Second Semester Examination
2016/2017 Academic Session

June 2017

CST234 – Network Programming
[Pengaturcaraan Rangkaian]

Duration : 2 hours
[Masa : 2 jam]

INSTRUCTIONS TO CANDIDATE:
[ARAHAN KEPADA CALON:]

- Please ensure that this examination paper contains **FOUR** questions in **SEVEN** printed pages before you begin the examination.

  *[Sila pastikan bahawa kertas peperiksaan ini mengandungi **EMPAT** soalan di dalam **TUJUH** muka surat yang bercetak sebelum anda memulakan peperiksaan ini.]*

- Answer **ALL** questions.

  *[Jawab **SEMUA** soalan.]*

- You may answer the questions either in English or in Bahasa Malaysia.

  [Anda dibenarkan menjawab soalan sama ada dalam bahasa Inggeris atau bahasa Malaysia.]

- In the event of any discrepancies, the English version shall be used.

  [Sekiranya terdapat sebarang percanggahan pada soalan peperiksaan, versi bahasa Inggeris hendaklah diguna pakai.]

1. (a) Figure 1 (on page 7) shows two different echo clients – A and B. The source code given does not contain any errors and is working fine.

Rajah 1 (pada muka surat 7) menunjukkan dua pelanggan gema berbeza – A dan B. Kod sumber yang diberikan tidak mengandungi sebarang ralat dan beroperasi dengan baik.

(i) Client B uses `Connect()` function (line 18), but A does not. Why the difference?

Pelanggan B menggunakan fungsi `Connect()` *(baris 18), tetapi A tidak. Mengapa berbeza?*

(ii) Client A uses `Sendto()` function (line 33), but B uses `Writen()` (line 33), Why the difference?

Pelanggan A menggunakan fungsi `Sendto()` *(baris 33), tetapi B menggunakan* `Writen()` *(baris 33). Mengapa berbeza?*

(iii) What will happen when client A tries to send data to a UDP echo server which is running its echo service on a different/unknown port (i.e. not 2345)?

Apa akan berlaku apabila pelanggan A cuba menghantar maklumat kepada pelayan gema UDP yang melaksanakan perkhidmatan gemanya pada port yang berbeza/tidak diketahui (iaitu bukan 2345)?

(iv) What will happen when client B tries to send data to a TCP echo server which is running its echo service on a different/unknown port (i.e. not 2345)?

Apa akan berlaku apabila pelanggan B cuba menghantar maklumat kepada pelayan gema TCP yang melaksanakan perkhidmatan gemanya pada port yang berbeza/tidak diketahui (iaitu bukan 2345)?

(20/100)

(b) Can an IPv4 client communicate with an IPv6 server using sockets? If yes, explain how it can be done. If no, explain why it is not possible.

Bolehkah suatu pelanggan IPv4 berkomunikasi dengan suatu pelayan IPv6 menggunakan soket? Jika ya, terangkan bagaimana ia boleh dilakukan. Jika tidak, terangkan mengapa ia mustahil.

(5/100)

2. (a) Briefly compare the following.

Banding secara ringkas yang berikut.

(i) `inet_aton()` function and `inet_pton()` function

fungsi `inet_aton()` dan fungsi `inet_pton()`

(ii) `close()` function and `shutdown()` function

fungsi `close()` dan fungsi `shutdown()`

(iii) UDP socket and TCP socket

soket UDP dan soket TCP

(iv) IPv4 server and IPv6 dual-stack server

pelayan IPv4 dan pelayan IPv6 dwi-timbunan

(v) DNS' A record and DNS' AAAA record

rekod A DNS dan rekod AAAA DNS

(10/100)

(b) One of the functions used in socket programming is `accept()`.

Salah satu fungsi yang digunakan dalam pengaturcaraan rangkaian ialah `accept()`.

(i) Who uses `accept()` – client or server program?

Siapa gunakan `accept()` – atur cara pelanggan atau pelayan?

(ii) What is the task performed by `accept()`?

Apakah tugas yang dilaksanakan oleh `accept()`?

(iii) What will happen if `accept()` is not used?

Apa akan berlaku jika `accept()` tidak digunakan?

