**Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti**
**Facultatea de Electronică, Telecomunicaţii şi**
**Tehnologia Informaţiei**

# COURSE DESCRIPTION

## 1. Program identification information

| | |
|---|---|
| 1.1 Higher education institution | **National University of Science and Technology Politehnica Bucharest** |
| 1.2 Faculty | **Electronics, Telecommunications and Information Technology** |
| 1.3 Department | **Electronic Devices, Circuits and Architectures** |
| 1.4 Domain of studies | Electronic Engineering, Telecommunications and Information Technology |
| 1.5 Cycle of studies | Masters |
| 1.6 Programme of studies | Advanced Computing in Embedded Systems |

## 2. Date despre disciplină

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.1 Course name (ro)<br>(en) | | | | Analiza performanţelor şi optimizare<br>Performance Analysis and Optimization | | | |
| 2.2 Course Lecturer | | | | Conf. Dr. Calin Bira | | | |
| 2.3 Instructor for practical activities | | | | Conf. Dr. Calin Bira | | | |
| 2.4 Year of studies | 1 | 2.5 Semester | 2 | 2.6. Evaluation type | E | 2.7 Course regime | Ob |
| 2.8 Course type | | S | 2.9 Course code | 3 | | 2.10 Tipul de notare | Nota |

## 3. Total estimated time (hours per semester for academic activities)

| | | | | | |
|---|---|---|---|---|---|
| 3.1 Number of hours per week | 4 | Out of which: 3.2 course | 2 | 3.3 seminary/laboratory | 2 |
| 3.4 Total hours in the curricula | 56 | Out of which: 3.5 course | 28 | 3.6 seminary/laboratory | 28 |
| Distribution of time: | | | | | hours |
| Study according to the manual, course support, bibliography and hand notes<br>Supplemental documentation (library, electronic access resources, in the field, etc)<br>Preparation for practical activities, homework, essays, portfolios, etc. | | | | | 90 |
| Tutoring | | | | | 3 |
| Examinations | | | | | 1 |
| Other activities (if any): | | | | | 0 |

| | |
|---|---|
| 3.7 Total hours of individual study | 94.00 |
| 3.8 Total hours per semester | 150 |
| 3.9 Number of ECTS credit points | 6 |

## 4. Prerequisites (if applicable) (where applicable)

| | |
|---|---|
| 4.1 Curriculum | C/C++ programming language. Data structures and algorithms |
| 4.2 Results of learning | Technical competencies in utilizing the PC |

## 5. Necessary conditions for the optimal development of teaching activities (where applicable)

| | |
|---|---|
| 5.1 Course | Classroom with projector and Internet-access |

| | |
|---|---|
| 5.2 Seminary/ Laboratory/Project | Classroom with projector and PCs with multi-core CPUs, at least 8th gen from Intel or AMD and dedicated GPU (running OpenCL) |

**6. General objective** *(Reffering to the teachers' intentions for students and to what the students will be thought during the course. It offers an idea on the position of course in the scientific domain, as well as the role it has for the study programme. The course topics, the justification of including the course in the currcula of the study programme, etc. will be described in a general manner)*

The student should get familiar with full optimization cycle of an application. The students will study optimization techniques for run-time performance, for profiling apps, and techniques to estimate maximum achievable performance

**7. Competences** *(Proven capacity to use knowledge, aptitudes and personal, social and/or methodological abilities in work or study situations and for personal and proffesional growth. They refflect the empolyers requirements.)*

| | |
|---|---|
| **Specific Competences** | Applying knowledge, concepts and basic methods regarding system architecture, CPUs, MCUs, programming languages and techniques |
| **Transversal (General) Competences** | - |

**8. Learning outcomes** *(Synthetic descriptions for what a student will be capable of doing or showing at the completion of a course. The learning outcomes reflect the student's acomplishments and to a lesser extent the teachers' intentions. The learning outcomes inform the students of what is expected from them with respect to performance and to obtain the desired grades and ECTS points. They are defined in concise terms, using verbs similar to the examples below and indicate what will be required for evaluation. The learning outcomes will be formulated so that the correlation with the competences defined in section 7 is highlighted.)*

| | |
|---|---|
| **Knowledge** | *The result of knowledge aquisition through learning. The knowledge represents the totality of facts, priciples, theories and practices for a given work or study field. They can be theoretical and/or factual.*<br>After this course, the students will understand for a software + hardware system, the chain of performance estimation, bottleneck analysis, optimizations and measuring the results of various applications. |
| **Skills** | *The capacity to apply the knowledge and use the know-how for completing tasks and solving problems. The skills are described as being cognitive (requiring the use of logical, intuitive and creative thinking) or practical (implying manual dexterity and the use of methods, materials, tools and intrumentation).*<br>Students will know to estimate achievable performance of certain apps, will be able to write optimal code that use x64 SIMD instructions, that use multi-core CPU and GPU, and in addition to optimize memory space in both program and RAM |
| **Responsability and autonomy** | *The student's capacity to autonomously and responsably apply their knowledge and skills.*<br>As a result of this classes, the students will be able to optimize any application, from writing code to profiling and deploying into production. |

**9. Teaching techniques** *(Student centric techniques will be considered. The means for students to participate in defining their own study path, the identification of eventual fallbacks and the remedial measures that will be adopted in those cases will be described.)*

Teaching will be performed in an interactive way, with practical examples and challenges, presented in real-time using simple software programs at start, and a real application in the end.

