

Open Source: an Overview

Dirk Vermeir

dvermeir@vub.ac.be

Dept. of Computer Science

Vrije Universiteit Brussel, VUB



Contents

- What
 - History
 - Motivation and Definitions
 - Open source packages
 - GNU/Linux
 - Other free operating systems
- Suppliers: Who, Why, How, Examples.
- Users: Why (not), How, Examples.
- Extensions beyond software?
- Discussion

What is Open Source Software (OSS)?

History

Background

- In the early days of computing, most software was "open source" in the sense that source code was exchanged (e.g. libraries for scientific computation, early unix) between users.
- Only later, most, but not all, software became closed (and, usually, commercial).

-1974

- 1969-1970 Birth of the Unix operating system (Ken Thompson, Denis Ritchie).
- 1971 Unix ed. 1, commands include `cat`, `mv`, `cp`, `chmod`, `chown`, ... (still heavily in use)
- 1972 Unix ed. 2
- 1972 The `C` language (Ritchie)
- 1973 16 unix sites
- 1973 Unix ed. 3, ed. 4
- 1974 Unix ed. 5,
- 1974 First published report on Unix.
- 1974 Thompson goes to UC Berkeley, B. Joy (grad. student) writes `em` (→ `vi`, `vim`)

1975-1980

1975 Richard Stallman writes the first version of [emacs](#).

1975 Unix ed. 6

1977 BSD 1 (UC Berkeley)

1978 BSD 2 (UC Berkeley)

1979 Unix ed. 7

1979 BSD 3

1980 BSD 4

1979 Eric Allman writes first version of [sendmail](#) (called delivermail).

1979 BSD 4.0 and 4.1 Unix from UC Berkeley

1980's Unix evolves: ATT SysIII, V, BSD (UC Berkeley), SunOs, Solaris, Irix, Aix, HP/UX, Ultrix ...

1981-1990

- 1983 TCP/IP adopted as military standard.
- 1983 Xenix (Intel HW, SCO and Microsoft – sic)
- 1984 RFC 882, 883 describe DNS (Domain Name System)
- 1984 Birth of the **GNU** project (**R. Stallman**): **gcc** etc.
- 1985 Unix ed. 8
- 1986 Unix ed. 9
- 1986 Initial development of **Perl** (Larry Wall)
- 1987 Andy Tanenbaum develops Minix
- 1987 First release of Perl.
- 1988 First release of **Tcl** (John Ousterhout)

1990-1994

1990 AIX (IBM)

1990 [Python](#) developed by Guido Van Rossum.

1991 Solaris 2 (SMP)

1991 BSD 386

1991 [Linux v. 0.0.1](#) release by Linus Torvalds.

1992 Linux v. 0.95

1993 [FreeBSD](#) 1.0

1993 [NetBSD](#) 1.0

1993 First release of [Samba](#)

1994 Linux v. 1.0

1994 [Red Hat](#) Linux, [Caldera](#) Linux

1995-now

1995 BSD 4.4

1995 First release of [Apache](#), based on NCSA's httpd daemon.

1995 FreeBSD v. 2.0, completely free code

1996 First release of the Hurd (GNU OS)

1997 [The Cathedral and the Bazaar](#) (E. Raymond)

1999 Linux kernel 2.2.0

1999 Red Hat goes public

2001 Linux kernel 2.4.0, IBM supports Linux on a range of machines (incl. mainframe)

2003 Linux kernel 2.6.0?, 177 distributions on [distrowatch.com](#).

Now

Google search	
query	hits
linux	51,700,000
windows xp	2,620,000
windows software	3,670,000
sex	82,600,000

- 65% of active (56% of all) sites use *apache* (may 2000, see <http://www.netcraft.com/survey/>).
- 33% of websites run linux, expected to grow to 41% in 2005.
- The 10 longest uptime servers all run apache on several BSD variants.

Thus: Open Source ..

- is based on a long (in CS terms) tradition.
- is hugely successful in that it has produced many packages/systems that
 - are of high quality
 - are in widespread use.(more on this further on).

