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 2016
Time Allowed	Two hours
Rubric	ANSWER All QUESTIONS FROM SECTION A, ONE QUESTION FROM SECTION B AND ONE QUESTION FROM SECTION C. IF MORE THAN ONE QUESTION IS ANSWERED IN SECTION B OR C THE MARKS FOR THE BEST QUESTION ANSWERED WILL BE USED. ALL QUESTIONS CARRY EQUAL MARKS. ANSWER EACH QUESTION ON A NEW PAGE OF YOUR ANSWER BOOK AND WRITE ITS NUMBER IN THE SPACE PROVIDED.
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

Section A: Answer both Questions in this Section (Question 1 and Question 2)

1. In this question you will build a Java simulation of a level crossing (a place where the trainline crosses a road) based on the following Java classes.

```
public class LevelCrossing {

  private boolean roadClosed = false;
  private String trainDirection = "west";
  private int vehiclesOnCrossing = 0;

  public void useCrossing(Vehicle v){
    System.out.println("Vehicle " + v.vehicleName() +
     " Using Crossing in Direction " + v.getDirection());
    try{
      Thread.sleep(100);
    } catch (Exception e){
      e.printStackTrace();
    }
    System.out.println("Vehicle " + v.vehicleName() +
     " Leaving Crossing in Direction " + v.getDirection());

  }
}
```

QUESTION 1 CONTINUES ON NEXT PAGE

```
public class Vehicle {

    String type, direction, name;
    LevelCrossing crossing;

    public Vehicle(String typeIn, String directionIn,
    String nameIn, LevelCrossing levelCrossing){
        type = typeIn;
        direction = directionIn;
        name = nameIn;
        crossing = levelCrossing;
    }

    public void cross(){
        crossing.useCrossing(this);
    }

    public String getType(){
        return type;
    }

    public String getDirection(){
        return direction;
    }

    public String vehicleName(){
        return name;
    }
}
```

QUESTION 1 CONTINUES ON NEXT PAGE

a. The programmer wants instances of the `Vehicle` class to be executed as concurrent threads, and when these are executed the cross method should run. Give two modifications that need to be made to the class Vehicle in order for this to be possible.

[3 marks]

b. Assuming the modifications required in part (a) have been made, write a main method that could be used to:

- Create a `LevelCrossing` object.
- Create and execute 5 car threads (3 north, 2 south) and 5 trains (3 east, 2 west), with the names car0, car1, ..., car4 and train0, train1, ..., train4 respectively (which exact car or train has which exact name is not important). These car/train threads should use the same level crossing object.

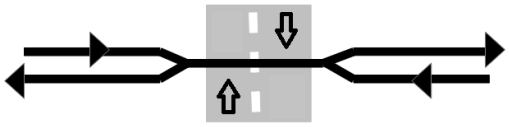
[4 marks]

c. In its current form (together with the the correct modifications in part (a)) the simulation will allow cars to crash into trains, and trains to crash into both cars and other trains. Propose a small modification to the code in the LevelCrossing class that will prevent more than one vehicle entering the crossing at the same time.

[2 marks]

QUESTION 1 CONTINUES ON NEXT PAGE

In reality the desired behaviour is that multiple cars can use the crossing at once in both directions. As there is a single train line only trains going in the same direction can use the crossing together:



The body of the cross method is modified thus:

```
crossing.enterCrossing(this);
try{
 Thread.sleep(100);
} catch (Exception e){ e.printStackTrace(); }
crossing.leaveCrossing(this);
```

d. Implement the method enterCrossing (including the method signature) according to the following specification:

- The method should take a Vehicle as an argument.
- Only one thread can execute `enterCrossing` at once.
- A car should wait until there are no trains on the crossing before it enters, you should **not** prevent cars from entering the crossing when other cars are using it.
- A train should wait until there are no cars on the crossing and there are no trains using the crossing in the opposite direction before entering the crossing (again you should not prevent trains from entering the crossing in the same direction).
- When a car enters the crossing it should set `roadClosed` to false, increment `vehiclesOnCrossing` then print a message stating it has entered the crossing along with its name and direction;
- When a train enters the crossing it should set `roadClosed` to true, increment `vehiclesOnCrossing`, set `trainDirection` to its direction, then print a message stating it has entered the crossing and giving its name and direction.

