CONFIDENTIAL

SET A

UNIVERSITI KUALA LUMPUR
Malaysia France Institute

FINAL EXAMINATION
SEPTEMBER 2014 SESSION

SUBJECT CODE	:	**FSB33904**
SUBJECT TITLE	:	**REAL-TIME SYSTEMS**
LEVEL	:	**BACHELOR**
TIME / DURATION	:	**9.00 AM – 12.00 PM (3 HOURS)**
DATE	:	**08 JANUARY 2015**

INSTRUCTIONS TO CANDIDATES

1. **Please read the instructions given in the question paper CAREFULLY.**
2. **This question paper is printed on both sides of the paper.**
3. **Please write your answers on the answer booklet provided.**
4. **Answer should be written in blue or black ink except for sketching, graphic and illustration.**
5. **This question paper consists of TWO (2) sections. Section A and B. Answer all questions in Section A. For Section B, answer three (3) questions only.**
6. **Answer all questions in English.**

THERE ARE 6 PAGES OF QUESTIONS, EXCLUDING THIS PAGE.

SECTION A (Total: 40 marks)

INSTRUCTION: Answer ALL questions.
Please use the answer booklet provided.

Question 1

(a) Explain the difference between preemptive and non-preemptive scheduling.

(4 marks)

(b) Define scheduler and briefly explain THREE (3) important aspect to understand how real-time scheduling works.

(8 marks)

(c) Explain FOUR (4) conditions that cause the deadlock.

(8 marks)

Question 2

(a) Use a diagram to show the differences between one-way data communication for both interlocked and non-interlocked in using the message queues within an application.

(5 marks)

(b) Illustrate a *Single shared-resource-access* synchronization, a type of semaphores that is useful for coordinating access to a shared resource by using a diagram.

(5 marks)

(c) Explain a typical FSM (*finite state machine*) for THREE (3) task execution states, with brief description of each state transition.

(10 marks)

SECTION B (Total: 60 marks)

INSTRUCTION: Answer THREE (3) questions only.
Please use the answer booklet provided.

Question 3

(a) There are four tasks ready to run in the system, where the arrival time and service time of the task is given in Table 1.

Table 1: Property of four tasks.

TASK	ARRIVAL TIME (ms)	SERVICE TIME / BURST TIME (ms)
T1	0	2
T2	1	3
T3	4	4
T4	6	5

i. Assuming that each task has the same priority, construct a figure that shows how all tasks can be completed using `round robin and preemptive` scheduling with the given time slices of `2ms`.

(5 marks)

ii. Assuming that the same task was assigned with priorities as shown in Table 2, construct a figure that shows how all tasks can be completed using `preemptive priority-based` scheduling.

Table 2: Property of four tasks.

TASK	ARRIVAL TIME (ms)	SERVICE TIME / BURST TIME (ms)	PRIORITY
T1	0	2	1
T2	1	3	3
T3	4	4	4
T4	6	5	2

(5 marks)

(b) Suppose there are two processes, *A* and *B*, which share semaphores *R* and *S*. At the starting time, semaphore *R* = 0 and *S* = 1. Process *A* and *B* are entering the sections of code as shown in Figure 1.

PROCESS A	PROCESS B
`P(R);`	`Operation B.1;`
`P(S);`	`V(R);`
`Operation A.1;`	`Operation B.2;`
`V(S);`	`P(S);`
`Operation A.2;`	`Operation B.3;`
`V(R);`	`V(S);`

Figure 1: Two processes that uses the same semaphores.

Determine whether the following statements are TRUE or FALSE. Justify your answer.

i. Operation A.1 cannot start until operation B.1 is complete.

(2 marks)

ii. Operations A.1 and B.2 cannot execute simultaneously.

(2 marks)

iii. Operations A.1 and B.3 cannot execute simultaneously.

(2 marks)

iv. Operations A.2 and B.3 cannot execute simultaneously.

(2 marks)

v. The system may be deadlock.

(2 marks)

Question 4

(a) Consider the following 3-process concurrent program in Figure 2 that uses semaphores *S1*, *S2*, and *S3*. The semaphore operation, which are sometimes called "*wait*" and "*signal*", are denoted here with the classical notation of "*P*" and "*V*".

```
Process-1            Process-2            Process-3
L1:  P(S3);          L2: P(S1);           L3:  P(S2)
     type ("T");         type ("U");           type ("B")
     V(S2);              V(S3);                V(S1)
     goto L1;            goto L2;              goto L3;
```

Figure 2: Three concurrent processes.

a. Identify and give the minimum initial values that can be given to the semaphores, so that the processes cooperate to print the string "BUTBUTBUTBU" (tell which value is to be used for which semaphore). Justify your answer.

(5 marks)

b. Suppose the initial values are S1=0, S2=0, S3=0. Determine whether "BBTTUTT" is a possible output sequence or not. Justify your answer.

(5 marks)

(b) Assume that there are three LEDs connected to port B0 to B2 on AVR Butterfly microcontroller. Write a C function on AVR Studio 4 called `LED_Blinking()`, that will blink each LED simultaneously for every 1s if the limit switch (connected to port D0) is triggered.

(Port B is declared as output and port D as input, with all initial value is set on 1)

(10 marks)

Question 5

(a) There are three periodic tasks ready to run in the real-time systems, where the properties of each tasks is as given in Table 3. Using the rate monotonic algorithm, check whether the task is schedulable or not.

Table 3: Property of three tasks.

TASK	EXECUTION TIMES (C_i)	PERIOD LENGTH (T_i)
A	2	6
B	2	8
C	2	12

(10 marks)

(b) There are three periodic tasks ready to run in the real-time systems, where the properties of each tasks is as given in Table 4. Using the rate monotonic algorithm, check whether the task is schedulable or not.

Table 4: Property of three tasks.

TASK	EXECUTION TIMES (C_i)	PERIOD LENGTH (T_i)
D	3	5
E	1	8
F	1	10

(10 marks)

Question 6

Table 5: Property of three tasks.

TASK	EXECUTION TIMES (C_i)	PERIOD LENGTH (T_i)
X	1	4
Y	2	6
Z	3	10

(a) Verify the schedulability and construct the schedule according to the rate monotonic algorithm for the given set of periodic tasks on Table 5.

(10 marks)

(b) In a particular application, there are three interacting processes, which used two shared semaphores, *U* and *V* as shown in Figure 3. Within each process, the statements are executed sequentially, but statements from different processes can be interleaved in any order that is consistent with the constraints imposed by the semaphores. Assume that once execution begins, the processes will be allowed to run until all three processes are stuck in a *wait()* statement, at which point execution is halted.

```
Semaphore U = 3;
Semaphore V= 0;

Process-1            Process-2          Process-3
L1:  wait (U)        L2: wait (V)       L3:  wait(V)
     type ("C")          type ("A")          type ("D")
     signal (V)          type ("B")          goto L3
     goto L1             signal (V)
                         goto L2
```

Figure 3: Three interacting processes that use two shared semaphores, *U* and *V*.

(b) When this set of processes runs, compute the smallest number of 'A' that might be printed. Justify your answer.

(5 marks)

(c) When this set of processes runs, determine whether 'CABABDDCABCABD' is a possible output sequence or not. Justify your answer.

(5 marks)

END OF QUESTIONS