Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Distributed Computing

Date: Wednesday 1st June 2011

Time: 09:45 - 11:45

Please answer Question ONE and any two from Questions TWO, THREE and FOUR

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

1. **Compulsory**

a) Describe one key difference between *client-server* and *peer-to-peer* applications. (2 marks)

b) Explain briefly what failures are known as *Byzantine failures* and what the *Byzantine generals problem* refers to. (2 marks)

c) Describe briefly the *two-phase commit protocol*. (2 marks)

d) Explain briefly why somebody would choose to use *Cloud Computing*. (2 marks)

e) Why is it practically impossible to achieve *strict consistency* in a distributed system? (2 marks)

f) Suppose the C function below is to be made available to remote processes using RPC. What particular implementation problem does this highlight? Why is the normal solution only partly satisfactory?

```
void doIt (int *p, int *q) { (*p)++ ; (*q)-- ; return ; }
```

(2 marks)

g) Distinguish carefully between a *name server* and a *directory server*. (2 marks)

h) What is meant by the term *causally ordered multicast*? (2 marks)

i) What properties are provided by a *secure channel*? (2 marks)

j) The Andrew File System (AFS) uses *callback promises*. Explain what this means. (2 marks)

2. a) Explain briefly why some applications are not parallelisable. Describe Amdahl's law and explain what it can be used for. (3 marks)

b) Explain briefly what the four properties commonly denoted by the acronym ACID are when referring to transactions. (3 marks)

c) A service is replicated onto 3 computers.

- The first computer, A, has a mean time between failures of 2 days.
- The second computer, B, has a mean time between failures of 3.5 days.
- The third computer, C, has a mean time between failures of 12 days.

When a failure occurs, it takes on average 12 hours to fix.

i) What is the availability of the replicated service? (2 marks)

ii) What would the availability of the replicated service be if only computers A and B were used? (2 marks)

iii) Describe how in the general case of n computers, each with a mean time between failures f_i and an average time to fix a failure t_i you would choose the two computers that provide the highest availability. (2 marks)

d) The following four processes access a shared variable x. Each process accesses a different replica of the store used to hold this variable. Before any process starts executing, the value of x is 0 in all the replicas.

<u>Process 1</u>	<u>Process 2</u>	<u>Process 3</u>	<u>Process 4</u>
x=1;	*x=2;*	*while(x==0);*	*while(x==0);*
x=4;	*x=3;*	*y=x;*	*z=x;*
		*y=5*x+y;*	*z=5*x+z;*

(i) When all four processes have completed executing the statements given, are 7 and 14 possible values of y and z respectively, if the replication uses the sequential consistency model? Justify your answer. (4 marks)

(ii) When all four processes have completed executing the statements given, are 7 and 14 possible values of y and z respectively, if the replication uses the causal consistency model? Justify your answer. (4 marks)

[PTO]

3. a) Describe in detail the Bully algorithm for the election of a leader in a distributed system. (8 marks)

   Carefully explain all the assumptions made in the Bully algorithm. (4 marks)

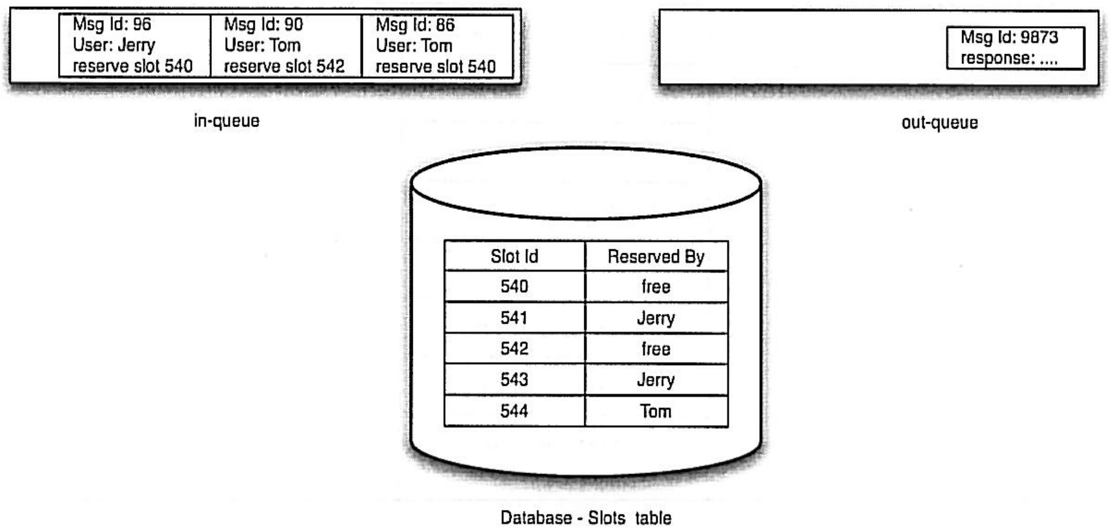
   b) Describe one application for which the Bully algorithm might be applied, indicating why a leader is needed. (2 marks)

