Introduction

CSCI 4140: Open-Source Software Project Development

Prof. Hong Xu

Objectives

- Not to teach you how to open your source
  - We will introduce software licenses used in open-source software
- Teach you techniques to develop “real” software projects
  - “Most of your previous assignments are just TOYS”
  - “We will be implementing ADVANCED TOYS”
  - Arouse the creativity and fun in developing software projects

What Will You Learn In This Class?

- Becoming a *full-stack* developer
  - How to host your application/service in the cloud
  - How to implement the back end of an application
  - How to write a dynamic and responsive frond end for both desktop and mobile devices
  - How to communicate between the frond end and the back end
  - How to perform data analytics and ML
- Collaborating with other developers

Schedule (subject to change)

	Lecture	
Week	**Notes No.**	**Contents**
1	Intro; Version control	Outline, Licenses, OSS development model
2	Web apps / Overview	Procedure of request processing. (HTML, CSS, JS, HTTP, CGI)
3	Go	Server side, similar to Nodejs.
4	Javascript	Client side (browser used)
5	Session; Security; Authentication	Cookies …
6	Scripting; Communication	Bash, python; AJAX, JSON, websocket
7	Networking, video	WebRTC, Quic, video streaming
8	Mobile apps	
9	Cloud; Container; Serverless	
10	Databases (SQL, NoSQL)	
11	Spark	
12	Machine learning	
13	Final review	

What's Involved in the Learning?

- Absorb material presented in lectures and tutorials
- 3 individual programming assignments (39% total)
- A group project (30%)
  - No fixed theme
  - Up to 2 group members
- A comprehensive final exam (31%)
- Suggestions?

What's Required

- Prerequisites:
  - CSCI 2100 / CSCI 2520 / ESTR 2102
  - Familiarity with Linux, Java, Python, SQL, HTML
- Engage in lectures & tutorials
- Participate in Piazza
  - Post all course-related questions/comments there
  - No email: it doesn't scale!
  - For private matters, contact Prof or TA using Piazza direct messages
  - Do not post specifics about homework/project

Class Policies

- Late homework with penalty (http://course.cse.cuhk.edu.hk/~csci4140/#policies-late-homework-policy)
  - 3-day grace period for an assignment: -25% if < 1 day, -50% if < 2 days, -75% if < 3 days, no credit >= 3 days
  - 2-day grace period for project: -15% if < 1 day, -30% if < 2 day, no credit >= 2 days
- Collaboration (http://course.cse.cuhk.edu.hk/~csci4140/#policies-collaboration)
  - Work on your own (except for project)! Never share solutions, code, etc., or let any other student see them.
- Other policies
  - http://course.cse.cuhk.edu.hk/~csci4140/#policies

A Software Engineering Course?

Software Engineering Courses	This Course
Focus on the formal steps in developing software, independent of the technologies.	Focus on how to leverage nowadays technologies to develop software.
Focus on the requirements and the outcomes.	The requirements may be coming from the lead programmer himself and are usually subject to change.
Focus on the testing of the software based on the requirement specification(s).	Most of the time, there is only a goal with a loosely written specification. So, we are going to learn how to write robust software.
Formal courses	**A casual courses**

Questions?

Open-Source Software vs. Proprietary Software

Is A Free Software An OSS?

NOT a “free software”

Google Docs

“Google Docs, Google Sheets, and Google Slides are a word processor, a spreadsheet and a presentation program respectively, all part of a free, web-based software office suite offered by Google within its Google Drive service.”
- https://en.wikipedia.org/wiki/Google_Docs,_Sheets,_and_Slides

What Is “Free Software”?

- The Free Software Definition
  - “Free software” means software that respects users' freedom and community. Roughly, it means that **users have the freedom to run, copy, distribute, study, change and improve the software.**
  - “Free software” is a matter of liberty, not price.

You should think of “free” as in “free speech,” not as in “free beer”.

Richard Stallman
Founded the GNU project in 1983

https://www.gnu.org/philosophy/free-sw.en.html

What Is “Free Software”? (Cont.)

