

**Birkbeck
(University of London)**

MSc Examination

Department of Computer Science and Information Systems

COMPUTER SYSTEMS (COIY060H7)

CREDIT VALUE: 15 credits

Date of examination: 13/06/2014

Duration of paper: 14:30–16:30

There are five questions in this paper; each of them is compulsory and worth 20 marks.
The paper is not prior-disclosed.
The use of electronic calculators is not permitted.

1. Consider the computation $(m1+m2)+(m3*m4)$ where $m1, \dots$ 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 $r1 \leftarrow r1 + r2$) and use at most three registers altogether. (8 marks)
 - (b) Briefly describe the main ideas for pipelining and for superscalar processors. (4 marks)
 - (c) Show the pipeline activity by drawing a diagram when the assembly code is executed on a superscalar processor. There are three pipeline stages: fetch-decode, execute and write-back. There are two units for each of fetch-decode, execute (one adder and one multiplier) and write-back. (8 marks)
2. Consider your assembly code for the previous question.
 - (a) Explain dependencies (true or otherwise) and show them in your assembly code. (6 marks)
 - (b) Reorganise your code from the previous item using the register renaming technique to remove false dependencies. (4 marks)
 - (c) Show the pipeline activity by drawing a diagram when the reorganised code is executed on the same superscalar processor as in the previous question. (10 marks)
3. Consider the following concurrent code, where \parallel indicates that the execution of processes P_1 and P_2 are interleaved:

input variables: $y_1 = 0 \ y_2 = 0$

$$P_1 \left[\begin{array}{l} \mathbf{while(true)\{ \\ \quad noncritical; \\ \quad y_1 = 1; \\ \quad while(y_2 = 0); \\ \quad critical; \\ \quad y_1 = 0; \\ \quad \} \end{array} \right] \parallel P_2 \left[\begin{array}{l} \mathbf{while(true)\{ \\ \quad noncritical; \\ \quad y_2 = 1; \\ \quad while(y_1 = 0); \\ \quad critical; \\ \quad y_2 = 0; \\ \quad \} \end{array} \right]$$

- (a) Briefly explain the concepts of *critical section*, *mutual exclusion* and *deadlock*. (4 marks)
- (b) Determine whether the above code avoids deadlock. (8 marks)
- (c) Determine whether the above code achieves mutual exclusion. (8 marks)

4. (a) Describe the main goals of the short-term scheduler (or dispatcher) in an interactive system. (6 marks)
- (b) Explain the main difference between preemptive and non-preemptive scheduling algorithms. Which type is used in a timesharing system? (4 marks)
- (c) Consider the following scenario. Processes A, B and E have priority 1, and processes C and D have priority 0 (the higher the priority, the sooner the process is scheduled). Their arrival times are 0, 1, 2, 3 and 4 seconds, and their run times are 2, 1, 1, 1 and 1 seconds, respectively. Compute the average turnaround time for the five processes A to E using each of the following short-term scheduling algorithms:
- first-come-first-served (4 marks)
 - multiple queues (priority queueing) using round-robin on each priority level. (6 marks)
5. (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. If the running process references an instruction that is in main memory, it takes 10 μ s to load into the IR. If the referenced instruction is on the hard disk, it takes on average 100 ms to load into main memory. The initial probability (after the process switch) that a referenced instruction is in main memory is 0.1. How many microseconds does it take to load the first referenced instruction into the IR on average? Explain how the situation changes for 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 and you can ignore the loading time from main memory into the IR in this case.) (10 marks)