

UNIVERSITY OF SOUTHAMPTON

COMP3006W1

SEMESTER 2 EXAMINATIONS 2011-2012

REAL-TIME AND EMBEDDED SYSTEMS

DURATION 90 MINS (1.5 Hours)

This paper contains 5 questions

Answer **three** questions .

An outline marking scheme is shown in brackets to the right of each question.

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

Task	T	C	P	$W_b = C_b = 2$	$W_a^0 = C_a = 5$
a	20	5	1	$W_c^0 = C_c = 4$	$W_a^1 = 5 + \lceil \frac{5}{7} \rceil \cdot 2 + \lceil \frac{5}{12} \rceil \cdot 4 = 11$
b	7	2	3	$W_c^1 = 4 + \lceil \frac{4}{7} \rceil \cdot 2 = 6$	$W_a^2 = 5 + \lceil \frac{11}{7} \rceil \cdot 2 + \lceil \frac{11}{12} \rceil \cdot 4 = 13$
c	12	4	2	$W_c^2 = 4 + \lceil \frac{6}{7} \rceil \cdot 2 = 6 = W_c^1$	$W_a^3 = 5 + \lceil \frac{13}{7} \rceil \cdot 2 + \lceil \frac{13}{12} \rceil \cdot 4 = 13 = W_a^2$

$R_c = 6 \quad 2$

$R_b = 13$

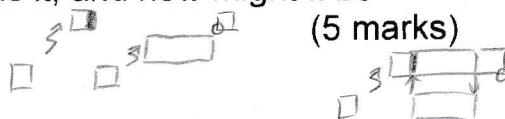
COMP3006W1

1. (i) Use a utilisation analysis to estimate whether the fixed priority technique will allow a set of 3 tasks with periods 20ms, 7ms and 12ms, which require computation times of 5ms, 2ms and 4ms respectively, to meet all their deadlines.
- $\frac{4}{20} + \frac{2}{7} + \frac{4}{12} = 0.814 < 0.83 \quad \checkmark$
- (6 marks)

- (ii) Use a response-time analysis to determine whether the fixed priority technique will allow the system of section (c) to meet all its deadlines.
- (12 marks)

- (iii) Priority inversion may be a problem if these tasks interact through critical sections. What is it, and how might it be avoided?

Priority - inheritance



(5 marks)

2. You are offered the following implementation of a thread-safe Counter in Java:

with public synchronized

```

import java.*;
public final class Counter {
    private AtomicInteger v=0;
    public int getValue() {
        return v;
    }
    public int increment() {
        return ++value;
    }
}
  
```

- (i) Correct the obvious mistakes in the implementation, without restructuring the code.
- (3 marks)

- (ii) Write down an alternative implementation of this class using Java 1.5's AtomicInteger.
- (10 marks)

- (iii) How might AtomicInteger be efficiently implemented on an Intel 486 architecture?
- (5 marks)

LOCK COMPXCHG 1h/32, 132

- (iv) What is the ABA problem?
- (5 marks)

Hidden modification in shared memory

3. Your embedded system is equipped with a memory-mapped byte wide output port at hexadecimal address **FFFF0000** and an associated byte-wide input port at hexadecimal address **FFFF0001**.
 - (i) How would you declare pointers to these two ports in order to make maximum usage of the C programming language's safety features? (6 marks)
 - (ii) The status register indicates that the port is willing to accept another byte by setting its low order bit to a 1. The value of the other bits is unspecified. Write a suitable C function to output a byte to the port. (6 marks)
 - (iii) What special precautions will you need to take if you wish to invoke C code from within an interrupt service routine you are writing for a 386 system? (11 marks)
4. (i) Explain the terms *polled*, *interrupt driven* and *DMA* in the context of a device driver in a real-time kernel. (9 marks)
- (ii) How might each of these driver techniques affect the system's responsiveness to other devices? (6 marks)
- (iii) What are the performance implications of the use of PVM and caches in a real-time system? Ignoring other caches, what would be the effect of a TLB miss on a data read in an iA386 architecture? (8 marks)

5. The following blocks of C code attempt to implement a critical section which maintains mutual exclusion between two threads.