[7 marks]

QUESTION 1 CONTINUES ON NEXT PAGE

e. Implement the method exitCrossing according to the following specificatoin:

- The method should take a Vehicle as an argument.
- Only one thread can execute `exitCrossing` at once.
- Print a message that the vehicle has left the crossing and giving its name and direction.
- Decrement the number of users of the crossing and, if this has now become zero, notify any threads that might be waiting.

[5 marks]

f. Does the specification given above for `enterCrossing`, allow starvation of trains, cars, both or neither? Explain your answer.

[4 marks]

The body of this page intentionally only contains this statement, in order to improve the layout of the exam paper.

2. a. Consider the following C code fragment (line numbers on the left):

```
 0  static bool success = false;
 1  void getPIN() {
 2    char PIN[8];
 3    for (int i = 0; i < 8; ++i) {  PIN[i] = ' ';  }
 4
 5    gets(PIN);
 6    if (strcmp(PIN,"1234") == 0) {
 7      success = true;
 8    } else {
 9      success = false;
10    }
11  }
12
13  void login(int maxTries) {
14    int trycount = 0;
15    while (trycount < maxTries) {
16      getPIN();
17      if (success) {
18        printf("Correct, have some money.\n"); return;
19      }
20      ++trycount;
21    }
22  }
23
24  void main() {
25    int maxTries = 3;
26    login(maxTries);
27    print("End of program\n");
28  }
```

QUESTION 2 CONTINUES ON NEXT PAGE

The program begins by running the main function, at line 24.

i. When the execution of code reaches line 4 for the first time, what is on the stack?

[4 marks]

ii. Assuming the code is running on a 32-bit machine – so integer variables and return addresses are each 4-bytes in size, and `char` variables are 1 byte each – how much stack space has been used at this point?

[1 mark]

iii. In what way is the code shown vulnerable to a buffer overflow attack? Give an example input that could be given to the program, to cause line 18 of the code to be executed, even if the correct PIN (as tested at line 6) was not known.

[4 marks]

iv. How does *stack protection* work? In this context, what are *canary words*?

[4 marks]

v. As in part (i), sketch out what would be on stack at line 4 of the code, but this time include a suitable canary word of your choice at the relevant points. Explain how your choice of canary word prevents the buffer overflow attack you detailed in part (iii).

[3 marks]

QUESTION 2 CONTINUES ON NEXT PAGE

b. It is proposed that the details of usernames and passwords on a computer system be saved in a textfile `/etc/users` with the following format:

```
uid=1,user=root,password=andrew
uid=2,user=brian,password=badluck
uid=3,user=kanye,password=interrupt
...
```

i. If the file is saved with the following permissions, which users will be able to access it?

```
#owner: root
#group: root
user::rw-
group::---
mask::rw-
other::---
```

[1 mark]

ii. How can the *set UID bit* on executables be used to make the file accessible to other users? What is the *real user ID* and what is the *effective user ID* when the user brian is running such an executable?

[4 marks]

iii. It is proposed that the executable `/bin/cat` has the set UID bit set, so developers can use the following command to test whether a user with username 'abc' and password 'xyz' can be found on the system:

```
cat /etc/users | grep 'user=abc,password=xyz$'
```

Why is it a security issue to set the set UID bit on `cat`, particularly if the `/etc/users` file is in the format described?

