unibz
Fakultät für Ingenieurwesen
Facoltà di Ingegneria
Faculty of Engineering

# COURSE DESCRIPTION – ACADEMIC YEAR 2024/2025

| | |
|---|---|
| **Course title** | Operating Systems and networks |
| **Course code** | 42408 |
| **Scientific sector** | ING-INF/05 |
| **Degree** | L8 – Bachelor in Electronics and Cyber-Physical Systems Engineering |
| **Semester** | 2nd |
| **Year** | 2 |
| **Credits** | 6+6 |
| **Modular** | Yes |

| | |
|---|---|
| **Total lecturing hours** | 40+40 |
| **Total lab hours** | 20+20 |
| **Attendance** | M1: Attendance is not compulsory, but non-attending students have to contact the lecturers at the start of the course to agree on the modalities of the independent study.<br>M2: Attendance is not compulsory, but highly recommended as many labs require an adequate software and hardware infrastructure; non-attending students may contact the lecturer **at the start of the course** to get support on the modalities of the independent study. |
| **Prerequisites** | M1: C programming at intermediate level, as taught in the Fundamentals of Programming II course; Data structures and algorithms key concepts, as taught in the Algorithms and Programming course.<br>M2: // |
| **Course page** | TEAMS |

| | |
|---|---|
| **Specific educational objectives** | M1: Students gain an understanding of the theoretical and practical concepts of operating systems with a focus on the algorithmic aspects orchestrating the underlying hardware structure of a computer and the basic parallel programming techniques. After this course the students should be able to understand the main principles for designing an OS, operate on an OS through a terminal, and develop code with elements of parallelism.<br><br>M2: By building an idea, designers are challenged to "build to think" and thus gain deeper insights. The integration of wireless communication technologies gives devices the ability to interact with each other and their environment, extending the possibilities of mobile applications further. This course will go beyond early physical prototyping and show how to implement smart sensing devices that can communicate together (from the design to the implementation). Participants learn basic electronics, microcontroller programming, and physical prototyping using the Arduino/ESP32 platform, then use digital and analog sensors, LED lights and motors to build, program and customize smart prototypes. Moreover, students will get enough theoretical background for designing & developing their own physical prototypes that at the end can communicate together. Therefore, students will gain a profound understanding of wireless network technologies and sensor technology fundamentals. |

| | |
|---|---|
| **Module 1** | Operating and real-time Systems |
| **Lecturer** | Prof. Alessandro Torcinovich |
| **Contact** | alessandro.torcinovich@unibz.it |
| **Scientific sector of lecturer** | INF/01 |
| **Teaching language** | Italian |
| **Office hours** | Will be communicated during the first lesson |
| **Lecturing assistant (if any)** | Prof. Alessandro Torcinovich |
| **Contact LA** | Same as above |
| **Office hours LA** | Same as above |
| **List of topics** | • Introduction to Operating Systems<br>• Processes, Input/Output<br>• CPU Scheduling, Threads and Concurrency<br>• Synchronization and Parallel Programming<br>• Central/Virtual/Mass Storage<br>• Real-Time OS Design |
| **Teaching format** | Frontal lectures, exercises, labs |

| | |
|---|---|
| **Module 2** | Networks of Electronic Devices |
| **Lecturer** | Prof. Michael Haller |
| **Contact** | michael.haller@unibz.it |
| **Scientific sector of lecturer** | ING-INF/05 |
| **Teaching language** | German |
| **Office hours** | Thursdays, 12:00-14:00 |
| **Lecturing assistant (if any)** | |
| **Contact LA** | |
| **Office hours LA** | |
| **List of topics** | M1<br>• Introduction to Operating Systems<br>• Processes, Input/Output<br>• CPU Scheduling, Threads and Concurrency<br>• Synchronization and Parallel Programming<br>• Central/Virtual/Mass Storage<br>• Real-Time OS Design<br>M2<br>• Arduino Output (with a strong focus on programming, digital, analog, PWM, ADC)<br>• Arduino Input: Switches, Debouncing, Playing with sensors<br>• AdvancedIO: FSR, stretch sensors, sensitivity, offset, accuracy, dynamic range, linearity and noise, filtering the signals (moving mean filter, EMA, WEMA etc.)<br>• Motors: ERM/LRA, Interrupts, Memory Handling<br>• Serial Communication (Sensors to Microcontroller, Microcontrollers to Computer, etc.), i2C, Series Peripheral Interface (SPI)<br>• Wireless Sensor Networks (e.g. communication architecture, protocols, sensor nodes etc.), WLAN/IEEE 802.11, Bluetooth/IEEE)<br>• MIDI/OSC communication<br>• ESP32 architecture & How to render widgets on a ESP32-Display<br>• Design Guidelines & Prototyping Skills for the design & development of interconnected sensing devices<br>• Technology Trends |

