The Australian National University
Final Examination – November 2017

Comp2310 & Comp6310
Systems, Networks and Concurrency

Study period:	15 minutes
Writing time:	3 hours (after study period)
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 (and also note inside the original answer box that your answer is continued at the end of the booklet).

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

Q1 mark	*Q2 mark*	*Q3 mark*	*Q4 mark*	*Q5 mark*	*Q6 mark*	*Total mark*

1. [18 marks] General Concurrency

(a) [3 marks] Which of the following hardware architectures require or are supportive for concurrent programming?

Pipelines, Vector processors, Hyper-threading

Give precise reasons.

(b) [4 marks] Explain the functionality of a network router. Which layers of the OSI model are implemented? Give reasons why a specific OSI layer needs to be implemented in a network router.

(c) [4 marks] Which layer(s) of the OSI model are specified by IEEE 802.3 (commonly known as Ethernet). Give reasons why a specific OSI layer needs to be specified.

(d) [7 marks] If you could design a programming languages which would lend itself to any form of concurrent systems while also providing high level of abstraction, what would be the core language feature(s) which you would include?

2. [22 marks] Synchronization and Communication

(a) [6 marks] In the context of concurrent programming explain what is meant by a race condition? Include in your answer 20 lines or less of pseudo code that shows a race condition.

(b) [8 marks] Emulate asynchronous message passing by means of synchronous message passing. Identify the limitations of your design (if there are any). You can provide your answer in any programming language of your choice (including pseudo-code). You can also add a diagram.

(c) [8 marks] Emulate local, asynchronous message passing by means of memory based synchronization. Identify the limitations of your design (if there are any). You can provide your answer in any programming language of your choice (including pseudo-code). You can also add a diagram.

3. [14 marks] Selective Synchronization

Read the following Ada program carefully. The program is syntactically correct and will compile without warnings. See questions below.

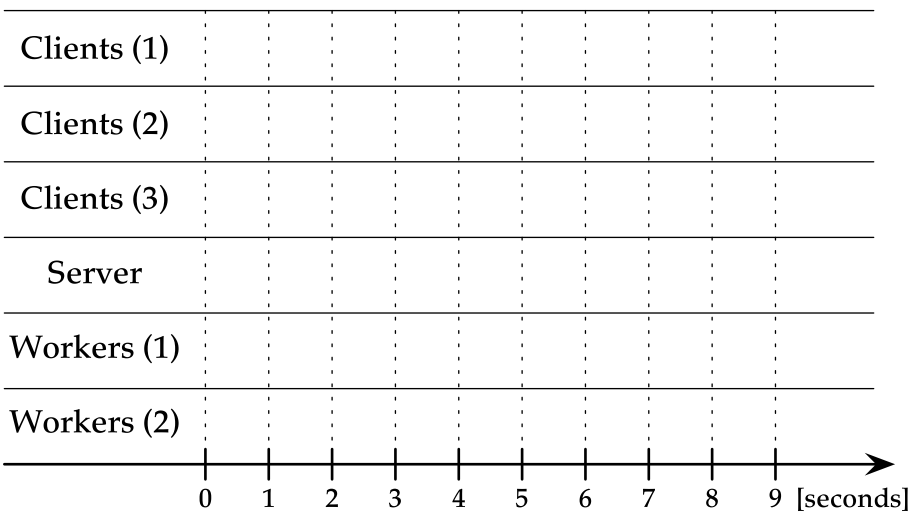
```
with Ada.Text_IO; use Ada.Text_IO;

procedure Working_Class is

   task type Worker is
      entry Ready;
      entry Service;
   end Worker;

   task Server is
      entry Service;
   end Server;

   task type Client;

   Workers : array (1 .. 2) of Worker;
   Clients : array (1 .. 3) of Client;
               pragma Unreferenced (Clients);

   task body Worker is

   begin
      loop
         select
            accept Ready;
            Put ("R"); --> Output!
         or
            accept Service do
               delay 2.0;
               Put ("W"); --> Output!
            end Service;
         or
            terminate;
         end select;
      end loop;
   end Worker;

   task body Server is

   begin
      loop
         select
            accept Service do
               for i in Workers'Range loop
                  select
                     Workers (i).Ready;
                     requeue
                         Workers (i).Service;
                  else
                     null;
                  end select;
               end loop;
               Put ("F"); --> Output!
            end Service;
         or
            terminate;
         end select;
      end loop;
   end Server;

   task body Client is

   begin
      Server.Service;
      delay 1.0;
      Put ("B"); --> Output!
      Server.Service;
      delay 3.0;
      Put ("T"); --> Output!
   end Client;

begin
   null;
end Working_Class;
```

(i) [2 marks] How many task queues are implemented in this program? Name them.

