September 15th,2021	Course Code: CSE 347	Time: 2 Hours
	Embedded System Design	

The Exam Consists of **5 Questions in 5 Pages** **Total Marks: 40 Marks**

Important Rules:	تعليمات هامة
• 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.	• 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: (8 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 8 break points (PB) to be hit, when GO is pressed. At each PB, Define the states of all tasks.**

Break Point at Line?	State of Sender 1	State of Sender 2	State of Receiver

```
59    int main( void )
60  {
61      xQueue = xQueueCreate( 1, sizeof( long ) );
62      if( xQueue != NULL )
63      {
64          xTaskCreate( vSenderTask,"SENDER1", 240, ( void * ) 100, 2, NULL );
65          xTaskCreate( vSenderTask,"SENDER2", 240, ( void * ) 200, 3, NULL );
66          xTaskCreate( vReceiverTask, NULL, 240, NULL, 1, NULL );
67          vTaskStartScheduler();
68      }
69      for( ;; );
70  }
71  static void vSenderTask( void *pvParameters )
72  {
73  long lValueToSend;
74      lValueToSend = ( long ) pvParameters;
75      for( ;; )
76      {
77          xQueueSendToBack( xQueue, &lValueToSend, 100 / portTICK_RATE_MS );
78          taskYIELD();
79      }
80  }
81  static void vReceiverTask( void *pvParameters )
82  {
83  long lReceivedValue;
84      for( ;; )
85      {
86          xQueueReceive( xQueue, &lReceivedValue, 100 / portTICK_RATE_MS );
87          vPrintStringAndNumber( "Received = ", lReceivedValue );
88      }
89  }
90  /*------------------------------------------------------------*/
```

Question 2: (8 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes.
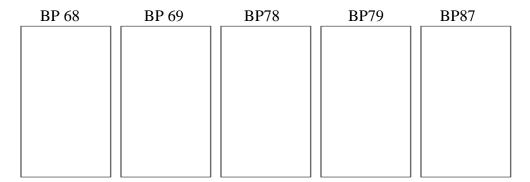**Sketch task switching timing diagram for the first 200ms approximately.**

```
82   int main( void )
83  {
84       vSemaphoreCreateBinary( xBinarySemaphore );
85       if( xBinarySemaphore != NULL )
86       {
87           prvSetupSoftwareInterrupt();
88           xTaskCreate( vHandlerTask, NULL , 240, NULL,3, NULL );
89           xTaskCreate( vPeriodicTask, NULL, 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           vPrintString( "Handler task - Processing event.\n" );
100      }
101  }
102  static void vPeriodicTask( void *pvParameters )
103  {
104      for( ;; )
105      {
106          vTaskDelay( 100 / portTICK_RATE_MS );
107          vPrintString( "Periodic task - About to generate an interrupt.\n" );
108          mainTRIGGER_INTERRUPT();
109          vPrintString( "Periodic task - Interrupt generated.\n\n" );
110      }
111  }
112  void vSoftwareInterruptHandler( void )
113  {
114  portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
115      xSemaphoreGiveFromISR( xBinarySemaphore, &xHigherPriorityTaskWoken );
116      mainCLEAR_INTERRUPT();
117      portEND_SWITCHING_ISR( xHigherPriorityTaskWoken );
118  }
```

Question 3: (8 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **Assume "heap2" FreeRTOS Heap memory algorithm is used. Show in the figures below how Heap memory looks like at designated break points when first hit.**

```c
56   int main( void )
57   {
58     xTaskCreate ( vTask1,NULL, 240, NULL, 1, NULL );
59     vTaskStartScheduler();
60     for( ;; );
61   }
62   /*------------------------------------------------------*/
63   void vTask1( void *pvParameters )
64   {
65     for( ;; )
66     {
67       vPrintString( "Task1 is running\n" );
68       xTaskCreate ( vTask2,NULL, 240, NULL, 2, NULL );
69       vTaskDelay( 100 / portTICK_RATE_MS );
70     }
71   }
72   /*------------------------------------------------------*/
73   void vTask2( void *pvParameters )
74   {
75     for( ;; )
76     {
77       vPrintString( "Task2 is running\n" );
78       xTaskCreate ( vTask3,NULL, 240, NULL, 3, NULL );
79       vTaskDelete (NULL);
80     }
81   }
82   /*------------------------------------------------------*/
83   void vTask3( void *pvParameters )
84   {
85     for( ;; )
86     {
87       vPrintString( "Task3 is running\n" );
88       vTaskDelete (NULL);
89     }
90   }
91
92
```

BP 68	BP 69	BP78	BP79	BP87

Question 4: (8 Marks)

For the following FreeRTOS based application snippet, **order the first 7 breakpoints** (in designated table) hit while debugging.

BP Order	BP1	BP2	BP3	BP4	BP5	BP6	BP7
Line Number							

```
53   int main( void )
54  {
55      xTaskCreate( vTask1, NULL, 240, NULL, 1, NULL );
56      xTaskCreate( vTask2, NULL, 240, NULL, 2, &xTask2Handle );
57      vTaskStartScheduler();
58      for( ;; );
59  }
60   void vTask1( void *pvParameters )
61  {
62   unsigned portBASE_TYPE uxPriority;
63      unsigned ux;
64      uxPriority = uxTaskPriorityGet( NULL );
65      for( ;; )
66      {
67          vTaskPrioritySet( xTask2Handle, ( uxPriority + 1 ) );
68          ux++;
69      }
70  }
71   void vTask2( void *pvParameters )
72  {
73   unsigned portBASE_TYPE uxPriority;
74      unsigned ux;
75      uxPriority = uxTaskPriorityGet( NULL );
76      for( ;; )
77      {
78          vTaskPrioritySet( NULL, ( uxPriority - 2 ) );
79          ux++;
80      }
81  }
```

Question 5: (8 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **Write down expected first 9 printed messages.**

```c
60  int main( void )
61  {
62      xMutex = xSemaphoreCreateMutex();
63      if( xMutex != NULL )
64      {
65          xTaskCreate( Task1, NULL, 240, NULL, 3, NULL );
66          xTaskCreate( Task2, NULL, 240, NULL, 2, NULL );
67          xTaskCreate( Task3, NULL, 240, NULL, 1, NULL );
68          vTaskStartScheduler();
69      }
70  }
71  void Task1(void *pvParameters)
72  {
73      while(1)
74      {
75      vTaskDelay( 100 / portTICK_RATE_MS );
76      xSemaphoreTake( xMutex, portMAX_DELAY );
77      printf( "Task 1 is running\n" );
78      xSemaphoreGive( xMutex );
79      }
80  }
81  void Task2(void *pvParameters)
82  {
83      while(1)
84      {
85      vTaskDelay( 50 / portTICK_RATE_MS );
86      printf( "Task 2 is running\n" );
87      }
88  }
89  void Task3(void *pvParameters)
90  {
91      int i,j;
92      while(1)
93      {
94      xSemaphoreTake( xMutex, portMAX_DELAY );
95      printf( "Task 3 is running\n" );
96      for(i=0;i<10000000;i++)
97          j++;
98      xSemaphoreGive( xMutex );
99      }
100 }
```