

## Specification for the book of courses

|  |  |   |                                |                      |
|--|--|---|--------------------------------|----------------------|
| <b>Study program</b>   |  | Electrical Engineering and Computer Science |                                |                      |
| <b>Module</b>  |  | Computing and Informatics                   |                                |                      |
| <b>Type and level of studies</b>   |  | Undergraduate Academic Studies              |                                |                      |
| <b>The name of the course</b>  |  | System Programming                          |                                |                      |
| <b>Lecturer (for lectures)</b>   |  | Stanimirović S. Aleksandar                  |                                |                      |
| <b>Lecturer/associate (for exercises)</b>  |  | Davidović P. Nikola                         |                                |                      |
| <b>Lecturer/associate (for OFE)</b>  |  | Davidović P. Nikola                         |                                |                      |
| <b>Number of ECTS</b>  | 5  | <b>Course status (obligatory/elective)</b>  | Elective                       |                      |
| <b>Prerequisites</b>   |  |   |                                |                      |
| <b>Course objectives</b>   | The outcome of the course is to introduce students to the basic concepts and principles of system programming. The course provide the knowledge necessary for the application development based on using low level operating system interfaces.  |   |                                |                      |
| <b>Course outcomes</b>   | Theoretical and practical understanding of system programming. The course goal is to provide students with knowledge how to use advanced operating system interfaces to develop various software solutions. As the outcome of the course, the students will have practical knowledge of using different techniques and tools for system programming.   |   |                                |                      |
| <b>Course outline</b>  |  |   |                                |                      |
| <b>Theoretical teaching</b>  | Introduction to system programming. The executable file format. Program code portability. Compute programme debugging and troubleshooting. Profiling the program. Memory allocation, memory leaks and automatic memory management (garbage collection). Linking and loading programming libraries. Version control systems. Development of system software using container environments. Concurrent programming: processes, threads and fibers (fibers). Race conditions, mutual exclusion and synchronization. Synchronization primitives in modern programming languages. Semafor and monitors. Message-based concurrent programming. Typical synchronization problems and their solutions. Deadlock and livelock. Concurrent programming with avoiding locking. Programming based on event processing. Synchronous and asynchronous input/output operations. Asynchronous task execution. |   |                                |                      |
| <b>Practical teaching (exercises, OFE, study and research)</b>                       | The students will be introduced with the modern tools and programming languages used for system programming. Practical examples illustrating various synchronization problems and their solutions. Practical examples illustrating advanced concurrent programming techniques. Practical examples that illustrate synchronous asynchronous input/output operations. Practical examples that illustrate the asynchronous task execution.  |   |                                |                      |
| <b>Textbooks/references</b>  |  |   |                                |                      |
| 1  | William Stallings, Operating Systems: Internals and Design Principles, 7th edition, Pearson Education  |   |                                |                      |
| 2  | M. Ben-Ari: Principles of Concurrent and Distributed Programming, 2nd Edition, Addison-Wesley  |   |                                |                      |
| 3  | R. Bryant, D. O'Hallaron, Computer Systems: A Programmer's Perspective, 3rd edition, Pearson, 2016, ISBN-13: 978-9332573901  |   |                                |                      |
| 4  | Different papers to discuss modern trends regarding system programming and concurrent programming.   |   |                                |                      |
| 5  |  |   |                                |                      |
| <b>Number of classes of active education per week during semester/trimester/year</b> |  |   |                                |                      |
| <b>Lectures</b>  | <b>Exercises</b>   | <b>OFE</b>                                  | <b>Study and research work</b> | <b>Other classes</b> |
| 2  | 2  | 1   | 0                              | 0                    |
| <b>Teaching methods</b>  | Lectures, auditory exercises, laboratory exercises. Individual work for homework and projects, student seminars (presentations of student work with discussion).   |   |                                |                      |
| <b>Grade (maximum number of points 100)</b>  |  |   |                                |                      |
| <b>Pre-exam duties</b>   | <b>Points</b>  | <b>Final exam</b>                           |                                | <b>Points</b>        |
| <b>Activity during lectures</b>  | 10   | <b>Written exam</b>                         |                                |                      |
| <b>Exercises</b>   | 30   | <b>Oral exam</b>                            |                                | 30                   |
| <b>Colloquia</b>   |  |   |                                |                      |
| <b>Projects</b>  | 30   |   |                                |                      |