.XXXX));
```

Figure 1 – *A snippet of code.*

3. Two applications, a client and server, are communicating with each other on the same host machine. The *netstat* command is executed twice on two separate occasions and produces the following outputs (labelled *Figure 2(i)* and *Figure 2(ii)*):

	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
1.						
2.						
3.	tcp	0	0	147.252.234.34:12345	0.0.0.0:*	LISTEN
4.	tcp	0	0	147.252.234.34:12345	147.252.234.34:44901	ESTABLISHED
5.	tcp	26	0	147.252.234.34:44901	147.252.234.34:12345	ESTABLISHED

Figure 2(i)

	Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
1.						
2.						
3.	tcp	0	0	147.252.234.34:12345	0.0.0.0:*	LISTEN
4.	tcp	0	0	147.252.234.34:12345	147.252.234.34:44901	TIME_WAIT

Figure 2(ii)

(i) In relation to *Figure 2(i)*, line 5. You are required to:

(a) Identify which application this line refers to, client or server. Justify your answer with reference to the data displayed in the figure. (3 marks)

(b) Explain what the number in the Recv-Q column refers to and identify what Primitive must be invoked in the associated application for this number to reduce to zero. (3 marks)

(ii) In relation to *Figure 2(ii)*, line 4. You are required to:

(a) Explain the current status of the connection (with reference to the State column) describing the purpose of this State. (4 marks)

(b) Identify which of the three *Phases of Communications* this State relates to. (2 marks)

(c) Identify which application (client or server) intiated the sequence of events that caused the connection to arrive at this state and what primitive was invoked. (3 marks)

4. Refer to Figure 3 – *Using Fork.*

```
16
17  for ( ; ; ) {
18
19              connfd = Accept(listenfd, (SA *) NULL, NULL);
20
21              if ( (childpid = Fork()) == 0)
22              {
23              Close(listenfd);
24
25              DATA EXCHANGE WITH CLIENT APPLICATION
26
27              Close(connfd);
28
29              exit(0);
30
31              }
32
33              Close(connfd);
34
35          }
36
37
```

Figure 3 – *Using Fork.*

(i) Describe the purpose and basic operation of the *Fork* function. In your answer explain the relationship between the *Child process* and *Parent process* and identify which lines of code refer to each process. (5 marks)

(ii) The *listening* socket is closed on line 23. Explain why it is closed here and explain whether this affects future connection requests. (5 marks)

(iii) Given that the connected socket is most likely closed within the Parent process before the Child process has exchanged any data with the Client application, explain how this is possible. (5 marks)

5. Consider the internal data buffers within a TCP entity supporting two Client applications:

   (i) Identify the internal data buffers used within a TCP entity and the total number of *data* buffers the TCP entity must manage internally for both applications. (2.5 marks)

   (ii) At what point within an application does the TCP entity create the internal data buffers? (2.5 marks)

   (iii) Is it possible for data to arrive for an application when one of the data buffers is full? If not, why not? (5 marks)

6. In relation to the *Daytime* application:

   (i) Outline the purpose of this application and describe the Semantics associated with the communications between the client and server applications. In your answer identify the correct sequence of socket primitive calls (without arguments) invoked by each application. (5 marks)

   (ii) What addressing information does the **TCP entity** on the client host require in order to make contact with the server application and from where does it obtain this information? (2.5 marks)

   (iii) What addressing information does the **TCP entity** on the server host need to return data to the client application and from where does it obtain this information? (2.5 marks)

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF CMPU 3006 VENUE: Courtyard, Aungier St

SUBJECT: Client-server Programming

DATE: 11/01/18

TIME:

SIGNED:

INSTRUCTIONS:

Page 3.

Q2 line 2. The term "sesvaddr" should read "chiaddr".

Damian R

COLLEGE EXAMINATIONS

AMENDMENTS TO EXAMINATION QUESTION PAPER

COURSE REF CMPU300 **| VENUE:** Room 1054 / 1002

SUBJECT: ADVANCED COMPUTER NETWORKS

DATE: 11/1/18

TIME:

SIGNED:

INSTRUCTIONS: PAGE 3

Q2 Line 2. The term "servaddr" should read "cliaddr"