Free and Open Source Software

- **Free Software**:
  - means free as in "*free speech*" (vs. as in "*free beer*")
  - *freedom* to use as desire, copy, distribute, modify
  - often also free as in "free beer" (zero cost)
  - preferred term of Richard Stallman
- **Open Source Software**:
  - source code (original program) is available to user
  - to be able to *modify*, must have source code
  - alternative term to Stallman's "free software"

Free/Open Source Software (contd.)

- **FOSS**: Free and/or Open Source Software
- **FLOSS**: Free/Libre/Open Source Software (to emphasize meaning of “free”)
- Not all people like these acronyms---particularly FLOSS (because of dental interpretation).
- Note that software can be mostly *free but not open source*—e.g., Java (in the beginning).
- Software can also be *open source but not free*—e.g., source may be made available to paying customers (usually at additional cost).

Proprietary Software

- Alternative to FOSS is **Proprietary, Closed Source Software**.
- Proprietary:
  - numerous restrictions on use, embodied in **EULAs (End User License Agreements)**
  - e.g., how many instances can be run at once, what machines can run on, illegal to reverse engineer, how many clients can connect, need activation, etc.
- Closed Source:
  - only binary/executable version of program is provided–cannot examine nor modify code

Proprietary Software (contd.)

- Windows XP Home EULA example:
  - You may install, use, access, display and run one copy of the Software on a single computer, such as a workstation, terminal or other device ("Workstation Computer"). The Software may not be used by more than one processor at any one time on any single Workstation Computer.
  - The license rights granted under this EULA are limited to the first thirty (30) days after you first install the Software unless you supply information required to activate your licensed copy in the manner described during the setup sequence of the Software. You can activate the Software through the use of the Internet or telephone; toll charges may apply. You may also need to reactivate the Software if you modify your computer hardware or alter the Software.
  - You may permit a maximum of five (5) computers or other electronic devices (each a "Device") to connect to the Workstation Computer to utilize one or more of the following services of the Software: File Services, Print Services, Internet Information Services, and remote access (including connection sharing and telephony services).
  - Without prejudice to any other rights, Microsoft may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the Software and all of its component parts.
  - Microsoft reserves all rights not expressly granted to you in this EULA. The Software is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Software. The Software is licensed, not sold.

FOSS Examples

- Linux (operating system)
- KDE, GNOME, Xfce (desktop environments)
- Android (phone operating system/environment)
- Apache (web server)
- MySQL, PostgreSQL (DBMS's/servers)
- Perl, PHP, Python (scripting languages)
- OpenOffice (office software suite)
- GCC (GNU compiler collection)

FOSS Examples (contd.)

- GNU toolchain: Autoconf, make, etc.
- Git, Subversion, CVS (version control systems)
- OpenSSH (SSH server)
- Sendmail, Postfix (email transport software)
- Octave (GNU Matlab clone)
- GIMP (image manipulation a la Photoshop)
- Wordpress (blogging)
- Drupal (content management system)

FOSS Characteristics

- Often built collaboratively via Internet.
- Support for FOSS is typically provided via maillists, newsgroups, and web forums.
- Most FOSS is free of cost (“free beer”), so no continual cost for “upgrades.”
- Not limited to running single instance or having to guarantee license provisions are being met.
- Many companies exist to provide support or customization for businesses using FOSS.

FOSS History

- 1960's and 1970's: software was largely provided by computer companies and freely shared.
- 1969: **UNIX** developed at AT&T Bell Labs.
- 1969: **ARPANET** created.
- 1970's: AT&T provides CS departments with UNIX source code and encouraged modifications (could not sell due to 1974 antitrust findings).
- 1975: **Microsoft** founded, first product is BASIC for MITS Altair (an early microcomputer).

FOSS History (contd.)

- 1976: **Bill Gates** accuses hobbyists of stealing his software, thus preventing “good software from being written” (of course he paid nothing for BASIC).
- 1976: US amends copyright law, no longer requires explicit registration, etc.
- 1980: US copyright law amended to cover software.
- 1980: Microsoft launches UNIX-clone XENIX for 16-bit microprocessors.
- 1981: Bill Gates makes deal to buy DOS for $50k (without mentioning pending IBM PC deal).

