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

Кафедра програмного забезпечення комп'ютерних систем

SYSTEM PROGRAMMING

Syllabus

Details of the educational component	
Level of higher education	***First (Bachelor)***
Branch of knowledge	*12 Information Technologies*
Specialty	*121 Software Engineering*
Educational program	*Software Engineering of Multimedia and Information-Retrieval Systems*
Status of the educational component	*Elective*
Form of education	*Full-time*
A year of training	*3rd year of training, 5th semester*
The scope of the educational component	*Lectures: 36 hours, computer workshop: 18 hours, self-study: 66 hours.*
Semester control / control measures	*Credit, modular test, calendar control*
Schedule of classes	*According to the schedule for the spring semester of the current academic year (http://roz.kpi.ua/)*
Language of instructions	*English*
Information about head of the course / teachers	Lecturer: Ph.D., Associate Professor Yuriy Volodymyrovych Vuntesmeri, yv-ee@lll.kpi.ua Computer workshop: Ph.D., associate professor Yuriy Volodymyrovych Vuntesmeri, yv-ee@lll.kpi.ua
Course location	Google classroom. Access is given to registered students.

Program of educational component

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

The educational discipline "System Programming" is aimed at forming the student's knowledge and skills regarding modern software design methods of computing platforms as part of electronic devices of various purposes.

Appropriate theoretical and practical training forms skills in the implementation of typical control mechanisms of hardware platforms of various purposes directly or using operating system interfaces, in the creation and application of programs of the corresponding direction and serves to successfully master professional disciplines.

The aim of the discipline is to inculcate in the student a systematic approach to software design of computing platforms of various types as part of electronic devices for solving applied problems.

The subject of study is modern methods of software design of computing platforms as part of electronic devices of various purposes using interfaces of operating systems.

*The study of the discipline "System Programming" contributes to the formation of **professional competences (PC)** in students, necessary for solving practical tasks of professional activity related to the development, improvement and support of intelligent information systems for processing multimedia data:*

PC 03 Ability to develop software systems architectures, modules and components.
PC12 Ability to carry out the system integration process, apply change management standards and procedures to maintain software integrity, overall functionality and reliability.
PC13 Ability to reasonably select and master software development and maintenance tools.

*The study of the discipline "System Programming" contributes to the formation in students of the following **program learning outcomes (PLO)** according to the educational program:*
PLO06 Ability to select and use the appropriate task of software development methodology.
PLO13 To know and apply methods of developing algorithms, designing software and data and knowledge structures.
PLO15 To choose programming languages and development technologies to solve the problems of creating and maintaining software.
PLO17 To be able to apply methods of component software development.
PLO19 To know and be able to apply software verification and validation methods.

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

To successfully master the discipline "System Programming" it is necessary and sufficient to have mathematical, algorithmic and programming training in the amount provided by the curricula of semesters 1-4. This discipline is supported by the disciplines "Fundamentals of Programming", "Algorithms and Data Structures", "Programming", which are previously taught.

Successful mastering of knowledge in this discipline provides support for further study of such disciplines of the educational program of bachelors, such as "Multimedia Systems Software", "Software Security", disciplines of the educational-scientific and educational-professional program of master's training

3. Content of the academic discipline

1. *Topic 1. Subject of the course. Introduction. Basic concepts and definitions.*
2. *Topic 2. Hardware virtualization.*
3. *Topic 3. Interfaces of operating systems.*
4. *Topic 4. I/O management.*
5. *Topic 5. Process management.*
6. *Topic 6. Planner management.*
7. *Topic 7. Signal management.*
8. *Topic 8. Methods of interprocess communication.*
9. *Topic 9. Virtual file system management.*
10. *Topic 10. Memory management.*
11. *Topic 11. System time management.*
12. *Topic 12. Stream management.*
13. *Topic 13. Management of network sockets*

4. Educational materials and resources

Basic literature:

1. Vuntesmeri Y.V. Educational and methodological complex. System programming: https://ee.kpi.ua/~yv/edu/pzks/system/

2. Vuntesmeri Y.V. Digital technologies in microelectronics. Practicum KPI named after Igor Sikorsky, 2017. Access mode: https://campus.kpi.ua/tutor/index.php?mode=mob&show&irid=151302

3. Official Linux documentation. Syscalls(2) section. Access mode: https://man7.org/linux/man-pages/man2/syscalls.2.html

Educational content

5. Methods of mastering the educational component

Lectures on the discipline are conducted using modern multimedia presentation technologies.