(ii) [4 marks] Considering the program structure, which of the entries in this program would you consider to be potentially blocking for a non-trivial amount of time? Assume that your underlying hardware supports running all concurrent entities in this program in parallel.

(iii) [4 marks] Will this program never / sometimes / always terminate? Explain your answer.

(iv) [4 marks] On the provided time-lines below, add the outputs which you expect from each entity at the correct time. If you think that there are multiple possible output sequences then pick one of them.



4. [20 marks] Safety and Liveness

(a) [5 marks] Does the exclusive usage of synchronous message passing prevent deadlocks? Give precise reasons why it would be free of deadlocks or a counter-example if you can construct a deadlock situation using only synchronous message passing.

(b) [5 marks] Suggest a synchronization scheme which is guaranteed to be free of deadlocks. If you nominated synchronous message passing above as deadlock preventing, you cannot mention it here again. If your synchronization scheme is only deadlock free under certain assumptions, then name those assumptions.

(c) [10 marks] Read the following Ada program carefully. The program is syntactically correct and will compile without warnings. See questions below.

```
with Ada.Text_IO; use Ada.Text_IO;

procedure Ring is

   type Ring_Ix is mod 5;

   task type Node is
      entry Provide_Id (Provided_Id : Ring_Ix);
   end Node;

   protected type Port is
      procedure Provide_Id (Provided_Id       : Ring_Ix);
      procedure Get_Port_A (Router_Nr, Port_B : Ring_Ix);
      procedure Get_Port_B (Router_Nr         : Ring_Ix);
   private
      Port_Id : Ring_Ix := Ring_Ix'Invalid_Value;
   end Port;

   Nodes : array (Ring_Ix) of Node;
   Ports : array (Ring_Ix) of Port;

   protected body Port is

      procedure Provide_Id (Provided_Id : Ring_Ix) is

      begin
         Port_Id := Provided_Id;
      end Provide_Id;

      procedure Get_Port_A (Router_Nr, Port_B : Ring_Ix) is

      begin
         Put_Line ("Router" & Ring_Ix'Image (Router_Nr) &
                     " aquired" & Ring_Ix'Image (Port_Id) & " as its port A ");
         Ports (Port_B).Get_Port_B (Router_Nr);
      end Get_Port_A;

      procedure Get_Port_B (Router_Nr : Ring_Ix) is

      begin
         Put_Line ("Router" & Ring_Ix'Image (Router_Nr) &
                     " aquired" & Ring_Ix'Image (Port_Id) & " as its port B ");
      end Get_Port_B;

   end Port;

   task body Node is

      Id : Ring_Ix := Ring_Ix'Invalid_Value;

   begin
      accept Provide_Id (Provided_Id : Ring_Ix) do
         Id := Provided_Id;
      end Provide_Id;

      Ports (Id).Get_Port_A (Id, Id + 1);
   end Node;

begin
   for Id in Ring_Ix loop
      Ports (Id).Provide_Id (Id);
   end loop;
   for Id in Ring_Ix loop
      Nodes (Id).Provide_Id (Id);
   end loop;
end Ring;
```

(i) [2 marks] How many tasks and protected objects are created by this program and how many protected objects have to be entered simultaneously by each task in order for it to complete?

(ii) [4 marks] Will this program never/certainly/potentially deadlock? Provide a precise reason.

(iii) [4 marks] If you answered with "certainly or potentially deadlocks" in the previous question then suggest changes to the program such that it never deadlocks. If you answered with "never deadlocks" then suggest changes to the program such that it will potentially deadlock. Which of the required deadlock conditions are you adding or removing with your suggestion?

