INTERNATIONAL EUROPEAN UNIVERSITY
Education and Research Institute "European Business School"
Department of Information Technology

Approved by
The Scientific and Methodical Council of the University_____

Chair of SMC _____

WORKING PROGRAM OF THE ACADEMIC DISCIPLINE:

OPERATING SYSTEMS

Knowledge area: 12 Information Technology

Specialty: 121 Software Engineering

Educational program: 121 Software Engineering

Discipline status: Compulsory

Kyiv – 2023

The working program of the Operating systems academic discipline is based on the 121 Software Engineering educational and professional program for the first (Bachelor) level of the 121 Software Engineering specialty approved by the University Academic Council on May 26, 2022, protocol No. 4.

Developer: Oleksandr Nesterenko, Doctor of Science (Techn.), professor

Reviewers: Ivan Kazachkov, Doctor Science (Techn.), professor
Oleksandr Falovskyi, PhD in Technology

Guarantor of the educational program: Oleksandr Nesterenko, Doctor Science (Techn.), professor

The working program of the academic discipline is reviewed and approved by the Department of Information Technology, protocol dd. August 31, 2023, No. 1.

Head of the Department
Doctor Science (Techn.),
professor  O.V. Nesterenko,

The program is reviewed and approved by the Academic Council of the European Business School, protocol dd. September 11, 2023, No. 1.

Chair of the Academic Council
PhD in Economics, associate professor,
Acting Director
of the European Business School  Y.S. Remyha,

Registration No. _13/23_

INTRODUCTION

The **program of the Operating systems academic discipline** is designed according to the Higher Education Standard of Ukraine (hereinafter referred to as the Standard) of the knowledge area: 12 Information Technology, specialty: 121 Software Engineering.

Discipline description (annotation). This academic discipline is one of the professional disciplines for future software developers.

Table 1

<table>
<tr><td rowspan="2">Criteria</td><td rowspan="2">Knowledge area, training program, educational level</td><td colspan="2">Discipline characteristics</td></tr>
<tr><td>full-time mode of study</td><td>part-time mode of study</td></tr>
<tr><td>Number of credits – 4</td><td>Knowledge area:
12 INFORMATION TECHNOLOGY</td><td colspan="2"><u>Compulsory</u></td></tr>
<tr><td>Sections – 1</td><td rowspan="9">Specialty:
121 SOFTWARE ENGINEERING</td><td colspan="2">Year of training</td></tr>
<tr><td>Content sections – 1</td><td>2023</td><td>2023</td></tr>
<tr><td rowspan="3">Individual research task:</td><td colspan="2">Semester</td></tr>
<tr><td>3rd</td><td>3rd</td></tr>
<tr><td colspan="2">Lectures</td></tr>
<tr><td rowspan="2">Total amount of hours – 120</td><td>16 hours</td><td>6 hours</td></tr>
<tr><td colspan="2">Practical and laboratory classes</td></tr>
<tr><td>32 hours</td><td>4 hours</td></tr>
<tr><td rowspan="4">Weekly load:
class hours – 3
independent work of students – 4</td><td rowspan="4">Educational level:
Bachelor</td><td colspan="2">Independent work</td></tr>
<tr><td>72 hours</td><td>110 hours</td></tr>
<tr><td colspan="2">Type of control:</td></tr>
<tr><td>exam</td><td>exam</td></tr>
</table>

Subject matter of the academic discipline: principles and algorithms underlying the development of operating systems, their internal structure.

Interdisciplinary links: The academic discipline is related to such disciplines as Fundamentals of programming, Object-oriented programming, Computer discrete mathematics, Computer networking organization, Data algorithms and structures.

1. GOAL AND OBJECTIVES OF THE ACADEMIC DISCIPLINE

1.1. The **goal** of the Operating systems academic discipline is to provide students with knowledge of basic principles and algorithms underlying the development of operating systems, exploration of their internal structure, skills and

practical abilities in working with the basic tools of operating systems used in programming activities.