   c) In a system containing 6 computers, identified by the integers 1-6, the leader is chosen by the Bully algorithm to be the live one with the highest identifier. Assume for this part that all messages are delivered promptly, and that the computers and the network are entirely reliable.

      i) How many messages in total are sent so that the computer with identifier 1 after it is rebooted can learn the identity of the leader by triggering an election? Take care to explain your working! (4 marks)

      ii) Repeat (i) with the computer rebooted and needing to discover the leader being that with identifier 5 instead. (2 marks)

4. This question is based on your understanding of transactions and distributed transactions, in the context of lab exercise two. The server that processes all requests (reservation, cancellation, availability and bookings retrieval) hosts a database for storing reservations and some message-oriented middleware, which provides an in-queue (to queue requests) and an out-queue (to queue responses to messages). Suppose there are three messages in the in-queue: two from Tom and one from Jerry. Suppose also that Tom already holds a reservation for slot 544 and Jerry holds a reservation for slots 541 and 543 (assume a total of 1000 slots). The state of the database and queues is shown below.



The pseudo-code that processes a reservation request is shown below. It assumes that all messages are delivered exactly once by the underlying messaging infrastructure. Also, assume that the maximum permitted number of reservations a student can hold at any one time is two, that is, `max_res=2`.

```
begin_tx;   // this is an ACID transaction to reserve a slot
      dequeue a request from the in-queue

      if requested slot is 'free' and
         number of reservations by student < max_res then
            reserve slot for the student;
            send 'slot reserved' response message;
      else if number of reservations by student >= max_res then
            send 'fail: too many reservations' response message;
            else if slot is not 'free' then
                 send 'fail: slot is not free' response message;
      endif
commit;     // transaction commits
```

(Question 4 continues on the following page)

(Question 4 continues from the previous page)

a) Show what the state of the slots table in the database would look like after all in-queue messages are executed (assuming that messages are executed left-to-right, that is, the first message executed is the request from Jerry). Your answer should also indicate clearly the response messages generated in each case. (4 marks)

b) How would your answer to a) above change if the last 6 lines of the pseudo-code on the previous page were as follows?

```
   . . .
   else if slot is not 'free' then
     send 'fail: slot is not free' response message;
     else if number of reservations by student >= max_res then
           send 'fail: too many reservations' response message;
   endif
commit;     // transaction commits
```

(2 marks)

c) Suppose that two threads are concurrently processing messages from the in-queue and the code is not enclosed in an ACID transaction. Based on the three messages already in the in-queue, would it be possible for Tom to hold more than the maximum permitted number of reservations? Would it be possible for Jerry? In both cases, you should show how this can happen (if it can) stating any assumptions you make. (5 marks)

d) What would the consequences of not using ACID transactions be when the server runs the code for: i) obtaining availability; (ii) retrieve bookings; (iii) cancelling a slot? State any assumptions you make. (3 marks)

e) Write pseudo-code that the server can run to respond to requests to cancel a slot. (3 marks)

f) Suggest a server configuration to handle the load of incoming messages if, on average, every incoming message requires CPU time of 0.1 seconds and, on average, there are about 100000 messages per hour. State any assumptions you make. (3 marks)

END OF EXAMINATION