Declared global in both threads:

```
static int turn = 1;
```

Thread 1:

```
while (turn == 1);
/* Critical region */
turn = 1;
```

Thread 2:

```
while (turn==0);
/* Critical region */
turn = 0;
```

- (i) Using the *FSP language* of the *Labelled Transition System Analyser* or a suitable alternative, construct a model of the two interacting threads and the turn variable. (9 marks)
- (ii) Construct a suitable safety property which, when model-checked, will assure that mutual exclusion is maintained (7 marks)
- (iii) Construct a further model-checkable test which a conventional critical section would be expected to pass, but which this implementation fails. (7 marks)

EXT

(7 marks)

$TURV1 = (\text{set1} \rightarrow TURV1 \mid \text{unset2} \rightarrow TURV0 \mid \text{isset2} \rightarrow TURV1),$

$TURV0 = (\text{set1} \rightarrow TURV1 \mid \text{unset2} \rightarrow TURV0 \mid \text{unset1} \rightarrow TURV0).$

$P1 = (\text{isunset1} \rightarrow \text{claim1} \rightarrow \text{release1} \rightarrow \text{set1} \rightarrow P1).$

$P2 = (\text{isset2} \rightarrow \text{claim2} \rightarrow \text{release2} \rightarrow \text{unset2} \rightarrow P2).$

END OF PAPER

property MUXEX = ($\text{claim1} \rightarrow \text{release1} \rightarrow \text{MUXEX} \mid \text{claim2} \rightarrow \text{release2} \rightarrow \text{MUXEX}$).

$\text{EXT} = (t \rightarrow \text{claim0} \rightarrow \text{release0} \rightarrow \text{EXT} \mid t \rightarrow \text{claim1} \rightarrow \text{release1} \rightarrow \text{EXT}) \setminus \{t\}$

→ The crude mutex actually fires strict alternation and to do better we need Peterson's algorithm.

Require that the mutex be able to offer the (free) critical section to either thread.

We check this by composing the mutex with code which makes an internal choice about which thread will request the mutex.

Possible deadlock would imply that the mutex fails to accept this imposed choice.

UNIVERSITY OF SOUTHAMPTON

COMP3006W1

SEMESTER 2 EXAMINATION 2012 - 2013

REAL-TIME COMPUTING AND EMBEDDED SYSTEMS

DURATION 120 MINS (2 Hours)

This paper contains 5 questions

Answer THREE questions.

An outline marking scheme is shown in brackets to the right of each question.

Ctt 11

This examination is worth 70%.

Criticism of Java's synchronized keyword

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

Page 1 of 6

Question 1.

You are asked to implement a Readers-Writers application using a shared buffer in multi-threaded C++11. You should provide two functions, `read_int()` and `write_int(i)` which will remove and insert a single integer into a 16 deep FIFO buffer but will block if the buffer is empty or full respectively. Many threads may be using each of the two functions at one time.

- (a) Write correct C++11 code for the `read_int()` and `write_int()` functions, and for the shared state. [11 marks]
- (b) How will your code perform as the number of threads becomes large? [6 marks]
- (c) Why is Peterson's algorithm unlikely to be useful in a modern implementation?

[WEB]

[6 marks]

In modern architecture, a CPU cache could screw up the mutual exclusion requirement. The problem is called cache-coherence, and it is possible that the cache used by Process 0 on CPU 0 sets flag [0] equal to true, while Process 1 on CPU 1 still thinks flag [0] is false. In this case, both critical sections would enter, and mutual exclusion fail [RACE condition].

Question 2.

There has been some criticism of Java's synchronized keyword and wait() method. Explain the potential problems with:

- (a) exceptions, [6 marks]
- (b) visibility of the locks, [6 marks]
- (c) fairness, and [6 marks]
- (d) use of the if(condition) wait() construct. [5 marks]

TURN OVER

Question 3.

How might you expect to use these x86 instructions in a library supporting multi-threaded execution?

(a) LOCK XCHG mem, reg

Exchanges the contents of the first and second operands.

[6 marks]

(b) LOCK CMPXCHG mem, reg

Compares the value in the accumulator register with the first operand. If the two values are equal, the second operand is loaded into the first operand. Otherwise, the first operand is loaded into the accumulator.