Motivation and Definitions

The Cathedral and the Bazaar

(E. Raymond, 1997)

- The Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches .. out of which a coherent and stable system could seemingly emerge only by a succession of miracles.
- Too often software developers spend their days grinding away for pay at programs they neither need nor love.
- Release early, release often.
- Given enough eyeballs, all bugs are shallow.
- While coding remains an essentially solitary activity, the really great hacks come from harnessing the attention and brainpower of entire communities.

Motivation

(opensource.org)

- The basic idea behind open source is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.
- This rapid evolutionary process produces better software than the traditional closed model, in which only a very few programmers can see the source and everybody else must blindly use an opaque block of bits.

Note on Motivation

- Compare this with scientific progress which is largely based on free exchange of theories/knowledge among participants who may use and modify (falsify, extend, ...) them.
- The ideas on software evolution are compatible with recent trends in software engineering such as “extreme programming” and “agile development”.

Open Source Definition

(B. Perens, 1997)

Rationale: to foster rapid evolution/improvement:

- free redistribution
- include un-obfuscated source code
- allow modifications and derived works
- may demand integrity of the author's source code (identification)
- no discrimination against persons or groups, or fields of endeavor, or product (e.g. os), or technology
- distribution of license: you cannot e.g. add a non-disclosure agreement
- do not restrict other software that is distributed with it

Free Software (R. Stallman, 1985)

Free software is a matter of the users' freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

- The freedom to **run** the program, for any purpose.
- The freedom to **study** how the program works, and **adapt** it to your needs. Access to the source code is a precondition for this.
- The freedom to **redistribute** copies so you can help your neighbor.
- The freedom to improve the program, and **release your improvements to the public**, so that the whole community benefits. Access to the source code is a precondition for this.

GPL (v2)

GNU GENERAL PUBLIC LICENSE

- You can copy and distribute verbatim copies of the source code, you can distribute binaries only with (access to) the source code.
- You may sell the source code or offer warranty protection for a fee.
- You may modify the source code if the result remains free ([self-propagating license](#)).
- The LGPL variant is used to allow proprietary software to use e.g. a free library.

(Free Software Foundation, 1991)

GPL vs Other Licenses

- Free Software and GPL have also a philosophical motivation.
- GPL puts the additional **copyleft** (**copyright reversed**) protection/restriction which stipulates that redistributed modifications must be released under the same license (“forbidden to forbid”: “cancer” or “protection” ?)
- Other definitions encompass “free software” but are incompatible with the GPL, usually on the copyleft requirement.
- E.g. “public domain”, and many other “free” licenses, have no copyleft restriction and are not GPL compatible.
- GPL is most popular (e.g. Linux).

GPL in Practice

- GPL has been “tested” in a (US) court: Mysql AB (Sweden) vs Nusphere:
 - Nusphere derived a closed product from the GPL'ed mysql (database management system) package
 - GPL seems enforceable and binding (Nusphere settled).
- IBM has sued SCO for, among others, GPL violation because SCO allegedly sells licenses for GPL software which it distributed (under GPL) earlier.

What is Available as Open Source

Example Open Source Packages (1/2)

- Operating systems: Linux, FreeBSD, OpenBSD, NetBSD, Hurd, ...
- Programming languages (compilers, interpreters, IDEs): all.
- Internet infrastructure: bind (DNS), sendmail, nfs, apache, mozilla, konqueror, firewalls, ...
- Window systems: X, Gnome, KDE, ...
- Database management systems: mysql, berkeley db, postgres, ...
- Office applications: openoffice, gnumeric, gnucash, koffice, abiword, ...

Example Open Source Packages (2/2)

- Content Management Systems: zope, ...
- Web Services Systems: gsoap, ...
- Multimedia: gimp, xmms, mplayer, gnomemeeting, ...
- Literally thousands of general or specialized applications, e.g. for
 - clustering (beowulf),
 - grid computing,
 - bioinformatics,
 - ...

