

UNIVERSITY OF DUBLIN

TRINITY COLLEGE

FACULTY OF ENGINEERING & SYSTEM SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

B.A. (Mod). in Computer Science
Senior Sophister Examination

Trinity Term 2003

4BA8 - Distributed Systems

Tuesday 3rd June

Sam.Beckett Room

09.30 - 12.30

Mr. Jim Dowling, Dr. Vinny Cahill, Mr. Brendan Tangney

Attempt five questions (all questions carry equal marks)

- Q1. Define the "happened-before" relation and draw a space-time diagram illustrating both (potentially) causally related and concurrent events.

(5 marks)

If the events of a distributed system are time stamped with logical times, explain why it is not, in general, possible to deduce the causal order of two events from their timestamps. Explain what is meant by a vector timestamp and describe how vector timestamps can be used to solve this problem.

(5 marks)

Outline an algorithm for implementing causally ordered multicast to a group of n processes.

(10 marks)

- Q2. Distributed multi-player games have become very popular on PCs, with games such as Quake and Counter Strike becoming increasingly popular. The next wave of distributed multi-player games to capture the imagination may well be deployed on interacting mobile computers operating in a wireless network, thus opening up new possibilities for the gaming experience, but also introducing new problems that have to be overcome. For example, such games may need to maintain consistency of data, such as the players' scores or other properties and the state of the game space, despite the possibility of the wireless network being partitioned when players move away from each other.

Quazoom is such a multi-player game designed for deployment in a wireless network. In Quazoom, there are 3 different game objects: players, medikits, and flags. Players can move and shoot. Players move around in the game by moving around in the real world. When a player shoots, the first player in their line of fire gets hit, and loses 50% of their health. When a player's health reaches 0, he is killed and the player that shot him gets a point. The first player to reach a score of 3 points wins the game. A dead player cannot do anything, or be shot, for 15 seconds. After 15 seconds, his health is reset to 100% and he can continue in the game. There are medikits in the game. A player can pick these up using a keyboard command, but only if he is within 2 meters of the kit. Only one player is allowed to pick up a kit at a time, after which it disappears for 30 seconds. The kit replenishes the player's health to 100%. Players can pick up medikits even if their health is already at 100%. There is a flag that a player can pick up or drop for a capture the flag type game. The flag has an initial position. When a player carrying the flag dies or leaves the game, it is reset to its original position.

Outline a possible design for Quazoom concentrating on the distributed systems issues arising rather than, for example, the user interface design. In particular, your design should:

- motivate the choice of an appropriate paradigm for building the application;
(4 marks)
- include a diagram illustrating the overall architecture of the system including where the player and game state is stored;
(4 marks)
- give an overview of the algorithms used to ensure consistency of updates;
(7 marks)
- describe how users joining and leaving (for example, because of failure) the game are supported;
(3 marks)
- describe the impact of players becoming temporarily disconnected from the game due to mobility.
(2 marks)

- Q3. Describe the main differences between engineering local and distributed objects. What is the role of middleware for distributed objects?

(8 marks)

One of the main problems with distributed object computing has been the tangling of object-services code and business logic code. In light of this statement, discuss how component-based technologies, such as Enterprise Java Beans,TM can make distributed server-side programming easier.

(12 marks)

- Q4. CORBA represents message data in a binary format called the CDR (Common Data Representation). SOAP uses XML to represent message data. Compare and contrast these approaches with respect to:

- Message Size
- Marshalling Complexity
- Message readability

(6 marks)

Describe a travel agent web services application that demonstrates inter-company business process integration using an architectural diagram, details of the steps involved in delivering the web service and some sample WSDL and SOAP syntax. Include the following requirements in your description:

- uses external airline, hotel and insurance booking web services when constructing a package for the customer
- provides a web-based interface to the customer.

(14 marks)

- Q5. What are the relevant advantages and disadvantages of implementing a distributed file system at user and operating system level.

(4 marks)

Assuming that a distributed file system is implemented at user level, e.g. at the UNIX library level, clearly describe the state information which would need to be maintained at both client and server when a file is open. Show clearly the flow of control when a user application performs a read operation.

(12 marks)

What mechanism is used to control access to files in NFS and what would be an appropriate policy for a system administrator to put in place to ensure transparent access.

(4 marks)

Q6. In what ways can scale impact upon a distributed system?

(4 marks)

Compare the approaches to handling scale adopted by the Domain Name Service, the NFS cache coherency algorithm, the AFS cache coherency algorithm.

(12 marks)

What would be the implications of increasing the maximum length of an Ethernet or Token Ring network to be 100km?

(4 marks)

Q7. Write brief notes on **ALL** of the following

- i) What will the following piece of code measure
- ```
T := GetTime(); F := FileOpen("RemoteFile");
For I:=1 to Max Do
 FileRead(F,Buffer,SizeOf(Buffer));
End
Print("Operation took", GetTime()-T);
```

(5 marks)

- ii) Critique the structure of file names in Windows.

(5 marks)

- iii) Comment on the assertion that the distributed transaction paradigm is all that is needed to develop reliable distributed applications.

(5 marks)

- iv) Comment on the importance of probability in fault-tolerant distributed computing.

(5 marks)