[6 marks]

(c) How would you use an *atomic integer* to provide thread-safe access to a bank balance? [6 marks]

(d) What is the ABA problem?

[5 marks]



Question 4.

A real-time system consists of three periodic processes, of periods 5ms, 19ms and 21ms and running times 2ms, 4ms and 3ms respectively.

- (a) Explain how you would attempt to schedule them onto a single processor using *rate monotonic scheduling*. What priorities would you assign? [5 marks]
- (b) Will all three processes meet their deadlines? [8 marks]
- (c) The system is modified so that the processes of period 5ms and 21ms both must access a single shared mutex which they each hold for $10\mu s$, once during their period. Will all processes still meet their deadlines? If not, how could you solve the problem? [10 marks]

(a) Task-based scheduling \rightarrow Fixed-Priority Scheduling (FPS)

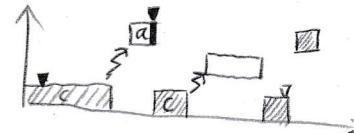
Task	T	C	Priority
a	5	2	3
b	19	4	2
c	21	3	1

(c) No

(b) Use utilization-based test

$$\sum_{i=1}^3 \left(\frac{C_i}{T_i} \right) = 0.753 < 2 \cdot (2^{\frac{1}{2}} - 1) = 0.83$$

As this test is sufficient, hence all three processes meet their deadlines.

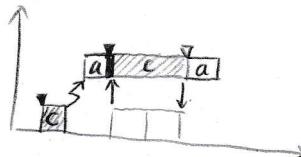


Assume that c is running, after several tick, task a preempts task c and contends for the mutex, task a is blocked.

Then c runs again and after one tick, task b preempts task c as it has higher priority. After competing for 4 tick, task misses its deadline.

Reduce priority inversion with a priority inheritance algorithm

When task a contends for the mutex and is blocked, task c will execute at the priority of the higher-priority task a



The system will be schedulable again with priority inheritance algorithm

TURN OVER

Question 5.

You are maintaining some embedded real-time code for a 486 processor and you see these two fragments. One is in a C file:

```
#define outb(port, data) \  
    asm volatile ("outb %%al,%%dx" : \  
    : "a" (data), "d" (port) )
```

and the other is in an assembler file:

```
section .text  
global _outb  
_outb: mov edx,[esp+4]  
       mov eax,[esp+8]  
       out dx,al  
       ret
```

- (a) What do they each do and how are they related? [6 marks]
- (b) Why do the `out` instructions differ? [4 marks]
- (c) Explain each line of the assembler code. [13 marks]

END OF PAPER

UNIVERSITY OF SOUTHAMPTON

COMP3006W1

SEMESTER 2 EXAMINATION 2013 - 2014

REAL-TIME COMPUTING AND EMBEDDED SYSTEMS

DURATION 90 MINS (1.5 Hours)

This paper contains 5 questions

Answer THREE questions.

An outline marking scheme is shown in brackets to the right of each question.

This examination is worth 70%.

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

Page 1 of 6

Question 1.

You are asked to implement a Readers-Writers application using a shared buffer in multi-threaded C11. You may use either the pthread or the C11 threading libraries. You should provide two functions, `read_int()` and `write_int(i)` which will remove and insert a single integer into a 16 deep FIFO buffer but will block if the buffer is empty or full respectively. Many threads may be using each of the two functions at one time.

- (a) Write correct C11 code for the `read_int()` and `write_int(i)` functions, and for the shared state.

[11 marks]

- (b) How will your code perform as the number of threads becomes large?

[6 marks]

- (c) Why is Peterson's algorithm unlikely to be useful in a modern implementation?

[6 marks]

D int read_int()
{
}
void write_int (int)
{
}
}

Question 2.

How might you expect to use these x86 instructions in a library supporting multi-threaded execution?

(a) LOCK XCHG mem, reg

Exchanges the contents of the first and second operands.

[6 marks]

(b) LOCK CMPXCHG mem, reg

Compares the value in the accumulator register with the first operand. If the two values are equal, the second operand is loaded into the first operand. Otherwise, the first operand is loaded into the accumulator.