[2 marks]

QUESTION 2 CONTINUES ON NEXT PAGE

iv. To avoid this security issue it is decided to move the data from the users file to a database, with columns 'user' and 'password'. The following code snippet is used to prepare the SQL query:

```
public String query(String user, String password) {
  String toReturn = "SELECT * FROM USERS WHERE ";
  toReturn += "user='" + user + "' AND ";
  toReturn += "password='" + password + "';";
  return toReturn;
}
```

Is the authentication system now secure? Justify your answer.

[2 marks]

Section B: Answer one Question from this Section (Question 3 and Question 4)

3. a. Consider the following proposed solution to the critical section problem. For each of the three required properties of a solution to the critical section problem state whether the property is satisfied. If so, give an informal proof that this is the case; if not, give an execution trace that violates the property.

turn ← 0	
p	q
loop forever:	loop forever:
p1: non-critical section	q1: non-critical section
p2: turn ← turn + 1	q2: await turn = 0
p3: await turn = 0	q3: turn ← turn + 1
p4: critical section	q4: critical section
p5: turn ← 0	q5: turn ← 0

[9 marks]

b. Consider the following code representing producer threads that perform some independent computation to produce data and then append that data to a buffer.

Integer Array buffer ← empty array, Integer nextItem ← 0	
p	q
Integer v1	Integer v2
loop forever:	loop forever:
p1: v1 ← produce	q1: v2 ← produce
p2: buffer[nextItem] = v1	q2: buffer[nextItem] = v2
p3: nextItem = nextItem + 1	q3: nextItem = nextItem + 1

i. Explain how data loss can arise if these two producers are executed as concurrent threads.

[4 marks]

QUESTION 3 CONTINUES ON NEXT PAGE

ii. Propose a synchronisation mechanism (i.e. some code before (preprotocol) and after (postprotocol) the critical section) to enforce mutual exclusion between p and q where necessary, whilst also guaranteeing freedom from deadlock and freedom from starvation. You can make use of any synchronisation mechanism you wish to do this (e.g. semaphores, test and set or another of your choice) and can assume atomic assignment operators if you require. Give your answer in pseudocode in the style above, making sure to initialise any variables that you introduce.

[4 marks]

c. This part of the question is about the Banker's Algorithm.

i. Use the Banker's algorithm to determine whether the following system is in a safe or unsafe state. If multiple processes are able to run at once, break ties by choosing the one with the lowest ID. If the system is safe, give the safe order in which the processes would execute, if the system is in an unsafe state list the processes that cannot run. The current resource availability in the system is 1,1,1 and the following processes are running:

- P0 max requirement 1,2,3 and allocation 0,0,0.
- P1 max requirement 3,1,3 and allocation 1,0,2.
- P2 max requirement 2,0,0 and allocation 1,0,0.
- P3 max requirement 0,4,2 and allocation 0,3,0.
- P4 max requirement 0,2,0 and allocation 0,0,0.

[6 marks]

ii. The Banker's algorithm can show that a system is in either a safe or an unsafe state. If a system is in a safe state does this mean it cannot deadlock? If a system is in an unsafe state does this mean it will definitely deadlock?

[2 marks]

4. a. Consider the following threads, all of which are to be executed concurrently.

Semaphore s1 = 0, s2 = 0, s3 = 1, int v ← 4		
p	q	r
p1: wait(s1)	q1: wait(s2)	r1: wait(s3)
p2: v = v/2	q2: wait(s3)	r2: v = v * 3
p3: v = v * 4	q3: v = v + 4	r3: signal(s3)
p4: signal(s2)	p5: signal(s2)	r4: signal(s1)
		r5: signal(s2)

i. Assuming each line of the code is executed atomically (including the division, addition and multiplication statements) what are the possible values of the variable v upon a successful complete execution of this program? Explain how the execution of the code can give rise to these values.

[7 marks]

ii. What problem could occur if the multiplication operator was not atomic? Explain your answer.

[4 marks]

iii. If the order of the lines q1 and q2 was swapped (i.e. q calls wait(s3) before wait(s2)) what problem could occur? Will this happen on every run of the program? Explain your answers.

