THE AUSTRALIAN NATIONAL UNIVERSITY

Mid Semester Examination – August 2007

COMP2310

Concurrent and Distributed Systems

Study Period: 0 minutes

Time Allowed: 50 hours

Permitted Materials: NONE

Questions are NOT equally weighted.

The questions are followed by labelled, framed blank panels into which your answers MUST be written. **No additional paper will be provided. Writing outside of the boxes will not be marked.**

This exam is marked out of 40. You should answer all questions.

Your mark for this exam will contribute at most 10% of your total course mark, according to the marking scheme given on the course web page.

Name (family name first):

Student Number:

The following are for use by your friendly examiner!

Q1 Mark	Q2 Mark	Q3 Mark	Q4 Mark	Q5 Mark	Total Mark

Name:..... UID: .....

Question 1 [6 marks] Basic Concurrency

(a) Define precisely under what circumstances two events should be considered to have occurred concurrently.

[**2 marks**]

(b) Explain the statement: "What appears sequential on a higher abstraction level, is usually concurrent at a lower abstraction level". Include in your answer at least one example illustrating how this statement can be true.

[**4 marks**]

Question 2 [8 marks] Processes and Threads

(a) Give four pieces of information that you might reasonably expect to find contained within a process control block.

[4 marks]

(b) Draw one or more labelled diagrams to illustrate the difference(s) between a process and a thread.

[4 marks]

Name:..... UID: .....

Question 3 [14 marks] Synchronization

(a) The following code attempts to implement Peterson's algorithm for mutual exclusion. It contains **two** different logical errors. Identify the **two** errors and correct them. For each logical error state what the effect would be if the code contained only that error.

```
type Critical_Section_State is (In_CS, Out_CS);
C1, C2 : Critical_Section_State := Out_CS;
Last : Positive range 1..2 := 1;
```

```
task body P1 is
begin
  loop
    -- non_critical_section_1;
    C1 := In_CS;
    Last := 1;
    loop
      exit when C2 = Out_CS
           or else Last /= 2;
    end loop;
    -- critical_section_1;
    C2 := Out_CS;
  end loop;
end P1;
```

```
task body P2 is
begin
  loop
    -- non_critical_section_2;
    C2 := In_CS;
    Last := 2;
    loop
      exit when C1 = Out_CS
           or else Last /= 1;
    end loop;
    -- critical_section_2;
    C1 := Out_CS;
  end loop;
end P2;
```

[8 marks]

(b) The PowerPC architecture provides hardware support for memory cell reservation. What is "memory cell reservation"?

[2 marks]

(c) The Java notification methods; `wait, notify` and `notifyAll`, provide a form of condition synchronization. Semaphores also provide a form of condition synchronization. List **two** significant differences between these two approaches to condition synchronization.

[4 marks]

Name:..... UID: .....

Question 4 [8 marks] Message Passing

(a) Give one advantage and one disadvantage of asynchronous message passing over synchronous message passing.

[4 marks]

(b) Show how you would synchronize four processes using only one-to-one message passing.

[4 marks]

Name:..... UID: .....

Question 5 [4 marks] Non-Determinism

(a) In terms of determinism how does the semantics of the Ada `select` statement differ from a `switch` statement in C or Java?

[2 marks]

(b) Describe two scenarios where it would be necessary to employ an Ada `select` statement.

[2 marks]

Name:..... UID: .....

Continuation of answer to Question ☐ Part ☐

Continuation of answer to Question ☐ Part ☐