[6 marks]

(c) How would you use an *atomic integer* to provide thread-safe access to a bank balance? [6 marks]

(d) What is the ABA problem?

[5 marks]

TURN OVER

Task	T	C	U
A	80	39	0.4875
B	40	9	0.225
C	20	5	0.25

As A, B, C have a common multiple period 20
They can be seen as a family thus they have a bound 1 ($N=1$)

Question 3.

- (a) Use a utilisation analysis to estimate whether the fixed priority technique will allow a set of 3 tasks with periods 80ms, 40ms and 20ms, which require computation times of 39ms, 9ms and 5ms respectively, to meet all their deadlines. Which priority would you assign to each task? [5 marks]

A
B
C
1
2
3

- (b) Explain the *Earliest Deadline First* (EDF) scheduling strategy. Would the tasks be able to meet all their deadlines under this strategy?

Priorities are assigned to processes based on the time remaining until the process's deadline

[5 marks]

$$\sum_{i=1}^3 U_i = 0.9625 < 1$$

- No deadline
- 7 (c) Draw a pair of Gantt charts to illustrate the scheduling of the tasks in section (a) under fixed priority and under EDF strategies. [4 marks]
- 7 (d) Which of the tasks from section (a) would in fact meet all their deadlines under fixed priority scheduling?

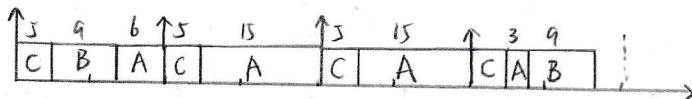
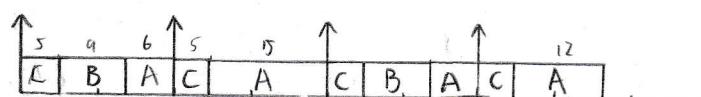
A B C

[4 marks]

- 7 (e) What demands do the three scheduling techniques

- Static table-driven cyclic scheduling, All tasks period must be a multiple of the minor cycle time
- Fixed priority pre-emptive scheduling, and The period is fixed
- Deadline scheduling, Can't be applied to multiprocessors.

place on a system kernel? [5 marks]



Question 4.

You are maintaining some embedded real-time code for a 486 processor and you see these two fragments. One is in a C file:

```
#define outb(port, data) \  
    asm volatile ("outb %%al,%%dx" : \  
    : "a" (data), "d" (port) )
```

and the other is in an assembler file:

```
section .text  
global _outb  
_outb: mov edx,[esp+4]  
       mov eax,[esp+8]  
       out dx,al  
       ret
```

- (a) What do they each do and how are they related? [6 marks]
- (b) Why do the out instructions differ? [4 marks]
- (c) Explain each line of the assembler code. [13 marks]

TURN OVER

Question 5.

The following blocks of C code attempt to implement a critical section which maintains mutual exclusion between two threads.

Declared global in both threads:

```
static int turn = 1;
```

Thread 1:

```
while (turn ==1);  
/* Critical region */  
turn = 1;
```

Thread 2:

```
while (turn==0);  
/* Critical region */  
turn = 0;
```

- (a) Using the FSP language of the Labelled Transition System Analyser or a suitable alternative, construct a model of the two interacting threads and the turn variable.

[9 marks]

- (b) Construct a suitable safety property which, when model-checked, will assure that mutual exclusion is maintained

[7 marks]

- (c) Construct a further model-checkable test which a conventional critical section would be expected to pass, but which this implementation fails.

[7 marks]

END OF PAPER

UNIVERSITY OF SOUTHAMPTON

COMP3215W1

SEMESTER 1 EXAMINATION 2014 - 2015

REAL-TIME COMPUTING AND EMBEDDED SYSTEMS

DURATION 120 MINS (2 Hours)

This paper contains 5 questions

Answer three questions.

An outline marking scheme is shown in brackets to the right of each question.

This examination contributes 70% of the marks for the module, the remainder coming from the laboratories.

University approved calculators MAY be used.

A foreign language translation dictionary (paper version) is permitted provided it contains no notes, additions or annotations.

