Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

8.4.30 Systems Software

Module Author

Paul Kelly

Module Description

This module introduces the students' to the various components of the UNIX/Linux operating systems from a system programmer's perspective including both the shell and programming interfaces.

In this course, students will learn to develop software using the C programming language in the UNIX programming environment. Topics covered will include the user/kernel interface, fundamental concepts of UNIX, advanced I/O, filesystems, device drivers and processes. Fundamental concepts of software development and maintenance on UNIX systems will also be covered.

Students are expected to have a good working knowledge of the C programming language and to be able to competently use a UNIX system with a command-line shell interface.

Module Aims

The aims of this module are to provide the student with skills in the advanced concepts, structures, mechanisms and techniques of UNIX systems programming.

Learning Outcomes

On completion of this module, the successful learner will be able to:

1. describe and employ the fundamental concepts, structures, mechanisms of systems libraries and calls of UNIX-based systems programming,
2. use the UNIX tools in developing software in C, including gcc, gdb, ddd, gprof, cvs, make
3. use signals at the command level and as part of a program
4. write software using inter-process communication (IPC) and appropriate system calls
5. program terminal I/O and relevant system calls
6. write concurrent programs using processes and threads

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

Learning and Teaching Methods

Lectures with demonstrations

Laboratory practicals based on lectures

Extensive use of a Virtual Learning Environment (VLE)

Module content

Software development tools in UNIX/Linux

Advanced scripting techniques

File Systems: File and directory structures, Permissions, Sequential and random file access, Accessing directories, I/O redirections

Processes: Process model, Process environment, Process creation and termination, Process control, Process times

Race conditions and deadlocks

Daemons: Characteristics, Coding, Error logging, Client-server model

Design and implementation of a UNIX-oriented Shell

Signals: concepts, Catching and handling signals, Signal system calls

Interprocess communication: Process synchronisation and communication concepts

Pipes: Programming Concepts, Limitations, Named pipes (FIFOs), Semaphores, Shared memory

Socket programming: APIs and their implementation.

Terminal I/O: Getting and setting terminal attributes, Canonical and non-canonical modes, Nonblocking I/O, Pseudo terminals

Advanced I/O: Record locking, Streams, I/O Multiplexing, Asynchronous I/O, Memory Mapped I/O

POSIX Threads: Concepts, Thread environment, Thread invocation and synchronisation

Module Assessment

Written Examination - 70%

Programming assignments, written tests, laboratory assessments, on-line tests - 30%

Essential Reading

K. Haviland, D. Gray, B. Salama (1999), Unix System Programming, Addison-Wesley.

Pre-Requisite Modules code(s)	Co-Requisite Modules code(s)	ECTS Credits	Module Code	**Systems Software**
		5	CMPU4051	

Supplemental Reading

R. Stevens, S.A. Rago (2005), Advanced Programming in the UNIX Environment, Second Edition, Addison-Wesley.Maurice

Bach (1986), The Design of the UNIX operating System, Prentice Hall.

S. Sarwar, R. Koretsky, S.Sarwar (2002), Linux: The Textbook , Addison-Wesley.

K. Robbins, S. Robbins (2003), UNIX Systems Programming: Concurrency, Communication, and Threads, Prentice Hall.

Web references, journals and other

The Open Group Base Specifications Issue 6 IEEE Std 1003.1-2001: http://www.opengroup.org/onlinepubs/009695399/mindex.html

GNU Online Documentation available online at: http://www.gnu.org/manual/manual.html

Further current web links as listed on VLE