



Fundamentals of Operating Systems

Syllabus

Requisites of the Course

Cycle of Higher Education	<i>First (Bachelor)</i>
Field of Study	<i>12 Information technologies</i>
Speciality	<i>121 Software Engineering</i>
Education Program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Type of Course	<i>Selective</i>
Mode of Studies	<i>full-time</i>
Year of studies, semester	<i>3 year, 5 semester</i>
ECTS workload	<i>36 hours for lectures, 18 hours for practicals, 66 hours for self-study</i>
Testing and assessment	<i>Credit, modular test, calendar control</i>
Course Schedule	<i>According to http://roz.kpi.ua/</i>
Language of Instruction	<i>English</i>
Course Instructors	<i>Lecturer: Ph.D., associate professor, Nataliya A. Rybachok, Linkedin: https://www.linkedin.com/in/nataliia-rybachok-0903bb252/ e-mail: rybachok.nataliia@i11.kpi.ua Telegram: https://t.me/Nataliia_Rybachok <i>Laboratory work: Ph.D., associate professor, Nataliya A. Rybachok</i></i>
Access to the course	<i>Google classroom.</i>

Program of educational discipline

1. Description of the educational discipline, its purpose, subject of study and learning outcomes

The study of the discipline "Fundamentals of Operating Systems" allows students to develop the competencies necessary for solving practical problems of professional activity, related to the use of IIS and Nginx web servers, which are installed on virtual machines with operating systems of the Windows and Linux family, located on Azure cloud platforms.

***The purpose** of studying the discipline "Fundamentals of Operating Systems" is to provide students with the ability to independently create virtual machines, install web servers, deploy and update static sites, test the load on web servers, manage the OS using RDP protocols, SSH and use a mobile device to manage the VM and access the OS.*

***The subject** of the discipline "Fundamentals of Operating Systems" is cloud technologies, operating systems, web servers, system utilities, access protocols to remote OS.*

*The study of the discipline "Fundamentals of Operating Systems" contributes to the formation of the following **General competencies (GC)** for students according to the educational program:*

GC5 Ability to learn and use modern knowledge.

GC6 Ability to search, process and analyze information from various sources.

The study of the discipline "Fundamentals of Operating Systems" contributes to the formation of the following **professional competence (PC)** in students according to the educational program:
PC8 Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.

The study of the discipline "Fundamentals of operating systems" contributes to the formation of the following program learning outcomes (PLO) for students according to the educational program:

PLO01 To analyze, purposefully search and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according to the relevant educational program)

The success of studying the discipline "Fundamentals of Operating Systems" does not depend on other disciplines of the curriculum of bachelor's training in the specialty 121 Software Engineering.

The theoretical knowledge and practical skills obtained during the mastering of the discipline "Fundamentals of Operating Systems" ensure the successful completion of course and diploma projects, master's theses in the specialty 121 Software Engineering.

3. Content of the academic discipline

The discipline "Fundamentals of Operating Systems" involves the study of the following topics:

Topic 1. Cloud technologies and Deploying Websites

Topic 2. Server load testing and OS resource monitoring

Modular test

Credit

4. Educational materials and resources

Basic References:

1. Azure documentation

<https://learn.microsoft.com/en-us/azure/>

2. AWS Academy Data Center Technician Course

<https://awsacademy.instructure.com/courses/12156/>

3. Methodological instructions for performing laboratory work in the discipline "Fundamentals of operating systems".

Use to master the practical skills of the discipline.

The materials are in Google classroom.

Additional References:

4. Local File Systems (Windows) | Microsoft Learn

[https://learn.microsoft.com/en-us/previous-versions/windows/desktop/legacy/aa364407\(v=vs.85\)](https://learn.microsoft.com/en-us/previous-versions/windows/desktop/legacy/aa364407(v=vs.85))

5. SSH Command - Usage, Options, Configuration

<https://www.ssh.com/academy/ssh/command>

6. Home : The Official Microsoft IIS Site

<https://www.iis.net/>

7. What is STRESS Testing in Software Testing?

<https://www.guru99.com/stress-testing-tutorial.html>

8. Windows Server performance troubleshooting documentation

<https://learn.microsoft.com/en-us/troubleshoot/windows-server/performance/performance-overview>

Educational content

5. Methodology

No	Training session type	Lesson description
<i>Topic 1. Cloud technologies and Deploying Websites</i>		
1	<i>Lecture 1. Introduction to cloud technologies</i>	<i>Models of cloud services. Major cloud providers (AWS, Azure, GCP). Comparison of public, private and hybrid clouds.</i> 1
2	<i>Lecture 2. Azure cloud platform services</i>	<i>Azure cloud platform services</i> 1
3	<i>Task 1. How to create a student account on the Azure platform and access the Coursera platform</i>	<i>Create an account on the Ill.kpi.ua domain; create an account on the Azure cloud platform and activate a student subscription; create an account on the Coursera platform</i> 3
4	<i>Lecture 3. Basic concepts of cloud computing and virtualization</i>	<i>The concept of the main components of architecture, regions of availability, virtual machines and their resources, Resource manager</i> 1
5	<i>Task 2. How to create a Windows Server 2019 Gen2 VM on the Azure portal and manage VM via mobile app</i>	<i>Create a Windows VM on the Azure cloud platform, configure RDP access, use mobile app for managing VM;</i> 3
6	<i>Lecture 4. Features of creation of VM on Azure</i>	<i>Subscription, resource group, resource, OS disk, temporary disk</i> 1
7	<i>Lecture 5. OSI model and protocol stack TCP/IP</i>	<i>MAC address, IP addressing, public and private addresses, main TCP and UDP ports, CIDR</i> 2
8	<i>Task 3. How to find out information about a VM on Azure portal and main OS</i>	<i>define VM parameters from the OS; define public and private addresses; define public IP addresses locations</i> 3
9	<i>Lecture 6. Windows File system structure</i>	<i>Windows File system structure</i> 4
10	<i>Task 4. How to set up IIS on Windows Virtual Machine hosted on Azure</i>	<i>Installing IIS, configuring VM network parameters (ICMP, HTTP)</i> 2
11	<i>Lecture 7. Windows File system structure and SSH</i>	<i>SSH. Structure, placement in the file system, settings</i> 5
12	<i>Task 6. How to set up IIS on Windows Virtual Machine hosted on Azure</i>	<i>Configure IIS</i> 3

