

Implementing Free Software Solution in Workstations

Case: Linux in Helia

Abstract

This paper examines possibilities of replacing proprietary software in workstations with Free software. The main point of interest is the use of Free software in medium to large organizations. In addition to the obvious benefit of saving licensing costs, other possible benefits of Free software are listed.

Method used in this thesis is constructive case research. The definitions and benefits are searched from previous research and the publications of Free software organizations. An installable example solution is produced. The example solution is then used in a test environment.

As the result, Free software was defined and a simple criteria for choosing Free software was discussed. A working and installable example implementation of a Free software workstation was created and tested. Tools required for administering a large network of workstations were used as system in a test environment, except for centralized user authentication. Even though the specific software versions and products change trough time, the evaluation criteria, the needs for workstations and even the protocols used are likely to be the same for a long time. It is possible to implement a whole workstation with Free software, and thus it is possible to get all software for workstations at no cost.

Keywords: Free software, Linux, GNU, license, distribution, workstation

Contents

- 1 Introduction.....4**
 - 1.1 Background.....4
 - 1.2 Previous Research.....5
 - 1.3 Goals and Research Problem.....8
 - 1.4 Method and Structure of the Thesis.....9
 - 1.5 Principle Findings.....10
- 2 Licenses and the Definition of Free Software.....11**
 - 2.1 Popularity of Licenses12
 - 2.2 Freedom vs Protection of Rights.....13
 - 2.3 Licensing risks14
 - 2.4 Recommended licenses17
- 3 Software and Operating System for Workstations.....18**
 - 3.1 Distributions.....18
 - 3.2 Software needs.....27
 - 3.3 Office suites.....32
 - 3.4 Web Browsers38
 - 3.5 Running Windows Software in Linux.....39
 - 3.6 Other Information Systems43
 - 3.7 File Formats.....43
- 4 Administration of Workstations.....47**
 - 4.1 Current solution47
 - 4.2 Methods of Software Installation and Update48
 - 4.3 User authentication.....53
 - 4.4 Remote Control.....56
- 5 Practical Recommendation for Case Organization.....59**
 - 5.1 Costs.....62
- 6 Conclusions.....65**
- 7 References.....67**
- 8 Appendixes.....74**
 - 8.1 Automated Installation.....74
 - 8.2 Estimating the Number of Users in Helia.....77
 - 8.3 Yum automated software installation and update for Red Hat Linux.....77
 - 8.4 Legal Notice.....80
 - 8.5 Currently Installed Software in Helia and Free Alternatives.....81
 - 8.6 Glossary.....83

1 Introduction

Software licensing costs and unilateral licensing terms have pushed many companies to seek new solutions for obtaining software for day to day business. Large organizations are either researching or already moving to Free software both in Finland and abroad. Some of the biggest countries of the world are running projects for wider adoption of Linux. The Chinese government is funding a Chinese Linux distribution, and many believe that the unified platform for Chinese e-government will be based on Linux (Shen 2005). Many government organizations have adopted Linux, such as the German national Railway Deutsche Bahn (Heise 2005). In Finland, many universities such as Helsinki University and Helsinki University of Technology have rolled out Linux on significant part of workstations to replace Microsoft Windows.

On the server side, Free software has had a strong position for a decade. For example, the Free Apache web server has been the most popular web server since 1996, currently being more popular than all competing web servers together (Netcraft 2005). Linux is the most popular Internet server in Europe (Zoebelein 1999). Other very successful Free Internet servers include BIND, the most popular domain name server and Majordomo, the most popular mailing list system (Castellucio 2000).

Desktop workstations are currently the field of most interesting competition between proprietary and Free software. Microsoft Windows clearly dominates the desktops of both homes and companies, claiming to have over 95% market share of desktop operating systems (OS). Because there is no requirement to pay anyone or register anywhere when running Free OS'es, they cannot be counted. However, practical experience shows that Microsoft is leading the desktop market and has at least nearly as high market share as estimated. Historically, Microsoft has been able to use and abuse its dominating position on desktop operating systems to put down competition using “embrace and extend” tactics. As software licenses can easily form half of the price of a workstation, Free software could result in considerable savings.

