National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Department of Computer Engineering

Network 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	*121 Software Engineering* *123 Computer engineering*
Education Program	*Computer Systems Software Engineering* *Computer Systems and Networks*
Type of Course	*Selective*
Mode of Studies	*Full-time*
Year of studies, semester	*4 year (8 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	

Outline of the Course

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

What will be studied. *Network programming in the Unix environment, that is, the development of system and server programs that work with the network 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 and server applications that work with the network will be provided. The discipline is not focused on network programming in any specific implementation of a Unix-like operating system, portable network programming will be studied. The discipline consists of the following topics: elementary sockets, names and addresses, advanced sockets, advanced network input/output, implementation of client-server programs in different ways.*

Why it is interesting/necessary to study. *It is advisable to study this discipline for those who will develop system and server programs that work with the network 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 network programming tasks for Unix-like operating systems in other programming languages.*

What you can learn (learning results). *Develop system and server programs that work with the network for Unix-like operating systems in the programming language C, or C++, or Rust, create client-server programs in various ways.*

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

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, Computer networks.

3. Content of the course

Topic 1. Program execution environment
Topic 2. Names and addresses
Topic 3. Elementary sockets
Topic 4. TCP sockets
Topic 5. UDP sockets
Topic 6. Input/output multiplexing
Topic 7. SCTP sockets
Topic 8. Unix domain sockets

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. *W. Richard Stevens, Bill Fenner, Andrew M. Rudoff. Unix Network Programming, Volume 1, 3rd edition. - Addison-Wesley Professional, 2003. - 1152 p.*
4. *Michael Kerrisk. The Linux Programming Interface: A Linux and UNIX System Programming Handbook, 1st edition. - No Starch Press, 2010. - 1552 p.*
5. *Kevin R. Fall, W. Richard Stevens. TCP/IP Illustrated: The Protocols, Volume 1, 2nd edition. - Addison-Wesley Professional, 2011. - 1056 p.*

Educational content

5. Methodology

Parts, topics	*Total, h*	*Lectures, h*	*Laboratory works, h*	*Self-study, h*
Topic 1. Program execution environment	*10*	*5*		*5*
Topic 2. Names and addresses	*15*	*5*	*3*	*7*
Topic 3. Elementary sockets	*12*	*5*		*7*
Topic 4. TCP sockets	*17*	*5*	*5*	*7*
Topic 5. UDP sockets	*16*	*4*	*5*	*7*
Topic 6. Input/output multiplexing	*11*	*4*		*7*
Topic 7. SCTP sockets	*16*	*4*	*5*	*7*
Topic 8. Unix domain sockets	*11*	*4*		*7*
Test	*3*			
Total	*108*	*36*	*18*	*54*

Laboratory works (tasks in the first book from the coursebooks):

1. Addressing in IP networks (3 h, pages 285-193)
2. Working with TCP sockets (5 h, pages 294-298)
3. Client-server (5 h, pages 304-309)
4. Routing in IP networks (5 h, pages 299-304)

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)