13	<i>Task 7. How to Deploy Website on Azure Windows VM</i>	<i>OpenSSH Server, Filezilla</i> 3
14	<i>Lecture 8. Windows File system structure and IIS</i>	<i>IIS. Structure, placement in the file system, settings</i> 6
<i>Topic 2. Server load testing and OS resource monitoring</i>		
15	<i>Lecture 9. Software for stress testing</i>	<i>Stress testing. Jmeter. Counters.</i> 7
16	<i>Task 8. How to create a Performance Test Plan in JMeter</i>	<i>Evaluation of response of sites at load testing by means of the JMeter utility</i> 3
17	<i>Lecture 10. Architectural features of modern OS</i>	<i>OS resources, modularity, basic OS subsystems</i> 2
18	<i>Task 9. Using Autoruns to Deal with Startup Processes and Malware</i>	<i>Assess the overall state of the system using the utility "Task Manager".</i> 3
19	<i>Lecture 11. Startup Processes on OS</i>	<i>Startup Processes on OS, Windows Register, System Internals Utilities</i> 3
20	<i>Lecture 11. Monitoring of OS subsystems</i>	<i>Counters of OS subsystems and their limit values</i> 1
21	<i>Lecture 12. System Software for load testing</i>	<i>Utilities "Task Manager", "Performance Monitor", "Resource Monitor".</i> 8
22	<i>Task 10. How to use the Resource Monitor in Windows</i>	<i>View and analyze the use of system resources for each process using the utility "Resource Monitor".</i> 3
	<i>Modular test</i>	

2. Self-study (supplementary reading)

No	<i>The name of the topic that is submitted for independent study</i>	<i>Hours of study</i>	<i>References</i>
1	<i>Preparation to the lectures</i>	20	1,2,4,5,6,7,8
2	<i>Preparation to the tasks</i>	26	2
24	<i>Preparation to the modular test</i>	10	1,2,4,5,6,7,8
25	<i>Preparation to the credit</i>	10	1, 2, 3, 4, 5, 6,7,8

Policy and Control

7. Course policy

Students are required to be on time to attend all the lessons according to the timetable, and adhere to the code of ethical conduct. To successfully complete the course students need to study coursebook

materials and use learning resources available in print, in digital form, and online. Regular attendance at the lessons, self-study and continuous revisions are essential for completion of the course.

According to the Code of Honour of the university (<https://kpi.ua/code>) students are expected to be self-disciplined, well-behaved, considerate, honest and responsible.

During the course, writing an essay/ a conference abstract, presenting at a conference, taking part in a language contest/ university project will bring rewarding points added to the student's performance score.

8. Monitoring and grading policy

During the semester, students perform 10 computer classes. The maximum number of points for computer class: 6 points.

Points for computer class are calculated for:

- quality of the computer classes: 0-4 points;
- answer during the defense of the computer classes: 0-2 points.

Performance evaluation criteria:

- 4 points – the computer class is done qualitatively, in full;
- 3 points – the computer class is done qualitatively, in full, but has flaws;
- 2 points – the computer class is completed in full, but contains minor errors;
- 1 points – the computer class is completed in full, or contains significant errors;
- 0 points – the computer class is not fully completed.

Answer evaluation criteria:

- 2 points – the answer is complete, well-argued;
- 1 point – in general, the answer is correct, but has flaws or minor errors;
- 0 points – there is no answer or there are significant errors in the answer.

The maximum number of points for performing and defending computer classes:

6 points × 10 = 60 балів.

The assignment for **the modular test** consists of 40 tests questions. The answer to each question is worth 1 points.

Evaluation criteria for each test question:

- 1 – the answer is correct;
- 0 point – no answer or the answer is incorrect.

The maximum number of points for a modular test:

1 points × 40 theoretical test question = 40 points.

The rating scale for the discipline is equal to:

$R = 60 \text{ points} + 40 \text{ points} = 100 \text{ points}.$

Calendar attestation of students (8 and 14 weeks of semesters) for the discipline is carried out according to the value of the current rating of the student at the time of attestation. If the value of this rating is not less than 50% of the maximum possible at the time of certification, the student is considered satisfactorily certified. Otherwise, in the attestation statement is set "unsatisfactory".

A necessary condition for obtaining a test by a student is the performance modular control work and defense of all laboratory works with a sum of at least 60%. Students who do not have academic debt can also increase their grades by taking a final test.

The final performance score or the results of the Final/ Pass Test are adopted by university grading system as follows:

<i>Score</i>	<i>Grade</i>
100-95	<i>Excellent</i>
94-85	<i>Very good</i>
84-75	<i>Good</i>
74-65	<i>Satisfactory</i>
64-60	<i>Sufficient</i>
Below 60	<i>Fail</i>
Course requirements are not met	<i>Not Graded</i>

9. Additional information about the course

Syllabus of the course

Is designed by teacher PhD, associate professor, Nataliya A. Rybachok,

Adopted by Computer Systems Software Department (protocol № 8 from 25.01.23)

Approved by the Faculty Board of Methodology (protocol № 6 from 27.01.23)