Open Source Software

Agenda

- Definitions of different “types” of software
- Key characteristics of each type
- Key points of difference
- Cost of free/open source software
- Historical overview
- Licenses
- Examples of free/open source alternatives in a few domains

Agenda (cont.)

- Extensions of the “open source” philosophy to other domains
- The anti-open-source brigade
- Reasons for using OSS

Open source software

- Software whose source code can be viewed, modified and re-distributed in its modified form after it meets certain licensing restrictions, e.g.,
  - indicating the authorship of various components
  - requiring that the modified version of the product also be open-sourced

Free software

- As defined by Richard Stallman, affords the following freedoms to its users:
  - Freedom 0: The freedom to run the program for any purpose.
  - 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 copies so you can help your neighbor.
  - Freedom 3: The freedom to improve the program, and release your improvements (and modified versions in general) to the public, so that the whole community benefits.

Source: http://www.gnu.org/philosophy/free-sw.html

Differences between open source and free software

- “The term 'open source' software is used by some people to mean more or less the same category as free software. It is not exactly the same class of software: they accept some licenses that we consider too restrictive, and there are free software licenses they have not accepted. However, the differences in extension of the category are small: nearly all free software is open source, and nearly all open source software is free.” (as stated by the Free Software Foundation)

http://www.gnu.org/philosophy/categories.html

Cost of free/open source software

- The “free” concept refers to the freedom(s) associated with modification and use of the source code, not to the cost of production and distribution
- Companies/individual developers/communities pay for the software through investments of time and/or money
- Often, end-users of such software contribute via donations of time (documentation, reporting and/or fixing bugs), money, advocacy, etc

Other “free” software

- *Freeware/Shareware/Trialware:*
  - software that is provided free of charge
  - may not have the freedoms provided by free software
  - may have limited functionality
  - may be available for use for limited time
  - usually provided to users to “sample” before they make a purchase

A brief historical overview

- The notion of “sharing” has existed well before computers and software came into being
- One of the earliest examples of “sharing” of commercially produced “intellectual property” in the US: the sharing of patents pertaining to automobiles in the early 1900s
- According to Richard Stallman (founder of the GNU Project) sharing of source-code, algorithms etc., related to computing was prevalent before 1970s in academic/research communities

A brief historical overview(cont.)

- With the founding of the GNU (GNU is Not Unix) project aimed to create a free operating system, a formal, community-driven effort began
- The Internet was and remains a key factor in the creation and spread of open source/free software. The Linux kernel and the GNU/Linux operating system are the most famous examples of free/open source software

A brief historical overview(cont.)

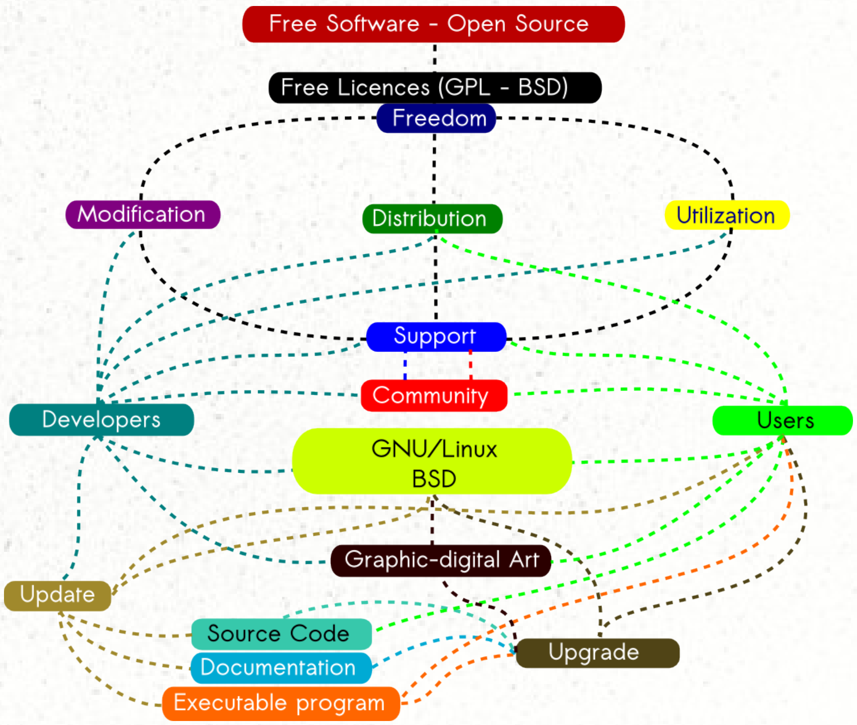
- Several projects have been and continue to be developed using the open source model:
  - Open source licensing (one of the several approved by the Open Source initiative (http://www.opensource.org)
  - Community-driven development and distribution
  - Sometimes, investments from commercial, non-profit, governmental, educational institutions

