Національний технічний університет України
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Емблема кафедри (за наявності)

Department of Computing Technics

Design and Implementation of Operating Systems

Syllabus

Requisites of the Course	
Cycle of Higher Education	***First cycle of higher education (Bachelor's degree)***
Field of Study	*12 Information Technologies*
Speciality	*123 Computer Engineering*
Education Program	*Computer Systems and Networks*
Type of Course	*Selective*
Mode of Studies	*Full-time*
Year of studies, semester	*4 year (7 semester)*
ECTS workload	*4 credits (ECTS). Time allotment – 108 hours, including 36 hours of lectures, 18 hours of practice, and 54 hours of self-study.*
Testing and assessment	*Test*
Course Schedule	*1.5 classes per week by the timetable http://rozklad.kpi.ua/*
Language of Instruction	*English*
Course Instructors	*Lecturer: senior lecturer, Andrey Simonenko, comsys.spz@gmail.com* *Practice: senior lecturer, Andrey Simonenko, comsys.spz@gmail.com*
Access to the course	*https://drive.google.com/drive/folders/17zhNyNEvwtkHPXgndj5gkn3OBfwjM9Fy*

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

What will be studied. *Principles of functioning, architecture and implementation of kernels of general-purpose operating systems will be studied. Implementations of Unix-like operating system kernels and implementations of other kernel types are taken as a basis, that is, the discipline is not focused on any one specific operating system. Thorough information on tasks and methods of solving them in the operating system kernel at a low level of implementation is provided. The discipline consists of the following parts: kernel and processes, support for multithreaded programs, file systems, memory management.*

Why it is interesting/necessary to study. *It is advisable to study this discipline for those who will develop operating system kernels or parts of operating system kernels. This discipline is also useful for system programmers and application programmers for an in-depth understanding of the functioning of the operating system, that will allow to develop more efficient programs.*

What you can learn (learning results). *Prepare to develop and understand the source code of parts of general purpose operating system kernels that implement process and thread management, system calls, file systems, memory management.*

How to use the acquired knowledge and skills (competencies). *The acquired knowledge can be used in the development of kernels of operating systems, to support the source code of existing operating system kernels, in the development of effective system and application programs.*

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

Understanding Assembler, understanding C or C++ to read examples in lectures. Basic knowledge of the disciplines Programming, Data Structures and Algorithms, Computer Architecture, System Programming.

3. Content of the course

Part 1. Kernel and processes

Topic 1.1. Definition of the OS kernel
Topic 1.2. System modes and contexts
Topic 1.3. Switching to kernel mode
Topic 1.4. Hardware interrupts handling
Topic 1.5. System calls
Topic 1.6. Involuntary context switching
Topic 1.7. Voluntary context switching
Topic 1.8. Signals
Topic 1.9. Monolithic kernel and microkernel
Topic 1.10. Process states diagram

Part 2. Support for multithreaded programs

Topic 2.1. Definition of multithreaded program
Topic 2.2. N:1 mode
Topic 2.3. 1:1 mode
Topic 2.4. N:M mode

Part 3. File systems

Topic 3.1. Definition of FS
Topic 3.2. Virtual File System
Topic 3.3. Stackable FS
Topic 3.4. FS for flash memory
Topic 3.5. Requirements for modern local FS

Part 4. Memory Management

Topic 4.1. Segmented memory organization
Topic 4.2. Virtual memory
Topic 4.3. Page replacement algorithms
Topic 4.4. Shared memory

4. Coursebooks and teaching resources

1. *Andrew Tanenbaum, Herbert Bos. Modern Operating Systems, 4th edition - Pearson, 2014. - 1136 p.*
2. *Daniel P. Bovet, Marco Cesati. Understanding the Linux Kernel, 3rd edition - O'Reilly Media, 2005. - 944 p.*
3. *Richard McDougall, Jim Mauro. Solaris Internals: Solaris 10 and OpenSolaris Kernel Architecture, 2nd edition - Prentice Hall, 2006. - 1020 p.*
4. *Chris Cooper, Chris Moore. HP-UX 11i Internals - Prentice Hall, 2004. - 432 p.*
5. *Amit Singh. Mac OS X Internals: A Systems Approach. - Addison-Wesley Professional, 2006. - 1641 p.*