FOSS History (contd.)

- 1981: Launch of the **IBM PC** with **MS-DOS**.
- 1980's: rise of **proprietary software**, companies quit sharing code and allowing modifications, and start charging lots of money for software.
- 1980's: **IBM** is #1 computer company with **DEC** #2 (DEC strongly associated with ARPANET but DEC anti-UNIX despite UNIX being developed on PDPs).
- 1982: AT&T divestiture (breakup) allows UNIX to be sold and the "**UNIX wars**" begin.
- 1982: **Sun Microsystems** born: UNIX **workstations**.

FOSS History (contd.)

- 1982: **Larry Wall** creates **patch** utility for UNIX, enables distributed, collaborative development.
- 1983: **DARPA**-funded **BSD** UNIX **TCP/IP** released.
- Early 1980s: ARPANET and UNIX hacker communities begin to converge on UNIX and C.
- 1984: MIT hacker Richard Stallman starts **GNU project** to promote "**free software**."
- 1984: **X Window** project begun at MIT to develop GUI for UNIX, supported by most UNIX vendors.
- 1985: POSIX starts to standardize UNIX.

FOSS History (contd.)

- Mid 1980's: DEC Vaxes running UNIX begin to take over ARPANET/NSFNET infrastructure duties.
- 1985: **NSFNET** created (ARPANET successor, and start of the civilian Internet).
- 1985: **Intel** releases **i386** chip, first 8086 CPU with flat address space that could support UNIX well.
- 1987: Larry Wall releases **PERL**, FOSS scripting language, for UNIX.
- 1987: first version of **GNU C compiler** released, and GNU development toolset largely complete.

FOSS History (contd.)

- Late 1980's: DEC and UNIX vendors continue to ignore the rise of Pcs and Intel microprocessors, allowing Microsoft to take over much of the market, and ultimately leading to the demise of DEC.
- 1987: source code (in C) for **MINIX** (mini UNIX-like OS) released for educational purposes by Prof. **Andrew Tanenbaum** (but does not become FOSS until 2000).
- 1990: first serious effort to port UNIX to i386 chips was begun: 386BSD (but project collapsed when sponsors wanted proprietary licensing).

FOSS History (contd.)

- 1990: Berkely begins effort to remove all proprietary AT&T code from BSD UNIX.
- 1991: Finnish CS grad student **Linus Torvalds** announces **Linux** project on **USENET**, with goal of producing a UNIX-like OS for Pcs (like MINIX), due to high cost of commercial UNIXes like Sun Solaris.
- 1992: AT&T sues Berkely over BSD UNIX, largely halting UNIX development at Berkely.
- 1994: AT&T/Berkely lawsuit settlement allows BSD UNIX to be released, free of AT&T code.

FOSS History (contd.)

- Mid 1990's: liberal **BSD license** allows companies (including Microsoft) to use BSD code in their products, leading to **Berkely sockets** becoming the de facto **network programming API**.
- Mid 1990's: Linux with GNU tools becomes the primary UNIX-like OS on PCs.
- 1995: **Red Hat Software** is founded, one of the first commercial **Linux distributions**.
- 1996: **KDE** desktop project started, but relied on non-free **Trolltech Qt** toolkit.

FOSS History (contd.)

- 1997: FOSS projects **GTK** toolkit and **GNOME** desktop are started over concerns about Qt.
- 1997: **Eric Raymond** publishes **The Cathedral and the Bazaar**, arguing that open source development models produce *better code*, which he summarized with what he termed "Linus Law":
  "with enough eyes, all bugs are shallow."
- 1998: Trolltech re-licenses Qt under "free" license.
- 1998: **Netscape** decides to open source its primary product, **Netscape Navigator** browser.

FOSS History (contd.)

- 1998: Eric Raymond and others found the **Open Source Initiative** (OSI) to promote "**open source software**" and to counter Stallman's extremism.
- Late 1990's: Intel-based platforms running Linux begin to **commoditize** the UNIX workstation market, starting the decline of UNIX workstation vendors such as Sun and Silicon Graphics.
- Late 1990's: **Apache web server** on Intel-based Linux machines begin to dominate the Internet web server market.

