
Exercise: Free RTOS and Priority Inheritance (Coding task 1.5h)

In the lecture we have learned that flying to Mars with the VxWorks Realtime Operating System can lead to possible deadline violations of tasks if the default settings of VxWorks are used, i.e., priority inheritance is switched off. A problem, called priority inversion, can occur.

In this exercise the learning goal is to deepen our understanding of priority inversion and priority inheritance. For this, FreeRTOS will be used.

FreeRTOS is a free real-time system operating system which is currently the most widely used real-time operating systems in embedded devices (source: UBM 2015 embedded markets study).

Download FreeRTOS, then try to understand one of the demo projects provided with FreeRTOS.

In the lecture we discussed a popular example of the priority inversion problem: the inversion problem that happened during the Mars Pathfinder mission.

Now build a simple FreeRTOS example program where two tasks (ASI/MET and bc_dist) of different priorities (high + low) share a common resource which is protected using a mutex. If the high priority task (bc_dist) wants to access the resource but it is locked by the task with the low priority (ASI/MET), the priority inheritance rule will be applied: the low priority task (ASI/MET) gets the priority of the high priority task (bc_dist) as long as it holds the resource / locks the mutex.

Let the tasks output their current priority information such that you can identify the situation in which the priority inheritance happens.