1.2. **Key objectives** of the Operating systems discipline:

- to expand students' knowledge of software management, both application and system, and the computer software and hardware interface;
- to introduce various types of operating systems and their architectural solutions to students;
- to reveal the place and role of operating systems in the development of information technology and systems;
- to make students understand the basic principles underlying the development of operating systems;
- to provide students with knowledge, skills, and abilities of software management using operating systems tools;
- to teach students to work with a certain class of system software;
- to develop students' skills in applying acquired knowledge to solve typical programming tasks.

1.3. **Competencies and learning outcomes** encouraged by the discipline (interrelation with the statutory content of student training stipulated in learning outcome terms of the Standard).

According to the Standard requirements, the discipline provides students with the following ***competencies:***

Table 2

Integral competence	Ability to solve complicated specialized tasks and practical problems in software engineering characterized by complexity and uncertainty of conditions.
General competencies	Understanding of the processes occurring in the computer's software environment. Ability to apply knowledge in practical programming situations.
Specialized (professional, subject) competencies	Understanding of the interactions in the computer's software and operating environment. Ability to think algorithmically and logically. Understanding of the possibilities of practical application of operating systems methods and tools in developing application software.

Specification of competencies according to the National Qualifications Framework descriptors in the Competency matrix form is given in Table 3.

Table 3

Competency matrix

No.	Competence	Knowledge	Skills / Abilities	Communication	Autonomy and responsibility
Integral competence					

1.	Ability to solve complicated specialized tasks and practical problems in software engineering characterized by complexity and uncertainty of conditions.	Theories of designing operating systems and interacting with application software	To use information technologies, basic system and application software to solve practical problems	Software interaction	Independent design and testing on the production site
General competencies					
2.	Understanding of the processes occurring in the operating system when processing information with application software. Ability to use knowledge in practical programming situations. Ability to search, process and analyze information for application in programming.	structure of operating systems, general principles of their functioning	to apply operations of interaction with the OS environment when developing application programs	Relation between theoretical and practical knowledge	Monitoring of information processing processes
Specialized (professional, subject) competencies					
3.	Understanding of the concept of processes and multitasking. Ability to think algorithmically and logically. Understanding of the practical application of operating system tools.	fundamentals of multiprogramming virtualization and distributed computing, relevant rules and functions of application programming	To use software tools in the operating environment	Application of parallel working technologies	Description of information processes

Integrated final program learning outcomes encouraged by the academic discipline:

Program learning outcomes Bachelor's qualifying paper

Learning outcomes:

After learning the discipline, students should

know:

– structure of operating systems, general principles of their functioning;

– purpose, functional capabilities and rules of using basic system programs;

– functional capabilities and rules of using the OS environment when creating general-purpose application programs;

– theory of building operating systems and its interaction with application software;

– best practices of using various operating system technologies.

be able to:

– use operating system tools, basic system calls to solve practical tasks of application programming;

– perform operations on interaction between OS and application programs using OS services;

– use functions and libraries of software tools for application development;

– apply technologies of working with the operating system, analyze system situations.

2. INFORMATION CAPACITY OF THE ACADEMIC DISCIPLINE

The Operating systems academic discipline consists of 120 hours / 4 ECTS credits.

TOPIC 1. PURPOSE, FUNCTIONS, AND ARCHITECTURE OF OPERATING SYSTEMS

TOPIC 2. PROCESSES AND FLOWS

TOPIC 3. MEMORY MANAGEMENT IN OPERATING SYSTEMS

TOPIC 4. INPUT/OUTPUT AND FILE SYSTEM

TOPIC 5. MULTIPROGRAMMING

TOPIC 6. PARALLEL OPERATION

TOPIC 7. VIRTUALIZATION

TOPIC 8. SECURITY

3. STRUCTURE OF THE ACADEMIC DISCIPLINE

Topics	Amount of hours				
	Total	including			
		Lectures	Practical/ seminars	Laboratory work	Independent work
1 2 3 4					
Topic 1. Purpose, functions, and architecture of operating systems	15	2		4	9
Topic 2. Processes and flows	15	2		4	9
Topic 3. Memory management in	15	2		4	9

operating systems					
Topic 4. Input/output and file system	15	2		4	9
Topic 5. Multiprogramming	15	2		4	9
Topic 6. Parallel operation	15	2		4	9
Topic 7. Virtualization	15	2		4	9
Topic 8. Security	15	2		4	9
Total	120	16		32	72
Consultations					
Exam					
Total hours					