- The Four Freedoms
  - Freedom 0: The freedom to **run** the program for any purpose as you wish.
  - Freedom 1: The freedom to **study** how the program works, and change it to make it do what you wish.
  - Freedom 2: The freedom to **redistribute** and make copies so you can help your neighbor.
  - Freedom 3: The freedom to **improve** and distribute your modified copies to the public, so that the whole community benefits.

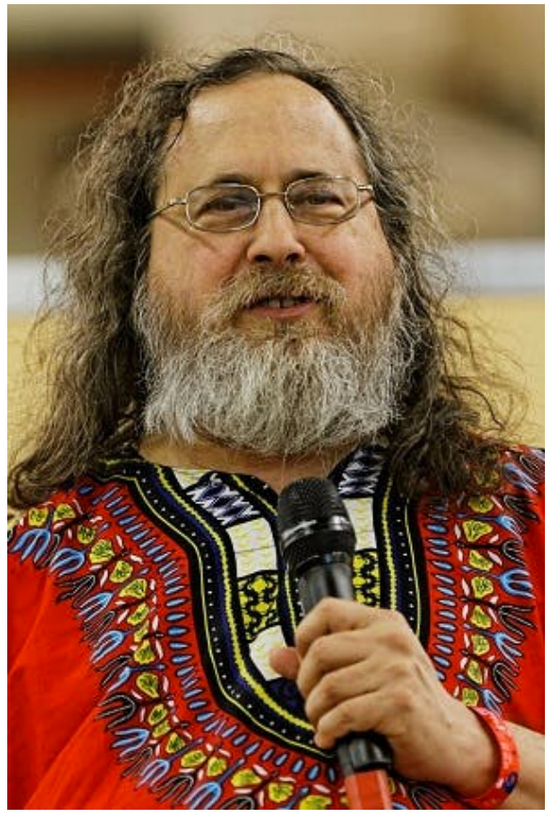
Requiring access to the source code

https://www.gnu.org/philosophy/free-sw.en.html

What Is Open-Source Software?

- Open-source software (OSS) is computer software with its source code made available with a license in which the copyright holder provides the rights to study, change, and distribute the software to anyone and for any purpose.
- Open-source software shares similarities with Free Software and is now part of the broader term Free and open-source software (FOSS).

Free software is a political movement; open source is a development model.



Richard Stallman
Founded the GNU project in 1983

A little detour: GNU

- GNU is a complete operating system; Linux is a kernel (check this out: https://www.gnu.org/gnu/linux-and-gnu.html)
- GNU has many essential software as part of an OS, most of which you use everyday without knowing it's from GNU
  - GCC, gdb, glibc, BASH, make, emacs, tar, …

Free and Open-Source Software Licensing

- Why do we need a software license?
  - To protect the holders of the intellectual rights.
- Free software license / Open-Source license
  - Types of license for computer software that allows the software and/or the source code to be used, modified and/or redistributed (under defined terms and conditions).
  - Explicitly grant actions that are usually prohibited by copyright laws.

