Birkbeck
(University of London)

MSc Examination

Department of Computer Science and Information Systems

COMPUTER SYSTEMS (COIY060H7)

CREDIT VALUE: 15 credits

Date of examination: 21/05/2018
Duration of paper: 14:30–16:30

There are four questions in this paper; each of them is compulsory and worth 25 marks.
The paper is not prior-disclosed.
The use of electronic calculators is not permitted.

1. Consider the computation `(m1+m2)+(m3*m4)` where `m1`, ... denote the content of memory locations.

   (a) Write assembly code typical of RISC machines for this computation. Use at most two operands for each instruction (e.g., `ADD r1 r2` for `r1 <- r1 + r2`) and use at most three registers altogether. The result of the computation should be stored in a register.

   (8 marks)

   (b) Briefly describe the main idea for superscalar processors.

   (5 marks)

   (c) Show the pipeline activity by drawing a diagram when the assembly code is executed on a superscalar processor. There are four pipeline stages: fetch–decode, register-read, execute and write-back, and there are two functional units for each pipeline stage. State you additional assumptions.

   (12 marks)

2. (a) List the most important registers in a generic CPU and briefly explain their functions.

   (6 marks)

   (b) Briefly describe the compiler-based register optimization technique (typically used for RISC machines).

   (8 marks)

   (c) Explain what register windows are and how they are used to improve performance.

   (11 marks)

3. Consider the following attempt to solve the critical section problem for two processes where the initial value of turn is 0 or 1:

```
while(TRUE){                    while(TRUE){
   while(turn!=0);                 while(turn!=1);
   critical-section();             critical-section();
   turn=1;                         turn=0;
   noncritical-section();          noncritical-section();
}                               }
```

Process 0 Process 1

   (a) Briefly explain the concepts of *race condition*, *critical section* and *mutual exclusion*.

   (7 marks)

(b) Explain whether the above code avoids deadlock.

(9 marks)

(c) Explain whether the above code is starvation free.

(9 marks)

4. (a) Explain the benefits of the virtual memory technique.

(5 marks)

(b) Explain the concept of *locality of reference*.

(5 marks)

(c) Consider a (RISC) machine that uses 4KB pages and 4-byte instructions, and assume that a process switch just occurred. Loading a page from the hard disk into memory takes 10 ms. The initial probability (after the process switch) that a referenced instruction is in main memory is 0.1. How many milliseconds does it take to load the first referenced instruction into the IR on average? (You can ignore the loading time from main memory into the IR.)

(9 marks)

(d) Explain how the situation changes for loading the second instruction, and give an estimate for the average load time in milliseconds for this case. (You can assume that the first instruction referenced is not a jump.)

(6 marks)

