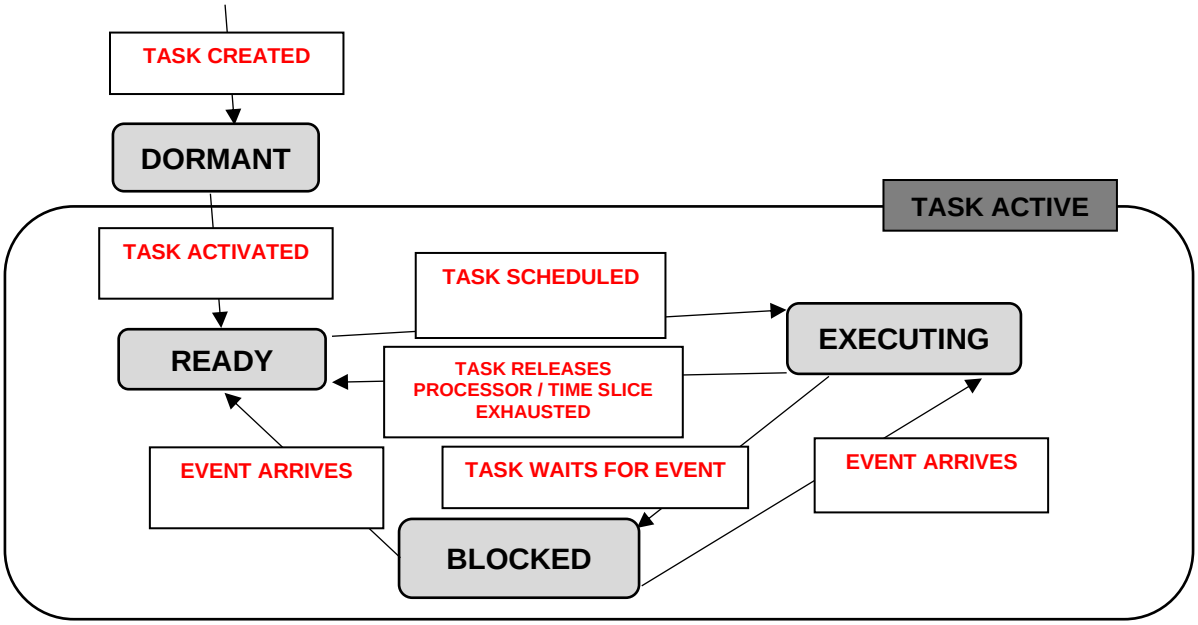
November 19th, 2021 **Course Code: CSE 347** ***Time: 1 Hour***

Mid Term; Embedded System Design

The Exam Consists of **4** Questions in **3** Pages ***Total Marks: 25 Marks***

Question 1: (7 marks)

The figure below shows a subset of RTOS task state machine. On "Cut" arrows, **give the condition of transition from one state to another**.



Question 2: (6 marks)

A. Having two tasks, each that executes its functionality within a while(1), **what is/are the downgrade(s) of using the Program Counter (PC) to switch the execution of the tasks?**

Hacking the PC leads to having the tasks executing from the beginning of their code and until their time slice ends only, hence the first part only will keep repeating while context is not recalled.

B. State at least two of the **CPU scheduling criteria**.

- CPU Utilization
- Throughput
- Turnaround time
- Waiting time
- Response

Question 3: (4 marks)

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **Sketch tasks timing diagram**. Vertical axe is the priority level while the horizontal axe is the time.

```
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  }
```

```
100 void vApplicationIdleHook( void )
101 {
102   ulIdleCycleCount++;
103 }
104
105
```

Question 4: (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 to be hit, when GO is pressed.**

```
 57  int main( void )
 58  {
 59    xTaskCreate( vTask1, "Task 1", 240, NULL, 2, NULL );
 60    xTaskCreate( vTask3, "Task 3", 240, NULL, 1, NULL );
 61    vTaskStartScheduler();
 62
 63    for( ;; );
 64  }
 65
 68  void vTask1( void *pvParameters )
 69  {
 70  const portTickType xDelay100ms = 100 / portTICK_RATE_MS;
 71
 72    for( ;; )
 73    {
 74      vPrintString( "Task1 is running\n" );
 75      xTaskCreate( vTask2, "Task 2", 240, NULL, 3, &xTask2Handle );
 76      vTaskDelay( xDelay100ms );
 77    }
 78  }
 82  void vTask2( void *pvParameters )
 83  {
 84    vPrintString( "Task2 is running and about to delete itself\n" );
 85    vTaskDelete( xTask2Handle );
 86  }
 87
 90  void vTask3( void *pvParameters )
 91  {
 92    char *pcTaskName;
 93    volatile unsigned long ul;
 94    pcTaskName = ( char * ) pvParameters;
 95
 96    for( ;; )
 97    {
 98      vPrintString( pcTaskName );
 99    }
100  }
125  void vApplicationIdleHook( void )
126  {
127    ulIdleCycleCount++;
128  }
```

1st Break Point Hit	2nd Break Point Hit	3rd Break Point Hit	4th Break Point Hit	5th Break Point Hit
74	75	84	85	76

6th Break Point Hit	7th Break Point Hit	8th Break Point Hit
98	98	98