**Study programme(s):** Information Technologies

**Level:** Bachelor

**Course title:** Operating Systems 2

**Lecturer:** Zoran D. Budimac

**Status:** elective

**ECTS:** 7

**Requirements:** completed course Data Structures and Algorithms 2, completed course Operating Systems 1

**Learning objectives**

Introduction to advanced concepts of operating systems. Presentation of the UNIX operating system. Analysis of a subset of the realization of an operating system, with emphasis on user interface.

**Learning outcomes**

*Minimum:* At the end of the course, it is expected that the successful student is able to use UNIX system calls and understand the basic principles of graphical user interface.

*Desirable:* At the end of the course, it is expected that the successful student is able to use UNIX system calls in an advanced way and to understand and demonstrate the application of implementation of interactive graphics software system.

**Syllabus**

*Theoretical instruction*

Deadlocks. Disk Management. Security system. Protection mechanisms. UNIX operating system. The structure of the operating system and system calls. Input and output. The input-output devices. Interrupts and device management software. Graphical user interface. Elements of word processors and graphic editors. Analysis of a subset of the realization of an operating system.

*Practical instruction*

The system calls of the UNIX operating system. The file system of UNIX operating system. Interprocess communication and synchronization with a special focus on the specifics of the UNIX operating system.

**Literature**

*Recomended*

1. Andrew S. Tanenbaum: Modern Operating Systems, 4th Edition, Prentice Hall, 2015.
2. Hanspeter Mössenböck: Object-Oriented Programming in Oberon-2, 2nd Edition, Springer, 1995.

**Weekly teaching load**

| Lectures: | Exercises: | Practical Exercises: | Student research: | Other: |
|---|---|---|---|---|
| 2 | 1 | 2 | 0 | 0 |

**Teaching methodology**

During lectures, classical methodology is applied, through usage of beam-projector and slides. During theoretical exercises, main principles are outlined and practiced and typical problems and their solutions are analyzed. During practical exercises, students independently apply the mastered techniques.

Knowledge of students is assessed through their ability to apply gained knowledge on appropriate real life problems and is shown during practical exercises. Practical exercises are designed so that two types of classes alternate weekly: classes on which, with the help of assistants, students practice the principles and techniques, discuss solutions and the like, and classes on which students work independently on small projects and their results are studied in detail and then evaluated. On the oral part of the exam students demonstrate a comprehensive understanding of concepts, data structures and algorithms presented on lectures and system call concepts with focus on the UNIX operating system.

The course is supported by many additional resources and specially prepared exercises available in the form of an electronic course on the site of the Department, with a goal of continuous innovation of teaching and encouragement of students to independently research current topics, critically think and to detect application areas of learned materials.

**Grading method (maximal number of points 100)**

| **Pre-exam oblications** | **points** | **Final exam** | **points** |
|---|---|---|---|
| Practical instruction | **50** | Oral exam | **50** |