4. TOPICS OF LECTURES

Topic No.	Lecture topic and list of key questions
1	PURPOSE, FUNCTIONS, AND ARCHITECTURE OF OPERATING SYSTEMS 1.1. Introduction to the discipline. 1.2. Functions, classification, and components of OS. 1.3. Operating systems structure. 1.4. System challenges. 1.5. Concept of the operating and software environment.
2	PROCESSES AND FLOWS. 2.1. Basic information about processes. Process management. 2.2. Process interaction. 2.3. Synchronization primitives. 2.4. Deadlock. 2.5. Interprocess interaction problems. 2.6. Flows. 2.7. Signals.
3	MEMORY MANAGEMENT IN OPERATING SYSTEMS. 3.1. General information about using computer memory. 3.2. Memory sharing technology. 3.3. Virtual memory.
4	INPUT/OUTPUT AND FILE SYSTEM. 4.1. Input/output system. 4.2. File system. Concept of file. 4.3. Logical organization of the file system.
Topic No.	Lecture topic and list of key questions
5	MULTIPROGRAMMING. 5.1. Key definitions and characteristics. 5.2. Deadlocks. 5.3. Interrupt-based multiprogramming.
6	PARALLEL OPERATION. 6.1. Organization of parallel operation of devices and processor. 6.2. Multiprocessor systems. 6.3. Multicomputers.

7	VIRTUALIZATION. 7.1. Principles of virtualization. 7.2. Efficient virtualization technology. 7.3. Memory and input/output virtualization.
8	SECURITY. 8.1. Threats and attackers. 8.2. OS security principles. 8.3. Managing access to resources. 8.4. Authentication. 8.5. Hacking methods. 8.6. Malicious software. 8.7. Protection methods and techniques.

5. TOPICS OF LABORATORY AND PRACTICAL CLASSES

No.	Laboratory work (computer workshop)	Amount of class hours
1	1. Exploration of Windows OS by designing a Windows application.	4
2	2. Exploration of Windows OS processes.	4
3	3. Exploration of the Windows memory management system.	4
4	4. Exploration of particularities of NTFS file system implementation.	4
5	Fundamentals of programming and synchronization tools in multitasking OS	4
6	Developing a framework for parallel programs in Windows	4
7	Installing Oracle VM Virtualbox on Windows	4
8	Microsoft Azure cloud service	4
	Total	32

6. INDEPENDENT WORK

No.	Topic for independent learning	Amount of hours
	1. Architecture of operating systems	
1	Architecture of operating systems (Windows, UNIX and Linux, Android).	5
2	The main stages of operating systems development.	4
	2. Tasks of deadlock processes	
3	Dekker's algorithm	3
4	Peterson's algorithm	3
5	Algorithm for solving the dining philosophers problem	3
	3. Memory page replacement problem	
6	Adaptive cache replacement algorithm (ARC)	4
No.	Topic for independent learning	Amount of hours
	4. File systems	
7	Features of UNIX and Linux file systems	6
	5. Deadlock tasks in multiprogramming	
8	Disk space deadlock	4
9	Resource deadlock	5

6. Multiprocessor systems		
10	Multiprocessor planning	9
7. Virtualization		
11	Exploring the VMware Workstation hypervisor	9
8. Security		
12	Exploring research on virtualization and clouds for security.	9
		72

7. TRAINING METHODS

Teaching the Operating systems discipline, one uses information and practical training methods: classical lectures, laboratory and practical classes using simulation laboratory workshops, as well as consultations on the accomplishment of independent work of students, written assignments.