Timeline of OSS

Year	OSS Project	Description
1976	Emacs	Text-based IDE
1982	TeX	Document formatting system
1984	X window system	Precursor to the X server
1985	GCC	The GNU compiler collection
1987	Perl	An interpreted programming language
1991	Linux Kernel	The core of the GNU/Linux operating system
1991	Python	An interpreted programming language
1992	386BSD	An open-source BSD based on Unix
1992	Samba	“Free” implementation of SMB/CIFS networking protocol
1993	Wine	Wine Is Not an Emulator – an environment for running MS Windows applications
1993	FreeBSD	A Free Unix-like operating system
1993	NetBSD	Another free Unix-like operating system
1995	GIMP	GNU Image Manipulation Program
1995	PHP	A free, “server-side” programming language
1996	Apache	A free webserver
1996	KDE	The K Desktop Environment
1997	GNOME	GNU Network Object Model Environment
1999	OpenOffice.org	A free office suite (word processor, spreadsheet, presentation software, etc.)
2002	MediaWiki	Precursor to the software infrastructure of Wikipedia
2003	Firefox	A popular web browser, created by the Mozilla Foundation

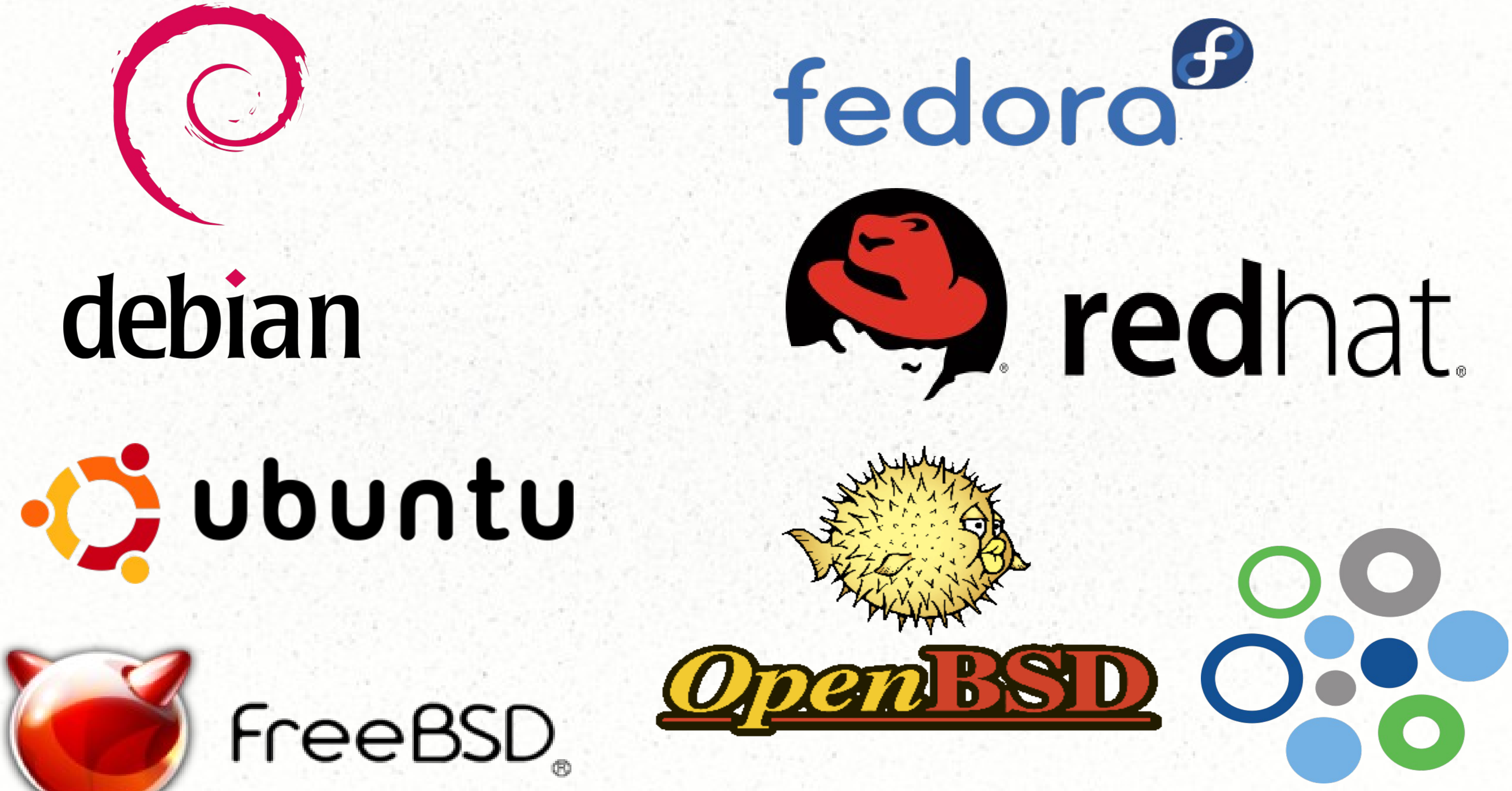
Licensing

- Several "open source" and "free software" licenses exist: GPL v2, v3, BSD, Apache, CDDL, MIT, etc.
- The key differences are in terms of distribution and use: commercially usable or not, whether the modified code should be released back to the community, etc.



http://upload.wikimedia.org/wikipedia/commons/2/2b/ConceptualMapFLOSS.png

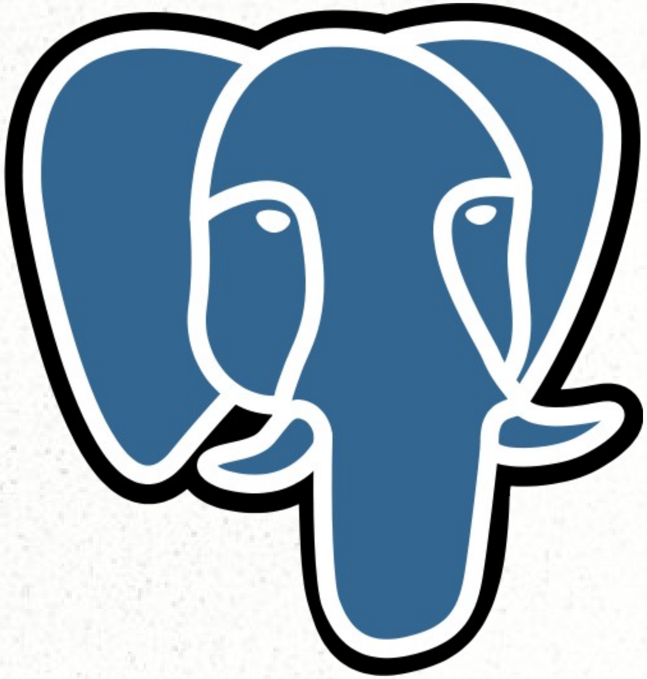
Domains of OSS prevalence – operating systems



