



August --th, 2021

Course Code: CSE 347
 Mid Term; Embedded System Design

Time: 1 Hour

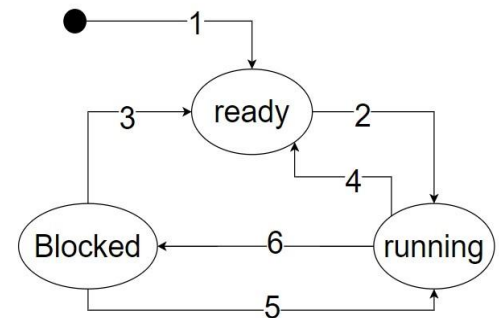
The Exam Consists of 4 Questions in 4 Pages

Total Marks: 25 Marks

Question 1: (6 marks)

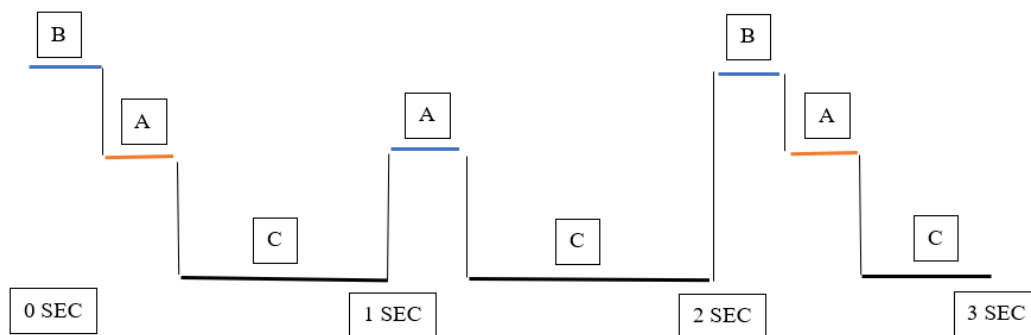
Match a transition-number with the sentence that is suitable to change the task state as it shown:

- A. Task is unblocked but is not the highest priority task **(3)**
- B. Task has the highest priority **(2)**
- C. Task no longer has the highest priority **(4)**
- D. Task is unblocked and it's the highest priority task **(5)**
- E. Task is waiting for an event **(6)**
- F. Task is initialized and activated **(1)**



Question 2: (3 marks)

In a FreeRTOS project, three short tasks were created (Task A, Task B and Task C). Task A and Task B are having the periods 1Sec and 2Sec, respectively. Task C is a continuous task. Their priorities are 2, 3, and 1, respectively. Sketch tasks timing diagram for the first 3 Seconds. Vertical axe is the priority level while the horizontal axe is the time in seconds.



Question 3: (9 marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 9 break points to be hit, when GO is pressed.**

```

54 int main( void )
55 {
56
57     xTaskCreate( vTask1, "Task 1", 240, NULL, 2, NULL );
58     xTaskCreate( vTask2, "Task 2", 240, NULL, 1, &xTask2Handle );
59     vTaskStartScheduler();
60     for( ;; );
61 }

```

```

64 void vTask1( void *pvParameters )
65 {
66     unsigned portBASE_TYPE uxPriority;
67     uxPriority = uxTaskPriorityGet( NULL );
68     for( ;; )
69     {
70         vPrintString( "Task1 is running\n" );
71         vPrintString( "About to raise the Task2 priority\n" );
72         vTaskPrioritySet( xTask2Handle, ( uxPriority + 1 ) );
73     }
74 }
75

```

```

78 void vTask2( void *pvParameters )
79 {
80     unsigned portBASE_TYPE uxPriority;
81     uxPriority = uxTaskPriorityGet( NULL );
82     for( ;; )
83     {
84         vPrintString( " Hi  \n" );
85         vPrintString( "About to lower the Task2 priority\n" );
86         vTaskPrioritySet( NULL, ( uxPriority - 2 ) );
87         vPrintString( " Bye  \n" );
88     }
89 }

```

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
70	72	84	86	70

6 th Break Point Hit	7 th Break Point Hit	8 th Break Point Hit	9 th Break Point Hit
72	87	84	86

Question 4: (7 marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 9 break points to be hit, when GO is pressed.**

```
59 int main( void )
60 {
61     xQueue = xQueueCreate( 5, sizeof( long ) );
62     if( xQueue != NULL )
63     {
64         xTaskCreate( vSenderTask, "Sender", 240, ( void * ) 100, 1, NULL );
65         xTaskCreate( vReceiverTask, "Receiver", 240, NULL, 2, NULL );
66         vTaskStartScheduler();
67     }
68     else
69     {
70         /* The queue could not be created. */
71     }
72     for( ;; );
73 }

76 static void vSenderTask( void *pvParameters )
77 {
78     long lValueToSend;
79     portBASE_TYPE xStatus;
80     lValueToSend = ( long ) pvParameters;
81     for( ;; )
82     {
83         xStatus = xQueueSendToBack( xQueue, &lValueToSend, 0 );
84         if( xStatus != pdPASS )
85         {
86             vPrintString( "Could not send to the queue.\r\n" );
87         }
88     }
89 }
```

Embedded System Design

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

Total Marks: 25 Marks

4/4

```

92 static void vReceiverTask( void *pvParameters )
93 {
94     long lReceivedValue;
95     portBASE_TYPE xStatus;
96     const portTickType xTicksToWait = 100 / portTICK_RATE_MS;
97     for( ;; )
98     {
99         if( uxQueueMessagesWaiting( xQueue ) != 0 )
100         {
101             vPrintString( "Queue should have been empty!\r\n" );
102         }
103         xStatus = xQueueReceive( xQueue, &lReceivedValue, xTicksToWait );
104
105         if( xStatus == pdPASS )
106         {
107             vPrintStringAndNumber( "Received = ", lReceivedValue );
108         }
109         else
110         {
111             vPrintString( "Could not receive from the queue.\r\n" );
112         }
113     }
114 }

```

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
103	83	105	103	84

6 th Break Point Hit	7 th Break Point Hit
83	105