



PRINCE OF SONGKHA UNIVERSITY

FACULTY OF ENGINEERING

Department of Computer Engineering

**Final Examination:** Semester 1

**Academic Year:** 2005-2006

**Date:** Monday 3rd October, 2005

**Time:** 13:30 – 16:30 (3 hours)

**Subject Number:** 240-322

**Room:** R 300

**Subject Title:** Client/Server Distributed Systems

**Lecturer:** Aj. Andrew Davison

---

**Exam Duration:** 3 hours

**This paper has 4 pages.**

**Authorized Materials:**

- Writing instruments (e.g. pens, pencils).
- Books (e.g. dictionaries) and calculators are **not** permitted.

**Instructions to Students:**

- *Answer questions in English.* Perfect English is **not** required.
- Attempt all questions.
- Write your answers in an answer book.
- Start your answer to each question on a new page
- Clearly number your answers.
- Any unreadable parts will be considered wrong.
- When writing programs, use good layout, and short comments; marks will not be deducted for minor syntax errors.
- The marks for a question appear in brackets (...) at the end of a question.

**Question 1**

(75 minutes; 75 marks)

- a) Assume the existence of a function:

```
int tcp_serv_sock(int port_number);
```

which returns a socket descriptor linked to the given port number of the machine.

Use `tcp_serv_sock()` and:

```
int accept(int sd, struct sockaddr *peer, int *len);
```

to write a code fragment showing the top-level of an *iterative* server. Document your code. (15)

*Note:* do **not** include the source code for `tcp_serv_sock()` in your answer.

- b) Use the iterative server of part (a) to write a server which sends back a client's message with the text reversed. For example, if the client sends the message "abcde\n", then the response would be "edcba\n". Document your code. (25)

*Note:* do **not** write out part (a) in your answer.

Use only `read()` and `write()` for communication:

```
int read(int sd, char *buf, int len);
int write(int sd, char *buf, int len);
```

Do **not** implement extensions to `read()` or `write()`, such as a `getline()` function.

- c) Many servers use the following
- `end_input()`
- function. Explain what the function does,
- and**
- how it affects the design of the communication protocol between the client and server. (10)

```
int end_input(char buf[], int len)
{
    if ((len == 2) &&
        (buf[0] == '\015') && (buf[1] == '\n'))
        return 1;

    if ((len == 1) && (buf[0] == '\n'))
        return 1;

    return 0;
}
```

- d) Explain the problems with using `read()` and `write()` in client/server programs. Use diagrams and small code fragments to help your explanation. (15)
- e) *Briefly* explain how the problems with `read()` and `write()` can be avoided. Use small pseudocode and/or code fragments to help your explanation. (10)

**Question 2**

(45 minutes; 45 marks)

- a) Explain RPC in words and diagrams. Do **not** include code fragments. (10)
- b) What is XDR, and why is it necessary? Explain in words, with data structure examples and diagrams. (10)
- c) Consider the following code fragment:

```
char *make_date_string(int day, int month, int year);

void main()
{
    char *dateStr = make_date_string(3, 10, 2005);
    :
}
```

The function `make_date_string()` takes three integers as input, and produces a string result. The string represents the date as specified by the day, month, and year integers, and is at most 30 characters long. For example, `dateStr` in the code fragment above would be assigned "3rd October 2005".

**Briefly** explain how `make_date_string()` would need to be rewritten to become more suited for conversion to an RPC function. Do **not** implement the code which converts the three integers into the date string. (10)

- d) Write down the XDR data types that would need to be passed between `main()` and the RPC version of `make_date_string()`. (10)
- e) Explain how the data types in part (d) would be translated to C code. (5)

**Question 3**

(30 minutes; 30 marks)

- a) Write a Java application which retrieves the Web page "http://www.amazon.com". The application should use a socket, and send a HTTP GET message to the server. (10)
- b) Describe the "no route to host" problem with the application in part (a), and how it might be solved. Do **not** include any code fragments. (5)
- c) Write a Java *applet* which retrieves the Web page "http://www.amazon.com" and displays it in a Web browser. (10)
- d) Explain in words the advantages and disadvantages of displaying a Web page using a Java applet. Do **not** include any code fragments. (5)

**Question 4 on Next Page.**

**Question 4**

(30 minutes; 30 marks)

- a) Outline the advantages and disadvantages of using a client/server model to implement a chat system. Do **not** include any code examples. (5)
- b) Explain how you would implement a threaded TCP client/server chat system in Java. Your explanation should include: system diagrams, the message protocols, how threads are used, and the importance of synchronization. Include code **fragments** only. Do **not** implement the entire system. (25)

--- *End of Examination* ---