Free licensing is more than just the technicalities of coding. A growing number of researchers see that the Free licensing model and its way of collaboration across organizational borders could extend to production of other information products (Benkler 2002. Cole & Lee 2003). Feasibility of Free licensing outside software has also been practically proved, as Free licensing has been used in texts outside software documentation. Wikipedia is the worlds largest Free encyclopedia. Having currently more than a million articles, it is also the largest encyclopedia ever. Project Gutenberg is a huge collection of classical literature with expired copyrights. Smaller projects have experimented with Free licensing in original photography (openphoto.net 2005) and music.

1.1 Previous Research

Traditionally, IT-projects have been run manager led, top down and with strict timetables. Brooks (1995) laid out the principles of running a traditional software project in the Mythical Man-Month. The first edition was published in 1975, when software development was a rare skill. Brooks' ideas are still vivid, having achieved a status of a proverb among IT-professionals, such as “adding programmers to late project makes it later”. Software projects fail because of the tar pit effect, the sum of many simple problems (3-9) and these problems cannot be tackled simply by adding man power (16-17) because of added requirements on

communication (babel tower, p 73-83). Even though there is no one single method to solve all these problems, the main forces keeping the project together are main architects conceptual integrity and separating designing from implementation (255 – 257).

The basic theory of how a Free software project works was put on paper by Raymond in 1997 in the first version of the Cathedral and the Bazaar. He noticed that many Free software projects contradicted the traditional wisdom of Brooks' cathedral, but were still very successful. Raymond's (2000) rhetorics of calling a traditional hierarchical, proprietary software project a cathedral and a Free software a bazaar have been very popular ever since. Cathedral and the Bazaar is cited on many texts on Free software (for example, its quoted on Shen 2005, Cole & Lee 2004, Zittrain 2004, Bretthauer 2002). Raymond wrote a Free email downloading program fetchmail. He took ideas and concepts from Linux kernel development, and generalized them into rules for running successful free software projects.

According to Raymond (2000), a Free software project should start by quickly creating a practical solution to a problem its programmer is facing. This solution should be published very soon, even unfinished, in order to get feedback from other users too (Raymond 2000a). This was contradictory to Brooks (1995: 116) view that pilots or beta versions should not be given to customers. After being seen by “many eyeballs”, bugs and problems vanish more easily (Raymond 2000b). Short development cycles, redesigning and rewriting when necessary create simple and elegant programming solutions (Raymond 2000a).

Linus Torvalds, the lead developer of Linux kernel, is one of the best known persons in Free software scene. Despite affecting Free software development model so much, he has not written any longer guides on how Free software should be developed. His messages on kernel-devel mailing list and newsgroups are often quoted, and sometimes he gives press interviews, but these mostly comment on some single issue related to Free software.

Weill and Broadbent (1998) describe a framework for managing information technology

portfolio by classifying IT investments into four categories. Infrastructure makes IT investments in other areas work quicker and cheaper, transactional creates instant savings, informational improves quality and strategic creates competitive advantage. They provide detailed instruction on each category and analyze the risks and possible benefits of different kinds of investments. (Weill & Broadbent 1998) The choice of Free vs proprietary software and between operating systems could fit into their infrastructure category. Choice of operating system affects a long time, and requires obtaining platform related skills through education and hiring, and dictates other software choices to a large extent. Specific pieces of software (other than operating system) could be put into other categories.

Linux is continuously inspiring researchers and historians. Several histories of Linux and Free software (Bretthauer 2002) exist. The city of Turku has run a research project on using Linux in the public sector (Onnela 2003), including three theses. There exists theory on choosing information technology products for companies.

Even though Free software development model has been of interest to researchers, less research is done about using Free software for purposes other than developing it further. On one hand, nearly all companies using information technology face the problems of software development - at least the choice between making or buying software. On the other hand, there must be some other goal for a company for adopting Free software than developing it for fun. Some research on Linux in a workstation has implicitly stretched the concept of Free licensing by considering and even recommending distributions that are practically not usable without non-free parts. This is strange, because one of the main reasons of moving to Linux (or any other Free platform) is getting rid of software licensing costs. What should an organization practically do to eliminate software licensing costs altogether? Are there other benefits for using Free licenses?

