POZNAN UNIVERSITY OF TECHNOLOGY

EUROPEAN CREDIT TRANSFER AND ACCUMULATION SYSTEM (ECTS)

pl. M. Skłodowskiej-Curie 5, 60-965 Poznań

COURSE DESCRIPTION CARD - SYLLABUS

Course name

Network Programming

Course

Field of study	Year/Semester
Computing	1/1
Area of study (specialization)	Profile of study
Distributed Systems	general academic
Level of study	Course offered in
Second-cycle studies	polish
Form of study	Requirements
full-time	compulsory

Number of hours

Lecture	Laboratory classes	Other (e.g. online)
30	30	
Tutorials	Projects/seminars	

Number of credit points

4

Lecturers

Responsible for the course/lecturer:

dr inż. Michał Kalewski

email: Michal.Kalewski@cs.put.poznan.pl

tel: 61 6652370

Faculty of Computing and Telecommunications
ul. Piotrowo 3, 60-965 Poznań

Responsible for the course/lecturer:

dr inż. Michał Sajkowski

email: Michal.Sajkowski@cs.put.poznan.pl

tel. 61 6653062

Faculty of Computing and Telecommunications
ul. Piotrowo 3, 60-965 Poznań

Prerequisites

A student starting this course should have basic knowledge of operating systems, concurrent programming and computer networks. He should also have the ability to obtain information from the indicated sources; should be able to use analytical and experimental methods to formulate and solve engineering tasks and simple research problems; should be able to integrate knowledge from various areas of computer science and apply a systemic approach, also taking into account non-technical aspects. The student should show such features as: honesty, responsibility, perseverance, cognitive curiosity, creativity, personal culture, respect for other people.

Course objective

1. Theoretical and practical aspects of network programming using modern computer networks

2. Technical solutions currently used in the implementation of network applications: server architectures, network clients and software-defined networking

3. Solving common problems encountered by a network application developer

4. Configuring protocols and system tools for designing and implementation of network software

Course-related learning outcomes

Knowledge

1. has a structured and theoretically founded general knowledge related to key issues in the field of network programming

2. has advanced detailed knowledge regarding selected computer networks and network programming issues

3. has knowledge about development trends and the most important cutting edge achievements in computer science and other selected and related scientific disciplines

4. has advanced and detailed knowledge of the processes occurring in the life cycle of network servers and clients

5. knows advanced methods, techniques and tools used to solve complex engineering tasks and conduct research in a selected area of network programming

Skills

1. is able to obtain information from literature, databases and other sources (both in Polish and English), integrate them, interpret and critically evaluate them, draw conclusions and formulate and fully justify opinions

2. can use analytical, simulation and experimental methods to formulate and solve engineering problems and simple research problems

3. can - when formulating and solving engineering tasks - integrate knowledge from different areas of computer science (and if necessary also knowledge from other scientific disciplines) and apply a systemic approach, also taking into account non-technical aspects

4. is able to assess the suitability and the possibility of using new achievements (methods and tools) and new IT products

5. can carry out a critical analysis of existing technical solutions and propose their improvements (streamlines)

6. is able to interact in a team, taking various roles in it

7. can determine the directions of further learning and implement the process of self-education, including other people

Social competences

1. understands that in the field of IT the knowledge and skills quickly become obsolete

2. understands the importance of using the latest knowledge in the field of computer science in solving research and practical problems

3. understands the importance of popularization activities concerning the latest achievements in the field of computer science

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Summative assessment:

a) the knowledge acquired in the course of the lecture is verified by a written test. The test consists of 5 open questions. The final grade is determined using the following scale: (90%, 100%] -> 5.0, (80%, 90%] -> 4.5, (70%, 80%] -> 4.0, (60%, 70%] -> 3.5, (50%, 60%] -> 3.0, (0%, 50%] -> 2.0.

b) verification of assumed learning objectives related to laboratories is based on:

- assessment of skills related to the implementation of laboratory exercises and project progress,

- continuous assessment during each class - rewarding the increase in the ability to use the learned skills and methods

Getting extra points for activity during classes, especially for:

- proposing to discuss additional aspects of the issue,

- effectiveness of applying the acquired knowledge while solving a given problem,

- ability to work within a team that practically performs a specific task in a laboratory,

- comments related to the improvement of teaching materials.