FOSS History (contd.)

- 2000's: Linux is increasingly widely used in corporate environments, particularly for servers.
- 2000's: Linux development is supported by numerous corporations that view it as commoditizing operating systems, reducing their reliance on Microsoft and eliminating the “Microsoft tax.”
- 2000's: open source software projects involving Internet-based collaborative programming become common, and commoditize many types of software.
- 2000's: virtually all **supercomputers** run Linux.

FOSS History (contd.)

- 2003: **SCO** sues IBM over claimed "UNIX IP" illegally transferred to Linux.
- 2007: SCO loses in court against **Novell** over ownership of UNIX IP, effectively ending IBM suit (plus repeatedly fails to prove UNIX is in Linux).
- 2007: Sun finally re-licenses Java under free license (but see below).
- 2007: **Google** releases **Android** OS based on Linux.
- 2010: Oracle sues Google over Java-related technology patents in Android!

FOSS History (contd.)

- Late 2000's on: Microsoft threatens Android over claimed Linux patent infringements (that it won't name!), extorts license fees from HTC and others.
- 2011: Android becomes the most widely sold OS on smart phones.
- 2011: Barnes & Noble makes Microsoft Android patent claims public (showing them to be trivial and possibly invalid patents), and initiates claims of anti-competitive behavior against Microsoft.

Richard M. Stallman (rms)

- Graduate student and hacker in the MIT AI lab during the late 1970's.
- One of the original authors of EMACS.
- Founder of the "**free software**" movement.
- Began GNU ("GNU's not UNIX") project (1984).
- GNU goal was to build a free complete UNIX-like system.
- Developed first "free software" license: GNU **General Public License (GPL)**

Richard Stallman (contd.)

- Much software in a Linux distribution is from GNU (rms insists Linux be called “GNU/Linux”).
- Stallman is very extreme and inflexible in his views, and as such has sometimes been a polarizing figure.
- The GNU project has made little progress on its UNIX kernel, Hurd, and Hurd has been largely supplanted by Linux (though work on it continues).

Linus Torvalds

- Linux began as a project when he was a CS grad student in Finland, as he wanted a UNIX-like OS he could use on PCs—and could afford.
- Originally modeled on **Minix** (a UNIX-like PC OS), and first postings were to **comp.os.minix**.
- Version 0.01 made available during 1991.
- One of the first Internet-based collaborative programming projects, and certainly one of the largest and most successful.

Linus Torvalds (contd.)

- Linus continues to serve as the main director for Linux kernel development.
- Linus owns the “Linux” **trademark**.
- Unlike Bill Gates, Linus has not become obscenely wealthy—though he is very well off now and very well known (lives in the US).
- Believes in FOSS (Linux licensed under GPL), but known as a pragmatist (used a proprietary VCS for Linux development for many years, until he developed the Git VCS).

Copyright

- Legal basis of free software requires understanding of how **copyright law** applies to software.
- Copyright is a legal mechanism that provides certain *exclusive rights* to the *author* of an *original work*.
- Works must meet a minimal test of *originality* to be eligible for protection (cannot copyright single words or someone else's book by changing a few sentences).
- International treaties on copyright are recognized by most governments in the world.
- Computer *software* is covered under copyright law.

Copyright (contd.)

- Prior to 1989, the US required an *explicit copyright notice* be included with a work for it to be covered.
- As of 1989, that requirement was dropped, so most programs you write are automatically copyrighted.
- Still a good idea to include explicit copyright notice, however, as damages for copyright infringement may be limited otherwise.
- Legal copyright notice format:
  - Copyright *years(s) name-of-copyright-holder*
  - © *year(s) name-of-copyright-holder*

Copyright (contd.)

- Generally, *copyright holder* is the *author*, but if work was done *for hire* (i.e., as an employee) then copyright holder may be *employer*.
- Note that copyright applies to a *program* (implementation) not to its **algorithm** (idea).
- To protect an algorithm, you would have to apply for a **patent**.
- Also, if someone else *independently* produces the exact same code as you (e.g., for a simple function), this is *not* copyright infringement.