1.2 Goals and Research Problem

The purpose of this paper is to examine the choice of Free software for workstations in a medium or large organization. Because the size of the organization creates requirements for administering workstations, special attention is paid to find a unified combination of software and administrative tools. Number of workstations and the number of people involved can easily make platform choice an infrastructural investment, so methods for evaluating vendor reliability, release cycle and a steady stream of future updates is considered. The findings are mostly useful for small networks too, even though just a couple of workstations would not need the level of automation and standardization discussed.

The research problem is thus

- What are the benefits of Free software in workstations and how can these benefits be realized?

The problem can be further divided to sub problems

- What is Free software, and what benefits does it offer
- How an organization chooses a combination of Free software to be rolled out to workstation

Servers are out of scope of this paper. Where servers are a requirement for basic workstation tasks, they are mentioned from the workstation point of view. It is the writers belief and experience from administering and teaching on both Windows and Linux platforms that Windows has little or no place in server room.

A case example of a 1000+ workstation organization running only Windows workstations is discussed to bring proposed ideas and theory to practice. If possible, a working demonstration of all related pieces of software is created. Plans for roll out, education and internal marketing and cost comparison in case organization are considered out of scope and left for future research. The case organization in this work is Helia, Helsinki Business Polytechnic. Helia has about 1500 workstations and 6000 active users. This paper (mostly an earlier 2004 release) forms the basis for future research on open source and free software in Helia. This work is part of “Avoin Helia Helia” / OpenHelia, a project to find out possibilities, problems

and benefits of moving workstations to open source software. There will be other projects in OpenHelia (outside this paper) such as testing the security of this software package, testing and choosing an office suite and analyzing the needs of legacy software. Possible costs and savings are analyzed to the degree possible in a work that is published. Some areas, such as research on existing systems in Helia, are intentionally very brief. As this is the first part of a larger project, I attempt to point out interesting starting points for future research. Server related questions are not very interesting in Helia, as Helia has Linux servers and Free software in production, and IT center has already obtained practical experience about Linux on the server side.

1.3 Method and Structure of the Thesis

The method used in this paper is constructive case research. The goal of realizing the benefits of Free software is approached by planning and building a workstation from Free software, including Free alternatives for most tools used in case organization. The size of the case organization is taken into account by also considering the demands of administering a large network of workstations, but massive roll-out is not tested in practice. Instead, workstation and installation tools are tested in small scale.

Chapter two of this paper attempts to define open source and free software and review the licenses implied by these definitions. If the results are clear enough, a recommendation for most suitable licenses or criteria for choosing a license will be provided. In chapter three lays out criteria for choosing software and lists the actual software to create a basic workstation. It will result in a list of programs and requirements for a suitable workstation. If possible, to simplify infrastructure, there will be only one basic workstation package for most users. Support tasks of installation, updating and administration are also considered, and a package of suitable software is listed. Issues affecting future costs are briefly discussed in chapter five. Finally, in chapter six, the above recommendations are compiled into a single recommendation. Main areas for future research are pointed out. If some parts of recommendations are not obvious, some secondary choices are pointed out too.

1.4 Principle Findings

Some criteria for selecting a combination of Free software to organizations workstations was pointed out. A sample combination of distributions, software and administrative tools matching this criteria were listed. Using this combination, a working sample meta distribution was produced and tested outside production environment. A strong second choice candidate was pointed out. Features and costs were briefly compared in a case example. It became apparent from this research that it is possible to completely eliminate licensing costs of workstations. Roll out, education and day to day administration cost comparison were discussed, but left for further research.

2 Licenses and the Definition of Free Software

Based on the level of limitations or fees on use, study and distribution of intellectual property, licenses are often categorized as

- proprietary
- free (beer)
- Free (speech)

A product that is "free as in beer" does not cost anything to use for some purpose, but may contain restrictions, for example on distribution or modification. Proprietary means that use is strictly limited by license and usually there are fees on use. Free as in speech, or "Free" with capital F, means that the license meets the strict freedom criteria as defined by various free software organizations.

For example, license of Microsoft Word is proprietary, license of Adobe Acrobat Reader is free (as in beer) and the license of Linux is Free (as in speech). The somewhat funny English terms Free speech and free beer are to make distinction between no cost software and Free software movement. Those words are also the ones used by the Free Software Foundation.

