Operating Systems 2016/17
Tutorial-Assignment 8

Prof. Dr. Frank Bellosa
Dipl.-Inform. Marc Rittinghaus

Question 8.1: Kernel Synchronization

a. On single-processor systems, why can mutual exclusion in the kernel be achieved by masking interrupts?

b. Why does this approach not work on multi-processor systems?

c. How can mutual exclusion within the kernel be achieved on multi-processor systems?

Question 8.2: Prerequisites for Deadlocks

a. Define the term deadlock and give some examples.

b. Enumerate and explain the necessary conditions for deadlocks.

Question 8.3: Searching for Deadlocks

Consider the following code fragment:

```
1  Spinlock s1, s2, s3 = FREE;
2  int counter = 0;

3  void Thread1() {
4      if (counter == 0) {
5          lock(s1);
6          counter++;
7          unlock(s1);
8      }
9      lock(s2);
10     lock(s3);
11     // update some more data
12     unlock(s3);
13     unlock(s2);
14 }
```

```
15 void Thread2() {
16     lock(s3);
17     counter++;
18     // update some data
19     if (counter == 2) {
20         lock(s2);
21         // update some more data
22         unlock(s2);
23     }
24     lock(s1);
25     // update even more data
26     unlock(s3);
27     unlock(s1);
28 }
```

a. Is the code vulnerable to race conditions?

b. Can a deadlock occur? Why, or why not?

Question 8.4: Deadlock Prevention and Avoidance

a. What is deadlock prevention and how does it differ from deadlock avoidance?

b. For each condition necessary for deadlocks, give an example of how deadlocks can be prevented by breaking the condition.

c. What is a safe state?

Question 8.5: Resource Allocation Graphs

a. What kinds of vertices exist in a resource allocation graph (RAG)?

b. How are resource types with multiple instances per resource represented in a RAG?

c. What is a request edge?

d. What is an assignment edge?

e. Does a cycle in a RAG always mean that a deadlock occurred?