Syllabus

Course Program

Computer architecture and operating systems

Specialty
121 – Software Engineering

Institute
Institute of Computer Science and Information Technology

Educational program
Software Engineering

Department
Software Engineering and Management Intelligent Technologies (321)

Level of education
Bachelor's level

Course type
Special (professional), Mandatory

Semester
2

Language of instruction
English, Ukrainian

Lecturers and course developers

Pavlo Smolin

pavel.smolin@khpi.edu.ua
Senior Lecturer of Department of Software Engineering Management Intelligent technologies, National Technical University "Kharkiv Polytechnic Institute

Google Scholar: https://scholar.google.com/citations?user=zCHB-xoAAAAJ&hl%20google%20scholar
ORCID: https://orcid.org/0000-0002-1290-9698
Scopus: https://scholar.google.com/citations?user=zCHB-xoAAAAJ&hl%20google%20scholar
More about the lecturer on the department's website

General information

Summary

The basic principles of functioning of the hardware part of modern computers are given. The functions of the operating system to manage application programs and manage computer resources, such as RAM and input-output devices, are considered.

Course objectives and goals

The purpose of the discipline is to teach students the concepts of building modern operating systems, features of process scheduling in multitasking operating systems, interprocess communication, methods of memory allocation, as well as acquiring skills in the practical application of system calls in software applications.

Format of classes

Lectures, laboratory classes, consultations, self-study. Final control in the form of a test.

Competencies

К01. Ability to think abstractly, analyze and synthesize.
К05. Ability to learn and master modern knowledge.
К06. Ability to search, process and analyze information from various sources.
К15. Ability to develop architectures, modules and components of software systems.
К20. Ability to apply fundamental and interdisciplinary knowledge to successfully solve software engineering problems.
К26. Ability to think algorithmically and logically.

Learning outcomes

PLO01. Analyze, purposefully search and select information and reference resources and knowledge necessary for solving professional problems, taking into account modern achievements of science and technology.
PLO07. To know and apply in practice the fundamental concepts, paradigms and basic principles of functioning of language, tools and computing tools of software engineering.

Student workload

The total volume of the course is 120 hours (4 ECTS credits): lectures - 28 hours, laboratory classes - 28 hours, self-study - 64 hours.

Course prerequisites

Fundamentals of programming.
Fundamentals of software engineering

Features of the course, teaching and learning methods, and technologies

Lectures, laboratory classes, self-study, consultations. Final control in the form of an test.

Program of the course

Topics of the lectures

Topic 1. Fundamentals of computer system architecture
Central processing unit. Execution of commands. Design principles of modern computer systems. Parallelism at the level of commands and at the level of processors. RAM. Memory address. Cache memory. External memory. Magnetic disks. Solid state drives. Input-output devices. System bus.
Topic 2. Basic concepts of operating systems
Concept of operating system. The operating system as an extended machine and as a resource allocator. Functional components of operating systems. Basic concepts of operating systems. The kernel of the system. Kernel mode and user mode. System software. Architectures of operating systems. Interaction of the operating system and hardware. Interaction of the operating system and executable software code.
Topic 3. Management of processes and threads
Basic concepts of processes and threads. Multithreading. States of processes and threads. Control blocks of processes and treads. Context switching and interrupt handling. Creation and termination of processes and threads. Synchronous and asynchronous execution of processes. Process management in Linux. Thread management in Linux. Process management in Windows. Thread management in Windows
Topic 4. Processes and threads scheduling
General principles of scheduling. Types of scheduling. Preemptive and non- preemptive multitasking. Scheduling algorithms. Scheduling in Linux. Scheduling in Windows.
Topic 5. Communication of threads
Race condition. Critical sections and locks. Semaphores. Mutexes. Conditional variables. Block read - write. Synchronization according to the barrier principle. Interaction of threads in Linux. Interaction of streams in Windows.
Topic 6. Interprocess communication
Types of interprocess communication. Shared memory methods. Message passing method. Technology of memory mapped. Channels. Pipes. Sockets.

