UNIVERSITY of NICOSIA

Course Syllabus

Course Code	Course Title	ECTS Credits
COMP-413	Systems Programming	6
Prerequisites	**Department**	**Semester**
COMP-212, COMP-354	Computer Science	Spring
Type of Course	**Field**	**Language of Instruction**
Elective	Computer Science	English
Level of Course	**Lecturer(s)**	**Year of Study**
1st Cycle	Dr Harald Gjermundrød	3rd or 4th
Mode of Delivery	**Work Placement**	**Corequisites**
Face-to-face	N/A	None

Course Objectives:

The main objectives of the course are to:
- introduce the Unix operating system and the environment and tools it provides for developing system programs
- introduce the tools, language and OS libraries needed to develop command line applications in an effective manner
- explain in detail structure and organization of the filesystem and develop applications that are using buffered and unbuffered I/O
- explain the concept of processes and develop applications that dynamically create processes and synchronize their interaction
- introduce signals and signal handlers as well as develop applications which are using signals
- make students aware of the concept of multi-threaded applications and various synchronization mechanisms
- expose the students to scripting and how it can be used as simple system programs.

Learning Outcomes:

After completion of the course students are expected to be able to:
1. use the Unix environment in order to develop applications on this platform
2. carry out simple system administrator tasks according to a security policy, such as creating accounts, and setting permissions

3. demonstrate the capability to use a range of software tools in support of the development of an application
4. defend the need for APIs and abstraction of services and summarize a typical set of system commands provided by an operating system
5. identify the difference between system calls and library calls and the trade-offs of executing in user mode/space compared to kernel mode/space
6. critically assess when to use buffered I/O and when to use unbuffered I/O
7. develop multi-process and multi-threaded applications
8. summarize the difference between creating multi-process applications and multi-threaded applications
9. tell what signals are and what they can be used for
10. explain why one or more language constructs or system calls may lead to security problems such as overflows
11. demonstrate the mechanisms for implementing scripts and the role of scripts on system implementation and integration.

Course Content:

1. Introduction to the Unix operating system: users, processes, filesystems, basic system commands, editors, compilers, build systems, versioning control systems, and Integrated Development Environments (IDE)
2. Develop command line applications: make utility, command line arguments, using man pages, include OS libraries and language libraries, use system calls and library functions, vulnerabilities or misuse of system calls that may lead to run-time problems
3. Files and I/O: structure and organization of directories and files, operations on files and directories, buffered and unbuffered I/O
4. Process: creation, termination, synchronization, process hierarchies, usage of shared memory, and inter-process communication
5. Signals: role of signals, registering signal handlers, and handling of signals
6. Threads: creation, termination, synchronization, mutual exclusion, semaphores, and condition variables
7. Scripts: creating and executing scripts with parameter passing.

Learning Activities and Teaching Methods:

Lectures, Practical Exercises, and Assignments

Assessment Methods:

Final Exam, Midterm Exam, Assignments, and Quizzes

Required Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Advanced Programming in the UNIX(R) Environment, 3rd Edition	R. Stevens, S. Rago	Addison Wesley	2013	978-0321637734

Recommended Textbooks / Readings:

Title	Author(s)	Publisher	Year	ISBN
Unix in a Nutshell	A. Robbins	O'Reily	2005	0-596-10029-9
How Linux Works: What Every Superuser Should Know, 2nd Ed.	B. Ward	No Starch Press	2014	978-1593275679