Exercise: Deadlocks (Reading/Understanding task 0.5h + Coding task 0.5-1h = 1-1.5h)

1. Introduction

We have talked about scheduling approaches for real-time systems (RMS, DMS, EDF, LLF). While these scheduling approaches often can guarantee that all deadlines are met, we have always assumed that the tasks are independently. However, in practice, tasks often share common resources. In this context, a problem can occur that is called **"deadlock"**.

2. Reading task

Search for sources and read into the topic and understand what a **deadlock** is.

(i) How can a deadlock occur if there are 2 tasks sharing 2 resources?
(ii) How can a deadlock occur if there are 3 tasks sharing 3 resources?

3. Coding task

Write a minimal C++ program that is able to demonstrate a deadlock situation for case (i) and write another minimal program that is able to demonstrate a deadlock situation for case (ii).