6 page examination paper.

Question 1.

It is proposed that the existing electromechanical consumer electricity meters be replaced with *smart meters* that can be read remotely and which provide further added-value features including consumer information about energy consumption and, for those who don't pay their bills, remote disconnection.

Carefully explain the important features and capabilities of such a meter.
What technology would you use for implementation? [33 marks]

Question 2.

You are asked to discuss the schedulability under *fixed priority scheduling* of sets of simple periodic tasks on a single core processor with a real-time kernel supporting pre-emption. For each example, define each process in the set by its period, worst case execution time, and priority. Priorities are natural numbers with 1 being the lowest priority. Also describe the tests you applied, and their results for this example.

- (a) Construct a set of two tasks with a total utilisation of > 0.70 which can be shown to be schedulable by the use of a simple test of total utilisation. [12 marks]
- (b) Construct a set of tasks which fail the simple total utilisation test but pass Bini's *hyperbolic bound*. [9 marks]
- (c) Construct a set of tasks which fail the *hyperbolic bound* but are nevertheless schedulable. [12 marks]

Task	Period, T	Computation time, C	Utilisation, U
a	10	2	0.2
b	15	3	0.2

$$\sum_{i=1}^2 \left(\frac{C_i}{T_i} \right) = \frac{C_a}{T_a} + \frac{C_b}{T_b} = 0.4 < 2 \cdot (2^{\frac{1}{2}} - 1) = 0.828$$

Task	Period, T	C	U
a	99	39	0.3939
b	47	6	0.1276
c	15	4	0.2667

$$\sum_{i=1}^3 \left(\frac{C_i}{T_i} \right) = 0.7882 > 3 \cdot (2^{\frac{1}{2}} - 1) = 78.0\%$$

Utilisation test fails

$$\sum_{i=1}^3 \left(\frac{C_i}{T_i} + 1 \right) = 1.991 < 2$$

Hyperbolic bound satisfies.

Task	T	C	P	U
a	7	3	3	$\frac{3}{7}$
b	12	3	2	$\frac{1}{4}$
c	20	5	1	$\frac{1}{4}$

$$\frac{10}{7} \cdot \frac{5}{4} \cdot \frac{5}{4} = 2.332 < 2 \quad \text{Hyperbolic bound fails}$$

$$w_a^0 = C_a = 3$$

$$w_b^0 = C_b = 3$$

$$w_b^1 = 3 + \lceil \frac{3}{7} \rceil \cdot 3 = 6$$

$$w_b^2 = 3 + \lceil \frac{6}{7} \rceil \cdot 3 = 6 = w_b^1 \Rightarrow R_b = 6$$

$$w_c^0 = C_c = 5$$

$$w_c^1 = 5 + \lceil \frac{5}{7} \rceil \cdot 3 + \lceil \frac{5}{12} \rceil \cdot 3 = 11$$

$$w_c^2 = 5 + \lceil \frac{11}{7} \rceil \cdot 3 + \lceil \frac{11}{12} \rceil \cdot 3 = 14$$

$$w_c^3 = 5 + \lceil \frac{14}{7} \rceil \cdot 3 + \lceil \frac{14}{12} \rceil \cdot 3 = 17$$

$$w_c^4 = 5 + \lceil \frac{17}{7} \rceil \cdot 3 + \lceil \frac{17}{12} \rceil \cdot 3 = 20$$

$$w_c^5 = 5 + \lceil \frac{20}{7} \rceil \cdot 3 + \lceil \frac{20}{12} \rceil \cdot 3 = 20 \Rightarrow R_c = 20$$

$$R_c = T_c = 20$$

which means Task c just meets its deadline. **TURN OVER**

Question 3.

You may answer this question using either the Java programming language, or using pthreads or cthreads with the C programming language.

- (a) Write down a simple pattern using mutexes and condition variables that will suffice to prevent *race conditions* in multi-threaded software. [12 marks]
- (b) How might your pattern be inappropriate for use in a real-time system? [9 marks]
- (c) Using the *dining philosophers* example, explain the advantages of an alternative approach using semaphores. In what way does the structure of this code differ from that using condition variables? Ensure your system does not deadlock. [12 marks]

Sem ← mutex inside
claim first

O : 45m 18s

Question 4.

