

Software Development on Linux Systems

Independent Study

by

Cody Van De Mark



Today

- Introduction
- What is open source?
- Why an independent study?
- Why a course?
- What are the goals of the course?
- What is in the course?
- How do I convey material?
- How do I know I've succeeded
- Deliverables
- What Would I Do with More Time?
- Questions

Introduction

- Cody Van De Mark
- Graduate Student in Human-Computer Interaction
- Focusing on User Experience and System Architecture

What is Open Source?

- Libre: *the state of being free; having liberty; having freedom*
- Gratis: *to be free of charge; supplied without payment*

What is Open Source?

- Open Source
 - Practical benefits to open source development in business
- Free Software
 - Ethical benefits to society and innovation
- Free Open Source Software (FOSS)
 - Union of Free Software and Open Source

Why an Independent Study?

- Interested in Open Source:
 - Development
 - Business Approaches
 - Legalities
 - Software Innovation
 - System Architecture with Linux

Why a Course?

- My Perspective:
 - Forces me to learn material to a point I can teach it
 - Provides material to others who are interested
 - Gives me something to give back to the community

Why a Course?

- College Perspective:
 - Provides school with a packaged course
 - Provides students with skills applicable to industry
 - IST lacks a base in open source unlike other major schools
IE: MIT, Cornell, Carnegie-Mellon,

Why a Course?

- Student Perspective:

Provides students with:

- Software design and development skills
- System architecture and integration skills
- Open source business approaches
- Software legalities experience
- Understanding of industry's most popular tools
(IE: Linux, Apache, Android, MySQL, SQLite, CouchDB, etc)

What are the Goals?

- Ensure students:
 - Are familiar with Linux
 - Are familiar with open source
 - Understand licensing, trademarks, ownership and rights
 - Understand the basics of open source in business and economics
 - Understand open source communities and politics
 - Explore software development in open source
 - Understand the benefits of open source software development tools
 - Understand the integration abilities of open source software
 - Understand publishing, bug tracking and patching

What is in the Course?

- Topics:
 - Open source history
 - How open source works
 - Licensing, trademarks, ownership & rights
 - Open source communities & politics
 - Open source relationship to businesses
 - Software development life cycles
 - Software development tools & frameworks for Linux

What is in the Course?

- Topics:
 - Automation & configurations
 - Distributed Programming Systems
 - Developer & end user documentation
 - Linux distribution/environment differences
 - Integration with systems & services
 - Publishing & code hosting
 - Bug tracking, forking, merging & maintenance
 - Development issues, current themes and future development planning

What is in the Course?

- Materials:
 - 14 Lectures
 - 11 Labs
 - 2 Homeworks
 - 1 Semester Project
 - 1 Practical & Exam

What is in the Course?

- Skills/Tools Learned:
 - Fedora and Ubuntu software architecture
 - Linux environment configuration
 - Application environment configuration
 - Bash, C++, C, Python, Perl, Java, C#
 - Automation techniques
 - Distributed Programming Systems (git, bzt)

What is in the Course?

- Skills/Tools Learned:
 - Documentation programming and generation
 - Linux varieties and purposes
 - Packaging and distributable binaries (DEB and RPM)
 - Service integration (MySQL, SQLite, CouchDB, Apache, Django)
 - System integration across applications and languages (D-Bus)
 - Code hosting, publishing and bug tracking (Github & Launchpad)

How Do I Convey Material?

- Applied Course Design:
 - Lecture → Lab Design
 - Lectures provide theory and applied code
 - Labs are directly related to the previous lecture
 - Labs grow, directly using the skills from the previous

How Do I Know I've Succeeded?

- Student Evaluation:
 - Labs, lectures and homeworks directly use the skills from lectures and can be used for evaluation
 - Exam & Practical designed to evaluate course goals directly
 - Project requires skills based on course goals directly

How Do I Know I've Succeeded?

<u>Goal</u>		<u>Evaluation</u>
▪ Linux Familiarity	→	Labs and Practical
▪ Open Source Familiarity	→	Exam and Project
▪ Licensing, Trademarks & Rights	→	Exam and Project
▪ Business and Economics	→	Exam
▪ Communities & Politics	→	Exam, Project & Labs
▪ Software Development	→	Project and Labs
▪ Open Source Development Tools	→	Project, Practical & Labs
▪ Integration Abilities of Open Source	→	Project & Labs
▪ Publishing, Bug Tracking & Patching	→	Project & Labs

Deliverables

- Syllabus
- 14 Lectures (446 lecture slides)
- 11 Labs (186 lab pages)
- 2 Homeworks (2 pages)
- 1 Semester Project (3 pages)
- Student files provided for every lab
- Answer keys provided for almost all labs
- CD containing all deliverables

*All licensed to Creative-Commons Attribution Share-Alike 3.0

What Would I Do With More Time?

- Change the name to “Open Source Software Development on Linux Systems” (OSSDLS)
- Better testing of each lab
(some labs were not tested on both Fedora AND Ubuntu, but should still work)
- Testing labs on individual students
- Graduate level research requirements & assignments

Random Tidbits

- All documents are in open document (AKA Google document), MS doc and PDF formats
- The entire course was created using only open source software
- Most code required in labs is Python
- There is a total 637 pages in labs, lectures, homeworks & project
- This does not include the syllabus, answer keys, scripts, configuration files, the course outline form or this presentation

Questions

- Questions?
- Feedback?