**10. Contents**

**COURSE**

| Chapter | Content | No. hours |
|---|---|---|
| 1 | Introduction, course planning, tools. | 2 |
| 2 | Moore's Law. Amdahl's Law. Discussions regarding L1, L2, L3 cache | 2 |
| 3 | Single-threaded and multi-threaded applications. C++ std::thread class. OpenMP library. | 2 |
| 4 | SIFT matching. L1-norm and L2-norm for points in 128-dimension space. Introduction to SIMD instructions. Utilization of SIMD instructions | 4 |
| 5 | IR communication waves for remote controls. Light-based communication. OOK modulation. | 4 |
| 6 | DES/AES encryption algorithms (software resistent). Bitslice optimizations. | 2 |
| 7 | Profiling an app. Estimating I/O performance and performance added by external accelerators. | 2 |
| 8 | Introduction to GP-GPUs as accelerators. Cuda and OpenCL | 2 |
| 9 | OpenCL: memory transfers, kernels, simple apps running in O(N) | 4 |
| 10 | Matrix-matrix multiplication optimizations | 2 |
| 11 | Conclusions | 2 |
| | **Total:** | 42 |

**Bibliography:**

1. Optimization guide based on x64: https://www.intel.com/content/www/us/en/content-details/671488/intel-64-and-ia-32-architectures-optimization-reference-manual-volume-1.html
2. OpenMP documentation: https://www.openmp.org/resources/refguides/
3. Intel Intrinsics Guide - https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html
4. GNU Make documentation - https://www.gnu.org/software/make/manual/make.pdf
5. CMake documentation - https://cmake.org/cmake/help/latest/
6. OpenCL documentation https://www.khronos.org/opencl/
7. PIC10F documentation: https://www.microchip.com/en-us/product/pic10f200

**LABORATORY**

| Crt. no. | Content | No. hours |
|---|---|---|
| 1 | Implementing bit permutations for 100 Million 32-bit numbers, naive implementation | 2 |
| 2 | Loop version, effect of branching in loops, loop unroll, compiler optimizations, and 8-bit look-up table | 2 |

| | | |
|---|---|---|
| 3 | 16-bit LUTs, 32-bit LUTs. Discussions regarding Big-Oh notation | 2 |
| 4 | SIFT matching utilizing L1 and L2 norm. Naive implementation. Optimized implementation with SIMD and OpenMP | 4 |
| 5 | Analysis and profiling an app. Estimation of acceleration if adding an external hardware | 4 |
| 6 | ML-L3 type remote control with IR led for Nikon DSLR camera in C/C++. Program space optimizations, loops, RAM space optimizations. | 2 |
| 7 | ML-L3 type remote control with IR led for Nikon DSLR camera, in asm for PIC10F200. | 4 |
| 8 | OpenCL, on O(N) apps. Setup and definitions | 4 |
| 9 | OpenCL apps running in O(N**3) - matrix-matrix multiplication | 4 |
| | **Total:** | 28 |

**Bibliography:**

1. Optimization guide based on x64: https://www.intel.com/content/www/us/en/content-details/671488/intel-64-and-ia-32-architectures-optimization-reference-manual-volume-1.html
2. OpenMP documentation: https://www.openmp.org/resources/refguides/
3. Intel Intrinsics Guide - https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html
4. GNU Make documentation - https://www.gnu.org/software/make/manual/make.pdf
5. CMake documentation - https://cmake.org/cmake/help/latest/
6. OpenCL documentation https://www.khronos.org/opencl/
7. PIC10F documentation: https://www.microchip.com/en-us/product/pic10f200

**11. Evaluation**

| Activity type | 11.1 Evaluation criteria | 11.2 Evaluation methods | 11.3 Percentage of final grade |
|---|---|---|---|
| 11.4 Course | The student was able to analyze a chosen app and he accelerated it | Spoken, by defending his work in a presentation | 50% |
| 11.5 Seminary/laboratory/project | The student was able to understand and use the techniques for accelerating execution time in weekly tasks of the lab | Accumulation of points on tasks received during the labs | 50% |
| 11.6 Passing conditions | | | |
| Student is able to get 50% of total points. | | | |

**12. Corroborate the content of the course with the expectations of representatives of employers and representative professional associations in the field of the program, as well as with the current state of knowledge in the scientific field approached and practices in higher education institutions in the European Higher Education Area (EHEA)**

**Universitatea Naţională de Ştiinţă şi Tehnologie Politehnica Bucureşti**

**Facultatea de Electronică, Telecomunicaţii şi**

**Tehnologia Informaţiei**

In the context of increasing demand of better and better hardware, capable of high performance, understanding compromises and the impact source code has, in running and harvesting parallel resources of the hardware. is a major advantage for any engineer. In addition, estimating performance correctly, may lead to great choices of software and hardware, in the beginning of the real-world project, thus avoiding costly mistakes.

| Date | Course lecturer | Instructor(s) for practical activities |
|---|---|---|
| 21.09.2026 | Conf. Dr. Calin Bira | Conf. Dr. Calin Bira |

| Date of department approval | Head of department |
|---|---|
| 22.09.2026 | Prof. Dr. Claudius Dan |

| Date of approval in the Faculty Council | Dean |
|---|---|
| 22.09.2026 | Prof. Dr. Mihnea Udrea |