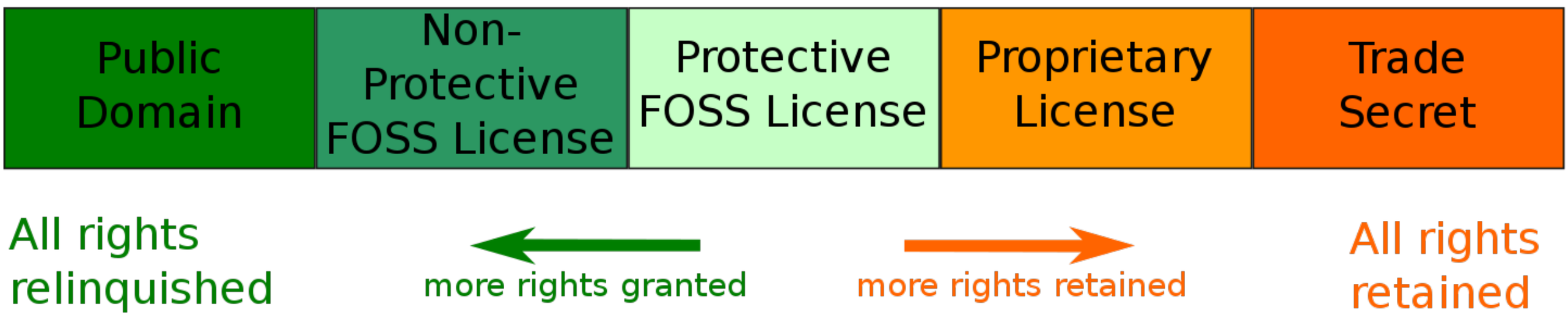
Copyleft 

- A form of licensing that requires the derivative works to be distributed with the same copyright conditions as in the original work.
- GNU General Public License (GPL) - written by Richard Stallman
  - Guarantees end users the freedom to run, study, share and modify the software.
  - Any modified versions / derivative works automatically carry the GPL.
- Other examples:
  - GNU Lesser General Public License, Mozilla Public License, AGPL

Permissive Software License

- A free software license with minimal requirements about how the software can be redistributed.
- “Non-copyleft license”
  - Doesn’t enforce the modified source to be opened when redistributed.
- Examples:
  - MIT License, BSD License, Apache License, Apple Public Source License

Software Licenses



Software licenses in context of copyright according to **Mark Webbink.**

FOSS vs Proprietary Software

- Benefits
  - Security
  - Affordability/Availability
  - Flexibility
  - Quality/Reliability
  - Longevity
- Drawbacks
  - Security
  - User-support
  - Bugs/Missing features
  - Compatibility
  - Licensing

Software Development Model

- The Cathedral (traditional) Model:
  - Centralized; roles are defined and managed
  - “A cathedral carefully crafted by individual wizards or small bands of mages working in splendid isolation”
- The Bazaar (open-source) Model:
  - Decentralized; roles are not clearly defined
  - “A great babbling bazaar of differing agendas and approaches”

“*The Cathedral and the Bazaar*”, Eric S. Raymond, 1997

OSS Development Model

- Users should be treated as co-developers
  - Linus’s Law: "Given enough eyeballs all bugs are shallow."
- Rapid prototyping
- Early releases
- Frequent integration
- Incremental and evolutionary development

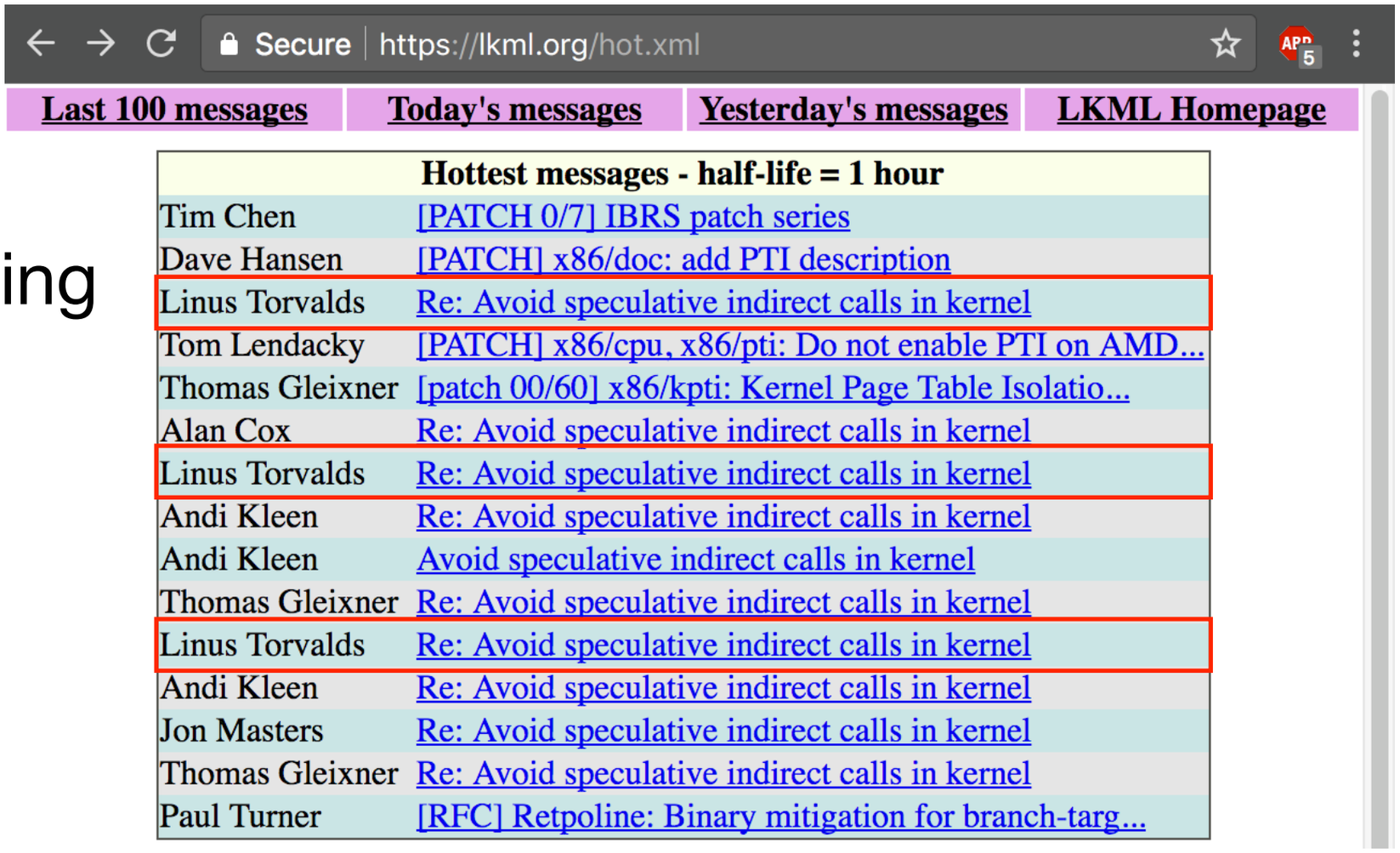
“Lessons for creating good open source software”