Educational content

5. Methodology

Parts, topics	*Total, h*	*Lectures, h*	*Laboratory works, h*	*Self-study, h*
Part 1. Kernel and processes *Topic 1.1. Definition of the OS kernel* *Topic 1.2. System modes and contexts* *Topic 1.3. Switching to kernel mode* *Topic 1.4. Hardware interrupts handling* *Topic 1.5. System calls* *Topic 1.6. Involuntary context switching* *Topic 1.7. Voluntary context switching* *Topic 1.8. Signals* *Topic 1.9. Monolithic kernel and microkernel* *Topic 1.10. Process states diagram*	*26*	*10*	*2*	*14*
Part 2. Support for multithreaded programs *Topic 2.1. Definition of multithreaded program* *Topic 2.2. N:1 mode* *Topic 2.3. 1:1 mode* *Topic 2.4. N:M mode*	*20*	*8*		*12*
Part 3. File systems *Topic 3.1. Definition of FS* *Topic 3.2. Virtual File System* *Topic 3.3. Stackable FS* *Topic 3.4. FS for flash memory* *Topic 3.5. Requirements for modern local FS*	*32*	*10*	*8*	*14*
Part 4. Memory Management *Topic 4.1. Segmented memory organization* *Topic 4.2. Virtual memory* *Topic 4.3. Page replacement algorithms* *Topic 4.4. Shared memory*	*30*	*8*	*8*	*14*
Test	*3*			
Total	*108*	*36*	*18*	*54*

Laboratory works (three laboratory works to choose from):

1. General purpose memory allocator using tags (5 h)
2. General purpose memory allocator using slab allocation (7 h)
3. File system, part 1 (7 h)
4. File system, part 2 (5 h)
5. Page replacement algorithms (6 h)

6. Self-study

In the process of understanding topics from lectures and performing laboratory works students must consolidate the knowledge gained during lectures and practical work, self-study certain topics using information from Internet, deepen their knowledge for further study.

Self-study is the following:

1. Studying and understanding topics from previous lectures.
2. Performing tasks given for self-study.
3. Performing laboratory works.
4. Writing reports for laboratory works.

Policy and Assessment

7. Course policy

Course policy completely corresponds to rules and regulations published by KPI. To pass a laboratory work one must score 60% of the maximum number of points for it. To be admitted to the test, one must pass all laboratory works. To obtain the first attestation it is necessary to have credited the first laboratory work. To obtain the second attestation it is necessary to have credited the first and second laboratory works. The number of attempts to pass any laboratory work is not limited. Checks of laboratory works are performed according to the group timetable. If in the performed laboratory work there are errors or non-compliance with the conditions of the laboratory work and if one refuses to correct errors or non-compliance, the laboratory work is not credited or credited with lower score.

8. Monitoring and grading policy

According to regulations published by KPI maximum number of 100 points is evenly divided between laboratory works. Students who have fulfilled all the conditions for admission to the test, i.e. have a rating of 60 points and above, receive a grade corresponding to their rating. Students who wish to improve their rating can write a credit control work at the final scheduled class of the discipline in the semester. The credit control work consists of four questions, the maximum number of possible points of 100 for the credit control work is evenly divided between these questions. The student receives the higher of the grades obtained by the results of the credit control work or by the rating.

The final performance score or the results of the Pass/Fail are adopted by KPI grading system as follows:

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

Syllabus of the course

Is designed by teacher senior lecturer, Andrey Simonenko

Adopted by Department of Computing Technics (protocol № 10 25.05.2022)

Approved by the Faculty Board of Methodology (protocol № 10 09.06.2022)