Operating Systems 2016/17
Tutorial-Assignment 1

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

Question 1.1: Basics

a. Enumerate the major tasks of an operating system.

b. What are some of the differences between a processor running in privileged mode (also called kernel mode) and user mode? Why are the two modes needed?

c. What are typical examples of privileged instructions? Why are they privileged?

Question 1.2: The User/Kernel Boundary

a. The operating system does not always run (even if multiple CPUs/cores are available). What three events can lead to an invocation of the kernel?

b. Kernel entries can be classified in at least two dimensions:

   (a) voluntary vs. involuntary

   (b) synchronous vs. asynchronous

   Associate each of the three kernel entries with a combination of the two parameters and discuss the unassigned combination with respect to potential use in current or future systems.

c. Some systems try to enhance kernel protection by introducing an extra kernel stack instead of using the application stack of the currently running application whilst performing kernel operations. Does this technique really improve kernel protection? Explain!

d. What is an interrupt vector?

e. What is an interrupt service routine (ISR)?

f. Some exceptions usually lead to the abortion of the executing thread/task while others can be repaired by the OS. Find examples for both types of exceptions!

Question 1.3: System Calls

a. Explain how a `trap` instruction is related to system calls.

b. What is a system call number and how does it relate to the operating system API?

c. How can you pass parameters to the kernel when performing a system call?

d. How are library calls related to system calls?

e. System calls enable the transition from user level to kernel level. Why do we have to be very careful when designing this transition?

f. What problem exists when a system call expects a pointer to a user-level buffer to which the kernel has to write data? Is there also a problem when the user-level buffer is only read?