Operating Systems 2016/17
Tutorial-Assignment 3

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

Question 3.1: Process Switching

a. What data is stored in a process control block (PCB)? Where is it located?

b. Describe the actions taken by the kernel to perform a context-switch between processes.

Question 3.2: Threads

a. Explain the terms process, address space, and thread. How do they relate to each other?

b. Compare the three thread models: One-to-One threads (kernel-level threads), Many-to-One threads (user-level threads), and Many-to-Many threads (hybrid threads). Point out advantages and limitations of each thread model.

c. Which types of events can trigger a One-to-One thread switch?

d. Which types of events can trigger a Many-to-One thread switch?

e. Discuss the following statement: “Jobs are either I/O-bound or compute-bound. In neither case would user-level threads be a win. Why would one go for pure user-level threads at all?”

f. The Unix system call `fork()` creates a new process (child), which is identical to its parent in most parts. Would it make sense for the new process to also contain copies of all the parent’s (other) threads?

g. What is the motivation for and purpose of kernel-mode threads?

h. Write a small program that creates five threads using the pthread library. Each thread should print its number (e.g., `Hello, I am 4`) and the main program should wait for each thread to exit.