(10/100)

(c) Explain the role played by Domain Name System (DNS) in a client-server application.

Terangkan peranan yang dimainkan oleh Sistem Nama Domain (DNS) dalam aplikasi pelanggan-pelayan.

(5/100)

3. (a) Assume a server is playing a number-guessing game with multiple clients simultaneously, as following.

Anggap suatu pelayan sedang bermain permainan teka-nombor dengan berbilang pelanggan secara serentak seperti berkut.

Server *Pelayan*	Client(s) *Pelanggan(-pelanggan)*
• Select a random number (different for each client). *Pilih suatu nombor rawak (berbeza untuk setiap pelanggan).* • Receive guess from client and check for correct answer. *Menerima tekaan dari pelanggan dan memeriksa jawapan betul.* • If the guess is correct, end the game for that client and store the number of tries it has taken. *Jika tekaan adalah betul, tamatkan permainan untuk pelanggan itu dan simpan bilangan percubaan yang diambilnya.* • If guess is wrong, give hint to help client make correct guess (i.e. too high, too low, almost there, a little more, a little less, etc.). *Jika tekaan adalah salah, berikan petunjuk untuk membantu pelanggan meneka dengan betul (contoh: terlalu tinggi, terlalu rendah, hampir-hampir, lebih sedikit, kurang sedikit, dan sebagainya).* • Once all clients completed their games, compare performance of clients (i.e. which client guessed correctly in least number of tries) and announce the winner to all clients. *Apabila semua pelanggan telah menamatkan permainan, banding pencapaian pelanggan (contoh: pelanggan mana yang meneka dengan tepat menggunakan bilangan percubaan yang terendah) dan mengumumkan pemenang kepada semua pelanggan.*	• Send guess to server. *Hantar tekaan kepada pelayan.* • If wrong, use the hints given by server to try again. *Jika salah, gunakan petunjuk yang diberikan oleh pelayan untuk cuba sekali lagi.* • If correct, wait for announcement of winner by server. *Jika betul, tunggu pengumuman pemenang oleh pelayan.*

(i) In your opinion, should this server use *fork()* or *select()* function to manage its clients?

Dalam pandangan anda, pelayan ini patut gunakan fungsi fork() atau select() untuk mengurus pelanggan-pelanggannya?

(ii) Does this client need to use *select()* function?

Perlukah pelanggan ini menggunakan fungsi select()?

Explain each answer.

Terangkan setiap jawapan.

(10/100)

(b) For each of the following error messages, indicate what causes it and who (client, server or both) receives it.

Untuk setiap mesej ralat berikut, nyatakan apa yang menyebabkannya dan siapa (pelanggan, pelayan atau kedua-duanya) yang menerimanya.

(i) ECONNRESET

(ii) ECONNREFUSED

(iii) ETIMEOUT

(10/100)

(c) Following is a list of common socket options:

Berikut ialah senarai opsyen soket yang sering digunakan:

```
SO_BROADCAST, SO_DONTROUTE, SO_KEEPALIVE, SO_LINGER, SO_RCVBUF,
SO_RCVLOWAT, SO_SNDLOWAT, SO_REUSEADDR, IP_TOS, IP_TTL,
IPV6_V6ONLY, TCP_MAXSEG, TCP_NODELAY.
```

(i) Which socket option ensures the connection between client and server remains open even when no data has been exchanged between them for some time?

Opsyen soket manakah yang memastikan sambungan di antara pelanggan dan pelayan tetap terbuka sungguhpun tiada data yang mengalir di antara mereka untuk suatu tempoh masa?

(ii) Which socket option determines how far a data packet can travel?

Opsyen soket manakah yang menentukan berapa jauh suatu bingkisan data boleh bergerak?

(iii) Which socket option manages TCP's flow control?

Opsyen soket manakah yang mengurus kawalan aliran TCP?

(iv) Which socket option is normally used with `close()` function?

Opsyen soket manakah yang biasa digunakan bersama fungsi `close()`?

(v) Which socket option ensures enough free space is available before data is sent out?

Opsyen soket manakah yang memastikan ruang kosong mencukupi sebelum data dihantar?

(5/100)

4. (a) A client-server application may communicate in various modes, such as unicast, anycast, multicast and broadcast.