[4 marks]

QUESTION 4 CONTINUES ON NEXT PAGE

b. This part of the question is about CPU scheduling.

i. What is meant by ageing in priority scheduling and how does it prevent starvation of low priority processes?

[3 marks]

ii. Explain how priority scheduling with ageing can be implemented using a multi-level feedback queue system.

[3 marks]

iii. A multi-level queueing system uses round robin with quantum 10ms when processes first enter the system, any processes not complete after they have exhausted their quantum, sent to a second (lower priority) queue which also uses round robin scheduling, but with quantum 50ms, finally the processes are sent to the lowest priority queue which executes first come first served. What are the advantages of this approach versus simply using round robin scheduling with a quantum of 10ms or 50ms?

[4 marks]

Section C: Answer one Question from this Section (Question 5 and Question 6)

5. a. Given five memory partitions (holes) of the following sizes (in bytes) in the following order:

p0) 192

p1) 160

p2) 256

p3) 128

p4) 512

i. Assuming Memory Control Blocks (MCBs) of 16 bytes in size, how would each of the best-fit and first-fit algorithms place memory allocation requests ordered as follows?

A 128

B 200

C 100

D 400

E 90

F 160

[8 marks]

ii. What are *external fragmentation* and *internal fragmentation*? With reference to Question 5.a.i, which of best-fit and first-fit did and did not demonstrate a consequence of these?

[6 marks]

QUESTION 5 CONTINUES ON NEXT PAGE

iii. The heap contains a large free block of memory. A process makes 128 consecutive allocation requests, each of size 4 bytes; each of these requests is met by taking 4 bytes from the free block of memory, and placing a new memory control block before the rest.
How many bytes would be used by Memory Control Blocks at this point?

[1 mark]

iv. What is an object pool (memory pool)? If our object pools can each store 64 objects of size 4 bytes, and a process makes 128 consecutive allocation requests, each of size 4 bytes – but these requests are now met using the object pools – how many bytes of memory would now be used by Memory Control Blocks?

[2 marks]

b. i. What is the role of a 'mode bit', and how does it change throughout the process of handling a software interrupt?

[3 marks]

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

[3 marks]

iii. A multi-user operating system provides a special interrupt '`yolo()`' that sets the mode bit to 0 for all processes running on the machine. Discuss the effect of this, and what could possibly go wrong.

[2 marks]

6. This question concerns memory paging and IO.

a. What is the difference between a *physical* memory address and a *logical* memory address? How can a page table be used to translate from logical to physical addresses?

[4 marks]

b. A 32-bit system has a page size of 2048 bytes (2^{11} bytes). If logical memory addresses are 32 bits, how many bits are used for the page number, and how many for the offset? How much space is needed to store the page table for each process?
(NB If you wish, you may give your answers as powers of two.)

[3 marks]

c. In the context of IO, what is Direct Memory Access (DMA)? What is an 'IO interlock', and why is it important when using DMA?

[4 marks]

d. Consider a system with four frames of memory, and the following scenario:

i An IO interlock is requested, using one of the frames of memory;
ii Pages are accessed in the order: 0,1,2,3,0,1
iii The IO interlock is released;
iv Pages are accessed in the order: 3,0,1,2

When do page faults occur in the two page-access sequences, using the Least Recently Used (LRU) and First In–First out (FIFO) page replacement algorithms?

[8 marks]

e. When using DMA, why is it a good idea to align the block of memory being used for DMA, with the start of a page boundary?

[2 marks]

QUESTION 6 CONTINUES ON NEXT PAGE

f. How can *dirty bits* be used to reduce the overheads of using virtual memory with swapping? How can additional bits of information about each page, such as dirty bits or read-only bits, be stored in the page table, without increasing the memory needed to store the page table? If the page size is 2^{11} bytes, how many such additional bits can be stored in this way?

[4 marks]