Programme content

The lecture program covers the following topics:

- Introduction: a reminder of the material on the layered model of computer networks and basic datagram (UDP) and stream (TCP) IPv4 network sockets

- Network socket buffers and methods and tools for encapsulating application data in stream transmissions

- System options and functions for network sockets and domain name system support

- Programmatic access to data link layer frames (libpcap and libnet libraries) and network packages in the operating system kernel (libnetfilter libraries)

- Raw network sockets for IPv4 and IPv6 protocols and SOCK_PACKET and PF_PACKET sockets

- Stream Control Transmission Protocol (SCTP) network sockets and multi-stream communication

- Datagram and stream IPv6 network sockets; interoperability of processes using IPv4 and IPv6 protocols; constructions of dual network servers

- Mechanisms and architectures of concurrent network servers: non-blocking network functions; input-output multiplexing (select, poll, epoll, and kqueue system functions); network functions triggered by system signals; concurrency with the use of child processes and threads; child process and thread pools; SO_REUSEPORT mechanism

- Methods and mechanisms of disconnection detection in stream internet communication

- Implementation of internet broadcast and multicast communication using network sockets

- Programming interfaces to manipulate routing and ARP tables and key management network sockets (PF_KEY type)

- Network sockets interface in Windows (winsock), Windows Phone, Android and iOS operating systems; using network sockets to implement communication in wireless mobile networks

- Network communication support mechanisms in applications with a graphical user interface

- Internet communication implementations using application layer protocols (libcurl library)

- Introduction to SDN (Software-Defined Networking), the OpenFlow protocol and the Mininet emulator; use of the OpenFlow protocol to implement network device controllers in SDN networks

.The laboratory program covers the following topics:

- Using libpcap, libnet and libnetfilter libraries to implement programs for intercepting data link layer frames and network packets in the operating system kernel

- Implementation of simple routing tools on IPv4 and IPv6 networks using raw network sockets (ICMPv4 and ICMPv6)

- Implementation of client and server applications using SCTP protocol network sockets and multi-stream connections and performance tests of such connections

- Implementation of client and server applications using IPv6 datagram and stream network sockets; dual server applications IPv4 / IPv6

- Implementation of selected architectures of concurrent network servers

- Implementation of broadcast and multicast communication in internet networks

- Implementations of programs to manipulate routing and ARP tables and ARP

- Implementation of network client applications for selected operating systems of mobile devices, support for the graphical user interfaces and communication in wireless mobile networks

- Using libcurl to implement internet communication using application layer protocols

- Support for the Mininet emulator and implementations of network device controllers in SDN networks using the OpenFlow protocol

Teaching methods

Lecture: multimedia presentation, illustrated with examples given on the board.

Laboratories: multimedia presentation, illustrated with examples given on the blackboard, and carrying out the tasks given by the teacher - practical exercises.

Bibliography

Basic

1. UNIX - programowanie usług sieciowych. 1, API: gniazda i XTI, Stevens W. R., Wydawnictwa Naukowo-Techniczne, 2002

2. Sieci komputerowe TCP/IP. 3, Programowanie w trybie klient-serwer, wersja BSD, Comer D., Stevens D., Wydawnictwa Naukowo-Techniczne, 1997

3. Computer Networks, A. S. Tanenbaum, Pearson, 2014

Additional

1. Data and Computer Communications. Networking and Internetworking, Hura G. S., Singhal M., CRC Press LLC, 2001

2. Client/Server Survival Guide, Harkey D., Wiley, 1999

3. Wireless Communications and Networks, Stallings W., Pearson, Prentice Hall, 2002

4. Algorithms and Protocols for Wireless, Mobile Ad Hoc Networks, Boukerche A., Wiley-IEEE Press, 2008

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,0
Classes requiring direct contact with the teacher	60	2,0
Student's own work (literature studies, preparation for laboratory classes/tutorials, preparation for tests/exam, project preparation) [1]	40	2,0

[1] delete or add other activities as appropriate