Your embedded system, based on a 32-bit Intel x86 architecture, is equipped with a memory-mapped byte wide output port at hexadecimal address FFFF0010 and an associated byte-wide input port at hexadecimal address FFFF0011.

- (a) How would you declare pointers to these two ports in order to make maximum usage of the C programming languages safety features?
[9 marks]
- (b) The status register indicates that the port is willing to accept another byte by setting its low order bit to a 1. The value of the other bits is unspecified. Write a suitable C function to output a byte to the port.
[10 marks]
- (c) What special precautions will you need to take if you wish to invoke C code from within an interrupt service routine? [14 marks]

TURN OVER

Question 5.

- (a) Explain the terms polled, interrupt driven and DMA in the context of a device driver in a real-time kernel. [12 marks]
- (b) How might each of these driver techniques affect the systems responsiveness to other devices? [9 marks]
- (c) What are the performance implications of the use of PVM and caches in a real-time system? Ignoring other caches, what would be the effect of a TLB miss on a data read in an iA386 architecture? [12 marks]

END OF PAPER

UNIVERSITY OF SOUTHAMPTON

COMP3215W1

SEMESTER 1 EXAMINATION 2015 - 2016

REAL-TIME COMPUTING AND EMBEDDED SYSTEMS

DURATION 120 MINS (2 Hours)

This paper contains 5 questions

Answer THREE questions out of FIVE.

An outline marking scheme is shown in brackets to the right of each question.

2013C, Ctt

This examination is worth 70%.



University approved calculators MAY be used.

A foreign language dictionary is permitted ONLY IF it is a paper version of a direct Word to Word translation dictionary AND it contains no notes, additions or annotations.

6 page examination paper.

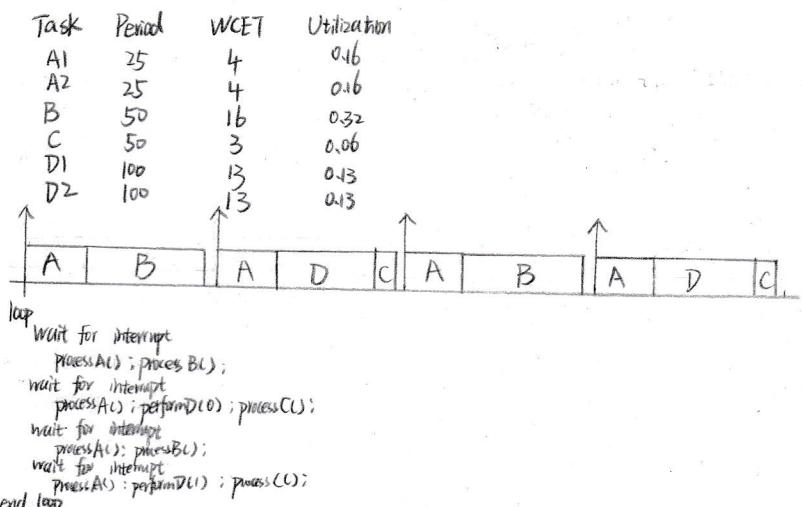
Question 1.

You are asked to schedule a uniprocessor system which is required to run four independent hard real-time tasks, A, B, C and D. Each must start and run to completion exactly once in each of its periods. The periods are 25ms, 50ms, 50ms and 100ms respectively. They have worst-case execution times (WCET) of 8ms, 16ms, 3ms and 26ms respectively. You have been shown the top-level code for two of the four tasks,

```
void processA() {
    int i;
    for (i=0; i<2; i++) {
        performA(i); } }
```



```
void processD() {
    int i;
    for (i=0; i<2; i++) {
        performD(i); } }
```



and, after further enquiries, you have determined that the WCET of the two inner functions do not depend on their arguments.