Free Software Foundation, the organization behind the most popular Free software license (GPL), defines Free Software as software that user can

- Run for any purpose (without costs or limitation)
- Study (requires source code)
- Distribute (and keep the money)
- Modify (and distribute modified)

(Free Software Foundation 2003)

Open Source Initiative, whose criteria is often implicitly seen as the definition of Free by categorizing licenses to OSI approved, gives a ten point criteria for license evaluation (Open Source Initiative 2005). In content, it is very similar to Free Software Foundations criteria.

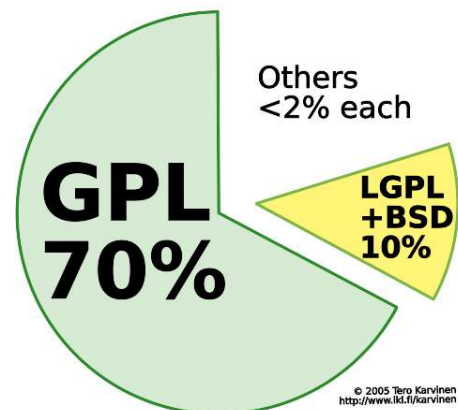
Much of what Helia produces and uses is not software. Teachers create materials for courses, students write reports of solved problems. Books and materials made elsewhere are used by courses.

Many free licenses are made for software, and even though they can be used for other intellectual property, their text can become confusing. For example, what is the source code of book, mentioned by GPL and other licenses. Because of this confusion, many non-software licenses have been created lately. Most popular Free non-software licenses are GNU Free documentation license (FDL) and Creative Commons licenses. Creative Commons, in addition to FDL, is now suggested by the Free Software Foundation (Free Software Foundation 2003).

2.1 Popularity of Licenses

Popularity of software licenses can be compared by looking at the total number the license is used in any project, or we can concentrate just on the most important and most popular projects.

An important Free software directory freshmeat.net keeps statistics of licenses used. The most used license is the GNU General Public License (GPL), which is more used than all the other licenses together. GPL is used by 70% of projects. All ten most popular projects in freshmeat.net use the GPL. (OSDN 2003)



GNU Lesser General Public License (LGPL) and the BSD license are next in popularity. Almost 10% of

Illustration 1. License distribution of open source projects according to freshmeat.net.

projects use either LGPL or the BSD license. (OSDN 2003) These licenses are similar in spirit so that they grant freedoms much like GPL, but don't put many restrictions to protect those freedoms. For example, you can take BSD licensed source code, put it in your proprietary, non-free product and never release the source. (FreeBSD Team 1994-2004) Free Software Foundation (gnu.org), the organization behind LGPL suggests this license for some software libraries. BSD license is used by a whole unix-like operating system, BSD.

LGPL and (clarified) BSD license are compatible with GPL. This means that you can take an LGPL program and include it in your fully GPL licensed work. This reduces license complexity even further. As the list made by GNU shows, many popular open source licenses are compatible with GPL.

Other licenses, such as the Artistic license or Mozilla Public license, have less than 2% share each. In addition to this, some licenses are either GPL-compatible or author as dual licensed the work with GPL. Thus, these other licenses have very little significance. (Illustration 1)

2.2 Freedom vs Protection of Rights

In my view, the most important criteria for choosing a license are freedoms as a user and protection for intellectual property. If we accept the basic assumption of free market, greedy consumers, any user would probably have a program for no cost, with source code and unlimited redistribution rights. When distributing the software, the user would become a vendor. Many vendors would like to limit the use of software to get maximal profits for selling licenses to use that software.