Open Source provides Choice

Note that for almost any (type of) functionality, there are several **alternative** open source solutions available. This leads to

- Competition on quality. E.g. the mail servers **sendmail**, **qmail** and **postfix** compete on "security record" and "ease of management".
- Products that fit (niche) user needs. E.g. for e-mail user agent programs, there are text-only (e.g. **mutt**) as well as GUI (**evolution**, **kmail**, ..), and web-based clients available. *If you don't like what's available, roll your own (reusing existing stuff).*
- Rapid take-up of useful novelties (e.g. tabbing in browsers).

GNU/Linux

The Kernel

- **Linux** is just the **kernel** operating system.
- A full distribution needs
 - A C compiler (**gcc**), linker-loader.
 - Lots of utilities re. files, text editing, networking, ...
 - A set of basic applications.
 - That is why R. Stallman insists on **GNU/Linux** as the name of the whole.
- A **distribution** is a package containing the kernel with selected (sometimes thousands) tools/applications (and the source code) as well as installation and administration tools.
- Distributions differ w.r.t. set of tools included, ease of use, expertise level of target user, initialization style (SysV or BSD).

Some Distributions (out of 177)

- Mandrake: multimedia, easy to install.
- Red Hat: the biggest market share (supposedly).
- Suse: the biggest (7CD's), very complete w.r.t. applications.
- Debian: extremely nice automatic update-via-internet service: `apt-get`. Non-commercial (you cannot buy it).
- Slackware: one of the older distributions, lean & mean, macho.
- **Gentoo**: automated get-and-compile selected packages (and dependencies) from the net yourself, customized and optimized for your machine.
- LFS (Linux From Scratch): compile everything yourself, the most “hard core” non-distribution.

Specialized and Derived Distributions

- Linuxinstall: no-click install (Red Hat-based)
- Lindows (Walmart's windows alternative)
- BBC-LNX is a mini Linux-distribution, small enough to fit on a CD-ROM that has been cut, pressed, or molded to the size and shape of a business card. See also "Damn Small Linux".
- En Garde Linux (and several others) stresses security.
- Smoothwall: "firewall out of the box" on dedicated (possibly old) machine.
- Knoppix: runs off a CD with full HW detection, office applications etc.
- Real time versions.
- It is not difficult to make your own (e.g. based on gentoo).

Compatibility

- No problem between distributions.
- Sometimes (seldom) between kernel versions although the kernel API changes extremely slowly.
- Dependencies between (applications and) libraries is a problem, e.g. video player V1.2 depends on library A1.2 and B3.2. If you have A1.2 and B2.4, V1.2 may not work (similar to “DLL Hell” in other systems?).

Customizing a Kernel

- There are many variables controlling e.g. which features to include or leave out (e.g. SMP, file system types, firewall support, . . .)
- Before compiling a new kernel, you just select the features you want using a text or graphical application ((x)menuconfig).
- Most device drivers can be inserted/deleted into/from a running (unmodified) kernel: **Loadable Kernel Modules**.
- There are many extra customized kernels available, e.g. with extra security (NSA, and many others).

Other Uses of the Linux Kernel

- Embedded systems (e.g. IBM's linux watch).
- Internet appliances (e.g. firewall, router, set-top box (Sony)).
- Clusters (Beowulf) and Grid computing.

Linux GUI

- Separate from the rest of the system in Linux (e.g. usually not useful on a server).
- Low level graphic library: X (Xfree), network-aware
- Window managers (many available) determine the look and operation of the desktop.
- Higher level libraries and desktops: mainly Gnome and KDE (both have a lot of graphical applications, which inter-operate (on the same desktop)).
- Many “raw” window managers are also available.

Other Open Source Operating Systems

Other Open Source Operating Systems

- BSD descendants ([FreeBSD](#), [NetBSD](#), [OpenBSD](#)) tend to evolve more slowly and in a more controlled manner.
- The [Hurd](#): the GNU operating system (based on Mach micro kernel, not monolithic like linux). Not ready for production (distributed by Debian).