- Design a cyclic scheduler that will satisfy the real-time requirements.
[16 marks]
- Would the tasks be schedulable under earliest deadline first? Why?
 $\sum \frac{C_i}{T_i} = 0.16 \times 2 + 0.32 + 0.06 + 0.13 \times 2 = 0.96 < 1$
Thus the tasks can be schedulable under EDF
[6 marks]
- Would the tasks be schedulable under rate-monotonic scheduling?
Why? What priorities would you assign?
[6 marks]
- What extra demands would EDF and RMS place on the system?
[5 marks]

(a) As all the tasks have multiple minor period thus they are one task family

Use the Utilization-based schedulability tests

$$\sum \frac{C_i}{T_i} = 0.96 < 1 \quad (2^{-1}) = 1$$

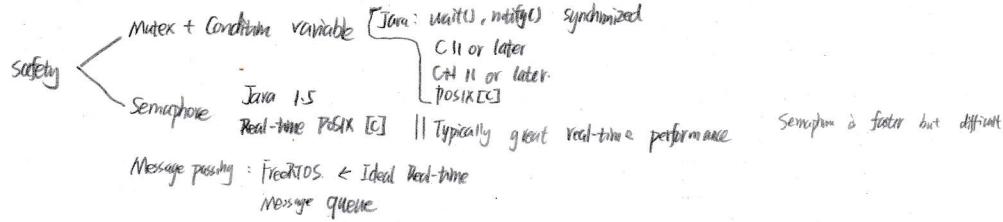
Thus the tasks are schedulable under rate-monotonic

RMS is preemptive which means that a high-priority task may be released during the execution of a lower-priority one.

EDF demands dynamic priority, where the runnable tasks are executed in the order determined by the absolute deadlines of the task. The next task to run being the one with shortest (nearest) deadline.

Priorities Assign:

	F	E	P
A	25	8	1
B	50	16	3
C	50	3	2
D	100	26	4



Question 2.

Five philosophers are seated round a circular table. In front of each is a plate of noodles. Between each pair of plates is a single chopstick. The philosophers wish to eat, but need to hold both the chopsticks that were beside their plate in order to do so. Having eaten, each philosopher will return both chopsticks to their original position; she will need to eat again later. Each philosopher is represented by a C (or C++) thread which runs the single function with signature `void philosopher(const int)`, where the parameter ranges over `0...4`. You are provided with two functions `void take(const int)` and `void drop(const int)` which take and drop the chopstick to the left. These functions show *undefined behaviour* if they are not invoked alternately on a given chopstick.

- (a) Use the language features of the 2013 C or C++ standards to write a correct `philosopher()` function which will allow eating to proceed. [27 marks]
- (b) Are there any *fairness* problems with your code? Why? [6 marks]

(b)

TURN OVER

Question 3.

You are asked to design a sensor module which receives an analogue signal over the range $0.5, \dots, 1.5$ Volts and transmits it over a digital bus to a controller every 1s. Ten modules are to be connected to a single bus.

- (a) How would you make design decisions about suitable module and bus architectures? What programming language and environment would you use? Give as many details as possible of your final design.

[19 marks]

- (b) How would your choices differ if the sensors were required to be wireless?

[14 marks]

Question 4.

What are the key advantages and disadvantages of using a RTOS such as FreeRTOS to implement a USB-based mouse? Which key FreeRTOS facilities would you use? [33 marks]

TURN OVER

Question 5.

You are presented with the following function:

```
unsigned int sumsq(const unsigned int n) {
    unsigned int i = 0;
    unsigned int result = 0;
    while (i < n) {
        i = i + 1;
        result = result + i * i; } ↗
    assert(result == n*(n+1)*(2*n+1)/6); ← control the induction variable
    assert(result == n*(n+1)*(2*n+1)/6);
    return result; }
```

An inductive hypothesis inside the loop
assert (result == i*(i+1)*(2*i+1)/6);
assert (i <= n); ← control the induction variable

- (a) What would be a suitable domain for the function? [4 marks]
- (b) How will the function first fail outside this domain? [4 marks]
- (c) Can you increase the domain without using other data types? [8 marks]
- (d) Create a suitable invariant and prove the function satisfies the assertion over its domain. [17 marks]