№	Type of training session	Description of the training session
1	Lecture 1. Concepts of modern system programming	Hardware platform architecture and controls. Hardware virtualization. Tasks on self-study: item 6
2	Lecture 2. Structure of the operating system	Interfaces of operating systems (API, ABI). Standards (POSIX, SUS). Tasks on self-study: item 6
3	Computer workshop 1	File system management and multiplexed I/O methods. Tasks on self-study: item 6.
4	Lecture 3. I/O management	File abstraction. Block and stream files. Tasks on self-study: item 6.
5	Lecture 4. Process management. Part 1	Hierarchy and imitation of processes. Process life cycle. Process metadata. Tasks on self-study: item 6
6	Computer workshop 2	Process management, separation of processes. Tasks on self-study: item 6
7	Lecture 5. Process management. Part 2	Creating a process. Process initialization. Separation of processes. Tasks on self-study: item 6
8	Lecture 6. Management of the planner	Separation of time. Prioritization. Scheduler policies. Real-time policies. Tasks on self-study: item 6
9	Computer workshop 3	Signals. Allocation of memory between processes. Tasks on self-study: item 6
10	Lecture 7. Signal management	Types of handlers. Signals with load. Signal masking. Critical sections. Tasks on self-study: item 6
11	Lecture 8. Methods of interprocess communication	Named channels. Notification queues. Shared memory. Semaphores. Mutexes. Conditional variables. Tasks on self-study: item 6
12	Computer workshop 4	Network sockets. Elementary mass service server. Tasks on self-study: item 6
13	Lecture 9. Virtual file system management. Part 1	Architecture of data warehouses. Types and history of file systems. Structures of file systems. An inode abstraction. Tasks on self-study: item 6
14	Lecture 10. Virtual file system management. Part 2	Calls for managing file system objects. File system metadata. Out-of-band events. File system notifications. Tasks on self-study: item 6
15	Computer workshop 5. Part 1	Implementation of a 4-level data transmission protocol. Tasks on self-study: item 6

16	*Lecture 10. Memory management. Part 1*	*Segmentation. Swap policy. Heap and stack based memory allocation methods.* *Tasks on self-study: item 6*
17	*Lecture 11. Memory management. Part 2*	*In-memory mapping and anonymous mapping. Alignment problem. Memory limits.* *Tasks on self-study: item 6*
18	*Lecture 12. System time management. Part 1*	*Implementation of a 4-level data transmission protocol.* *Tasks on self-study: item 6*
19	*Lecture 12. System time management. Part 1*	*History of time measurement technology. Time and frequency standards. Types of system clocks. Clock resolution. Methods of determining the system time. Methods of setting the system time. Calendar time and time conversion methods.* *Tasks on self-study: item 6*
20	*Lecture 13. System time management. Part 2*	*Methods of making corrections to system clocks. Synchronization protocols. Methods of putting the process to sleep. Timers. Timer scheduler and timer signals.* *Tasks on self-study: item 6*
21	*Computer workshop 6. Part 1*	*Computer workshop sh. Part 1*
22	*Lecture 14. Flow management. Part 1*	*Dynamics of algorithms on multiprocessor platforms. Threads and coroutines. Polymorphic algorithms. The libpthread library. Creating, waiting, and separating the stream.* *Tasks on self-study: item 6*
23	*Lecture 14. Flow management. Part 2*	*Management of thread groups. Methods of communication and synchronization of flows. Semaphores and mutexes.* *Tasks on self-study: item 6*
24	*Computer workshop 6. Part 2*	*A multi-threaded HTTP server with a thread pool* *Tasks on self-study: item 6.*
25	*Lecture 15. Management of network sockets.*	*Types of sockets. Address families. Listeners and transmitters. Local sockets. Calls to work with sockets.* *Tasks on self-study: item 6*
26	*Lecture 16.*	*Principles of building mass service servers.* *Tasks on self-study: item 6.*
Modular control work		

6. Student's self-training

The independent work of students consists of the performance of individual tasks on the topics assigned to laboratory work and homework, as well as the processing of theoretical material based on the provided lecture material and recommended literature, including the topics assigned to independent study (according to Table 1) .

A student should spend a number of hours on independent work that is commensurate with the number of hours he spent in classroom classes.