FreeBSD

- Technical simplicity.
- The FreeBSD installation program is widely regarded as the simplest Unix installation tool in existence.
- Its third party software system, the Ports Collection, has been modeled by NetBSD and OpenBSD and remains the most powerful application installation tool available. Through simple one-line commands, entire applications are downloaded, integrity checked, built, and installed making system administration amazingly simple.
- FreeBSD governance and development model lead to a very stable and easy to use system. As one of the most reliable operating systems for the x86 platform, FreeBSD's mark lies in sustaining an air of simplicity and stability.

NetBSD

- NetBSD's focus lies in providing a stable, multi-platform, research oriented operating system.
- NetBSD's portability leads it to run on 33 platforms as of January 2001, including exotic hardware.

OpenBSD (spin-off from NetBSD)

- Aims to be the most secure operating system available. Includes strong cryptography including RSA, Blowfish, and other advanced algorithms. User passwords are encrypted by default. OpenBSD developers also spear-headed the development of OpenSSH, a multi-platform clone of the wildly popular protocol for secure communications.
- OpenBSD also advanced the state of code auditing. Beginning in 1996, the OpenBSD team began a line-by-line analysis of the entire operating system searching for security holes and potential bugs.
- OpenBSD's mantra of "secure-by-default" has produced one of the most robust operating systems available.
- 4 years without a remote root-exploit allowing many system administrators to sleep better at night

Open Source Providers

Observation

“Open source software tends to be developed by loosely-organized ad-hoc communities with contributors from all over the world who have never met face-to-face .. this mish-mash of people coheres and effectively accomplish an extremely complex task: building high-quality software” (Kim, 2003)

- Who are these people?
- How are they organized?
- Why do they contribute?

Statistics about Projects

(March 2003)

Repository	projects
SourceForge	58,245
FreshMeat	27,514
GNU/FSF repository	2,017

Many large projects (e.g. Mozilla, Gnome, Kde, XFree, Tex ...) have their own repository.

Some Companies Contribute

Examples:

- XFS journaling file system (SGI)
- JFS, OpenAFS journaling file systems (IBM)
- Lots of stuff (IBM)
- Openoffice (Sun)
- Device drivers (many manufacturers).
- Kernel contributions re. security, clustering (Oracle, IBM)
- IBM pumped 1bn US\$ into Linux in 2001.
- ... (some fund projects secretly :-)

Some HW Companies offer Linux

Examples:

- Usually on servers: Dell, HP, SGI, Sun.
- On PC: small firms or only for “large” customers (e.g. Thailand initiative HP).
- IBM offers Linux (up to thousands of instances) on a Z series mainframe as a cheaper alternative to a server park.
- In US, people succeeded, by going to court, in getting a refund for an unused operating system that was pre-installed on a PC.

Statistics about People

The typical open source developer is:

- male (98%)
- young (70% between 22-37)
- living in Europe or the US (80%)
- an IT professional (>50%), or a student (20-30%)
- a college graduate (33-46%), or only has a high school degree (17-24%).
- part-time involved: 34-48% spends <5 hrs/week, 9-15% spends 20-40 hrs/week, 5-7% spends >40 hrs/week.

(FLOSS 2002, Boston Consulting Group 2002, WIDI 2001, Kim2003)

How are they organized?

There seem to be two types of active project organization models:

- The Cave model: few (<4, often 1) people contribute and control the project, with input (requests, bug reports) coming from the users. Most projects follow this model. (Krishnamurthy, 2002).
- The Community model: anyone can contribute but contributions are vetted for quality etc. by an “inner core” (10-30) of people who are responsible for sub-projects.

TouchGraph: Cave

(Kim 2003)

- Visualization and (graph) navigation component to be used in end-user packages such as GoogleBrowser.
- Started by A. Shapiro (US), an AI researcher, who wanted to test his ideas. Shapiro worked as a consultant and now works full-time on the project.
- Only indirect contributions to source code from others, most from users with requests (Europe, for use in semantic web applications). Direct contributions are difficult because the system is not modular enough.

