

June 3rd, 2023

Course Code: CSE411
Real-Time & Embedded Systems Design

Time: 2 Hours

The Exam Consists of **5 Questions in 4 Pages**

Total Marks: 40 Marks

تنبيه هام جدا: يجب على كل طالب الحل فقط هنا في ورق الاسئلة ولن يلتفت لأي إجابة في الكراسة المرفقة والتي تحتوي على البار كود الخاص بكل طالب

Important Rules:

- Having a (mobile -Smart Watch- earphones) inside the examination hall is forbidden and is considered as a cheating behavior.
- It is forbidden to have any references, notes, books, or any other materials even if it is not related to the exam content with you in the examination hall.
- This is an answer sheet.
- Assume missing data if any – Read it all well, at first.

تعليمات هامة

- حيازة (المحمول- الساعات الذكية - سماعة الأذن) داخل لجنة الامتحان يعتبر حالة غش تستوجب العقاب.
- لايسمح بدخول أي كتب أو ملازم أو أوراق داخل اللجنة والمخالفة تعتبر حالة غش.
- هذه ورقة إجابة أيضا - أقرأها أولا جيدا – أفترض الناقص إن وجد

Question 1: (5 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. In the given table, order the first 5 break points to be hit, when GO is pressed.

1 st Break Point Hit	2 nd Break Point Hit	3 rd Break Point Hit	4 th Break Point Hit	5 th Break Point Hit

```

82 int main( void )
83 {
84     vSemaphoreCreateBinary( xBinarySemaphore );
85     if( xBinarySemaphore != NULL )
86     {
87         prvSetupSoftwareInterrupt();
88         xTaskCreate( vHandlerTask, "Handler", 240, NULL, 3, NULL );
89         xTaskCreate( vPeriodicTask, "Periodic", 240, NULL, 1, NULL );
90         vTaskStartScheduler();
91     }
92 }
93 static void vHandlerTask( void *pvParameters )
94 {
95     xSemaphoreTake( xBinarySemaphore, 0 );
96     for( ;; )
97     {
98         xSemaphoreTake( xBinarySemaphore, portMAX_DELAY );
99     }
100 }
101 static void vPeriodicTask( void *pvParameters )
102 {
103     for( ;; )
104     {
105         vTaskDelay( 500 / portTICK_RATE_MS );
106         mainTRIGGER_INTERRUPT();
107     }
108 }
109 void vSoftwareInterruptHandler( void )
110 {
111     portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
112     xSemaphoreGiveFromISR( xBinarySemaphore, &xHigherPriorityTaskWoken );
113     mainCLEAR_INTERRUPT();
114 }

```

Question 2: (10 Marks)

For the following FreeRTOS based application snippet, in the given table, order the first 20 break points to be hit, when GO is pressed. Assume any missing data.

1 st Break Point Hit	2 nd Break Point Hit	3 rd Break Point Hit	4 th Break Point Hit	5 th Break Point Hit
6 th Break Point Hit	7 th Break Point Hit	8 th Break Point Hit	9 th Break Point Hit	10 th Break Point Hit
11 th Break Point Hit	12 th Break Point Hit	13 th Break Point Hit	14 th Break Point Hit	15 th Break Point Hit
16 th Break Point Hit	17 th Break Point Hit	18 th Break Point Hit	19 th Break Point Hit	20 th Break Point Hit

```

73 int main( void ){
74     xPrintQueue = xQueueCreate( 5, sizeof( char * ) );
75     xTaskCreate( prvPrintTask1, "Print1", 240, ( void * ) 0, 1, NULL );
76     xTaskCreate( prvPrintTask2, "Print2", 240, ( void * ) 1, 2, NULL );
77     xTaskCreate( prvSpool, "Spooler", 240, NULL, 3, NULL );
78     vTaskStartScheduler();
79     static void prvSpool( void *pvParameters )
80     {
81         char *pcMessageToPrint;
82         for( ;; ){
83             xQueueReceive( xPrintQueue, &pcMessageToPrint, portMAX_DELAY );
84             printf( pcMessageToPrint );
85         }
86     }
87     void vApplicationTickHook( void ){
88         static int iCount = 0;
89         portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
90         iCount++;
91         if( iCount >= 200 ){
92             xQueueSendToFrontFromISR( xPrintQueue, &(amp; pcStringsToPrint[ 2 ] ), &xHigherPriorityTaskWoken );
93             iCount = 0;
94         }
95     }
96     static void prvPrintTask1( void *pvParameters ){
97         int iIndexToString;
98         iIndexToString = ( int ) pvParameters;
99         for( ;; ){
100             xQueueSendToBack( xPrintQueue, &(amp; pcStringsToPrint[ iIndexToString ] ), 0 );
101             vTaskDelay( 100 );
102         }
103     }
104     static void prvPrintTask2( void *pvParameters ){
105         int iIndexToString;
106         iIndexToString = ( int ) pvParameters;
107         for( ;; ){
108             xQueueSendToBack( xPrintQueue, &(amp; pcStringsToPrint[ iIndexToString ] ), 0 );
109             vTaskDelay( 300 );
110         }
111     }

