ISO 9001 CERTIFIED
SLC. MS.040. LSO9001. 1693

INTERNATIONAL EUROPEAN UNIVERSITY

SYLLABUS

INTERNATIONAL EUROPEAN UNIVERSITY

EUROPEAN SCHOOL OF BUSINESS

Operating systems

2022-2023

SYLLABUS

Discipline	
	Operating systems
Lecturer (s)	
	Oleksandr Nesterenko, Doctor of Technology, associate professor, professor at the Department of Information Technology
Lecturer's profile	
	https://it.ieu.edu.ua/pro-yeash/struktura-yeash/kafedra-informatsiinykh-tekhnolohii/sklad#zzz-001
Consultations	
offline consulting	–
online consulting	Thursday 3 p.m. – 5 p.m. online consulting https://meet.google.com/mow-siow-uhj
Contact number	
	+380977572796
E-mail	
	oleksandr_nesterenko@ieu.edu.ua
Discipline page	
	https://dist.ieu.edu.ua/course/view.php?id=571

Form of final control	test	differentiated test	exam
	☐	☐	☒

SYLLABUS

1 Brief discipline annotation

The Operating systems academic discipline is aimed at training Bachelors of knowledge area: 12 Information technology, specialty: 121 Software engineering. It is one of the professional disciplines for future software developers.

2 Background for studying discipline

The discipline program is based on knowledge obtained while learning such disciplines as Fundamentals of programming, Computer discrete mathematics, Information technology methods and tools.

3 Goal and objectives of the discipline

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.

Objectives of the 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.

4 Learning outcomes

PLO1. To analyze, purposefully search for, and select information and reference resources and knowledge required to solve professional tasks, taking into account the latest achievements of science and technology.

PLO7. To know and apply fundamental concepts, paradigms, and underlying principles of software engineering language, hardware, and computational tools.

PLO17. To be able to use component-based software development techniques.

5 Credits

4 Credits

6 Discipline structure

Topics	Type of classes/hours		
	Lectures	**Laboratory work**	**Independent work**
Topic 1. Purpose, functions, and architecture of operating systems	**2**	**4**	**9**
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			
1.6. Operating systems architecture (Windows, UNIX and Linux, Android).			
1.7. Key stages of OS development			
Topic 2. Processes and flows	**2**	**4**	**9**
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			
Topic 3. Memory management in operating systems	**2**	**4**	**9**
3.1. General information about using computer memory			
3.2. Memory sharing technology			

6 Discipline structure

Topics	Type of classes/hours		
	Lectures	Laboratory work	Independent work
3.3. Virtual memory			
Topic 4. Input/output and file system	**2**	**4**	**9**
4.1. Input/output system			
4.2. File system. Concept of file			
4.3. Logical organization of the file system			
Topic 5. Multiprogramming	**2**	**4**	**9**
5.1. Key definitions and characteristics			
5.2. Deadlocks			
5.3. Interrupt-based multiprogramming			
Topic 6. Parallel operation	**2**	**4**	**9**
6.1. Organization of parallel operation of devices and processor			
6.2. Multiprocessor systems			
6.3. Multicomputers			
Topic 7. Virtualization			
7.1. Principles of virtualization			
7.2. Efficient virtualization technology			
7.3. Memory and input/output virtualization			
Topic 8. Cloud computing operating systems	**2**	**4**	**9**
8.1. Distributed systems			

6 Discipline structure

Topics	Type of classes/hours		
	Lectures	**Laboratory work**	**Independent work**
8.2. Cloud as a service			
8.3. Migration of virtual machines			
8.4. Current research in virtualization and clouds			

7 List of obligatory tasks

1. Exploration of Windows OS by designing a Windows application.
2. Exploration of Windows OS processes.
3. Exploration of the Windows memory management system.
4. Exploration of particularities of NTFS file system implementation.
5. Fundamentals of programming and synchronization tools in multitasking OS.
6. Modeling of parallel systems and elimination of deadlocks.
7. Installation of the operating system on a virtual machine.
8. Development of a web app using the Windows Azure cloud service.

8 List of elective tasks

1. To conduct the research of Dekker's and Peterson's algorithms; solve the dining philosophers problem and cache replacement algorithm. To describe the features of the subject area under study, the basic necessary data, the decision-making process in OS based on the current tasks.
2. Features of UNIX and Linux file systems.
3. Disk space deadlock.

9 Discipline features

Period of teaching	**Semester**	**International discipline integration**	**Year of study**	**Courses: general training/ professional training/elective**
1 semester	3	available	2	professional training

10 Hardware and software

Personal computer, Windows and Linux OS, office software packages, system utilities

11 Assessment system and requirements

As part of discipline teaching, one carries out the current and final control of students' knowledge. The final grade is given according to the total rating of students.
The results of the current control of students' knowledge is assessed in general between 0 and 60 points.
Students are admitted to the 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 students' knowledge is conducted in the form of exam.
The maximum amount of points that can be obtained during the exam is 40 points.
The overall points of the discipline are 100. The total grade for the discipline is given according to the national and European scale.

12 Discipline policy

Teaching of the discipline is based on cutting-edge educational technologies aimed at increasing the level of students' interest in the course, providing theoretical and practical knowledge of the discipline.
To activate the learning and cognitive activity of students, the discipline includes the consolidation of knowledge obtained at the lecture and acquisition of practical skills in lecture topics during laboratory works and performance of independent work.

13 Absence policy

Points are not given for missed lectures. If students miss a laboratory work, they should perform all tasks of the missed laboratory work before the next laboratory work and present the results to the lecturer. The grade is reduced by one point.
Students who have missed classes without valid reasons and have not participated in current control activities are not admitted to the final semester control. In this case, a mark 'non-admission' is put in the exam record on the day of the exam.
Repeated taking of the exam of the discipline is appointed in case of accomplishing all types of educational and independent work stipulated by the working program of the academic discipline and is carried out according to the approved schedule of academic failure liquidation.

14 Policy of late task performance

Tasks and laboratory works submitted later are assessed with a lower grade. The grade is reduced by one point for each week of lateness.

15 Academic integrity policy

Participants in the educational process rely on the academic integrity principles. One should provide references to sources of information when using someone else's ideas, statements, data, as well as verified information.

16 Recommended

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.

Internet resources:
1. Microsoft website. – Available at: https://www.microsoft.com/uk-ua/
2. Online textbooks for students. – Available at: https://stud.com.ua/informatika/
3. IT website. – Available at: https://dou.ua/
4. Information Technology. Analytical Materials Journal. – Available at: http://it.ridne.net/
5. Linux website. – Available at: https://www.kernel.org

17 Tips on successful study during the course

Note: examine lecture materials and perform tasks and laboratory works synchronously with the curriculum. Thus, your abilities and insistence will be the key to success!