Systems Software

Week 1: Introduction to the Module

Notes By: Jonathan McCarthy

Introduction

- **About Jonathan:**
- Jonathan McCarthy
- Lecturer – School of Computing
- **Contact Details:**
- If you have any questions or queries regarding the Software Systems module please send me a mail: **jonathan.mccarthy@dit.ie**

Module Timetable

Course schedule

- 2 hours lectures
  - Friday 9am to 10am - KA3-022
  - Friday 11am to 12pm – KA3-011
- 2 hours labs
  - Friday 1pm to 3pm - Aungier Street, 1-005
  - Friday 1pm to 3pm - Aungier Street, 1-006
  - Friday 1pm to 3pm – Kevin Street, KA3-005

Module Notes

- All course content will be available through Webcourses.
- Self Enrol Key: **debian**

- If you cannot access the content please email Jonathan asap!!
- **jonathan.mccarthy@dit.ie**

Module Aim (from Course Document)

- 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 (from Course Document)

- Describe and employ the fundamental concepts, structures, mechanisms of systems libraries and calls of UNIX-based systems programming
- Use the UNIX tools in developing software in C, including gcc, gdb, ddd, gprof, cvs, make
- Use signals at the command level and as part of a program

Learning Outcomes (from Course Document)

- Write software using inter-process communication (IPC) and appropriate system calls
- Program terminal I/O and relevant system calls
- Write concurrent programs using processes and threads

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

Module Content

- **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

Module Content

- **Interprocess communication:** Process synchronisation and communication concepts
- **Pipes:** Programming Concepts, Limitations, Named pipes (FIFOs), Semaphores, Shared memory
- **Socket programming:** APIs and their implementation.

Module Content

- **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

- This module has the following:
  - 70% weighting for the examination
  - 30% weighting for the continuous assessment

- Examination = 70%
- Assessment = 30%
  - CA 1: 15% (week 9) (Provisional)
  - CA 2: 15% (week 12) (Provisional)

Late Submissions

- **Rules for late submissions:**
- **Week 1:** 4% for the first day, 2% for each day thereafter.
- **Week 2:** 3% for each day thereafter.
- **Week 3:** No submissions accepted, zero grade.

- **Note:** All penalties are calculated per day started.

Essential Reading

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

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.

Questions

Notes By: Jonathan McCarthy