```

Question 3: (8 Marks)

For the following FreeRTOS based application snippet, consider one time slice elapsed during the dummy software delay loop. In the following table, order the first 16 break points to be hit, when GO is pressed. Assume any missing data.

```

58 int main( void )
59 {
60     xQueue = xQueueCreate( 1, sizeof( long ) );
61     xTaskCreate( vSenderTask, "Sender", 240, ( void * ) 100, 2, NULL );
62     xTaskCreate( vReceiverTask, "Receiver", 240, NULL, 1, NULL );
63     vTaskStartScheduler();
64 }
65 static void vSenderTask( void *pvParameters )
66 {
67     long lValueToSend;
68     portBASE_TYPE xStatus;
69     lValueToSend = ( long ) pvParameters;
70     for( ;; )
71     {
72         xStatus = xQueueSendToBack( xQueue, &lValueToSend, 1 );
73         taskYIELD();
74     }
75 }
76 static void vReceiverTask( void *pvParameters )
77 {
78     long i;
79     long lReceivedValue;
80     portBASE_TYPE xStatus;
81     for( ;; )
82     {
83         if( uxQueueMessagesWaiting( xQueue ) != 0 )
84         {
85             vPrintString( "Hello" );
86         }
87         for( i = 0; i < 3000; i++ );
88         xStatus = xQueueReceive( xQueue, &lReceivedValue, 0 );
89     }
90 }

223 void xPortSysTickHandler( void )
224 {
225     unsigned long ulDummy;
226     #if configUSE_PREEMPTION == 1
227         *(portNVIC_INT_CTRL) = portNVIC_PENDSVSET;
228     #endif
229     ulDummy = portSET_INTERRUPT_MASK_FROM_ISR();
230     {
231         vTaskIncrementTick();
232     }
233     portCLEAR_INTERRUPT_MASK_FROM_ISR( ulDummy );
234 }

```

1 st Break Point Hit	2 nd Break Point Hit	3 rd Break Point Hit	4 th Break Point Hit	5 th Break Point Hit
6 th Break Point Hit	7 th Break Point Hit	8 th Break Point Hit	9 th Break Point Hit	10 th Break Point Hit
11 th Break Point Hit	12 th Break Point Hit	13 th Break Point Hit	14 th Break Point Hit	15 th Break Point Hit
16 th Break Point Hit				

Question 4: (7 Marks)

Answer the following Context-Switch questions: (Only Shortest Answers)

- A. What is the technical name of “ARM HW optimization” important in context switching
- B. When should PENDSV-interrupt be triggered?
- C. Which is of higher priority; SYSTICK or PENDSV?
- D. How many bits assigned for interrupt priority?
- E. What are the main two things necessary to do in SYSTICK ISR?
- F. What data alignment that should be taken care of during RTOS-stack-initialization?
- G. What are registers-range that should not be stacked while switch-context from main?

Question 5: (10 Marks)

Assume a FreeRTOS “Heap_2.c” based application that:

- A. Create Task 1 & Task 2 that need from heap memory 100 and 200 bytes respectively
- B. Delete Task 1 & Task 2
- C. Create Task 3 that needs from heap memory 300 Bytes
- D. Create Task 4 that needs from heap memory 150 Bytes
- E. Create Task 5 that needs from heap memory 50 Bytes

Using the following Heap memory relevant sketches, show how FreeRTOS Heap looks like after each of the previous steps.

A	B	C	D	E
				