[Base case]
when $n=0$
 $0 \cdot 1 \cdot 1/6 = 0 == 0$

[Inductive step]
Let $n=i$
we have $y = i \cdot (i+1) \cdot \frac{2i+1}{6}$

- (a) $n \cdot (n+1) \cdot (2n+1) \leq$ maximum value of unsigned integer
- (b) The smallest integer value of n which leads to $n \cdot (n+1) \cdot (2n+1) > \text{MAX_Unsigned_INT}$
- (c) $n \cdot (n+1) \cdot (2n+1) / 6 = (n^3 + 3n^2 + n) \cdot \frac{1}{6} = (2n^3 + 3n^2 + n) \cdot \frac{1}{6} = \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n$
We can increase the domain by 6 times Assume $y = n(n+1)(2n+1)$

For previous function, y overflows first
For the current function, $\frac{1}{6}y$ overflows first

$$\begin{aligned}
 \text{let } n = i+1 \\
 y &= i \cdot (i+1) \cdot \frac{2i+1}{6} + (i+1)^2 \\
 &= (i^2 + i) \cdot \frac{2i+1}{6} + i^2 + 2i + 1 \\
 &= (i^2 + i) \cdot \left(\frac{1}{3}i + \frac{1}{6}\right) + i^2 + 2i + 1 \\
 &= \frac{1}{3}i^3 + \frac{1}{6}i^2 + \frac{1}{3}i^2 + \frac{1}{6}i + i^2 + 2i + 1 \\
 &= \frac{1}{3}i^3 + \frac{3}{2}i^2 + \frac{13}{6}i + 1 \\
 &= \frac{1}{6}(2i^3 + 9i^2 + 13i + 6)
 \end{aligned}$$

END OF PAPER

	T	C	P
a	80	40	1
b	40	10	2
c	20	5	3

RoundRobin

$$W_c^0 = 5 \quad W_b^0 = 10$$

$$W_b^1 = 10 + \lceil \frac{10}{20} \rceil \cdot 5 = 15$$

$$W_b^2 = 10 + \lceil \frac{15}{20} \rceil \cdot 5 = 15$$

$$R_b = 15$$

$$W_a^0 = 40$$

$$W_a^1 = 40 + \lceil \frac{40}{40} \rceil \cdot 10 + \lceil \frac{40}{20} \rceil \cdot 5 = 60$$

$$W_a^2 = 40 + \lceil \frac{60}{40} \rceil \cdot 10 + \lceil \frac{60}{20} \rceil \cdot 5 = 75$$

$$W_a^3 = 40 + \lceil \frac{75}{40} \rceil \cdot 10 + \lceil \frac{75}{20} \rceil \cdot 5 = 80$$

$$W_a^4 = 40 + \lceil \frac{80}{40} \rceil \cdot 10 + \lceil \frac{80}{20} \rceil \cdot 5 = 85$$

$$\sum_{i=1}^N \frac{C_i}{T_i} \leq N(2^{\frac{1}{N}} - 1)$$

$$\frac{N}{T} \left(1 + \frac{C_i}{T_i} \right) \leq 2$$

$$W_i^{n+1} = C_i + \sum_{j \in \text{dep}_i} \lceil \frac{W_i^n}{T_j} \rceil \cdot C_j$$

Semaphore

sem_t
sem_init

sem_wait
sem_post

reader writer

```

    ↓           ↓
    sem_wait(m);
    rc = rc + 1;
    if(rc == 1)
        sem_wait(&db);
    sem_post(m);
    → rc = rc - 1;
    if(rc == 0)
        semWait(&db);
    → post
  
```

C
 pthread_t
 pthread_create
 pthread_join
 pthread_mutex_t
 pthread_mutex_init
 pthread_mutex_lock
 pthread_mutex_unlock

++
 thread th[]
 thread [th[], func, data]
 th[] join
 mutex m
 condition_variable cv

Java

```

Object o = new Object();
synchronized(o){
    while (can't ...)
        o.wait();
}
  
```

```

unique_lock<mutex> ul(m);
while (can't ...)
    cv.wait();
  
```

producer invent_data(); sem_wait(&empty) sem_wait(&mutex) insert_data sem_post(&mutex) sem_post(&full)	consumer sem_wait(&full) sem_mutex mutex sem_wait(empty)
---	---