Topic 7. RAM management
Virtual memory technology. Memory segmentation. Page organization of memory. Linux Main Memory Management. Windows Main Memory Management. Communication with the disk during memory management. Dynamic memory allocation.
Topic 8. File systems
File and file system. Organization of information in the file system. Physical organization of the file system. POSIX file operations.
Topic 9. I/O device management
Methods of performing input - output operations. The input-output subsystem of the operating system kernel. Input - output in user mode.

Topics of the workshops

Practical classes within the discipline are not provided

Topics of the laboratory classes

Topic 1 Basic information about working with the Linux OS. Working with files in UNIX/Linux
Topic 2. Creation and compilation of programs in the Linux OS
Topic 3. Transferring command line parameters to the program. Work with processes. Obtaining system information
Topic 4. Background processes and signals
Topic 5. Application of threads.
Topic 6. Linux API - introduction to interprocess communication. Channels.
Topic 7. Linux Api - introduction to interprocess communication. Sockets..

Self-study

The curriculum provides for the implementation of a calculation task. At the beginning of the semester, students choose the topics of the assignment and agree on them with the teacher. The calculation task is completed during the semester and is defended during the credit week.
Students are recommended additional materials (videos, articles) for independent study

Course materials and recommended reading

Key literature

1. Jim Ledin. Modern Computer Architecture and Organization: Learn x86, ARM, and RISC-V architectures and the design of smartphones, PCs, and cloud servers, 2nd Edition, Packt Publishing Ltd., 2022.
2. Andrew A. Chien, Computer Architecture for Scientists: Principles and Performance, New Edition, Cambridge University Press, 2022.
3. Tanenbaum, E., Bos, H., Modern operating systems. Firth ed. Pearson Prentice-Hall.,2022.
4. William Stallings.(2017). Operating Systems: Internals and Design Principles 9th Edition. Pearson2017.
5. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Silberschatz's Operating System Concepts, John Wiley & Sons Inc,, 2018.

Additional literature

1. Avi Silberschatz, Peter Baer Galvin, Greg Gagne, Operating System Concepts. John Wiley & Sons, Inc., 2018.
2. Greg Tomsho, Guide to Operating Systems (MindTap Course List), Cengage Learning, 2020.
3. Richard Blum, Christine Bresnahan, Linux Command Line and Shell Scripting Bible, 4th Edition., Wiley, 2021.
4. Jack-Benny Persson, Linux System Programming Techniques: Become a proficient Linux system programmer using expert recipes and techniques, Packt Publishing, 2021.
5. Will Fuqua, Windows Terminal Tips, Tricks, and Productiv ity Hacks: Optimize your command-line usage and development processes with pro-level techniques, Packt Publishing, 2021.

Assessment and grading

Criteria for assessment of student performance, and the final score structure

100% of the final grade consists of the results of the assessment in the form of an test (30%) and current assessment (70%):
- 7 laboratory works (4% each);
- 2 tests (6% each);
- calculation task (30%).

Grading scale

Total points	National	ECTS
90–100	Excellent	A
82–89	Good	B
75–81	Good	C
64–74	Satisfactory	D
60–63	Satisfactory	E
35–59	Unsatisfactory (requires additional learning)	FX
1–34	Unsatisfactory (requires repetition of the course)	F

Norms of academic integrity and course policy

The student must adhere to the Code of Ethics of Academic Relations and Integrity of NTU "KhPI": to demonstrate discipline, good manners, kindness, honesty, and responsibility. Conflict situations should be openly discussed in academic groups with a lecturer, and if it is impossible to resolve the conflict, they should be brought to the attention of the Institute's management.
Regulatory and legal documents related to the implementation of the principles of academic integrity at NTU "KhPI" are available on the website: http://blogs.kpi.kharkov.ua/v2/nv/akademichna-dobrochesnist/

Approval

Approved by	08.06.2023	Head of the department Ihor HAMAIUN
	08.06.2023	Guarantor of the educational program Uliya LITVINOVA