063174

THE UNIVERSITY *of York*

Degree Examinations ***

DEPARTMENT OF COMPUTER SCIENCE

Real-Time Systems and Programming Languages

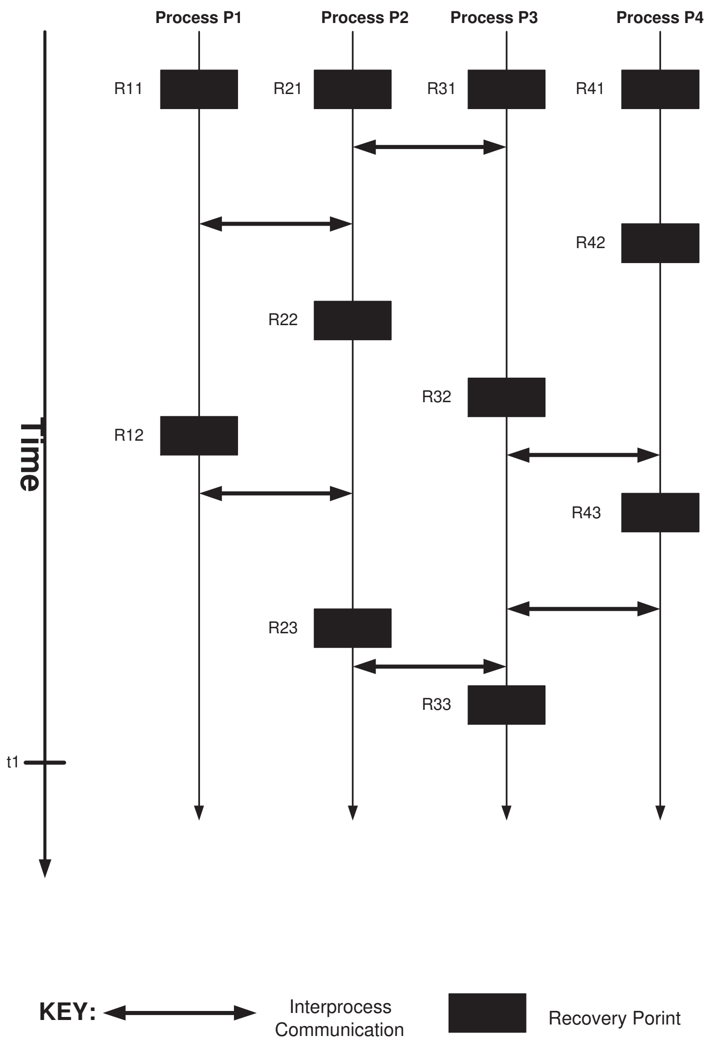
Time allowed: **Three (3) hours**

Candidates should answer not more than **four** questions

1 (25 marks)

(i) [8 marks] Explain the terms: **recovery point**, **checkpointing**, the **domino effect** and **recovery lines**.

(ii) [7 marks] The following diagram illustrates the concurrent execution of four communicating processes (P1, P2, P3 and P4) and their associated recovery points (for example, R11 is the first recovery point for process P1).



Explain what happens when an error is detected by

- Process P1 at time t1.

- Process P2 at time t1.
- Process P3 at time t1.
- Process P4 at time t1.

(iii) [10 marks] Consider the following code:

```
I : Integer := 10;
K : Integer := 5;

procedure Primary is
begin
    K := I * 2;
    I := I + 9;
end Primary;

procedure Secondary is
begin
    I := K;
    K := K + 15;;
end Secondary;

procedure Tertiary is
begin
    I := 2*K + I;
    K := 0;
end Tertiary;
```

This code is to be executed in a recovery block environment AND an exception handling environment. The following is the recovery block environment:

```
ensure I = 20
by
    Primary;
else by
    Secondary;
else by
    Tertiary;
else
    I := 20;
end;
```

The following is the exception handling environment:

```
Failed : exception;
type Module is (P,S,T);

for Try in Module loop
    begin
        case Try is
            when P =>
                Primary;
                if I /= 20 then
                  raise Failed;
                end if;
                exit;
            when S =>
                Secondary;
                if I /= 20 then
                  raise Failed;
                end if;
                exit;
            when T =>
                Tertiary;
                if I /= 20 then
                  raise Failed;
                end if;
                exit;
            end case;
    exception
        when Failed =>
            if Try = T then
              I := 20;
            end if;
    end;
end loop;
```

Assuming the **Termination model** of exception handling, discuss the execution of BOTH the recovery block environment and the exception handling environment code fragments.