SquirrelMail: Small, Community

- Web-based IMAP (email) client.
- Started by high-school student in 1999.
- At least 1 million users, among which ISPs, Universities, ...
- Supports 30 languages, plugins (spam filtering).
- Core team of 10 project leaders and 5 active contributors. One overall project lead (not the original author).
- Divided into 7 projects: stable release, development release, internationalization, plug-ins, user support, documentation and evangelism, system administration.
- The organization evolved **gradually**, adapting to the explosive interest.

(Kim 2003)

The Kernel: Large, Community

- There is a responsible person for each branch (2.2, 2.4, 2.5). He decides what goes in (benevolent dictatorship). Anyone (companies or individuals) can contribute (e.g. SGI contributed XFS).
- Some Key people:
 - Linus Torvalds (fi, now us, “top dog”),
 - Alan Cox (us?),
 - Andrea Arcangeli (Imola, It, works for SUSE while being also a CS student at Bologna),
 - ... (many others).
- Communication is mainly by mailing list: e.g. between 1-7 sep. 2003, there were 2096 messages sent to the list.

Kernel Versions and Evolution

(May 2002)

- Versions: *X.Y.Z*. Stable versions have *Y* even, development versions have *Y* odd.
- Current versions: 2.4.18 (stable), 2.5.17 (development).

- Development goes fast:

2.5.16 18/5/2002

2.5.17 21/5/2002

- After enough tests and new features, 2.5.X will become a 2.6.0 (or 3.?) stable series.

Why do companies contribute?

- To foster acceptance of new technology: an open standard with a reference free implementation has a better chance for quick take-up.
- To lower cost of HW system that can be sold unencumbered by the need for an accompanying SW license (e.g. embedded systems).

Why do people contribute?

What motivates developers to voluntarily provide, support and maintain a public good?

- Need for software (Raymond 1999, ...)
- Fun & challenge (Torvalds 1998, Raymond 1999, ...)
- Labor Market Signaling, Career Concerns, Learning (Lerner 2000, Lakhani 2003)
- Status Signaling (Raymond 1999, Kollock 1999)
- Ideology & Social Movements (Stallman 2001)

A Short Term Contributor

“I was using the software for work. It’s excellent but there was a feature that I wanted that was missing . . . I searched the documentation and mailing lists for information and to see if I had overlooked something, finally I asked a question. That spurred some conversation and someone suggested a beautiful way to implement the idea.” (us, 29)

(Shah 2003)

A Long Term Participant

“I pick and choose the work that’s most interesting to me . . . it’s great when you find a challenging problem to work on – either on your own or because someone needs it – you can spend hours on it . . . The routine stuff is okay, but I don’t do much unless I just want to hack for a while and there are no really interesting problems around . . . When I get bored, I’ll leave . . . ” (fr, 28)

(Shah 2003)

Another survey

- Why did you join an OSS project?
 - Learn and develop new skills (78%)
 - Share knowledge and skills (50%)
 - Improve existing OSS product (33%)
 - Solve new problems (30%)
- Why do you stay in an OSS project?
 - Learn (48%)
 - Personal sense of accomplishment and contribution (25%)

(FLOSS 2002, Boston Consulting 2002)

Motivation and Quality

- Intrinsic motivation:

“People will be most creative when they feel motivated primarily by the interest, enjoyment, satisfaction and challenge of the work itself ... and not by external pressures or inducements.” (Amabile, 2003)

⇒ This may explain the high quality of many OSS packages.

Rational Decision on Participation

- Decision 1:

- Join a collective development process; or
- Build in isolation.

If a developer joins and contributes, his work will automatically benefit other joiners (who may not contribute).

- Decision 2: within the collective:

- Contribute; or
- Free-ride.

- In what circumstances is it rational to join and contribute?

