Syllabus

Course Description

Course Title	Operating Systems
Course Code	76270
Course Title Additional	
Scientific-Disciplinary Sector	NN
Language	English; Italian
Degree Course	Bachelor in Computer Science
Other Degree Courses (Loaned)	
Lecturers	Prof. Alessandro Artale, Alessandro.Artale@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/3026 dr. Abdul Haleem Butt, AbdulHaleem.Butt@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/54504
Teaching Assistant	
Semester	Second semester
Course Year/s	1
CP	9
Teaching Hours	50
Lab Hours	40
Individual Study Hours	135
Planned Office Hours	27
Contents Summary	Module 1: Operating Systems and Networking • Operating Systems Structures • Processes, Threads and Concurrency • CPU Scheduling and Synchronization • Memory and Mass-Storage • I/O, File Systems • Networks and Distributed Systems

	Module 2: Programming in C • Data types, variables, operators, control structures, functions, dat structures: arrays and pointers • Dynamic memory allocation and correct memory management, dynamic data structures (e.g. linked lists, stacks, and queues), • Debugging, file handling, multi-processing (i.e. fork/exec/etc.), and multi-threading
Course Topics	This course belongs to the type "Attività formative di base" and the subject area is "Informatica". The goal of the module M1: Operating Systems and Networking, of this modular course, is to provide students with a solid understanding of operating systems and their core components and functionalities, scheduling algorithms, process management and synchronization, as well as memory management techniques, computer Networking principles and the TCP/IP model, and distributed file system principles. The second module, Programming in C, is devoted to teach the fundamentals of programming in C, and algorithmic techniques implementing some of the functionalities of Operating Systems.
Keywords	Operating Systems, CPU and Memory, I/O, Computer Networks, C programming
Recommended Prerequisites	We recommend the following courses as a prerequisite: - Introduction to Programming; - Computer Systems Architecture.
Propaedeutic Courses	
Teaching Format	The course includes frontal lectures and lab sessions.
Mandatory Attendance	Attendance is not compulsory; non-attending students may contact the lecturer at the start of the course to get support on the modalities of the independent study.
Specific Educational Objectives and Learning Outcomes	Knowledge and Understanding - D1.2 Know in details the fundamental principles of programming. - D1.3 Have a solid knowledge of the most important data structures and programming techniques. - D1.9 Know the principles of computer networks - D1.20 Know the innovative aspects of the last generation of

	operating systems. Applying knowledge and understanding - D2.2 Be able to develop small and medium size programs using different programming languages and paradigms. - D2.3 Be able to solve problems using programming methodologies. - D2.4 Ability to develop programs to interact with microcontrollers and the operating systems of modern computers. Make judgements - D3.1 Be able to collect and interpret useful data and to judge information systems and their applicability. - D3.2 Be able to work autonomously according to the own level of knowledge and understanding. Communication skills - D4.1 Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately. Learning skills - D5.1 Have developed learning capabilities to pursue further studies with a high degree of autonomy. - D5.3 Be able to follow the fast technological evolution and to learn cutting edge IT technologies and innovative aspects of last generation information systems.
Specific Educational Objectives and Learning Outcomes (additional info.)	
Assessment	Assessment for module M1, is based on a written exam designed to evaluate the understanding of fundamental concepts and in-depth knowledge of operating systems. The exam features open-ended questions that cover both theoretical topics and practical Lab exercises. Assessment for module M2 includes programming assignments to be done in front of a Laptop aimed at implementing either the management of the various data structures used in the C

	Language or operating system functionalities.
Evaluation Criteria	The final grade is a weighted average of the grades obtained in the two modules: Operating Systems and Networking (70%) and Programming in C (30%). The Operating Systems and Networking module is assessed through a single written exam, divided into two parts: Part 1: Theory (70%of the module grade). Written questions on the concepts delivered during the lectures. This part assesses conceptual understanding, problem-solving skills, and the ability to explain key principles. Part 2: Lab (30% of the module grade). Written questions and exercises on the topics covered in the lab sessions, in which students write, complete or explain programming code. This part assesses the practical understanding of the theory concepts, and is evaluated based on the correctness of the solution, the structure and readability of the code, and the ability to explain the implementation. Attending students can cover the practical Lab part during the semester by participating in Lab sessions and submitting exercises. The Programming in C module is assessed via a programming activity to be done during the exam in front of a Laptop. The program is assessed based on the clarity of the program, the correct output generated when running the program, the kind of data structures used and the way they are accessed. A passing grade in both parts is required to pass the module. All grades reflect both correctness and, to a lesser extent, clarity of answers. Attending and non-attending students are assessed in the same way.
Required Readings	• Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, and Riccardo Melen. Sistemi operativi. Concetti ed esempi. Ediz. Mylab. Con Contenuto digitale per accesso online. Pearson, 10th edition, February 2019. ISBN 978-88-919-0455-3. • C tutorial at https://www.w3schools.com/c/