2 (25 marks)

(i) [10 marks] Compare and contrast protected objects in Ada 95 and objects with synchronized methods and statements in Java.

(ii) [5 marks] Explain the synchronisation imposed by the following Ada protected object.

```
protected type Barrier(Needed : Positive) is
    entry Wait;
private
    Releasing : Boolean := False;
end Barrier;

protected body Barrier is
    entry Wait when Wait'Count = Needed or Releasing is
    begin
        if Wait'Count = 0 then
            Releasing = False;
        else
            Releasing := True;
        end if;
    end Wait;
end Barrier;
```

(iii) [10 marks] Show how the above Barrier can be implemented in Java.

3 (25 marks)

(i) [15 marks] Compare and contrast the facilities provided by Ada and Real-Time Java for detecting deadline overrun and for detecting worst-case execution time overrun.

(ii) [5 marks] Discuss the relationship between a task over-running its worst-case execution time and missing its deadline. Why is there no direct correlation?

(iii) [5 marks] Why might a system miss deadlines even though all its tasks are taking less than their worst-case execution times.

4 (25 marks)

Figure 1 shows three periodic tasks (A, B and C) that monitor environmental inputs and log these readings in a central store. A further periodic task (D) reads the store and updates an operator's screen. The final task (E) is sporadic, it is released for execution if two of the input readings are too high. It then sounds a warning bell for 1 minute. Task E will then wait 1 further minute before it calls back into the store. Hence, the minimum inter-arrival time for E is 120 seconds.

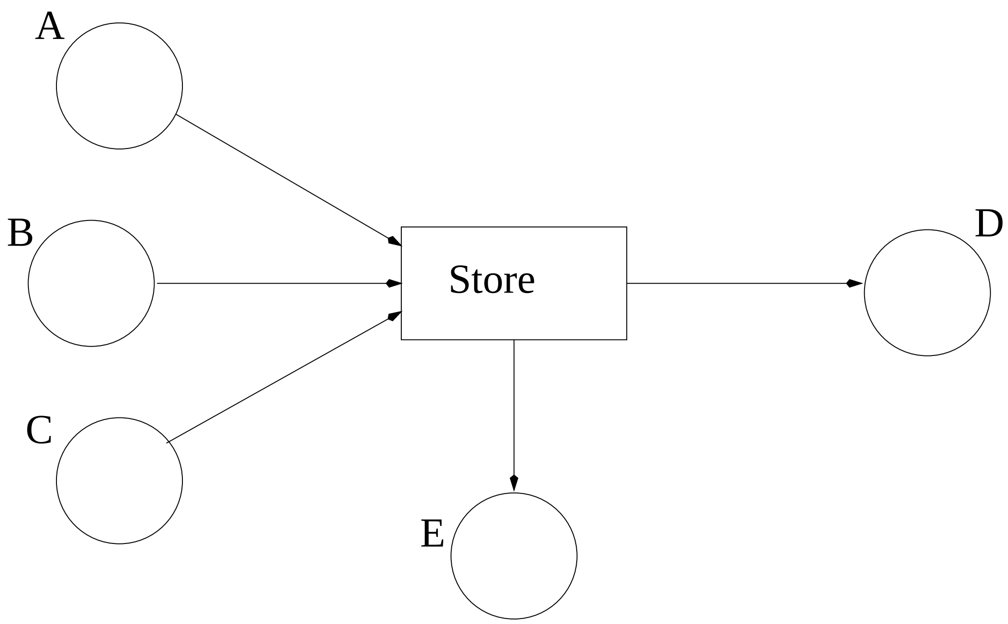


Figure 1:

Table 1 gives the temporal characteristics of the five tasks. The units are tenths of a second. One unit of time is spent in the store by each task (but this time has already been added into the computation time for each task).

(i) [7 marks] Give the code in Ada for a protected object that would implement the Store.

(ii) [4 marks] Sketch the structure of task E (using Ada).

(iii) [4 marks] What would the optimal priority assignment be for these five tasks if preemptive priority-based scheduling is used.

(iv) [10 marks] Using response time analysis decide if this system of tasks would

Task	Period	Deadline	Computation Time
A	15	15	1
B	18	18	2
C	12	12	3
D	35	35	10
E	1200	5	3

Table 1: Task Characteristics

meet all its deadlines.