(Baldwin & Clark, 2003)

Modular Architecture \Rightarrow Participation

- A modular design splits a system in several modules that can evolve (improve) independently; i.e. a module may be substituted for an newer version (option value).
- Design is a search under uncertainty.
- For a system with a modular design with option values, it is beneficial to join and contribute: more alternatives are explored with less effort (for the individual).
- This fits observed practice (e.g. TouchGraph).

(Baldwin & Clark, 2003)

Motivation for Field Support

OSS support is usually based on FAQs and forums. Why do people do such a boring job for free?

- They **learn** from reading the questions and answers (98% of time)
- If one knows the answer, it is easy and cheap (2% of time) to provide an answer.
- One gains reputation.
- One expects to be helped in return, if need be.

(Lakhani & Hippel, 2003)

Making money or a living from OSS

- Earlier idea: package and sell (Red Hat, SUSE, ...). Only moderately successful.
- Provide support or special extensions/customizations.
- Sell documentation, courses (e.g. O'Reilly).
- Consultants (e.g. IBM, local firms).
- Lower price of system by using OSS (IBM, Oracle).

Open Source Users

Factors in Open Source Adoption

- + HW cost (commodity)
- + SW cost
- +/- Perceived reliability (FUD?).
- + Availability of 3rd party applications.
- ? Portability of own applications.
- ? **IT worker skills**
- ? Fit to task (e.g. server).
- + Triability (reduces risk).
- + Central IT service (better IT skills)
- Large budget (can choose more expensive options).
- + Manpower available to evaluate new technology.
- + Innovation-oriented. (Dedrick & West, 2003)

Factors in Open Source Adoption

- + Avoid vendor lock-in: adherence to open standards is necessary for an efficient software market to satisfy software needs.
- +/- Generate in-house expertise.

Example: machine producers may prefer OSS to ensure easy interoperability and open competition.

Software Licensing Headaches?



Use Linux.

Price Hikes? Forced upgrades? Audit threats? Legal hassles?

We have the cure for your pain. A Cybersource study shows Linux and Open Source platforms have between 25% and 34% Total Cost of Ownership advantage over proprietary equivalents.

Cybersource Australia's leading specialist IT Services Firm. www.cyber.com.au

© 2002 Cybersource Pty Ltd. Linux is a trademark of Linus Torvalds. The goal of pain is to provide the right to easily be able to use the Linux & Open Source software. Linux & Open Source software, including the red & blue logos, is a registered trademark of Linus Torvalds.

Total Cost of Ownership

- Source: <http://www.cyber.com.au/> has a full report with detailed costings, soft- and hardware choices.
- Assumptions: 250 users, PCs, servers, internet connectivity, HW, cabling, 3 support staff (1 sr, 2 jr). e-business system

- Results (3 year period):

Scenario	Microsoft	Open Source
Use existing HW	\$733,973	\$482,580
New HW	\$1,042,110	\$790,717

- Interesting: linux staff is a bit more expensive.

Case Study: Beaumont Hospital (Dublin)

- B. Fitzgerald, T. Kenny, *“Open Source Software can Improve the Health of the Bank Balance - The Beaumont Hospital Experience”*,
<http://opensource.mit.edu/papers>.
- staroffice office applications
suse mail mail server
zope content management
claroline (ucl, be) e-learning
jboss application server
... medical images
- Total savings: 13MEURO over 5 yr (1.5MEURO without medical image systems) over academic discounts system.

Support for OSS Users

- Extremely quick support from
 - the **author** of the software: open source authors have an interest in finding bugs and be famous for their software's quality ("**fame, not fortune**"),
 - **colleague-users**, via a mailing list or a forum: most distributions and packages have dedicated forums and/or mailing lists where users provide 24hr service to each other.
- Good **commercial support** is also available from IBM, HP, other smaller companies. Most of them (e.g. HP) don't care about the HW and provide round-the-clock service.

