Universitatea Națională de Știință și Tehnologie Politehnica București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației

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) (en)				Calcul distribuit și de înaltă performanță Distributed and High Performance Computing			
2.2 Course Lecturer				Prof. Dr. Ing. Emil-Ioan Slusanschi			
2.3 Instructor for practical activities				Prof. Dr. Ing. Emil-Ioan Slusanschi			
2.4 Year of studies	2	2.5 Semester	I	2.6. Evaluation type	E	2.7 Course regime	Ob
2.8 Course type	DA	2.9 Course code		UPB.04.M3.O.26-21		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	2.00	3.3 seminary/laboratory	2
3.4 Total hours in the curricula	56.00	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 Supplemental documentation (library, electronic access resources, in the field, etc) Preparation for practical activities, homework, essays, portfolios, etc.					45
Tutoring					20
Examinations					4
Other activities (if any):					70

3.7 Total hours of individual study	69.00
3.8 Total hours per semester	125
3.9 Number of ECTS credit points	5

4. Prerequisites (if applicable) (where applicable)

4.1 Curriculum	Completion and/or promotion of the following subjects: Parallel and Distributed Algorithms, Architecture of Computing Systems, Architectures and Parallel Processing, Operating Systems, Compilers.

4.2 Results of learning	The ability to compile and evaluate the performance of some applications in Linux.

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

5.1 Course	The course is based on the presentation in the form of slides, as support for an interactive course with questions and discussions about the topics presented on the slides. The material included in the presentation is dense enough to provide good support for individual study. The material is rich in practical examples for correlating theoretical concepts with their use in laboratory activities. Students are encouraged to provide feedback on the structure and content of the course material.
5.2 Seminary/ Laboratory/Project	The laboratory consists of applying fundamental concepts relevant to the respective practical activities, and solving the associated exercises. Also, a part of individual study of some articles or scientific studies made available within the project will be taken into account, in order to deepen the knowledge gained through practical exercises. Questions or discussions about the structure of the course or the project can be asked directly during the classes, or on the course website. Students are encouraged to provide feedback on the structure and content of the course or project material.

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)*

Study of parallel programming models with distributed and shared memory. Description of the modes of communication, synchronization and parallelization of parallel programming paradigms. Detailing the coherence of memory, productivity, performance and portability of several parallel programming models.

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	
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.* Students must be able to: • Differentiate between distributed memory and shared memory programming models. • Describes the communication, synchronization and parallelization modes of the parallel programming paradigms described in the course. • Defines the level of coherence of the memory used. • Classify the level of productivity, performance and portability of parallel programming models covered in the course.
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).* The skills acquired by the students will be the following: • Selection of relevant information in the context of parallel programming paradigms. • Working effectively in a team. • Elaboration of technical and/or scientific reports. • Experimental verification of the solutions identified for the approach projects. • Solving practical applications in industry and research. • Appropriate interpretation of causal relationships • Identifying solutions and developing plans to solve the projects addressed. • Formulation of conclusions based on the experiments carried out. • Argumentation of the solutions identified in the project • Presentation of the ways of solving the implemented projects.
Responsability and autonomy	*The student's capacity to autonomously and responsably apply their knowledge and skills.* Students will be able to: • Select appropriate bibliographic sources and analyze them. • Respect the principles of academic ethics, correctly citing the bibliographic sources used. • Demonstrates responsiveness to new learning contexts • Manifest collaboration with colleagues and teaching staff in teaching activities. • Demonstrates autonomy in organizing the context and solving the problem addressed. • Raise awareness of the value of personal contribution in the field of computer engineering by identifying viable and sustainable solutions to solve problems in the social and economic life of the academic community. • Apply ethical principles and a professional deontology in the analysis of the technological impact of the proposed solutions in the field of computers. • Demonstrate technical team management skills in real-life situations.

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.)*

The teaching activity will consist of lectures using Keynote/Power Point presentations and video clips that will be made available to the students. In each course, there will be a recapitulation of the concepts and chapters covered previously, in order to establish the importance of those presented in the current course. Schemes, diagrams, images and animations will be used in the presentations so that the information is easier for the students to assimilate. In the laboratories, those presented in the theoretical courses will be implemented. Laboratory activities will be carried out in teams of 2 or 3 students, where different parallel programming paradigms applied to the same problem will be studied within each team. Students' tasks will be individual within the teams and will be evaluated weekly.

