SYLLABUS

1. Data about the program of study

1.1 Institution	Technical University of Cluj-Napoca
1.2 Faculty	Automation and Computer Science
1.3 Department	Computer Science
1.4 Field of study	Computer Science and Information Technology
1.5 Cycle of study	Master of Science
1.6 Program of study / Qualification	Cybersecurity Engineering / Master
1.7 Form of education	Full time

2. Data about the subject

2.1 Subject name	***Programming security mechanisms on the x86-64 architecture***			Subject code	**4.20**
2.2 Course responsible / lecturer	Prof. dr. eng. Anca Hangan - anca.hangan@cs.utcluj.ro				
2.3 Teachers in charge of seminars / Laboratory / project	Prof. dr. eng. Anca Hangan - anca.hangan@cs.utcluj.ro				
2.4 Year of study	I	2.5 Semester	1	2.6 Type of assessment (E - exam, C - colloquium, V – verification)	E
2.7 Subject category	Formative category: DA – advanced, DS – speciality, DC – complementary				DA
	Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)				DO

3.Estimated total time

3.1 Number of hours per week	4	of which:	Course	2	Seminar	-	Laboratory	2	Project	-
3.4 Total hours in the curriculum	56	of which:	Course	28	Seminar	-	Laboratory	28	Project	-
3.7 Individual study:										
(a) Manual, lecture material and notes, bibliography										20
(b) Supplementary study in the library, online and in the field										18
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays										54
(d) Tutoring										0
(e) Exams and tests										2
(f) Other activities										0
3.8 Total hours of individual study (sum (3.7(a)…3.7(f)))										94
3.9 Total hours per semester (3.4+3.8)										150
3.10 Number of credit points										6

4. Pre-requisites (where appropriate)

4.1 Curriculum	Assembly Language Programming, Operating Systems
4.2 Competence	Computer Architecture

5. Requirements (where appropriate)

5.1 For the course	blackboard, beamer, computers
5.2 For the applications	blackboard, beamer, computers

6. Specific competences

6.1 Professional competences	implement ICT risk management develop information security strategy perform ICT security testing manage disaster recovery plans execute ICT audits develop contingency plans for emergencies manage system security implement ICT recovery system manage IT security compliances identify ICT security risks define security policies perform risk analysis provide ICT consulting advice establish an ICT security prevention plan implement ICT security policies ensure information privacy monitor developments in field of expertise keep up with the latest information systems solutions
6.2 Cross competences	develop an analytical approach taking a proactive approach developing strategies to solve problems being open minded coordinate engineering teams

7. Expected Learning Outcomes

Knowledge	The student has knowledge of: • Operating systems • Computer programming • Cyber attack counter-measures • Embedded systems • Security engineering • ICT encryption • ICT security standards • ICT network security risks • Cloud technologies • Ethical hacking principles • Information security strategy • Computer forensics • Risk management • Security threats • Attack vectors • software anomalies • cyber security

Skills	The student is able to: • Analyse ICT systems • Develop ICT device drivers • Develop an information security strategy • Define technology strategy • Debug software • Use scripting languages for programming • Perform ICT security testing • Identify ICT security risks • Implement ICT security policies • Define security policies • Perform risk analysis • Monitor system performance • Execute software tests • Interpret technical texts • Keep up with the latest information systems solutions • Utilise computer-aided software engineering (CASE) tools • Provide user documentation • Respond to incidents in cloud environments • Manage IT security compliances • Solve ICT system problems
Responsibilities and autonomy	The student has the ability to work independently in order to: • develop an analytical approach • take a proactive approach • develop strategies to solve problems • be open minded • coordinate engineering teams

8. Discipline objectives (as results from the *key competences gained*)

8.1 General objective	Deeper understanding of the x86-64 architecture from the security perspective, understanding the low-level mechanisms of an operating system, its components as well as the basic elements necessary for its development.
8.2 Specific objectives	1. Understanding the x86-64 architecture at the structural and functional level. 2. Understanding the different security mechanisms offered by the x86-64 architecture as well as how to use them within an operating system. 3. Knowing the different low-level components of an operating system; understanding their role and functionality as well as the relationships between them. 4. Knowledge of the techniques of designing and implementing the different components of an operating system. 5. Acquiring experience of programming some hardware components at the level of hardware-software interface.

