King's College London

This paper is part of an examination of the College counting towards the award of a degree. Examinations are governed by the College Regulations under the authority of the Academic Board.

Degree Programmes	BSc, MSci
Module Code	5CCS2OSC
Module Title	Operating Systems and Concurrency
Examination Period	May 2017
Time Allowed	Two hours

Rubric **ANSWER ALL QUESTIONS.**

ANSWER SECTION A ON THE MULTIPLE CHOICE ANSWER SHEET PROVIDED, CAREFULLY FOLLOWING THE INSTRUCTIONS ON THE SHEET. YOU MUST USE A PENCIL TO WRITE ON THIS SHEET AND AN ERASER TO CORRECT ANY MISTAKES. WRITE YOUR CANDIDATE NUMBER IN THE SPACE PROVIDED, PRECEDED BY 000 AND ALSO REPLACING THE LETTER WITH 0, E.G. IF YOUR CANDIDATE NUMBER IS X12345, YOU WOULD WRITE 000012345. WRITE YOUR CANDIDATE NUMBER IN THE "NAME" BOX ON THE ANSWER SHEET (DO NOT WRITE YOUR REAL NAME). THE TEST NUMBER IS 506 AND THE COLLEGE NUMBER IS 10. THERE IS NO PENALTY FOR INCORRECT ANSWERS.

ANSWER SECTION B IN AN ANSWER BOOKLET START EACH QUESTION ON A NEW PAGE.

Calculators	Calculators may be used. The following models are permitted: Casio fx83 / Casio fx85.
Notes	Books, notes or other written material may not be brought into this examination

PLEASE DO NOT REMOVE THIS PAPER FROM THE EXAMINATION ROOM

© 2017 King's College London

Section B: Answer both Questions in this Section in your Answer Booklet

35. Concurrency (25 marks)

a. In this question you will develop some existing code to produce a working concurrent program. The given code models a web server, and two types of users, admin and non-admin users.

```
public class User extends Thread {

    WebServer server;
    String name;
    boolean adminUser;

    public User (WebServer toUse, String nameIn,
                 boolean isAnAdmin){
        server = toUse;
        name = nameIn;
        adminUser = isAnAdmin;
    }

    public boolean isAdmin(){
        return adminUser;
    }

    public String getUserName(){
        return name;
    }

    public void run(){
        server.connectToServer(this);
        server.interact();
        server.disconnectFromServer(this);
    }
}
```

QUESTION 35 CONTINUES ON NEXT PAGE

SEE NEXT PAGE

```
public class WebServer {

   private int activeUsers = 0;
   private int userCount = 0;
   private int adminsWaiting = 0;
   boolean adminUsing = false;

   public synchronized void connectToServer(User u){
       if(u.isAdmin()){
           ++adminsWaiting;
       }
       //your code goes here (1)
       if(u.isAdmin()){
           adminUsing = true;
           --adminsWaiting;
       } else {
           ++userCount;
       }
       System.out.println(u.getUserName() + " connected");
   }

   public void interact(){
       try {
           Thread.sleep(100);
       } catch (InterruptedException ex) {
           ex.printStackTrace();
       }
   }
```

QUESTION 35 CONTINUES ON NEXT PAGE

```
  public synchronized void disconnectFromServer(User u){
        if(u.isAdmin()){
            adminUsing = false;
        } else {
            --userCount;
        }
        System.out.println(u.getUserName() + " disconnected");
        //your code goes here (2)
    }
}
```

i. Write a main method that will:

- Create a web server object to be used by all users;
- Create 10 non-admin users (i.e. users with adminUser set to false), named user0, user1,...user9 and execute these as concurrent threads;
- Create 2 admin users, named admin1 and admin2 and execute these as concurrent threads.

[4 marks]

QUESTION 35 CONTINUES ON NEXT PAGE

ii. The intended behaviour of the program is that an unlimited number of non-admin users can use the server simultaneously; but, an admin user must not access the server at the same time as any other user (admin or non-admin). Write some code to be added at the place marked `//your code goes here (1)` that will ensure:

- No user may connect to the server if an admin is connected;
- No admin user may connect to the server if another user is connected;
- If an admin is waiting to connect non-admin users must not be allowed to connect until the admin has finished using the server (to prevent starvation of admins).

You will also need to add some code at the place marked `//your code goes here (2)`s to ensure your implementation works correctly.

[7 marks]

iii. Assuming the modifications have been made to correctly enforce the rules in part (ii), describe what problem might occur if the connectToServer method was not synchronized.

[3 marks]

iv. What would happen if the interact method was synchronized?

[2 marks]

.

QUESTION 35 CONTINUES ON NEXT PAGE

b. A variable in memory has the value of 10. Explain by showing a problematic execution, why two threads adding 2 to this variable without enforcement of mutual exclusion can lead to an incorrect resulting value.

[4 marks]

c. Show how the atomic processor instruction swap, which swaps the values of two boolean variables, can be used enforce mutual exclusion. In your answer you should show the pre- and post-protocol that is used and explain how this prevents processes entering when others are in the critical section, but allows processes to enter when no others are in the critical section.

[5 marks]

36. Operating Systems (25 marks)

a. i. What is a virtual machine? In what way does a virtual machine protect the underlying hardware?

[4 marks]

ii. What is an *arbitrary code execution* vulnerability? How does using a virtual machine help lessen the impact of such a vulnerability?

[3 marks]

b. i. Operating system kernels provide several possible software interrupts. How does the process indicate which interrupt it wishes to call? Identify two ways in which parameters can be passed to an interrupt.

[3 marks]

ii. What are *user mode* and *kernel mode*? What is the *mode bit* for each of these, and how does it change when handling a software interrupt?

[4 marks]

iii. What is the key design difference between a *monolithic kernel* and a *microkernel*? Why might a microkernel offer worse performance, due to this design?

[4 marks]

QUESTION 36 CONTINUES ON NEXT PAGE

c. i. Which of the following can be represented using the standard POSIX file permissions model (without using access control lists)? For those that can be represented, state the file permissions you would use, and if relevant, any user groups you would create.

a 'rickroll.html' should be read-writeable to rick, and to andrew; but should be read-only to anyone else.

b 'keats.html' should be read-writable to andrew and amanda; should be read-only by students; and should not be readable by anyone else.

c 'trollface.jpg' should be read-writeable by dave, and not readable by anyone else.

d 'answer.java' should be read-writeable by andrew; read-only by amanda; and not readable by anyone else.

e 'messages.txt' should be read only by andrew; and anyone should be able to write to it.

[5 marks]

ii. What is the purpose of setting the UID bit on system commands, for instance the '/usr/bin/passwd' executable that changes a user's password?

[2 marks]