Table 1. Questions submitted for independent study

№	*The name of the topic submitted for independent processing*

1	***I/O management***. *Types of input-output multiplexing.*
2	***Process management.*** *Executable header structures for different ABIs.*
3	***Management process.*** *Executable header structures for different Abis.*
4	***Signal management.*** *Group operations on signals.*
5	***Virtual file system management.*** *Methods of traversing file lists in directories.*
6	***Memory management.*** *Stack boundary adjustment methods.*
7	***System time management.*** *IEEE1588 synchronization protocol.*
8	***Flow management.*** *OpenMP specification.*
9	***Management of network sockets.*** *Principles of using Unix domain socket.*

Policy and control

7. Policy of academic educational component

The student must study the discipline during the sixth semester, adhering to the calendar plan for completing laboratory tasks, studying the topics of the lecture material, and performing calculation and graphic work. The student must complete all tasks independently and on time.

The laboratory assignment is considered completed if the student defended it with the teacher (showed ability to work, compliance with the individual assignment, answered all questions) and submitted a report on the performance of this assignment. Completion of a task with a delay of more than 1 week is considered untimely. Penalty points are provided for late submission of laboratory work. Such restrictions encourage the student to organize the systematic performance of tasks and prevent a significant accumulation of failed assignments at the end of the semester.

Student evaluation is carried out according to the rating assessment of the level of training of students in the discipline. The current state of student success is displayed in the "Electronic Campus" system, which students have access to.

8. Types of control and rating system for evaluating learning outcomes (RSO)

The current monitoring of learning results involves the students' performance of laboratory work, writing modular test, homework test, surveys at lectures.

Evaluation criteria for laboratory work include the quality of program development, error-free execution, quality of protection of the developed program and preparation of a report on the work performed (see Table 2). Weighted points for each task of laboratory work (10 points in total) are determined according to the criteria in the table. 2.

Table 2. Weighted points and criteria for evaluating laboratory work

		Efficiency, compliance task	Quality of software implementation	Interface quality	protection	report
1	Lab №1	2	2	2	2	2
2	Lab №2	2	2	2	2	2

3	Lab №3	2	2	2	2	2
4	Lab №4	2	2	2	2	2
5	Lab №5	2	2	2	2	2
6	Lab №6	2	2	2	2	2
Together for laboratory work			10*6 =60 points			

The homework test (HT) is conducted in the form of an independent written work, the questions of which can be both theoretical and practical.

HT provides for the performance of individual tasks according to the established options. The purpose of HT is to deepen and consolidate knowledge on certain topics of the discipline. HT consists of two tasks. The weighted score of the HT task is 5 points.

Evaluation criteria:

5 points – the answer/solution is correct, complete;

4 points – the answer/decision is correct, but incomplete (insufficiently substantiated);

3 points – the answer/decision is correct, but not substantiated, has minor inaccuracies;

2 points – the answer/solution is incomplete or has significant inaccuracies;

1 point – the answer\decision is incorrect, but the course of the decision is correct;

0 points - no answer\solution or completely incorrect.

Calendar control is carried out twice a semester as a monitoring of the current state of meeting the requirements of the work program.

At the first certification, the student receives "credited" if his current rating is at least 50% of the maximum number of points that the student can receive before the first certification.

At the second certification, the student receives "passed" if his current rating is at least 50% of the maximum number of points that the student can receive before the second certification.

Modular test (MT). MT is held twice a semester and covers the topics of lecture classes from 1 to 6 and from 7 to 13, respectively. MT is conducted in the form of a test on the platform https://ee.kpi.ua/~yv/app/

The MT weighted score is 15 points.

Semester control of study results is carried out in the form of credit.

The rating scale for the discipline is determined by:

R = RS = 60 points + 10 points + 30 points

A necessary condition for admission to credit is the completion and defense of all laboratory work (the sum of points is not less than 30 points), and the semester rating (rC) is not less than 60% of RS, i.e. not less than 60 points. Otherwise, the student must do additional work and improve his rating. With a semester rating (rC) of not less than 60% of RS, i.e. not less than 60 points, the student receives credit "automatically" according to the table:

Scores	Rating
100-95	Excellent
94-85	Very Good
84-75	Good
74-65	Satisfactory
64-60	Fair
Less than 60	Unsatisfactory
Admission conditions not met	Not allowed

A student can try to improve his grade by writing a credit test, while his semester rating is canceled (with the exception of points for laboratory work), after which points are awarded based on the results of the credit test ("hard" grading system).

Working program of the academic discipline (syllabus):

Compiled *by Ph.D., Assoc. Vuntesmeri Yu.V.*

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)*