Queen Mary
University of London

MSCI HONS / BSC HONS / BSC(ENG) HONS / BN631IE **Examination by course unit**

Friday, 01 May 2015 10:00 am

ECS404U Computer Systems and Networks Duration: 2 hours 30 minutes

YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR

Answer ALL FOUR questions

ONLY Non-programmable calculators are permitted in this examination. Please state on your answer book the name and type of machine used.

Complete all rough workings in the answer book and cross through any work that is not to be assessed.

Possession of unauthorised material at any time when under examination conditions is an assessment offence and can lead to expulsion from QMUL. Check now to ensure you do not have any notes, mobile phones, smartwatches or unauthorised electronic devices on your person. If you do, raise your hand and give them to an invigilator immediately. It is also an offence to have any writing of any kind on your person, including on your body. If you are found to have hidden unauthorised material elsewhere, including toilets and cloakrooms it will be treated as being found in your possession. Unauthorised material found on your mobile phone or other electronic device will be considered the same as being in possession of paper notes. A mobile phone that causes a disruption in the exam is also an assessment offence.

EXAM PAPERS MUST NOT BE REMOVED FROM THE EXAM ROOM

Examiners:

Graham White

Edmund Robinson

Question 1

This question is about the various stages from source code to the execution of a program.

a) What is machine code? What is assembly language? What are the differences between them?

[5 marks]

b) What do the *assembler*, *linker*, and *loader* do when they translate assembly language to machine code and load it into memory? Do not worry too much about the precise distinctions between assembler, linker and loader.

[5 marks]

c) Give an example of a *high level language*. What is a *variable* in such a high level language?

[5 marks]

d) What is the *load-store paradigm*? Explain how this paradigm is used when the CPU calculates with data. Give a *very small* code fragment showing how two variables, `x` and `y`, are added together and the result put in a variable `z`.

[10 marks]

The relevant MIPS instructions are:

```
lw $t1 p              # loads an integer from memory location p
                      # to register $t1
sw $t2 q              # stores an integer from register $t2
                      # to memory location q
add $t1 $t2 $t3       # adds registers $t2 and $t3 together
                      # and puts the result in $t1
```

Question 2

This question is about digital representation

Marks will be awarded for working as well as for correct results, and you should therefore include all working in your answers.

a) This part is about conversion between binary and hexadecimal representations.

i) Explain why it is easier to convert between hexadecimal and binary, or binary and hexadecimal than between hexadecimal and decimal representations.

ii) Convert the following *hexadecimal* expressions to *binary*:

1) $3A1$
2) $ABCDE458$

iii) Convert the following bit sequences to hexadecimal (sequences arranged in groups of eight bits):

1) 11110000 11100001 11000011
2) 10011101 00111100 01111000

[5 marks]

b) This part combines knowledge of 2's complement representations and binary arithmetic. It uses 8-bit representations throughout.

i) Carry out the following multiplication, taking the numbers as ordinary positive binary:

```
 11011101
 00000011  *
-----------

-----------

-----------
```

ii) In 8-bit 2's complement, what are the numbers represented by:

1) 11011101
2) 00000011
3) The 8 rightmost bits of your answer for the sum in i) above.

In each case explain how you get this answer.

iii) What is the connection between the numbers you have given as answers in 1), 2) and 3)?

iv) State the property of the 2's complement representation that ensures that the relation you have just given holds.

[10 marks]

c) This part is about character sets.

i) How many bits are used in the ASCII character set? Given that 'A' is character number 65 and 'a' is character number 97, give the bit patterns corresponding to the ASCII representations of 'B' and 'b'.

ii) How many bits are used in the character set ISO-8859-1? Give the ISO-8859-1 representations of 'C' and 'c' as bit patterns.

iii) What would be a quick way of checking whether a character was upper or lower case?

[5 marks]

d) This part is about floating point representation.

i) Give an example of a number in (ordinary decimal) scientific notation that uses 8 significant figures, and explain what the term *significant figures* means.

ii) Give an example of two numbers that can both be represented accurately by numbers with 8 significant figures, but whose sum can not be.

iii) Explain why your example is relevant to scientific computation on existing computers, even though they use binary, not decimal.

[5 marks]

Question 3

This question is about race conditions and deadlock

a) Suppose we have two processors executing at the same time, and both doubling the same variable, `z`. Assume that each of them must first load the value of the variable, then double it by adding it to itself, and then store the doubled value of the variable. There is a potential *race condition*: i.e. the result of this calculation depends on the way that the instructions interleave. Give two different orders of execution which arrive at two different results.

[5 marks]

b) Describe what a *lock* is, and write a code fragment (using operators `lock` and `unlock` which operate on memory locations) which doubles `z`, and which will avoid a race condition

[10 marks]

c) Describe what *deadlock* is, and write a pair of code fragments, one for each processor, using operators `lock` and `unlock` as above, which lead to deadlock when executed against each other, and explain how the deadlock occurs. (There is no need to write any code inside the lock-unlock pairs: simply show how the locks are configured)

[10 marks]

Question 4

This question is about computer networks

a) When a webserver is sending a webpage to a browser, the packets typically have four layers corresponding to different protocols being used. List the layers in order, starting at the innermost, giving the name of the layer and the protocol you would expect to see there in this instance.

[8 marks]

b) Draw a simple block-structure diagram of a typical packet including headers and footers, but not detail about the fields in the headers.

[2 marks]

c) Briefly explain the function of the Domain Name Service using the following output from the `dig` command to illustrate your points.

```
[edmundr@bert ~]$ dig mysis.qmul.ac.uk

; <<>> DiG 9.8.2rc1-RedHat-9.8.2-0.30.rc1.el6_6.1 <<>> mysis.qmul.ac.uk
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51079
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;mysis.qmul.ac.uk.              IN      A

;; ANSWER SECTION:
mysis.qmul.ac.uk.       300     IN      A       138.37.29.35

;; Query time: 1002 msec
;; SERVER: 138.37.0.88#53(138.37.0.88)
;; WHEN: Fri Feb  6 12:07:54 2015
;; MSG SIZE  rcvd: 50

[edmundr@bert ~]$
```

Figure 1: `dig` output

[4 marks]

d) At which packet layers (if any) would you expect to find the following data. Explain your answers in terms of the functionality of the layer and the data.

i) The symbolic name of a webserver.

ii) An IP address.

iii) A port number.

iv) A MAC address.

[8 marks]

e) Describe briefly what the following traceroute output is telling you about the route from `bert.student.eecs.qmul.ac.uk` to `mysis.qmul.ac.uk` (the server for the College's student database).

```
[edmundr@bert ~]$ traceroute mysis.qmul.ac.uk
traceroute to mysis.qmul.ac.uk (138.37.29.35), 30 hops max, 60 byte packets
 1  belt503.eecs.qmul.ac.uk (138.37.37.253)  0.250 ms  0.182 ms  0.185 ms
 2  eecs-sub-backbone-46.core-net.qmul.ac.uk (138.37.2.46)  0.588 ms  0.699 ms  0.718
ms
 3  gooseberry-ebr3.core-net.qmul.ac.uk (138.37.3.207)  0.834 ms  0.821 ms  0.801 ms
 4  138.37.2.22 (138.37.2.22)  1.028 ms  1.018 ms  1.050 ms
[edmundr@bert ~]$
```

Figure 2: `traceroute` output

[3 marks]

End of Paper