5. [11 marks] Data Parallelism

Read this syntactically correct Chapel expression and then proceed to the questions below:

```
sqrt (+ reduce ((Vector_1 - Vector_2)**2))
```

where you should assume the following declarations for `Vector_1` and `Vector_2`:

```
config const n = 1000;

const Index = {1 .. n};

var Vector_1, Vector_2 : [Index] real;
```

(i) [1 mark] What is the type of this expression?

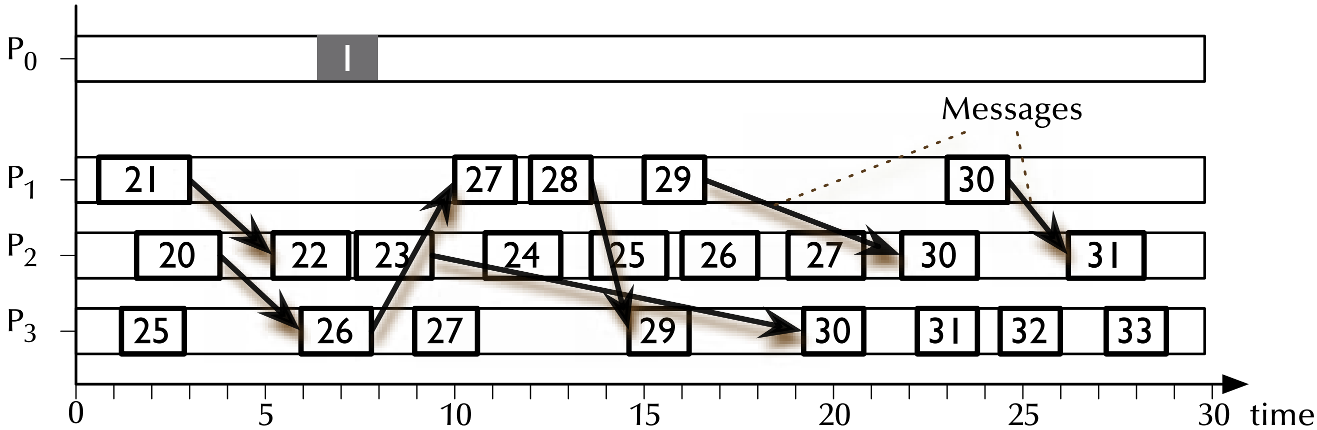
(ii) [6 marks] Enumerate and explain the potentially data parallel operations which are implemented by this Chapel expression. Also state for each operation the degree of potential data parallelism in terms of the maximum number of utilized cores.

(iii) [4 marks] Assume an infinite number of available computing cores. How does the processing time complexity of the above expression scale with `n` (in terms of overall time passed – not in terms of the sum of all executed machine instructions)?

6. [15 marks] Distributed Systems

(a) [10 marks] Process 0 in the diagram below has been tasked to take a snapshot of the processes set process 1 to process 3. Only message passing is available to perform this task. Events inside each tasks are carrying a logical time-stamp.

(i) [4 marks] Assume finite message speeds (meaning they cannot be received instantaneously) and draw on the diagram below how the snapshot will be assembled.



(ii) [2 marks] Which time-stamp out of each process will be the latest time stamp from the past with respect to the recorded snapshot?

(iii) [4 marks] Can a snapshot in a distributed system which has been assembled by means of a message passing system relate to a single, global time? Explain why this would be possible or not be possible. If you need to assume something for your answer then state your assumptions.

(b) [5 marks] What can you conclude about the events a and b (including whether they happened on the same or on different processors) if the relations between the logical times $C(a)$ and $C(b)$ associated with these events are:

(i) [1 mark] $C(a) < C(b)$

(ii) [1 mark] $C(a) = C(b)$

(iii) [1 mark] $C(a) \neq C(b)$

(iv) [2 marks] Is it true that if $C(a) > C(b)$ then there always exists an event c, such that: $C(a) > C(c) > C(b)$? Will your answer change if you measure time in calendar (or "real") time instead of logical time? Give precise reasons for your answers.

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* [ ]