Forum Support

- Example: the Gentoo forum has 29,000 registered users, of which there are typically 200 on-line (24hrs 7/7). The database contains 500,000 articles.
- (Own experience) Comparing open-source support via forum/e-mail vs. commercial support (from a reputable vendor):
 - + much faster (max. 1 hr vs. at least half a day, if you're lucky)
 - + much more specific: you usually only get answers from people that solved your exact problem; they are proud to share their knowledge.
 - + cheaper.
 - if you are the first to experience a problem, you may have to get involved in solving it yourself (and report it on the forum).

Public Organizations and OSS

For the public sector, there are extra factors in favor of using OSS: Munich officials motivate the switch to Linux on 14K PC's as follows

“... officials said that the decision (to switch to Linux) was a matter of principle: the municipality wanted to control its technological destiny. It did not wish to place the functioning of government in the hands of a commercial vendor with proprietary standards which is accountable to shareholders rather than to citizens”

(The Economist, sep 2003)

Motivation for Public Sector to use OSS

Same as for other organizations, plus:

- Open standards for data communication and storage (archive).
 - Refusal to blindly trust systems with secret source code.
 - Adopting open source encourages the growth of the local IT industry (as opposed to box shifters that simply import commercial offerings), e.g. to tailor OS products for local use (e.g. in de).
- ⇒ Save on capital, invest in people and knowledge.

Some Recent Events

Re. OSS for the Public Sector:

- German government funds extension of OSS packages for use by government agencies.
- Laws re. OS adoption by public sector have been proposed in several countries (including Belgium).
- Thailand offers cheap PC's with Linux to citizens.
- China, Japan, Korea announce joint development of open source operating system (probably Linux-based).
- ...

A number of considerations are nicely explained in the rebuttal by senator Nunez to complaints re. a proposed law requiring OSS for the public sector.

Simple Example Linux Deployments

Linux Deployments

Some simple general-purpose examples:

- Typical server.
- Typical utility box.
- Workstation for developer.
- Office workstation.

Typical Server

- Apache with mod_ssl, proftpd.
- Php or Perl or Python as a scripting language.
- Mysql or Postgres for database management.
- Postfix or sendmail or qmail as a Mail Transport Agent
- ProcMail or ... as a mail delivery agent with Bayesian anti-spam plug-in (many possibilities).
- Mailman or majordomo or Smartlist as a mailing list handler, mailman or hypermail for mailing list archives.
- Swish++ or ... as a full text indexing tool (search your site or mailing list archives).
- phpBB or ... as a forum system.
- Zope or ... as a CMS
- Samba as a windows server (even PDC).
- ...

Firewall Utility Box

- Even on older HW (486)
- Iptables to filter packets.
- Tcpwrapper
- E.g. run from CD to make corruption impossible.
- ... (no experience)

Developer Workstation

- Editor: `vim` or `emacs` (or `pico` for children).
- The gnu compiler suite: C++, C, java, objective C, fortran, . . . , scripting languages.
- `Make`, GNU autotools: `automake`, `autoconf`, `libtool`.
- Configuration management using `cv`s.
- `Doxygen` to generate documentation in many formats.
- Network backup using `rsync`.
- IDE's: e.g. `kdevel`oper, . . . (I don't use them)
- The `Qt`, `KDE` or `gtk++` GUI libraries. (Qt is also commercially available under windows).
- User mode Linux for testing..
- . . .

Office workstation

- Openoffice (replacement for Office), staroffice (Sun, commercial support available), abiword (gnome), gnumeric, gnucash ...
- Viewers: konqueror (kde), mozilla, firebird, galeon (gnome) opera (not free), netscape (not free)
- Multimedia: gimp, video/tv viewing and editing, ...
- Email clients, drawing programs, plotting and statistics, project management, ...
- etc. (e.g. Suse has 2000 applications)

Extending Open Source

Extending Open Source

- Information/knowledge is a non-rival good: can be duplicated at extremely low marginal cost. Often, its value increases if more people use it (e.g. forum).
 - Transaction costs hamper optimal matching of information demand/supply
- ⇒ There is a rationale for open access to knowledge.

Examples:

- Open Courseware (MIT)
- Open encyclopedia's: Nupedia, Wikipedia.
- ...

The End