Supplementary Readings	• Dennis Ritchie and Brian W. Kernighan. The C Programming Language. • Andrew Tanenbaum and Herbert Bos. Modern Operating Systems. Pearson, Boston, 4th edition, March 2014. ISBN 978-0-13-359162-0.
Further Information	The following software will be used during the course: - Ubuntu (https://ubuntu.com) - C compiler (https://gcc.gnu.org) - Oracle VirtualBox (https://www.virtualbox.org) - Visual Studio Code (https://code.visualstudio.com) - Code Runner (https://marketplace.visualstudio.com/items?itemName=formulahendry.code-runner) - clang-format (https://docs.kernel.org/dev-tools/clang-format.html)
Sustainable Development Goals (SDGs)	Quality education

Course Module

Course Constituent Title	Operating Systems M1:Operating Systems and Networking
Course Code	76270A
Scientific-Disciplinary Sector	INFO-01/A
Language	English
Lecturers	dr. Abdul Haleem Butt, AbdulHaleem.Butt@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/54504
Teaching Assistant	
Semester	Second semester
CP	6
Responsible Lecturer	
Teaching Hours	40
Lab Hours	20
Individual Study Hours	90
Planned Office Hours	18

Contents Summary	- Operating Systems Structures - Processes, Threads and Concurrency - CPU Scheduling and Synchronization - Memory and Mass-Storage - I/O, File Systems - Networks and Distributed Systems
Course Topics	The goal of this module is to provide students with a solid understanding of operating systems, their core components, and their principal functionalities. Students should develop a strong grasp of scheduling algorithms, process management and synchronization techniques, memory management strategies, and the fundamentals of networking and distributed systems building the conceptual and practical foundation needed to understand how operating systems manage hardware resources, coordinate concurrent execution, and communicate over networks. Topics Covered 1-Operating Systems Structures Architecture, design approaches, OS services, and system calls. 2- Processes, Threads and Concurrency Process concept and states, process control blocks, threads, and multithreading. 3- CPU Scheduling and Synchronization Scheduling algorithms and synchronization mechanisms for coordinating concurrent processes. 4- Memory and Mass Storage Techniques for memory allocation and management, and management of mass storage devices. 5- I/O, File Systems I/O subsystems, device management, and file system concepts. 6- Networks and Distributed Systems Fundamentals of networking and distributed system concepts.
Teaching Format	The course includes 40 hours frontal lectures and 20 hours lab sessions.
Required Readings	• Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, and Riccardo Melen. Sistemi operativi. Concetti ed esempi. Ediz. Mylab. Con Contenuto digitale per accesso online. Pearson, 10th edition, February 2019. ISBN 978-88-919-0455-3.

Supplementary Readings	Operating System Concepts - slides

Course Module

Course Constituent Title	Operating Systems M2: Programming in C
Course Code	76270B
Scientific-Disciplinary Sector	INFO-01/A
Language	Italian
Lecturers	Prof. Alessandro Artale, Alessandro.Artale@unibz.it https://www.unibz.it/en/faculties/engineering/academic-staff/person/3026
Teaching Assistant	
Semester	Second semester
CP	3
Responsible Lecturer	
Teaching Hours	10
Lab Hours	20
Individual Study Hours	45
Planned Office Hours	9
Contents Summary	• Data types, variables, operators, control structures, functions, dat structures: arrays and pointers • Dynamic memory allocation and correct memory management, dynamic data structures (e.g. linked lists, stacks, and queues), • Debugging, file handling, multi-processing (i.e. fork/exec/etc.), and multi-threading
Course Topics	This course belongs to the type "Attività formative di base" and the subject area is "Informatica". The goal of this module is to provide students with a solid understanding of operating systems and their core components and functionalities, the fundamentals of programming in C, scheduling algorithms, process management and synchronization, as well as memory management techniques.
Teaching Format	The course includes frontal lectures and lab sessions.

Required Readings	• C tutorial at https://www.w3schools.com/c/ • C. Programming Language, Brian W. Kernighan e Dennis M. Ritchie
Supplementary Readings	Lecture Notes