NATIONAL UNIVERSITY OF SINGAPORE

CG2271 – REAL-TIME OPERATING SYSTEMS

(Semester 2: AY2016/17)

Time Allowed : 2 Hours

INSTRUCTIONS TO STUDENTS

1. Please write your Student Number only. Do not write your name.
2. This assessment paper contains **THREE** questions and comprises **TWENTY ONE** printed pages.
3. Students are required to answer **ALL** questions.
4. Students should write the answers on the question booklet.
5. This is an OPEN BOOK assessment.
6. This assessment permits electronic calculators.

STUDENT NO: _____

This portion is for examiner's use only

Question	Marks	Remarks
Q1 [20 marks]		
Q2 [20 marks]		
Q3 [20 marks]		
Total [60 marks]		

QUESTION 1: Real-Time Scheduling [20 marks]

(A) Four periodic, independent tasks have the following timings. Assume the all the tasks are ready at time zero and the period is equal to the deadline.

Task	WCET	Period
T1	2	10
T2	1	5
T3	1	20
T4	2	4

(i) What is the hyper-period for the task set? [1 mark]

(ii) What is the utilization of the task set? [1 mark]

(iii) Assign priorities to the tasks according to the Rate Monotonic Scheduling (RMS) policy. Assume priority level 1 represents the lowest priority and priority level 4 represents the highest priority. [2 marks]

Task	WCET	Period	Priority
T1	2	10	
T2	1	5	
T3	1	20	
T4	2	4	

(iv) Prove the existence or non-existence of a feasible RMS schedule for the task set using utilization bound and critical instant analysis. [3 marks]

(v) If the task set is schedulable under RMS, then show the RMS schedule in the following table for the first 20 time units with minimum number of context switches. Otherwise, show the RMS schedule till a task misses its deadline. [3 marks]

1	2	3	4	5	6	7	8	9	10

11	12	13	14	15	16	17	18	19	20

(vi) Is the task set schedulable using EDF policy? Justify your answer. [1 mark]

(vii) If the task set is schedulable under EDF, then draw the EDF schedule in the following table for the first 20 time units with minimum number of context switches. Otherwise, show the EDF schedule till one of the tasks misses its deadline. [3 marks]

1	2	3	4	5	6	7	8	9	10

11	12	13	14	15	16	17	18	19	20

(viii) Assume that the tasks T1 and T4 both need to lock the same resource R for the entire duration of their execution. The tasks T2 and T3, on the other hand, do not require lock for any resource. Can the execution of the task set lead to priority inversion under Rate Monotonic Scheduling policy? Justify your answer. If priority inversion is possible for the task set, explain how the priority inversion problem can be solved for this task set. [3 marks]

(ix) Is it easy to create a cyclic executive schedule for the task set? Justify your answer. [1 mark]

(B) In a real-time system, suppose we have tasks with different periods; deadline equals period; and utilization is less than 1. Also, all the tasks need the same resource. Each task locks the resource in the beginning of execution and unlocks the resource at the end of execution. Explain whether RMS and EDF scheduling policies make sense in this scenario. [2 marks]

QUESTION 2: Synchronization [20 marks]

(A) Assume you have four tasks to produce H_2O (water) molecules from Hydrogen (H) and Oxygen (O) atoms. Two Hydrogen tasks (Task_H1 and Task_H2) generate Hydrogen atoms and put them in a circular queue Queue_H of length 2L. The Oxygen task (Task_O) generates Oxygen atoms and puts them in a circular queue Queue_O of length L. The Water task (Task_W) needs two Hydrogen atoms and one Oxygen atom to generate a water molecule. The tasks are all periodic in nature and each instance generates one hydrogen atom, oxygen atom, or water molecule, respectively.

Write the pseudo code for the four tasks using Semaphores and Mutex to implement synchronization. The boundary conditions for the queues (full or empty) should be handled properly from scratch (i.e., you cannot rely on enqueuer and dequeuer functions.) You may use P( ) and V( ) primitives as presented in the lecture notes. Define and initialize all the semaphores correctly. [4 marks]

BLANK PAGE

(B) Assume you have four tasks to produce H_2O (water) molecules from Hydrogen (H) and Oxygen (O) atoms. Two Hydrogen tasks (Task_H1 and Task_H2) generate Hydrogen atoms. The Oxygen task (Task_O) generates Oxygen atoms. The Water task (Task_W) needs two Hydrogen atoms and one Oxygen atom to generate a water molecule. The tasks are all periodic in nature and each instance generates one hydrogen atom, oxygen atom, or water molecule, respectively.

