

The Australian National University
Midterm Examination – September 2010

COMP 2310 / COMP 6310

Concurrent and Distributed Systems

Study period: 15 minutes
Time allowed: 1.5 hours
Total marks: 100
Permitted materials: None

Questions are **not** equally weighted – sizes of answer boxes do **not** necessarily relate to the number of marks given for this question.

All your answers must be written in the boxes provided in this booklet. You will be provided with scrap paper for working, but only those answers written in this booklet will be marked. Do not remove this booklet from the examination room. There is additional space at the end of the booklet in case the boxes provided are insufficient. Label any answer you write at the end of the booklet with the number of the question it refers to.

Greater marks will be awarded for answers that are simple, short and concrete than for answers of a sketchy and rambling nature. Marks will be lost for giving information that is irrelevant to a question.

Student number:

--

The following are for use by the examiners

<i>Q1 mark</i>	<i>Q2 mark</i>	<i>Q3 mark</i>	<i>Q4 mark</i>	<i>Q5 mark</i>	<i>Q6 mark</i>

<i>Total mark</i>

1. [7 marks] General Concurrency

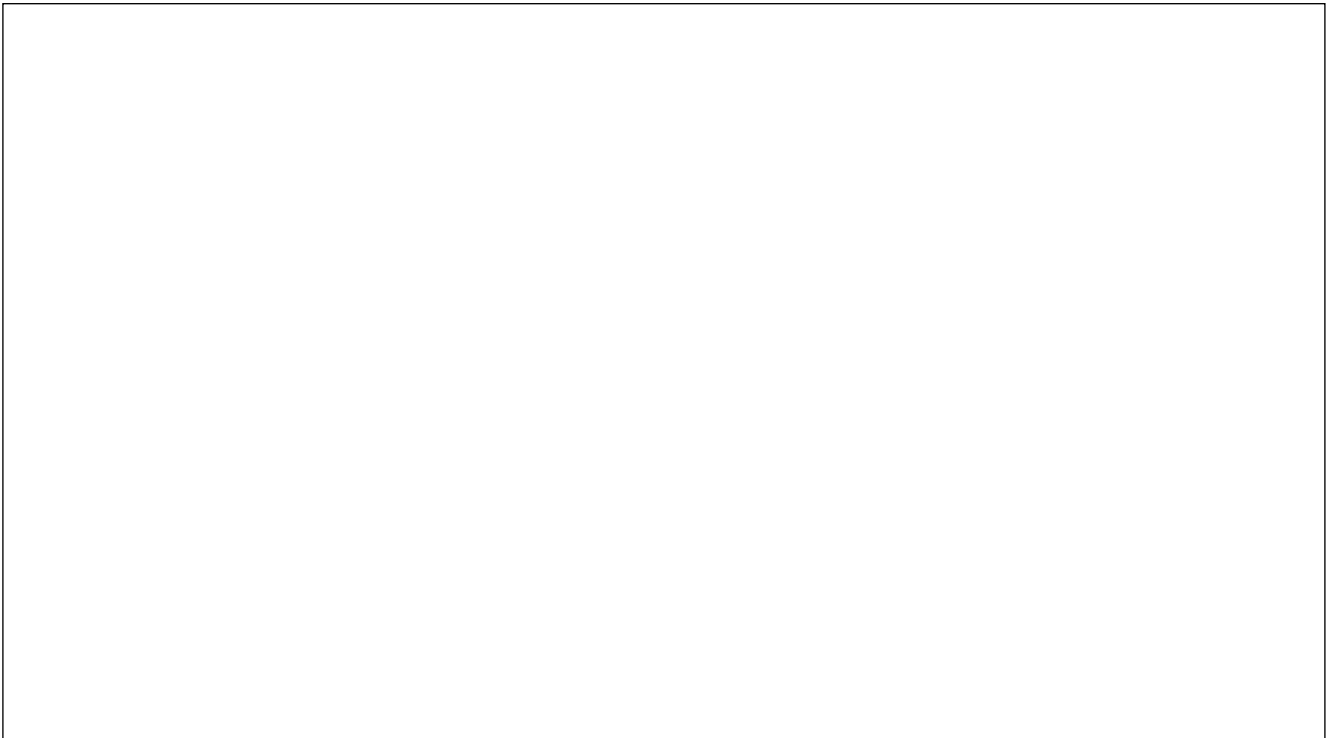
- (a) [3 marks] “What appears to be sequential on a higher level of abstraction can be concurrent on a lower level.” Is this statement true or false? Give a detailed answer. If the statement is true, provide at least one example; if the statement is false, explain why.

- (b) [4 marks] Draw a diagram showing the 5 basic states of a process, and all valid transitions between states (including their names). What events can trigger transitions that will lead to a running process being taken out of the CPU? Name at least three.

2. [18 marks] Mutual Exclusion

- (a) [12 marks] Implement the two methods P(S) and V(S) of a binary semaphore using the atomic test-and-set instruction offered by a particular CPU (assume that there is no scheduler or OS).

(b) [6 marks] How does the implementation of a Semaphore as offered by an operating system with scheduler differ from a Semaphore as implemented in a) ?



3. [30 marks] Synchronisation

- (a) [5 marks] Explain the differences and common features of Conditional Critical Regions, Monitors and Protected Objects.

- (b) [5 marks] POSIX offers `pthread_mutex` as a monitor-like construct. Which basic concepts of a Monitor does this POSIX interface implement, and which concepts are missing, according to the original Monitor definition by Hoare?

- (c) [20 marks] Read carefully through the following Ada program. It compiles without warning (Questions are on the next page).

```

procedure Protected_Example is

  type Priority is (High, Low);

  protected Trigger is
    entry Trigger_High;
    entry Trigger_Low;
    entry Queue (P : in Priority);
  private
    Current_Priority : Priority := Low;
    Trigger          : Boolean  := False;
  end Trigger;

  protected body Trigger is
    entry Trigger_High when True is
      begin
        Current_Priority := High;
        Trigger:=True;
      end;

    entry Trigger_Low when True is
      begin
        Current_Priority := Low;
        Trigger:=True;
      end;

    entry Queue (P : in Priority) when Trigger is
      begin
        if P /= Current_Priority then
          requeue Queue;
        end if;
        Trigger:=False;
      end;
  end Trigger;

  task type Priority_Task(P: Priority);
  task body Priority_Task is
  begin
    Trigger.Queue (P);
    if P = High then
      Trigger.Trigger_High;
    end if;
  end Priority_Task;

  T1          : Priority_Task (Low);
  T2, T3, T4 : Priority_Task (High);

begin
  delay 1.0;          --seconds
  Trigger.Trigger_High;
  delay 1.0;          --seconds
  Trigger.Trigger_Low;
end Protected_example;

```

(i) [2 marks] How many tasks are created by this program?

(ii) [8 marks] How many tasks are released immediately after the first call to `Trigger.Trigger_High` in the main program, and which one(s)? Give details to justify your answer. Which guard(s) need to be evaluated, and when?

(iii) [10 marks] How many, and which tasks of this program will not terminate? Give precise reasons why, and give an explanation if it is a deadlock or a livelock.

4. [10 marks] Message Passing

(a) [6 marks] Is it possible to emulate synchronous message passing with asynchronous message passing? Explain how and to what extent, or explain why not, depending on your answer.

(b) [4 marks] Where would you use asymmetric message passing? Provide at least one example.

Student number:.....

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part

continuation of answer to question

part