Copyright (contd.)

- *Exclusive rights* granted to copyright holders include the ability to:
  - copy/reproduce the work (including electronically)
  - sell the work or copies of the work
  - display or make public the work
  - create **derivative works**
  - assign or sell any/all of the above rights to others

Copyright (contd.)

- In terms of computer *software*, copyright law prevents anyone else from doing the following with your code (without your permission):
  - using it or selling copies of it
  - including parts/all of it into their own code
  - starting with your code and modifying it
  - posting it on the Internet or publishing it in a book

Copyright (contd.)

- A copyright holder may transfer all or certain of his exclusive rights to another person/organization.
- A copyright holder also may grant a **license** to persons/organizations to engage in normally protected activities with his work.
- *Exclusive* transfer of rights or licenses must be done *in writing* in the US.
- *Non-exclusive* licenses need not be in writing (e.g., I can simply tell CS 306 students they are allowed to work by modifying my code template).

Copyright (contd.)

- Violation of copyright law is referred to as **copyright infringement**.
- Infringement is a **civil matter** (not **criminal**), and enforcement is up to the copyright holder (via a lawsuit).
- The US **Digital Millennium Copyright Act** (**DMCA**) includes a provision that protects ISPs and the like from liability for copyright infringement by their customers, as long as they remove infringing content when properly informed (via a "**takedown notice**").

FOSS Licenses

- Because of copyright law, software is *non-free* by default.
- Making software “free” requires that the author *license* people to use, copy, etc. without restriction.
- Due to differing goals and concerns, a variety of “free software” licenses have been created.
- It is important to understand that a “free license” is not the same as a transfer of copyright, since the author maintains copyright and thus some control.

GPL

- **GNU General Public License (GPL).**
- One of the oldest and most popular FOSS licenses.
- Developed by Richard Stallman as part of GNU.
- Considered one of the most restrictive FOSS licenses in the sense that it restricts what modifiers of the software are allowed to do.
- Makes work free software so allows modification, but requires that any modified versions must remain free under the GPL.

GPL (contd.)

- Furthermore, source versions of modified GPL code are required to be made available to anyone that receives only binary versions.
- This requirement causes GPL licensed code to be unsuitable for use in commercial software that vendors want to keep secret and proprietary.
- The GPL has been characterized as "**viral**."
- Stallman refers to the GPL as "**copyleft**" in that it ensures rights rather than restricting rights (as is the intention with copyright).

GPL (contd.)

- There are actually two versions of the GPL that are in use:
  - GPLv2
  - GPLv3

LGPL

- A different Stallman/GNU license is the **GNU Lesser General Public License (LGPL)**.
- This was developed to deal with issues that arose with GNU libraries, where the GPL was inappropriate, and it was originally named the GNU **Library** General Public License.
- Basically, does not apply GPL requirements to software that merely **links** with LGPL software.

BSD Licenses

- Originally created in connection with BSD UNIX.
- Considered permissive FOSS licenses in that there are few restrictions placed on what can be done with licensed code.
- This has led BSD-licensed code to be widely used.
- For example, Microsoft has included BSD-licensed code in Windows, as has Apple in Mac OS X.
- Of course, anyone can take your code and use it to make themselves a fortune—yet owe you nothing!

Other FOSS Licenses

- MIT (as used with X)
- Apache License
- Mozilla Public License
- Eclipse Public License
- Common Development and Distribution License (CDDL), developed by Sun

FOSS Organizations

- GNU Project (gnu.org)
- Free Software Foundation (fsf.org):
- Open Source Initiative (opensource.org)
- SourceForge (sourceforge.net)
- GitHub (github.com)

How to get Involved with FOSS

- Starting/managing a FOSS project can be complex and require many skills, so best to first assist with an existing project.
- Some projects are single-person, but many have communities of participants and will welcome new volunteers.
- Become familiar with project by using the software and subscribe to development maillists or the like to get up to speed with current needs.
- Try to find niche where you can really assist.

How to get Involved with FOSS

- Larger projects generally require people with a range of skills:
  - development (coding)
  - testing
  - documentation
  - translation
  - website management
  - software packagers