

Some of the answers to these questions can be found using the lecture slides, the recommended textbooks or other sources (question marked with \*). Some questions may have more than one possible answer, or be more or less open for discussion. Note that no answers (solutions) will be given to these questions, but if help is needed the assistants will be available to answer questions. The concepts marked with yellow are important and should be fully understood.

## Operating Systems Basics

1. What are the two main functions of an operating system (OS)?
2. What are two OS structures presented in the lecture? Briefly discuss commonalities and differences and which OS component belongs to either kernel mode or user mode!
3. In a multiprogramming and time-sharing environment, several users share the system simultaneously. This situation can result in various security problems.
  - (a) What are such problems?
  - (b) Which concepts are provided by modern OSs to support secure multi-user computing?
  - (c) (How) may virtualization help solving these problems?
4. Which of the following instructions should be allowed *only* in kernel mode?
  - (a) Disable all interrupts
  - (b) Set time-of-day clock
  - (c) Change the memory map

## Processes & Threads

1. What is the difference between a program and a process? Discuss various properties of the OS process concept!
2. What kind of OS process management activities were mentioned in the lecture? Give examples!
3. Discuss the conditions for transiting between following process states:
  - (a) `ready`  $\rightarrow$  `running`
  - (b) `running`  $\rightarrow$  `ready`
  - (c) `running`  $\rightarrow$  `waiting`
  - (d) `ready`  $\rightarrow$  `waiting`

- 4.\* Try to find out which process states exist in Linux, and what they mean.
- 5.\* Enumerate and discuss the following reasons for providing an environment that allows **process cooperation**:
  - (a) Information sharing
  - (b) Computation speedup
  - (c) Modularity
  - (d) Convenience
6. Process management details
  - (a) How many parents does a process have?
  - (b) On child process creation, what decisions can be made by the parent?
  - (c) Give advantages for using process hierarchies.
  - (d) Discuss what is stored inside a **process control block (PCB)**.
  - (e) Describe the actions taken by a kernel to **context-switch** between processes.