Methods of learning and cognitive activity: explanatory and illustrative method, reproductive method, problem presentation method, partially exploratory or heuristic method, research method.

Methods of stimulation and motivation of learning and cognitive activity: inductive and deductive teaching methods; methods of stimulation and motivation of learning.

8. CONTROL METHODS

The plan of the Operating systems discipline implies carrying out of current and final control.

Current control is the assessment of the level of knowledge, skills and abilities of students carried out during the educational process by conducting a written survey at the end of sections (module colloquium). Final control is carried out in the form of an exam.

9. FORM OF STUDENT PERFORMANCE FINAL CONTROL

The form of final control is the **exam** taken on-campus (or in the form of computer test in case of a specific situation) in the period stipulated by the Dean's office or according to the individual schedule stipulated by the curriculum.

10. SCORING SYSTEM

Scoring during the semester

No.	Type of activity	Number of points per didactic unit	Number	Total points
1	Accomplishment of tests	2	8	16
2	Accomplishment of laboratory	4	8	32

	works			
3	Accomplishment of independent tasks	1.5	8	12
Maximum grade				60

General assessment of student knowledge due to current control

The results of current control of student knowledge are assessed in general ranging from **0** to **60** points.

Students are allowed to final control if they fulfil the requirements of the training program and obtain at least **36** points for the current learning activity.

Final assessment of student knowledge

Final assessment of student knowledge is conducted in the form of **exam**.

Allocation of assessment points during final control in the academic discipline

Grade in points for final assessment	Grade according to the national scale
35-40	Excellent
21-34	Good
10-20	Satisfactory
less than 10	Fail

Assessing the answer to the particular question, one takes into account the following gaps and mistakes:

– untidy preparation of work (nonconventional abbreviations, unclear handwriting, use of pencils instead of clear inks) (minus **2** points);

– incorrectness in certain economic categories and definitions (minus **4** points).

Assessment criteria for answers to theoretical questions of the exam card:

1. The full answer to the question rated as ***excellent (40 points)*** should correspond to the following requirements:

– detailed, comprehensive representation of the content of the given problem;

– full list of economic categories and laws required to reveal the question;

– ability to carry out a comparative analysis of various theories, concepts, approaches and make logical conclusions and generalizations;

– ability to apply methods for the scientific analysis of economic phenomena, processes and characterize their features and forms of appearance;

– demonstration of the ability to express and reason your own attitude to alternative views on this question;

– use of relevant actual and statistical data, knowledge of dates and historical periods that prove key points of the answer.

2. The answer to the question is rated as ***good (30 points)*** if:

– the answer for the highest grade does not reveal at least one of the above-mentioned points (if it is definitely required to reveal the question comprehensively), or if:

– revealing the question correctly in general according to the above-mentioned requirements, one makes some mistakes while using digital materials.

3. The answer to the question is rated as ***satisfactory (20 points)*** if:

– the answer for the highest grade does not reveal four and more points specified in its requirements (if they are required to reveal the question comprehensively);

– there are four or more gaps characterizing individually assessment criteria;

– conclusions made during the answer do not correspond to correct or generally defined ones with the absence of evidence for opposite facts given in the answer;

– the character of the answer gives reason to state that persons fail to understand the question properly or do not know the correct answer, and that is why fail to answer in actual fact, making serious mistakes.

