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

Емблема кафедри (за наявності)

Department of Computing Technics

# System programming in the Unix environment

## 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*<br>*Practice: senior lecturer, Andrey Simonenko, comsys.spz@gmail.com* |
| **Access to the course** | *https://drive.google.com/drive/folders/16sTKRto-CYGMicfvVLhjKBBl3kt5tTwZ* |

## Outline of the Course

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

***What will be studied.*** *System programming in the Unix environment, that is, the development of system programs for Unix-like operating systems at the level of using system calls to interact with the kernel will be studied. Thorough information on the POSIX functions (and sufficient information on the implementation of the corresponding system calls) used in the development of system programs will be provided. The discipline is not focused on system programming in any specific implementation of a Unix-like operating system, portable system programming will be studied. The discipline consists of the following topics: program execution environment, process management, working with files, working with signals, working with pipes, advanced input/output, process memory management, pseudo-terminal programming and others.*

***Why it is interesting/necessary to study.*** *It is advisable to study this discipline for those who will develop system programs for Unix-like operating systems. The tasks are programmed in C, or C++, or Rust, but the acquired knowledge will be useful in solving some system tasks for Unix-like operating systems in other programming languages.*

***What you can learn (learning results).*** *Develop system programs for Unix-like operating systems in the C, or C++, or Rust programming language that control processes, work with files, work with signals, use advanced input/output, work with pipes, work with pseudo-terminals.*

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

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

*Ability to program in C, or C++, or Rust and ability to work in a Unix-like operating system at the user level. Basic knowledge of programming, data structures and algorithms.*

## 3. Content of the course

*Topic 1. Program execution environment*
*Topic 2. File system tree*
*Topic 3. File descriptor*
*Topic 4. Users and credentials*
*Topic 5. Signals*
*Topic 6. Process control*
*Topic 7. Pipe and FIFO*

## 4. Coursebooks and teaching resources


1. *The Open Group. Single UNIX specification, version 4 - IEEE and The Open Group, 2018.*
2. *W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment, 3rd edition. - Addison-Wesley Professional, 2013 - 1032 p.*
3. *Michael Kerrisk. The Linux Programming Interface: A Linux and UNIX System Programming Handbook, 1st edition. - No Starch Press, 2010. - 1552 p.*


**Educational content**

## 5. Methodology

| *Parts, topics* | *Total, h* | *Lectures, h* | *Laboratory works, h* | *Self-study, h* |
|---|---|---|---|---|
| *Topic 1. Program execution environment* | *17* | *6* | *4* | *7* |
| *Topic 2. File system tree* | *12* | *5* | | *7* |
| *Topic 3. File descriptor* | *17* | *5* | *4* | *8* |
| *Topic 4. Users and credentials* | *12* | *5* | | *7* |
| *Topic 5. Signals* | *14* | *5* | | *9* |
| *Topic 6. Process control* | *18* | *5* | *5* | *8* |
| *Topic 7. Pipe and FIFO* | *18* | *5* | *5* | *8* |
| *Test* | *3* | | | |
| *Total* | *108* | *36* | *18* | *54* |

*Laboratory works:*

*1. Setting up the development environment (4 h)*
*2. File system tree (4 h)*
*3. Command interpreter (5 h)*
*4. Distributed shared memory (5 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 thress 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)