9. Contents

9.1. Lecture (syllabus)	Hours	Teaching methods	Notes
Introduction to Hardware-Level Programming and Security	2	Blackboard illustrations and explanations, beamer presentations, discussions, short challenges e	
Computer Architecture - Review	2		
Assembly Language and System Programming	2		
Firmware and Boot Process	2		
Memory Management and Security	2		
Hardware Vulnerabilities	2		

Side-Channel Attacks	2		
Trusted Execution Environments	2		
Hardware Security Features	2		
Secure Programming at Low Levels	2		
Virtualization and Security	2		
Security in Embedded and IoT Systems	2		
Emerging Topics in Hardware Security	2		
Student presentations	2		

Bibliography:

1. Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 1-3 (Intel – 2014 – electronic)
2. Operating System Concepts (Silberschatz, Abraham – 2012 – Wiley) (9th ed)
3. Optimizing subroutines in assembly language: An optimization guide for x86 platforms (Fog, Agner – 2013 – electronic, http://www.agner.org/optimize/)
4. Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic, MSDNAA)
5. Presentations (slides) of the course (https://users.utcluj.ro/~sebestyen/cursuri_lab.htm)
6. Development sites for operating system components(e.g. http://wiki.osdev.org/).

9.2 Applications - Seminars / Laboratory / Project	Hours	Teaching methods	Notes
Introduction to the OS starting template used: installation, compilation, execution and testing	2	Brief reviews, blackboard illustrations and explanations, tutorials, roadmaps, short live demos and guidance of code development, discussions, homework.	
Transitioning to long mode. Configuring CPU control structures, memory spaces and paging for 4 level paging	4		
IDT configuration for exception and interrupt handling. Implementing assembly stubs and C ISR routines for handling exceptions and interrupts. Dumping the trap frames for debugging.	2		
PIC programming for interrupt handling. Programming the PIT and keyboard and handling their interrupts.	2		
Implementing interactive I/O e.g., command interpreter	2		
Programming ATA hard drive for PIO access	2		
Memory Management: physical, virtual and heap memory allocators	2		
Intel SMP 1.4 trampoline for booting AP processors	2		
Implementing a synchronization primitive (spinlock). Updating the code to use the primitive: display access, doubly link list access, etc	4		
SMP threads, context switching, scheduling. Mutex. FPU/SEE context saving.	4		

Bibliography:

1. Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 1-4 (Intel – 2022 – electronic)
2. Operating System Concepts (Silberschatz, Abraham – 2012 – Wiley) (9th ed)
3. Optimizing subroutines in assembly language: An optimization guide for x86 platforms (Fog, Agner – 2013 – electronic, http://www.agner.org/optimize/)
4. Windows Operating System Internals Curriculum Resource Kit (CRK) (Microsoft – 2006 – electronic, MSDNAA)
5. Several sites dedicated to OS development (e.g. http://wiki.osdev.org/).
6. Several specifications regarding HW interfaces or devices (e.g. ATA, RTC, PIC, ..)

10 Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

This course was designed as a structure and content based on discussions with representatives of companies (e.g. BitDefender) directly involved in the development of security solutions. This course covers a series of knowledge that is necessary in developing methods to secure systems at a level close to the physical machine.

11. Evaluation

Activity type	Assessment criteria	Assessment methods	Weight in the final grade
Course	Ability to solve domain-specific problems, attendance, activity during class	Written exam - summative and presentation of different subjects / paper in the course's field during semester time - continuous.	70%
Laboratory	Ability to solve domain-specific problems, attendance, activity during class	Evaluate lab activity. Evaluate lab assignments (homework). Evaluate solutions of problems given in a final lab exam.	30%

Minimum standard of performance:

Lecture. Attending minimum 50% of lecture classes, to be allowed to take the final examination. Knowledge of the main protection mechanisms offered by the x86-64 architecture. Knowledge of the main principles of design of operating systems. Minimum final grade must be 5 for the exam to be considered passed.

Lab. Attending all lab classes (one lab could be recovered during the semester, and one more during re-examination sessions). The ability to use the acquired knowledge to develop components within an operating system. This kind of assessment could happen in relation to assignments given during semester or subjects given during the final lab evaluation.

Minimum laboratory grade 5.

Minimum exam grade 5.

Final grade=Note exams*0.7+Laboratory grade*0.3

Promotion criterion: minimum 5 at the final grade

Date of filling in: 01.09.2025	**Responsible**	**Title First name Last name**	**Signature**
	Course	Prof. dr. eng. Anca HÂNGAN	
	Applications	Prof. dr. eng. Anca HÂNGAN	

Date of approval in the department 17.09.2025	Head of department, Prof.dr.eng. Rodica Potolea
Date of approval in the faculty 19.09.2025	Dean, Prof.dr.eng. Vlad Mureșan