National and ECTS grading scale

<table>
<tr><td rowspan="2">Sum of points for all types of educational activities</td><td rowspan="2">ECTS grade</td><td colspan="2">Grade according to the national scale</td></tr>
<tr><td>for exam, term paper, practical training</td><td>for Pass/Fail test</td></tr>
<tr><td>90-100</td><td>A</td><td>excellent</td><td rowspan="5">pass</td></tr>
<tr><td>82-89</td><td>B</td><td rowspan="2">good</td></tr>
<tr><td>74-81</td><td>C</td></tr>
<tr><td>66-73</td><td>D</td><td rowspan="2">satisfactory</td></tr>
<tr><td>60-65</td><td>E</td></tr>
<tr><td>30-59</td><td>FX</td><td>fail with possible repeated pass</td><td>fail with possible repeated pass</td></tr>
<tr><td>1-29</td><td>F</td><td>fail with obligatory repeated learning of the discipline</td><td>fail with obligatory repeated learning of the discipline</td></tr>
</table>

The overall final grade in points according to the national and ECTS scales is put into the examination and test register, academic card and credit book of students.

11. METHODICAL SUPPORT:

- working program of the discipline;
- electronic course on the e-learning platform;
- plans of lectures, practical classes and independent work of students;
- key points of discipline lectures;
- methodical guidelines to laboratory and practical classes for students;
- methodical materials for independent work of students;
- test tasks for lecture topics;
- list of questions for the exam.

12. RECOMMENDED READING

Primary:

1. Bacon J., Harris T. Operating systems. / Translated from English. K.: BHV Publishing group; St. Petersburg: Piter, 2004. 800 p.
2. Deitel P.J. et al. Operating systems. Basics and Principles. 3rd edition: Translated from English by Binom-Press, 2011. 1024 p.
3. Molay B. Understanding UNIX/LINUX Programming: A Guide to Theory and Practice / Translated from English. M.: Kudic-Obraz, 2004. 576 p.
4. Pomerantz O. The Linux Kernel Module Programming Guide / Translated from English. M.: Kudic-Obraz, 2000. 112 p.
6. Stevens, W.R. UNIX: Process Interaction / Translated from English. St. Petersburg: Piter, 2003. 576 p.
7. Stevens, W.R. et al. UNIX Network Programming, 3rd edition / Translated from English. St. Petersburg: Piter, 2007. 1040 p.
8. Stallings W. Operating Systems. 4th edition.: Translated from English. – M.: Williams Publishing House, 2002. 848 p.
9. Tanenbaum A. Operating Systems. 3rd edition / Translated from English. St. Petersburg: Piter, 2010. 1120 p.
10. Operating systems: textbook / M.F. Bondarenko, O.H. Kachko. Kh.: SMITH Company, 2008. 432 p.
11. Habrusiev V. Y., Lapinskyi V.V., Nesterenko O.V. Fundamentals of operating systems: Core, process, flow. Ternopil: Bohdan, 2007. 94 p.
12. Hart J. Win32 System Programming / Translated from English. – M.: Williams, 2001. 464 p.

Additional:

1. Mitchell M. et al. Advanced Linux Programming / Translated from English. – M.: Williams, 2002. 288 p.
3. Matviienko M.P., Rozen V.P., Zakladnyi O.M. Computer architecture. K.: Lira-K Publishing house, 2013. 264 p.
4. Fusco, J. The Linux Programmer's Toolbox / Translated from English. St. Petersburg: Piter, 2010. 448 c.
5. Haseman C. Android Essentials. Apress, 2008. 116 p.
6. Ward B. How Linux Works / Translated from English. Piter, 2016. 864 p.
7. Shotts W. The Linux Command Line: A Complete Introduction / Translated from English. Piter, 2017. 480 p.
8. Operating systems: study guide for students of 123 Software engineering / V. H. Zaitsev, I. P. Drobiazko; Igor Sikorsky Kyiv Polytechnic Institute. 2019. – 240 c. [Electronic resource]
9. Operating systems: study guide [edited by V. M. Rudnytskyi] / I. M. Fedotova-Piven, I. V. Myronets, O. B. Piven, S. V. Sysoienko, T. V. Myroniuk; Cherkasy State Technological University. – Kharkiv.: DISA PLUS

LLC, 2019. 216 p.

Information resources

1. https://www.microsoft.com/uk-ua/
2. https://stud.com.ua/informatika/
3. https://dou.ua/
4. http://it.ridne.net/
5. https://www.kernel.org/