10. Contents

COURSE		
Chapter	**Content**	**No. hours**
1	Parallel computing architectures – computing systems vs. programming paradigms	2
2	Fundamental notions of design and implementation of parallel structures (SMP/Cluster/MPP)	2
3	Parallel programming models	2
4	Communication models	2
5	Timing patterns	2
6	Memory consistency models	2
7	Resource virtualization systems	2
8	Programming in shared memory systems	4
9	Programming in systems with distributed memory	4
10	Shared memory programming models	2
11	Programming languages with global memory - PGAS GP-GPU Programming	4
	Total:	28

Bibliography:

1. The Sourcebook of Parallel Computing; J. Dongarra, I. Foster, W. Grapp, K. Kennedy; MorganKaufmann 2002.
2. Introduction to Parallel Computing: Design & Analysis of Algorithms; V. Kumar, A. Grama, A.Gupta, G. Karypis; Addison Wesley; 2nd edition 2003.
3. Parallel Programming: Concepts and Practice, B. Schmidt, J. Martinez, C. Hundt, M. Schlarb, Morgan Kaufmann, 2017.
4. Programming Massively Parallel Processors: A Hands-on Approach 4th Edition, W. Hwu, D. Kirk, I. El Hajj, Morgan Kaufmann, 2022.
5. A Framework for Adaptive Algorithm Selection in STAPL, Thomas, Tanase, Tkachyshyn, Perdue, Amato, Rauchwerger, In Proc. ACM SIGPLAN Symp. Prin. Prac. Par. Prog. (PPOPP), pp. 277-288, Chicago, Illinois, Jun 2005.
6. Adve, S. V. and Gharachorloo, K. 1996. Shared Memory Consistency Models: A Tutorial. Computer 29, 12 (Dec. 1996), 66-76. DOI=http://dx.doi.org/10.1109/2.546611
7. Introduction to Parallel Programming, Subodh Kumar, Cambridge University Press, 2023.
8. Parallel Programming: for Multicore and Cluster Systems, Thomas Rauber, Gundula Runger, 3rd ed. Springer 2023.

LABORATORY		
Crt. no.	**Content**	**No. hours**
1	Parallel programming using OpenMP	4
2	Parallel programming using TBB	4

3	Parallel programming using MPI	2
4	Parallel programming using Charm++	2
5	Parallel programming using Cilk+	2
6	Parallel programming using CUDA	4
7	Parallel programming using OpenCL	2
8	Parallel programming using Chapel	2
9	Parallel programming using PThreads	2
10	Parallel programming using MapReduce	2
11	Parallel programming using UPC	2
	Total:	28

Bibliography:

1. http://www.openmp.org
2. Programming With POSIX Threads. D. Butenhof. Addison Wesley
3. http://www.mpi-forum.org/
4. http://charm.cs.uiuc.edu/research/charm/
5. http://chapel.cs.washington.edu/
6. http://hpff.rice.edu/
7. http://upc.gwu.edu/
8. http://x10.codehaus.org/

11. Evaluation

Activity type	11.1 Evaluation criteria	11.2 Evaluation methods	11.3 Percentage of final grade
11.4 Course	The correctness of solving the theoretical questions	Oral assessment	50 points (minimum 25 points to pass)
11.5 Seminary/laboratory/project	Laboratory activity	Oral assessment	10 points (minimum 5 points to pass)
	Correctness of solving homework	Evaluation of the documentation and the written programs for the four homework assignments	40 points (minimum 20 points to pass)
11.6 Passing conditions			
Obtaining a minimum of 50% in each scoring category			

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)

Date	Course lecturer	Instructor(s) for practical activities

	Prof. Dr. Ing. Emil-Ioan Slusanschi	Prof. Dr. Ing. Emil-Ioan Slusanschi
Date of department approval	Head of department	
31.10.2024	Prof. Dr. Claudius DAN	
Date of approval in the Faculty Council	Dean	
01.11.2024	Prof. Dr. Mihnea Udrea	