Suatu aplikasi pelanggan-pelayan mungkin berkomunikasi menggunakan pelbagai mod, seperti "unicast", "anycast", "multicast" dan "broadcast".

(i) Briefly describe each of these modes, showing how they differ from one another.

Terangkan secara ringkas setiap mod ini, menunjukkan bagaimana mereka berbeza di antara satu sama lain.

(ii) Broadcasting is not commonly used in client-server applications. In your opinion, what could be the reason?

"Broadcasting" tidak biasa digunakan dalam aplikasi pelanggan-pelayan. Dalam pandangan anda, apakah alasan yang mungkin?

(10/100)

(b) Threads are widely used in client and server applications.

Bebenang digunakan secara meluasnya dalam aplikasi-aplikasi pelanggan dan pelayan.

(i) Describe **two** (**2**) advantages of using threads.

*Terangkan **dua** (**2**) kebaikan menggunakan bebenang.*

(ii) Describe **two** (**2**) disadvantages of using threads.

*Terangkan **dua** (**2**) keburukan menggunakan bebenang.*

(iii) Which part of Echo Client B in Figure 1 can be threaded? Justify your answer.

Bahagian mana Pelanggan Gema B dalam Rajah 1 boleh dijadikan bebenang? Berikan alasan untuk jawapan anda.

(15/100)

Echo Client A
Pelanggan Gema A

```
1  #include  "unp.h"
2
3  int main (int argc, char **argv)
4  {
5     int                  sockfd;
6     struct sockaddr_in   servaddr;
7
8     if (argc != 2)
9        err_quit("usage: tcpcli <IPaddress>");
10
11    sockfd = Socket(AF_INET, SOCK_DGRAM, 0);
12
13    bzero(&servaddr, sizeof(servaddr));
14    servaddr.sin_family = AF_INET;
15    servaddr.sin_port   = htons(2345);
16    Inet_pton(AF_INET, argv[1], &servaddr.sin_addr);
17
18    str_cli_A(stdin, sockfd, (SA *) &servaddr, sizeof(servaddr));
19
20    exit(0);
21 }
22
23
24
25 void str_cli_A (FILE *fp, int sockfd, const SA *pservaddr, socklen_t
26 servlen)
27 {
28    int  n;
29    char  sendline[MAXLINE], recvline[MAXLINE];
30
31    while (Fgets(sendline, MAXLINE, fp) != NULL) {
32
33       Sendto(sockfd, sendline, strlen(sendline), 0, pservaddr, servlen);
34
35       n = Recvfrom(sockfd, recvline MAXLINE, 0 ,NULL, NULL);
36       if (n==0)
37          err_quit("str_cli: server terminated prematurely");
38
39       recvline[n] = 0;
40       Fputs(recvline, stdout);
41    }
42 }
```

Echo Client B
Pelanggan Gema B

```
1  #include  "unp.h"
2
3  int main (int argc, char **argv)
4  {
5     int                  sockfd;
6     struct sockaddr_in   servaddr;
7
8     if (argc != 2)
9        err_quit("usage: tcpcli <IPaddress>");
10
11    sockfd = Socket(AF_INET, SOCK_STREAM, 0);
12
13    bzero(&servaddr, sizeof(servaddr));
14    servaddr.sin_family = AF_INET;
15    servaddr.sin_port   = htons(2345);
16    Inet_pton(AF_INET, argv[1], &servaddr.sin_addr);
17
18    Connect(sockfd, (SA *) &servaddr, sizeof(servaddr));
19
20    str_cli_B(stdin, sockfd);
21
22    exit(0);
23 }
24
25
26 void str_cli_B (FILE *fp, int sockfd)
27 {
28   int  n;
29    char sendline[MAXLINE], recvline[MAXLINE];
30
31    while (Fgets(sendline, MAXLINE, fp) != NULL) {
32
33       Writen(sockfd, sendline, strlen(sendline));
34
35       n = Readline(sockfd, recvline, MAXLINE);
36       if (n==0)
37          err_quit("str_cli: server terminated prematurely");
38
39       Fputs(recvline, stdout);
40    }
41 }
```

Figure 1
Rajah 1