Write the pseudo code for the four tasks using message passing. [2 marks]

(C) Consider a semaphore S initialized to value 2 in FreeRTOS. Now consider the following sequence of P() and V() operations on semaphore S from three different tasks T1, T2, T3 where priority (T1) > priority (T2) > priority (T3). Assume that all the tasks are ready in the beginning. Fill up the table below to show the state of the semaphore variable and the task state changes after each operation. The first row has been filled up for you. [6 marks]

Operation	**Value of S after operation**	**Semaphore queue after operation**	**State change of any task after operation**	**Any other comments**
T1: P(S)	1	Nil	None	
T2: P(S)				
T2: P(S)				
T1: P(S)				
T3: V(S)				
T3: V(S)				
T3: V(S)				

(D) In the context of real-time systems, explain why busy (active) waiting is not used when a task is unable to acquire a lock and instead the task is moved into blocked state (passive waiting).

[2 marks]

(E) In the lecture, we presented the producer-consumer example with semaphores as shown below. [2 + 2 = 4 marks]

```
#define N 8
int buf[N];
int in=0, out = 0;
sem_t mutex = 1;
sem_t empty = N; // number of empty slots
sem_t full  = 0; // number of full slots


void prod (int d){
  P(empty);
  P(mutex);
      buf[in] = data;
      in = (in + 1) % N;
  }
  V(mutex);
  V(full);
}
```

```
void cons(void){
   int c;
   P(full);
   P(mutex);
      c = buf[out];
      out = (out + 1) % N;
   V(mutex);
   V(empty);
   return c;
}
```

The questions in the next two pages deal with variations of this producer-consumer example.

(i) Consider the variation below. Assume that the definitions and initializations of the variables remain the same. Will this solution work? Justify your answer.

```
void prod (int d){
  P(mutex);
  P(empty);
      buf[in] = data;
      in = (in + 1) % N;
  }
  V(full);
  V(mutex);
}
```

```
void cons(void){
    int c;
    P(mutex);
    P(full);
       c = buf[out];
       out = (out + 1) % N;
    V(empty);
    V(mutex);
    return c;
}
```

(ii) Now consider the following variation. Assume that the definitions and initializations of the variables remain the same. Will this solution work? Justify your answer.

```
void prod (int d){
  P(mutex);
  P(empty);
      buf[in] = data;
      in = (in + 1) % N;
  }
  V(full);
  V(mutex);
}
```

```
void cons(void){
    int c;
    P(full);
    P(mutex);
       c = buf[out];
       out = (out + 1) % N;
    V(mutex);
    V(empty);
    return c;
}
```

(F) Consider the following two tasks running concurrently in a shared memory (all variables are shared between the two tasks).

Task A	Task B

for (i=1; i<=2; i++)
  x = x+1;
	for (i=1; i<=2; i++)
	  x = x+1;

Assume that x is initialized to zero. What are the possible values of x after both threads have completed execution. [2 marks]

QUESTION 3: Memory Management [20 marks]

(A) A computing system has 32-bit (4GB) virtual address space. It has 512MB of physical memory and 256KB page size. [6 marks]

(i) How many bits are there in the page offset?

(ii) How many virtual pages are there in the address space?

(iii) How many physical pages are there in the address space?

(iv) How many bits are there in the virtual page number?

(v) How many bits are there in the physical page number?

(vi) How much memory would be required for the page table?

(B) State at least one benefit and one drawback of increasing page size? [2 marks]

(C) Assume that for a given system, virtual addresses are 40 bits long and physical addresses are 30 bits long. The page size is 8KByte. The TLB in the address translation path has 128 entries. What is the maximum number of distinct virtual addresses that can be quickly translated by the TLB at any point in time? [2 marks]

(D) Consider a virtual memory system with a TLB. Assume that the entire page table and all the pages are in the physical memory. It takes 10 milliseconds to search the TLB and 80 milliseconds to access the physical memory. If the TLB hit ratio is 0.6, what is the effective memory access time (in milliseconds)? [2 marks]

(D) Consider the following code fragment.

```
int * f (int buffer_size ) {
    static int *buffer;
    buffer = (int *) malloc (buffer_size);
    return buffer;
}

int main ( ){
   int *buffer_pointer;
   buffer_pointer = f (100);
}
```

Where (text, data, BSS, stack, heap) would the variables be allocated? [2 marks]

Variable	**Location in process memory**
buffer_size	
buffer pointer	
100 buffer elements	
buffer	

(E) (i) Assume there is an initial 512 byte segment where memory is allocated using the Buddy system with all allocations in power of 2 bytes. Illustrate how the following memory requests are handled (a) request 100 bytes (b) request 60 bytes (c) request 260 bytes (d) request 36 bytes (e) free 60 bytes (g) free 36 bytes. If a memory allocation request cannot be satisfied, the system returns error and move on to the next request (i.e., the allocation request is ignored). Perform coalescing whenever possible. [3 marks]

(ii) Given the memory allocation and deallocation requests in Question 3E(i) and the same 512 byte memory segment, would it be better to use free list instead of buddy system? Justify your answer. [2 marks]

(F) State an advantage of the buddy system over multiple free lists. [1 mark]

END OF ASSESSMENT PAPER