OpenSolaris

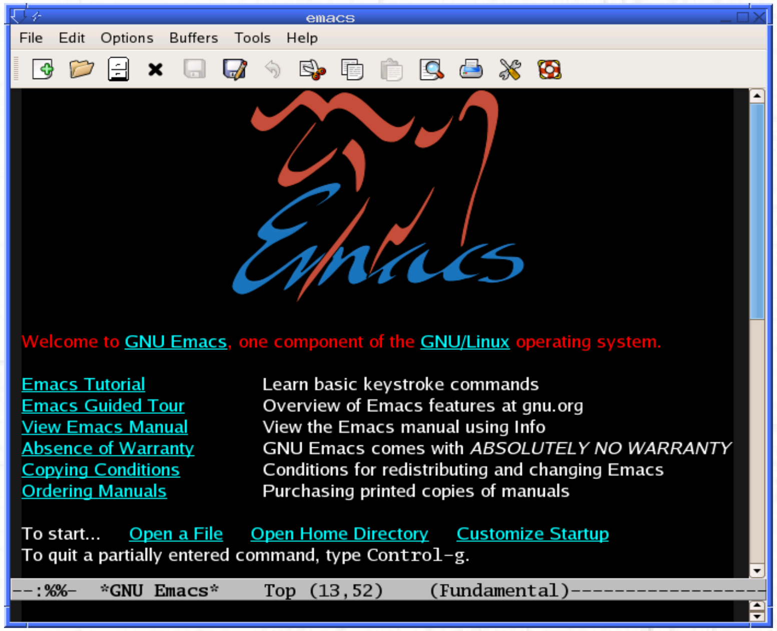
Source: wikimedia.org

Domains of OSS prevalence - RDBMSs



PostgreSQL

Domains of OSS prevalence – IDEs



Welcome to GNU Emacs, one component of the GNU/Linux operating system.

```
Emacs Tutorial          Learn basic keystroke commands
Emacs Guided Tour       Overview of Emacs features at gnu.org
View Emacs Manual       View the Emacs manual using Info
Absence of Warranty     GNU Emacs comes with ABSOLUTELY NO WARRANTY
Copying Conditions      Conditions for redistributing and changing Emacs
Ordering Manuals        Purchasing printed copies of manuals

To start...   Open a File   Open Home Directory   Customize Startup
To quit a partially entered command, type Control-g.
--:%%-  *GNU Emacs*    Top (13,52)    (Fundamental)-----------------
```

Domains of OSS prevalence – programming languages, compilers, interpreters

- Common Lisp
- Scheme
- Mono (open-source implementation of C# runtime + libraries)

Ruby

Domains of OSS prevalence – wordprocessing, spreadsheets...

OpenOffice.org

Domains of OSS prevalence – (digital) art

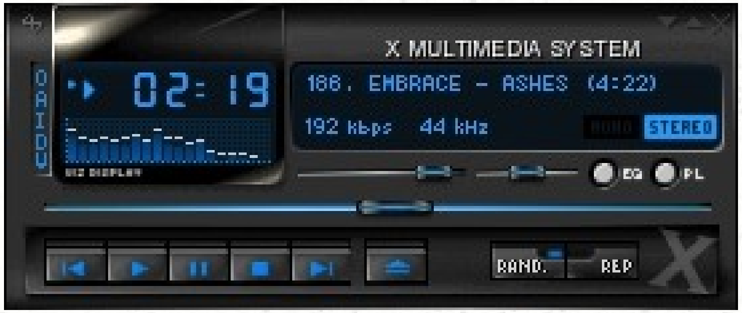
GIMP – GNU Image Manipulation Program

Blender - animation

Domains of OSS prevalence - science-related

- Science/research/engineering:
  - Libraries, packages for scientific computations: LAPACK, OpenMPI, Hadoop, Octave, GraphViz...
- Statistical computation:
  - R project
- Mathematics:
  - Sage, Macsyma, etc.

