AGH University of Science and Technology

Code:	UBPJO-234	Module name:	Programming in UNIX systems

Academic year:	2017/2018	Semester:	Spring, Fall	ECTS credits:	5

Programme:	Physics and Applied Computer Science

Course homepage:		Lecture language:	English

Responsible teacher:	dr hab. inż. Bołd Tomasz (tomasz.bold@fis.agh.edu.pl)
Academic teachers:	dr hab. inż. Bołd Tomasz (tomasz.bold@fis.agh.edu.pl)

Module summary

The course covers the Unix (Linux, Mac OS) system programming.

Description of learning outcomes for module

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Method of learning outcomes verification (form of completion)
Social competence		
M_K001	The student is able to explain the technologies used in the project and defend the choices he/she made.	Presentation
M_K002	The student will be able to participate constructively in the discussions about work presented by others.	Participation in a discussion
Skills		
M_U002	The student is accustomed to the UNIX programming environment and knows how to use various libraries in a more advanced project.	Execution of a project
Knowledge		
M_W002	The student knows UNIX process management and intercommunication mechanisms. He/she understands advantages of various available mechanisms.	Execution of a project

FLO matrix in relation to forms of classes

MLO code	Student after module completion has the knowledge/ knows how to/is able to	Form of classes										
		Lectures	Auditorium classes	Laboratory classes	Project classes	Conversation seminar	Seminar classes	Practical classes	Fieldwork classes	Workshops	Others	E-learning
Social competence												
M_K001	The student is able to explain the technologies used in the project and defend the choices he/she made.	+	-	-	-	-	-	-	-	-	-	-
M_K002	The student will be able to participate constructively in the discussions about work presented by others.	-	-	-	-	-	-	-	-	-	-	-
Skills												
M_U002	The student is accustomed to the UNIX programming environment and knows how to use various libraries in a more advanced project.	-	-	-	+	-	-	-	-	-	-	-
Knowledge												
M_W002	The student knows UNIX process management and intercommunication mechanisms. He/she understands advantages of various available mechanisms.	+	-	-	+	-	-	-	-	-	-	-

Module content

Lectures

Introduction
volution of Unix systems. System architecture. Standards evolution. POSIX. System V standard. Elementary concepts.
Components of UNIX system administration. UNIX program compilation and debugging.
Utilities: Timing. Logging. Date and time.

IO
File I/O. File abstraction in UNIX systems. Device files. File system interactions.
File access permissions.

Processes 1
Process vs program. Initialisation and termination. Process environment.

Processes 2
Controlling processes: signals, initiating new processes, daemons. A stack of processes.

Threads

POSIX threads. Controlling threads lifetime. Reentrancy. Thread data. Synchronisation primitives.

IPC 1

Overview of available methods. Signals.

IPC 2

Named and unnamed stream pipes. Messages. Asynchronous I/O.

IPC 3

Shared memory. Semaphores.

Basic networking

Network based process communication. Addressing. Sockets. UDP vs TCP. SSL/TSL

Modern IPC libraries

Examples of few libraries based on the elementary mechanisms.

Project presentations

Remaining time will be devoted to student projects presentations. Every student will be asked to present the two programs. The presentation has to cover the description of the problem to solve, the solution (presentation of relevant code snippets) and discussion of difficulties encountered.

Project classes

Simple unix program

In the first, simple, unix program elementary aspects will be covered. Typically process management and simple IO + some algorithmic part.

Advanced unix program

The level of complexity of the second program will be adjusted individually for each student (more advanced problems will be rewarded with the higher final grade). Subjects will cover the second part of the lectures, namely the IPC & threads.

Method of calculating the final grade

The final grade will be based on three notes: for the first, program, the second program and the quality presentation (each judged in 0-10 points scale). They will be weighted 30/50/20 correspondingly for the final grade.

Prerequisites and additional requirements

Good knowledge of C programming language and decent Unix user skills.

Recommended literature and teaching resources

Advanced UNIX programming and UNIX Network programming by Richard Stevens.

Scientific publications of module course instructors related to the topic of the module

Additional scientific publications not specified

Additional information

Unix (Linux, Mac OS) system-level programming is used typically behind the scenes of the high-level libraries. In typical user code, the later tend to be used nowadays. However, understanding the underlying mechanisms allows to better comprehend the high-level designs as well as gives an insight into the legacy systems. Last but not least some of the Unix system mechanism provide much more performant solutions than those offered by high-level libraries.

Student workload (ECTS credits balance)

Student activity form	Student workload
Participation in lectures	30 h
Completion of a project	60 h
Preparation for classes	60 h
Summary student workload	150 h
Module ECTS credits	5 ECTS