Document No : UniKL MFI_SD_AC41
Revision No: 02
Effective Date: 01 December 2008

CONFIDENTIAL

SET A

UNIVERSITI KUALA LUMPUR
Malaysia France Institute

FINAL EXAMINATION
JANUARY 2014 SESSION

SUBJECT CODE	:	**FSB33904**
SUBJECT TITLE	:	**REAL-TIME SYSTEMS**
LEVEL	:	**BACHELOR**
TIME / DURATION	:	**x.xx pm – x.xx pm (3 HOURS)**
DATE	:	

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) Discuss TWO (2) types of tasks.

(6 marks)

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

(8 marks)

(c) Explain the characteristics and concept of *periodic* and *aperiodic* real-time tasks with the help of suitable example for each concept.

(6 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.

(6 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.

(6 marks)

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

(8 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) Construct a typical FSM (*finite state machine*) for task execution states, with brief description of state transitions.

(9 marks)

(b) Consider the two processes in Figure 2, where A and B are arbitrary computations. Assume that an unbounded number of processes are invoking either of the processes and the initial values are:

$$S1 = S2 = d = 1,\ c1 = c2 = 0.$$

Process P1	Process P2

P(S1);
c1 = c1 + 1;
if(c1 == 1) P(d);
V(S1);
A;
P(S1);
c1 = c1 - 1;
if(c1 == 0) V(d);
V(S1);
	P(S2);
	c2 = c2 + 1;
	if(c2 == 1) P(d);
	V(S2);
	B;
	P(S2);
	c2 = c2 - 1;
	if(c2 == 0) V(d);
	V(S2);

Figure 2: Pair of processes that share a resource.

i. Evaluate how many invocations of the computation A can proceed concurrently and determine the values of $S1$, $c1$, and d at that time.

(3 marks)

ii. If another process is currently attempting to invoke A, determine the current values of $S1$ and $c1$.

(2 marks)

iii. While A is running, compute how many invocations of B can proceed concurrently and determine the values of $S2$, $c2$, and d at that time.

(4 marks)

iv. Check whether A and B can lead to starvation or not. Justify your answer.

(2 marks)

Question 5

Table 3 describes four tasks with their respective worst-case execution times and periods of execution. Answer the questions based on the given property for each tasks.

Table 3: Property of four tasks.

TASK	EXECUTION TIMES (C_i)	PERIOD LENGTH (T_i)
X	10	100
P	20	50
S	20	150
G	25	80

(a) Calculate the total CPU utilization of tasks.

(4 marks)

(b) Rearrange the task in rate monotonic order with the highest priority first.

(2 marks)

(c) Based on new ordering (your answer in question 5(b)), compute whether the tasks with the two highest priorities is schedulable or not.

(4 marks)

(d) Explain whether all four tasks are schedulable or not by using both basic RMA (*Rate Monotonic Analysis*) schedulability test and a timeline figure for this task set.

(10 marks)

Question 6

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.

(a) Assuming execution is eventually halted, identify the number of 'C' are printed when the set of processes runs. Justify your answer.

(5 marks)

(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)

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

(5 marks)

END OF QUESTIONS