

W249/403

DUBLIN INSTITUTE OF TECHNOLOGY  
KEVIN STREET, DUBLIN 8.

---

# **BSc Information Systems / Information Technology**

**Stage 4**

---

**SEMESTER 1 EXAMINATIONS 2009**

---

**DISTRIBUTED SYSTEMS**

MR. C. O'LEARY  
DR. D. LILLIS  
MR. P. LAWLESS

2 HOURS

ATTEMPT **3** QUESTIONS

ALL QUESTIONS CARRY EQUAL MARKS

1. (a) Demonstrate, using examples, the difficulty associated with synchronising clocks in a distributed system.

**(8 marks)**

- (b) Logical clocks are a method used in distributed systems to organise events according to their being either concurrent with, or having happened before, other events.

Using an example of your choice, show how the *vector clock* implementation of logical clocks captures the important *happened-before* relationship between events.

**(10 marks)**

- (c) An algorithm is required to coordinate a set of processes that all have access to the same file. It is required that only one process access the file at any point in time, and it is desirable that processes are granted access in the order with which they request it.

Provide a design for such an algorithm, and evaluate your algorithm according to its strengths and weaknesses.

**(15 marks)**

2. (a) Explain what is meant by garbage collection in an object oriented system, and demonstrate how garbage collection can be carried out effectively in a distributed object system.

**(8 marks)**

- (b) Demonstrate, using examples, the difference between passing primitive types, objects and remote objects as arguments in distributed object systems implemented using Java RMI.

**(10 marks)**

- (c) A distributed object system is required to provide the following functionality for a multi-player Java game:

- Players have a client application which they use to guess a number between 1 and 100, chosen randomly by the server.
- Any number of players can participate in the game.
- When a player makes a guess, the guess is sent to the server and the server returns true if the guess is correct or false if the guess is incorrect.
- If the correct number is guessed, the server informs all clients that the game is over.

Provide a design for such a distributed object system, by describing clearly the interface for each of the remote objects in your system, as well as the steps which take place during the game in terms of the involvement of the naming service, the passing of references and the invocation of methods. *Note: You do not need to provide code for anything other than the interfaces.*

**(15 marks)**

3. (a) Identify the various states that can be occupied by a thread in a Java process, and describe how threads move between these states.

**(8 marks)**

- (b) Consider the code shown below:

```
public class Q3 {

    public static void main(String args[]) {
        new Q3(Integer.parseInt(args[0]));
    }

    public Q3(int keyholdersRequired) {
        SecretChest secretChest = new SecretChest(keyholdersRequired);
        for(int i = 0; i < keyholdersRequired; i++) {
            new KeyHolder(secretChest).start();
            try {
                Thread.sleep((int)(Math.random() * 5000));
            } catch (InterruptedException ie) {}
        }
    }
}

class KeyHolder extends Thread {

    private SecretChest secretChest;

    public KeyHolder(SecretChest secretChest) {
        this.secretChest = secretChest;
        System.out.println("New Keyholder Arrived...");
    }

    public void run() {
        if(secretChest.unlock() == true)
            System.out.println("Secret chest unlocked");
    }
}

class SecretChest {

    private int keysStillRequired;

    public SecretChest(int keysRequired) {
        keysStillRequired = keysRequired;
    }

    public boolean unlock() {
        keysStillRequired--;
        return true;
    }
}
```

In this program, new `KeyHolder` threads are started intermittently. Each `KeyHolder` then calls the `unlock` method on the `SecretChest` to which it has been given a reference. The `unlock` method should return when the `SecretChest` is opened. The `SecretChest` should not be opened until the required number of `KeyHolder` threads have called the `unlock` method. The number of `KeyHolders` required is passed to the program as a command line argument.

As it stands, the output from the program, for 3 `KeyHolders` is:

```
java Q3 3
New Keyholder Arrived...
Secret chest unlocked
New Keyholder Arrived...
Secret chest unlocked
New Keyholder Arrived...
Secret chest unlocked
```

The desired output is:

```
java Q3 3
New Keyholder Arrived...
New Keyholder Arrived...
New Keyholder Arrived...
Secret chest unlocked
Secret chest unlocked
Secret chest unlocked
```

Using Java's object monitors, make the necessary changes to the `SecretChest` class above to produce the desired behaviour.

You do not need to copy the code into your answer book. Only provide the necessary amendments.

**(10 marks)**

- (c) An algorithm is required such that a set of processes can determine the highest priority, non-failed, process in a group. Processes have static priorities, assigned at startup time.

Processes can detect when another process has failed due to a time bound on message passing which can result in a timeout. Failed processes can reenter the system with their previous priority.

Provide a design for the required algorithm, outlining the information that is required by each of the processes, the messages that can be sent between processes, and the meaning attached to each of these messages.

Evaluate your algorithm according to its strengths and weaknesses.

**(15 marks)**

4. (a) Outline the security challenges associated with the exchange of mobile code in a distributed system, and describe the methods Java employs to address these challenges.

**(8 marks)**

- (b) Describe, compare and contrast the methods employed by the DES and RSA encryption algorithms.

**(10 marks)**

- (c) A method is required such that messages sent between processes in a distributed system can be verified by the receiving process whereby the receiving process can be certain that the sender is the process identified in the message, and not some other process pretending to have the identity contained in the message.

Provide a design for such a method, describing in detail the requirements for each of the components of the method, and evaluate your method according to its strengths and weaknesses.

**(15 marks)**