Domains of OSS prevalence - entertainment

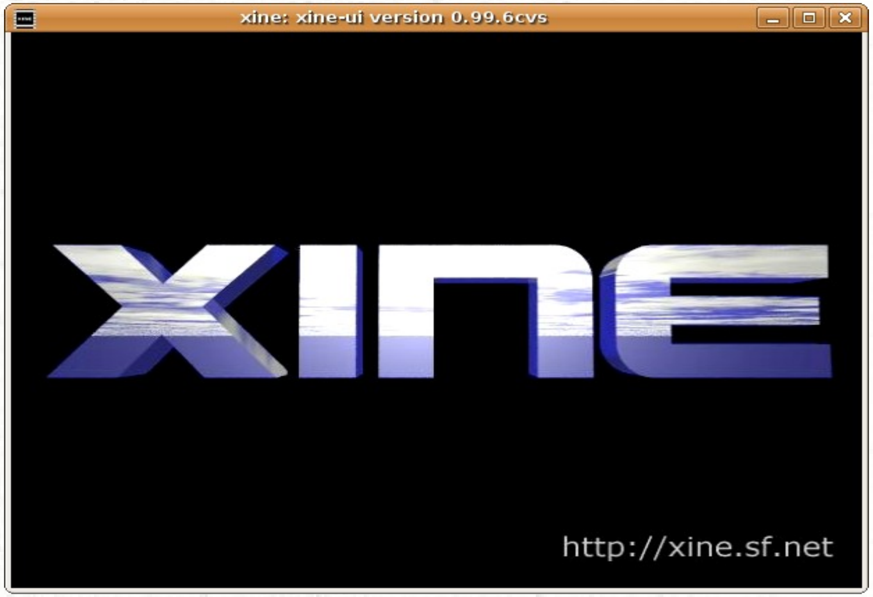


X MULTIMEDIA SYSTEM

02:19

186. EMBRACE - ASHES (4:22)

192 kbps 44 kHz MONO STEREO

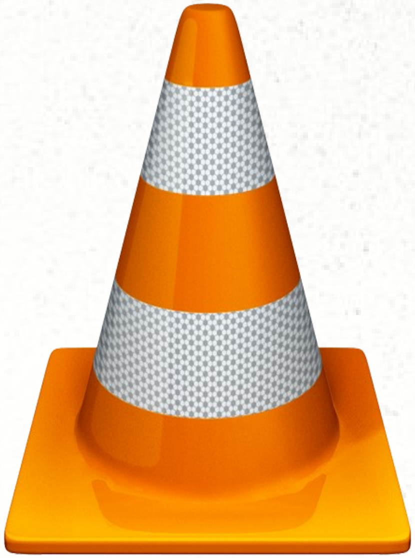


xine: xine-ui version 0.99.6cvs

http://xine.sf.net

xine-ui version 0.99.6cvs

NUM:00:00:00 AUD:auto SUB:auto



VLC Player

Domains of OSS prevalence – desktop environments

The OSS philosophy applied to other domains - documents/images/videos

- The GPL and its associated licenses informed other “licenses”:
  - Creative Commons license
  - GNU documentation license
- E.g., when searching for images to include in your project/document/etc., Flicker and Google let you search for “free” images

The OSS philosophy applied to other domains - hardware

- Publication of hardware specifications facilitates wider acceptance:
  - The community can create/modify/distribute drivers (in the case of computer hardware)
  - Collaboration among competitors leads to lower development costs, access to each others' expertise, etc., (e.g., some of the microchips in the “Sparc” range)

The OSS philosophy applied to other domains - education

- Sharing of course materials:
  - Textbooks
  - Videos
  - Projects-related software
  - Lesson plans
- Lowered cost of development and maintenance of materials
- Better quality (reviewed by a community of peers)
- Lower cost to students

Forces opposing “openness” of software, research/educational work, etc.

- Education
  - coalition of book publishers (Association of American Publishers) working to curtail the spread of “free” books through libraries, restricting the rights of authors to freely distribute their works, etc.
- Business
  - coalitions of proprietary software companies, e.g., the Business Software Alliance
  - pro-OOXML groups working against the adoption of ODF

cf.: http://www.opensource.org/node/510

Why use OSS - businesses

- Avoid lock-in that proprietary software vendors impose via their software
- Reduced licensing costs: most free/open source software does not have a 'per-seat' license fee; commercial support is available, but not mandated and charged
- Lower total cost of ownership
- More information, including case studies, can be obtained at: http://www.opensource.org/advocacy/case_for_business.php

Why use OSS - businesses

- Patents and proprietary software have been shown to impede, not facilitate, innovation
- *Commercial open source* has proven itself to be a viable business model, e.g., RedHat, IBM, Oracle, SugarCRM, etc.
- Commercial support providers have themselves proven to be reliable in supporting open source software
- Anecdotal evidence and survey-based research on online user communities indicates that the quick, high-quality feedback is available at no monetary cost

Why use OSS – educators/students

- Avoid “lock-in” (similar case made for the use of OSS by businesses)
- Lower cost of ownership of the software
- Understand how a particular functionality is implemented in the software without fear of repercussions of “reverse engineering” (e.g., refer to the EULA of a proprietary software that you installed recently)

Why use OSS - Government

- Sharing its data under the “open data” project – expected to promote *democracy*

  http://www.data.gov

- Lower cost of operations
- Better reliability of software
- Avoid vendor lock-in (similar to the case made for businesses)

http://www.oss-institute.org/newspdf/walker_oss_white_paper_2292004.pdf

Why use OSS - education

- Sharing of course materials, leads to a wider dissemination of educators' knowledge at relatively low cost
- Students and educators can access the “internals” of a software package without any negative outcomes associated with “reverse engineering” (refer to a recent EULA to which you agreed while installing some proprietary application)
- Educators who distribute “free”, online versions of their textbooks/tutorials reduce the cost of obtaining such materials to everyone

See, for example, http://www.ocw.mit.edu

Why use OSS – scientific community

- To ensure “correctness” of the software they use by being able to examine the implementation of various algorithms
- To suggest modifications to and/or implement modifications of incorrect algorithms
- Spend lesser amount of publicly-funded research grants on software

Further readings

- Coase's Penguin (analyzes the effects of free software on the economy):

  http://www.yale.edu/yalelj/112/BenklerWEB.pdf
- Defective by design: provides updates of various DRM-related issues (proprietary hardware/software combinations that affect users' rights) : http://www.defectivebydesign.org/
- Open source Institute and Free Software Foundation: advocacy and information pertaining to free/open source software: http://www.opensource.org/ and http://www.fsf.org/

Questions/comments?