5 (25 marks)

(i) [15 marks] Explain the facilities in Ada that support SYNCHRONOUS communication between tasks: you should consider task entries, the accept statement and the select statement. You should also explain how the requeue facility can be used inside an accept statement.

(ii) [10 marks] Nowadays, most airports have a security counter where passengers must report before being allowed to check-in. Consider the following task specifications which model the check-in process:

```
type Airline_Desk_T is (Air_France, American,
                        British_Airways, Virgin, Any);

subtype Airline_T is Airline_Desk_T range Air_France .. Virgin;
type Ticket is new Integer;

task Security_Counter is
  entry Check_In(A: Airline_T; T : Ticket);
  entry Desk_Opened(Ad : Airline_Desk_T);
  entry Desk_Closed(Ad : Airline_Desk_T);
end Security_Counter;

task type Check_In_Desk_T(For_Airline : Airline_Desk_T) is
  entry Check_In(A: Airline_T; T : Ticket);
end Check_In_Desk_T;
```

Passengers are also represented by tasks.

```
task type Passenger;
task body Passenger is
  A : Airline_T := ...;
  T: Ticket := ...;
begin
  Security_Counter.Check_In(A,T);
end Passenger;
```

Show how the body of `Security_Counter` can be implemented assuming that

- There is one check-in desk for each airline. These desks open and close at arbitrary times.
- There is one check-in desk which will deal with any airline. It also opens and closes at arbitrary times. However, it is always open if any of the other desks are open and it is always the first desk to open if all the desks are closed.
- The `Security_Counter` tasks should forward the passengers to an appropriate desk when desks are open.

6 (25 marks)

(i) [10 marks] Describe the facilities provided by Modula-1 for the programming of device drivers.

(ii) [12 marks] An embedded computer controls a pedestrian crossing. The crossing has traffic lights for controlling the movement of cars, a button for pedestrians to press when they wish to cross the road, an illuminated pedestrian figure and a bleeper for signaling to pedestrians that it is safe to cross. The required control algorithm is as follows:

- when the system is initialised, the traffic lights are set to green; the illuminated figure is set to red, the bleeper is off, and the button is enabled;
- when the button is pressed, the button is disabled;
- after 20 seconds, the traffic lights are turned to amber;

- after a further 10 seconds, the traffic lights are turned to red, the illuminated figure is turned to green and the bleeper is turned on;
- after a further 30 seconds, the bleeper is turned off, the illuminated figure is set to flashing green, and the traffic lights are set to flashing amber;
- after a further 10 seconds, the illuminated figure is set to red, the traffic lights are set to green and the button is enabled.

The embedded computer has the following 16-bit memory-mapped I/O registers:

- Button Control Register at address Octal 167444 of which bit 6, when set, enables interrupts (an interrupt signaling that the button has been pressed). When bit 6 is cleared, all but the last button press is ignored. The last button press is remembered and an interrupt is generated when bit 6 is again set.
- Bleeper Control Register at address Octal 167446 of which bit 8, when set, turns on the beeper. When bit 8 is cleared, the bleeper is turned off.
- Traffic Lights Data Buffer Register at address Octal 167450. The status of the traffic lights is determined by the following bits patterns:
  - octal 0000 - all lights off
  - octal 0010 - traffic lights are red
  - octal 0014 - traffic lights are amber
  - octal 0020 - traffic lights are green
  - octal 0024 - traffic lights are flashing amber
- Illuminated Pedestrian Data Buffer Register at Octal location 167452. The status of the illuminated person is determined by the following bits patterns:
  - octal 0000 - the figure is not illuminated
  - octal 0010 - the figure is illuminated red
  - octal 0014 - the figure is illuminated green
  - octal 0020 - the figure is illuminated green and flashing
- Interrupts from button presses are vectored and occur through location octal 40 and occur at hardware priority 6.

Continued.

Sketch the implementation of a Modula-1 device module for implementing the required control algorithm. You may assume that the following module is available and that the signal "tick" is sent every second.

```
MODULE timing;
  DEFINE tick;
  VAR tick : signal;
  . . .
END timing
```

(iii) [3 marks] Explain why the timing module given at the end of Part (ii) defines a signal as its interface rather than:

```
PROCEDURE delay(Sec : INTEGER);
```