https://en.wikipedia.org/wiki/The_Cathedral_and_the_Bazaar#Lessons_for_creating_good_open_source_software

OSS Development Team

OSS Development Communication Channels

- Email
  - Mailing list
- Instant Messaging
  - IRC
- Forums
- Wikis



OSS Development Tools

- Version control systems
  - The Subversion revision control system (SVN)
  - git (by linus)
- Bug trackers and task lists
  - Bugzilla

History of FOSS

- Launch of the free software movement
  - Richard Stallman launched the GNU Project in 1983
  - GNU GPLv1 was published in 1989
- Linux
  - Linus Torvalds released the Linux kernel first in 1991, then in 1992 under GPL
  - Debian GNU/Linux was started in 1993
- FreeBSD and NetBSD were released as free software in 1993
- The LAMP systems powered the development of the World Wide Web in the dot-com years (late 1990s)
- Eric Raymond published The Cathedral and the Bazaar in 1997
- Netscape Communicator was released as free software in 1998
- The Open Source Initiative was founded in 1998
- Google released Android in 2008
- Google released Chromium OS in 2009

Trends in OSS

- New HTML5 specifications
- Cloud computing & mobile computing
- Internet-of-things
- Artificial Intelligence applications
- Augmented reality
- Virtual reality
- Cross-device applications

Focus of This Course

- Technologies around the World Wide Web
  - Web is the primary tool that connects billions of people and devices on the Internet
  - Web browsers are ubiquitous, versatile clients that are available on multiple devices
- The cloud infrastructure that empowers the web
  - virtualization, distributed storage, data analytics, ML

How To Start An OSS Project?

- Announces the intent to develop a project in public
- Releases a limited-but-working project to the public
- The source code of a mature project is released to the public
- Fork an existing OSS project

- It is very important to investigate what's already there!
- It is often better to contribute to an existing similar project than building one from scratch!
- Do not re-invent the wheels yourself!

Start Your Project Now!

- http://course.cse.cuhk.edu.hk/~csci4140/#project
- Form a group of up to 2 people by 23:59 January 30, 2021
  - Use Piazza as a platform to search for teammates
  - Post any idea, or even prototypes, that you want to build
  - Comment on other's post to express your interest and discuss the idea
  - Find people with similar interests
- Write a project proposal and present to the instructor and TA in February
  - We will estimate the score that your proposed project is worth