|  |
| --- |
| Course: CMPE244  Name: Sarika Satish Natu  ID: 014529138(EE)  Lab: #2  Date: 9/5/2020 |

Multiple Tasks

Scenario 1: task1 and task2 both running at low priority

Subcase 1: task1 is created before task2

Relevant code:

A screenshot of a cell phone

Description automatically generated

Telemetry output:

A screenshot of a cell phone

Description automatically generated

Explanation of “How come 4(or 3 sometimes) characters are printed from each task? Why not 2 or 5, or 6?”

As the speed is defaulted to 38400bps and 10 bits of data used to send 1 byte, i.e. 3840 characters can be sent per second.

FreeRTOS tick rate is configured to be 1kHz, i.e. preemptive scheduling is occurring every 1ms repeatedly.

If 3840 characters can be sent in 1 sec, then how many characters can be sent in 1msec?

No of characters can be sent in 1 msec = 3840/1000 = 3.84 character.

Therefore, characters that can be printer can only be either of 4 or 3 of each task, but not 2 or 5 or 6.

Subcase 1: task2 is created before task1

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

Explanation:

Even though the tasks are created in any order, the execution of the tasks depends on the scheduler and order of task execution cannot be predicted. The control over the order of task execution is out of scope and it completely depends on the scheduler.

Scenario 2: Same priority; Task 1 and Task 2 both running at medium priority

Relevant code:

A screenshot of a cell phone

Description automatically generated

Telemetry output:

A screenshot of a cell phone

Description automatically generated

Observation:

As both the tasks have same priority, task1 gets equal time of execution as task2 when task2 calls delay.

The order of the task execution completely depends on the scheduler. Therefore both the strings of task1 and task2 can be printed in any fashion, depending on which tasks does scheduler decide to execute.

Scenario 3: Task 1 has high priority and Task 2 has low priority

Relevant code:

A screenshot of a cell phone

Description automatically generated

Telemetry output:

A screenshot of a cell phone

Description automatically generated

Explanation:

Here task1 has high priority and task2 has low priority.

Therefore, in output screen, it can be seen that task1 gets executed first printing ‘A’ sequence later followed by task2 printing ‘b’ sequence.

Scenario 4: Task 1 has low priority and Task 2 has high priority

Relevant code:

A screenshot of a cell phone

Description automatically generated

Telemetry output:

A screenshot of a cell phone

Description automatically generated

Explanation:

Here task2 has high priority and task2 has low priority.

Therefore, in output screen, it can be seen that task2 gets executed first printing ‘b’ sequence later followed by task1 printing ‘A’ sequence.