AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
ICHEP; MCT-CESS
Fall 2022

January 26th,2022 **Course Code: CSE347-CSE345** **Time: 2 Hours**

Embedded System Design & Real-Time Embedded System Design

The Exam Consists of **5 Questions in 5 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: (7 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	57	78	67	79	78	68	67

```
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 2: (10 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 10 break points to be hit, when GO is pressed.**

1st Break Point Hit	2nd Break Point Hit	3rd Break Point Hit	4th Break Point Hit	5th Break Point Hit
95	98	106	108	115

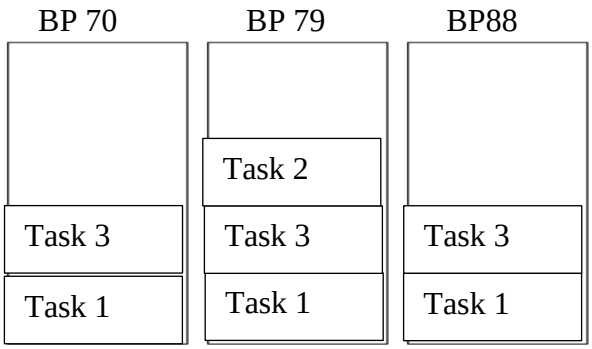
6th Break Point Hit	7th Break Point Hit	8th Break Point Hit	9th Break Point Hit	10th Break Point Hit
116	109	106	99	98

```
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 }
```

Question 3: (6 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 after starting the schedular.**

```
54  /*-----------------------------------------------------------*/
55
56  int main( void )
57  {
58      xTaskCreate( vTask1,NULL, 240, NULL, 2, NULL );
59      xTaskCreate( vTask3,NULL, 240, NULL, 3, NULL );
60      vTaskStartScheduler();
61      for( ;; );
62  }
63  /*-----------------------------------------------------------*/
64  void vTask1( void *pvParameters )
65  {
66      for( ;; )
67      {
68          vPrintString( "Task1 is running\n" );
69          xTaskCreate( vTask2,NULL, 240, NULL, 4, NULL );
70          vTaskDelay( 200 / portTICK_RATE_MS );
71      }
72  }
73  /*-----------------------------------------------------------*/
74  void vTask2( void *pvParameters )
75  {
76      for( ;; )
77      {
78          vPrintString( "Task2 is running\n" );
79          vTaskDelete(NULL);
80      }
81  }
82  /*-----------------------------------------------------------*/
83  void vTask3( void *pvParameters )
84  {
85      for( ;; )
86      {
87          vPrintString( "Task3 is running\n" );
88          vTaskDelay( 200 / portTICK_RATE_MS );
89      }
90  }
91  /*-----------------------------------------------------------*/
```



Question 4: (10 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. (Hint: Sender2 will be the first to execute once Scheduler starts)**

Break Point at Line?	State of Sender 1	State of Sender 2	State of Receiver
77	ready	running	ready
78	ready	running	ready
77	running	ready	ready
77	Block	running	ready
86	Block	Block	running
78	running	Block	ready
77	running	Block	ready
87	Block	Block	running
86	Block	Block	running
78	Block	running	ready

```
59  int main( void )
60  {
61    xQueue = xQueueCreate( 1, sizeof( long ) );
62    if( xQueue != NULL )
63    {
64      xTaskCreate( vSenderTask,"SENDER1", 240, ( void * ) 100, 3, NULL );
65      xTaskCreate( vSenderTask,"SENDER2", 240, ( void * ) 200, 3, NULL );
66      xTaskCreate( vReceiverTask, NULL, 240, NULL, 2, 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 5: (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	69	75	76	92	78	94	85

```
66  int main( void ){
67    xmutex = xSemaphoreCreateMutex();
68    xTaskCreate(vtask1,"Task 1",200, NULL, 1,NULL);
69    vTaskStartScheduler();
70  }
71  void vtask1(void *pvParameters)
72  {
73    for(;;)
74    {
75      xSemaphoreTake(xmutex, portMAX_DELAY);
76      xTaskCreate(vtask3,"Task 3",200, NULL, 3,NULL);
77      xTaskCreate(vtask2,"Task 2",200, NULL, 2,NULL);
78      xSemaphoreGive(xmutex);
79    }
80  }
81  void vtask2(void *pvParameters)
82  {
83    for(;;)
84    {
85      vTaskDelay(200);
86    }
87  }
88  void vtask3(void *pvParameters)
89  {
90    for(;;)
91    {
92      xSemaphoreTake(xmutex, portMAX_DELAY);
93      xSemaphoreGive(xmutex);
94      vTaskDelay(200);
95    }
96  }
```

AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
ICHEP; MCT-CESS
Fall 2022

January 26th,2022 **Course Code: CSE347-CSE345** **Time: 2 Hours**

Embedded System Design & Real-Time Embedded System Design

The Exam Consists of **5 Questions in 5 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: (7 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 2: (10 Marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 10 break points to be hit, when GO is pressed.**

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

6th Break Point Hit	7th Break Point Hit	8th Break Point Hit	9th Break Point Hit	10th Break Point Hit

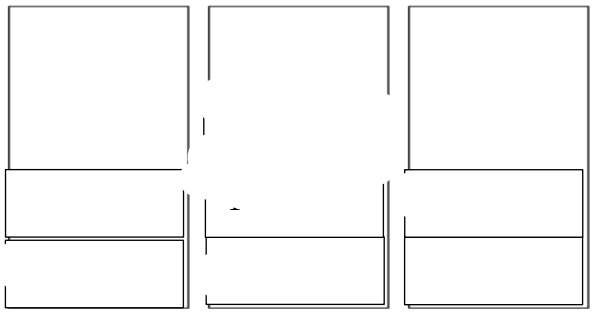
```
82  int main( void )
83  {
84      vSemaphoreCreateBinary( xBinarySemaphore );
85      if( xBinarySemaphore != NULL )
86      {
87          prvSetupSoftwareInterrupt();
88          xTaskCreate( vHandlerTask, NULL , 240, NULL,3, NULL );
89          xTaskCreate( vPeriodicTask, NULL, 240, NULL,1, NULL );
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 }
```

Question 3: (6 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 after starting the schedular.**

```
54  /*-----------------------------------------------------------*/
55
56  int main( void )
57  {
58      xTaskCreate( vTask1,NULL, 240, NULL, 2, NULL );
59      xTaskCreate( vTask3,NULL, 240, NULL, 3, NULL );
60      vTaskStartScheduler();
61      for( ;; );
62  }
63  /*-----------------------------------------------------------*/
64  void vTask1( void *pvParameters )
65  {
66      for( ;; )
67      {
68          vPrintString( "Task1 is running\n" );
69          xTaskCreate( vTask2,NULL, 240, NULL, 4, NULL );
70          vTaskDelay( 200 / portTICK_RATE_MS );
71      }
72  }
73  /*-----------------------------------------------------------*/
74  void vTask2( void *pvParameters )
75  {
76      for( ;; )
77      {
78          vPrintString( "Task2 is running\n" );
79          vTaskDelete(NULL);
80      }
81  }
82  /*-----------------------------------------------------------*/
83  void vTask3( void *pvParameters )
84  {
85      for( ;; )
86      {
87          vPrintString( "Task3 is running\n" );
88          vTaskDelay( 200 / portTICK_RATE_MS );
89      }
90  }
91  /*-----------------------------------------------------------*/
```

BP 70 BP 79 BP88



Question 4: (10 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. (Hint: Sender2 will be the first to execute once Scheduler starts)**

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, 3, NULL );
65        xTaskCreate( vSenderTask,"SENDER2", 240, ( void * ) 200, 3, NULL );
66        xTaskCreate( vReceiverTask, NULL, 240, NULL, 2, 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 5: (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							

```
66  int main( void ){
67    xmutex = xSemaphoreCreateMutex();
68    xTaskCreate(vtask1,"Task 1",200, NULL, 1,NULL);
69    vTaskStartScheduler();
70  }
71  void vtask1(void *pvParameters)
72  {
73    for(;;)
74    {
75      xSemaphoreTake(xmutex, portMAX_DELAY);
76      xTaskCreate(vtask3,"Task 3",200, NULL, 3,NULL);
77      xTaskCreate(vtask2,"Task 2",200, NULL, 2,NULL);
78      xSemaphoreGive(xmutex);
79    }
80  }
81  void vtask2(void *pvParameters)
82  {
83    for(;;)
84    {
85      vTaskDelay(200);
86    }
87  }
88  void vtask3(void *pvParameters)
89  {
90    for(;;)
91    {
92      xSemaphoreTake(xmutex, portMAX_DELAY);
93      xSemaphoreGive(xmutex);
94      vTaskDelay(200);
95    }
96  }
```