| | |
|---|---|
| **Teaching format** | -Frontal lectures |

| | |
|---|---|
| **Learning outcomes** | M1<br><br>**Knowledge and understanding**<br>Know the fundamental principles and programming techniques of operating and real-time systems;<br>Know the innovative aspects of the last generation of operating systems.<br><br>**Applying knowledge and understanding**<br>Ability to develop programs to interact with microcontrollers and the operating systems of modern computers.<br><br>**Making judgements**<br>Be able to work autonomously according to the own level of knowledge and understanding.<br><br>**Communication skills**<br>Be able to use one of the three languages English, Italian and German, and be able to use technical terms and communication appropriately.<br><br>**Ability to learn**<br>Have developed learning capabilities to pursue further studies with a high degree of autonomy.<br><br>M2<br>• Arduino Output (with a strong focus on programming, digital, analog, PWM, ADC)<br>• Arduino Input: Switches, Debouncing, Playing with sensors<br>• AdvancedIO: FSR, stretch sensors, sensitivity, offset, accuracy, dynamic range, linearity and noise, filtering the signals (moving mean filter, EMA, WEMA etc.)<br>• Motors: ERM/LRA, Interrupts, Memory Handling<br>• Serial Communication (Sensors to Microcontroller, Microcontrollers to Computer, etc.), i2C, Series Peripheral Interface (SPI)<br>• Wireless Sensor Networks (e.g. communication architecture, protocols, sensor nodes etc.), WLAN/IEEE 802.11, Bluetooth/IEEE)<br>• MIDI/OSC communication<br>• ESP32 architecture & How to render widgets on a ESP32-Display<br>• Design Guidelines & Prototyping Skills for the design & development of interconnected sensing devices<br>• Technology Trends |

| | |
|---|---|
| **Assessment** | M1: Written exam. The mark for each part of the exam is 18-30, or insufficient.<br><br>The written exam comprises verification questions and open questions to test knowledge application skills on the arguments discussed during the course.<br>In case of doubts on the final mark. the lecturer reserves the right to ask for an additional oral examination.<br><br>The oral exam comprises verification questions and open questions to test knowledge application skills. It can increase or decrease the mark obtained in the written exam.<br><br>M2: The major activity of the class is centered around a group project (in pairs of two), but there will be individual assignments early in the semester. The goal of these assignments is to ensure everyone in the class gains experience and understanding of the design and implementation of connected sensing devices, without which creating an interesting and sophisticated project will be difficult. |

| | |
|---|---|
| **Assessment language** | Italian (Module 1)<br>German (Module 2) |
| **Assessment Typology** | M1: Written + Oral (if needed)<br>M2: - |
| **Evaluation criteria and criteria for awarding marks** | M1 (50%): The final mark is computed in proportion of the correct answers given in the written exam.<br>The optional oral exam can increase or decrease the mark obtained in the written exam.<br>Relevant for the oral exam: clarity of answers, ability to recall principles and methods, and deep understanding about the course topics presented in the lectures.<br>Non-attending students have the same evaluation criteria and requirements for passing the exam as attending students.<br><br>M2 (50%): Each student group is provided with a physical computing kit including an Arduino/ESP32 compatible board as well as everything needed to learn how to use sensors, displays, and actuators. Through hands-on experiences during class periods, students acquire basic skills and learn to build a range of typical circuits that will communicate to each other. Along with basic skill acquisition, students are involved in a semester-long group assignment in which they develop a complex project from start to finish. Students are encouraged to quickly arrive at a working prototype at which point they can fine-tune their project through testing. At the end of the semester, the projects are presented to the rest of the students. |

| | |
|---|---|
| **Required readings** | M1<br>• Silberschatz, P. B. Galvin, G. Gagne, Operating Systems Concepts<br><br>Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it and Ilaria Miceli, Ilaria.Miceli@unibz.it<br><br>M2<br>• Steffen Wendze; IT-Sicherheit für TCP/IP- und IoT-Netzwerke: Grundlagen, Konzepte, Protokolle, Härtung; ISBN: 3658334223<br>• Walter Trojan; Das MQTT-Praxisbuch: Mit ESP8266 und Node-RED; ISBN: 3895763241<br>• Pradeeka Seneviratne; Beginning LoRa Radio Networks with Arduino: Build Long Range, Low Power Wireless IoT Networks; ISBN: 1484243560<br>• Ayhan Polat; Das Internet of Things. Ein Literatur Review zum aktuellen Forschungsstand; ISBN: 3668536503<br>• The Official Raspberry PI Projects Bool; WEB: https://magpi.raspberrypi.org/books/projects-1<br><br>Subject Librarian: David Gebhardi, David.Gebhardi@unibz.it and Ilaria Miceli, Ilaria.Miceli@unibz.it |
| **Supplementary readings** | M1: Web resources provided by the lecturer during the lectures<br>M2: - |
| **Software used** | M1: C compiler and debugger, Bash shell, GNU/Linux<br>M2: Different microcontrollers and microelectronics kits are used. Only participant students, who attend classes, can use them during class time. Moreover, we will mainly use ProtoPie, Visual Studio Code, Arduino IDE, PlatformIO, and Unity, all of which are available for the students. Further information is provided are on TEAMS. |