In the illustration, I have categorized proprietary and some of the most common Free software licenses according to this criteria. Typically, proprietary software licenses limit user rights as much as law permits and even more – often any resale is prohibited and vendor could deny use at any time without reason. On the other hand, BSD (Berkley System Distribution License) and LGPL (Lesser GPL) allow unlimited use with almost no

restrictions. Contrary to a typical initial impression, software licensed this way does not usually end up free. Instead, someone exercises the rights given by the license and redistributes the software under his own, more restrictive license, making his version proprietary. Apple OSX is a good example of this, as it is a highly popular proprietary operating system and a distribution based on BSD software. GNU General Public License (GPL) grants users many rights, but states rules to protect those rights. It states that when redistributing software under the GPL, all recipients must be given the same GPL rights that redistributor got when he received a copy of the software. (Illustration 2)

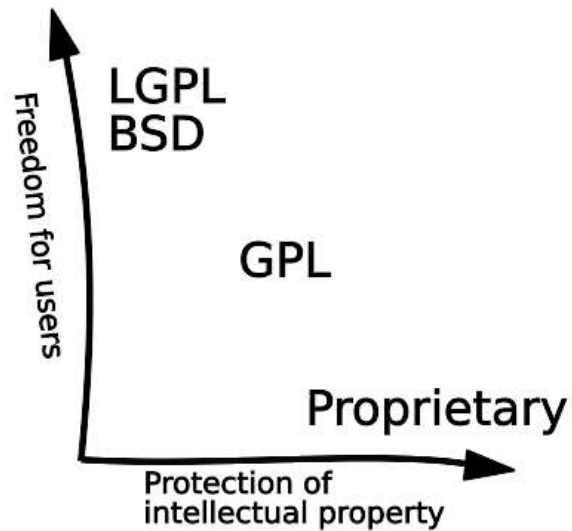


Illustration 2. Freedom for users versus protection of intellectual property.

2.3 Licensing risks

Using a licensed product is a legal agreement and not without risk. Risks common to Free and non-free licenses include trademarks, license conflicts and patents.

Trademarks can limit what you can call a product. With some licenses, you can have the program and source free, but if you distribute it, you must come up with another name.

Examples of trademark protected Free software include the most popular database management system MySQL and the most popular Linux distribution Red Hat (Freshmeat 2003, MySQL AB 2003, Red Hat Inc 2003). It is not fully clear if this kind of limitations to GPL licensed software are valid. Theoretically, this could be changed by Free Software Foundation (gnu.org) for GPL licensed products. The GPL usage notice allows use under GPL version 2 or any later version (Free Software Foundation 2003). In practice, most organizations will likely want to be on the safe sides to enjoy all public relations benefits of Free software, and because removing trademarks from software is often quite straightforward.

Releasing other's property could result in the products license to be invalid. For example, if I sold you the right to broadcast next summer Olympics, you still would not have the rights, because I can not sell something I don't own. At the time of writing, a failed Linux distributor SCO claims to own parts of Linux kernel. As they have failed to provide any proofs, many consider their claims false but harmful to Linux image. (Reuters 2003, Taylor 2003) BSD, a free operating system similar to Linux, won a similar case with a symbolic settlement in 1995 (McKusick 1999).

License conflict is just another form of somebody releasing code he does not own. Licenses often put some restrictions on what you can do with the software. For example, most licenses demand you don't claim credits for other's work. Parts of software with this restriction cannot be mixed with software whose license allows removal of credits. GPL requires modified software to be released with the same Free GPL license as the original. Because one cannot just take GPL code and put restrictions on it, Microsoft has called GPL a "viral license" and "cancer" (Microsoft 2004, BBC 2003, ZDNet UK 2005). In my opinion, the restrictions provided by GPL benefit Helia - when we develop and publish new features for software, we expect other organizations using our modifications give their improvements back to us.

Software patents are allowed in many countries, such as USA and most parts of Europe. In Finland, patenting a mathematical method (software) is not allowed by law, but is beginning to be possible because of the surprising new policies in National Board of Patents and

Registration of Finland (Electronic Frontier Finland 2003, National Board of Patents and Registration of Finland 2003). Patenting software has led to many ridiculous cases in USA and other countries. Examples include a patent for electronic commerce (Amazon 1997) and a patent for hyperlink (Sargent 1980). Also many formats are patented, such as GIF (images) and mp3 (music). Usually, a better patent-free format exists, in this case PNG for images and OGG for music.

2.3.1 Risks of Proprietary Licenses

Proprietary licenses, such as the Microsoft End User License (MS EULA), are created to by vendor, to protect vendors interest. Proprietary licenses can contain high risks for client companies. Vendors of proprietary software often aim for customer lock-in, so that even if customer is not satisfied with service or product quality, changing has become too expensive. (McHugh 1999)

The big risk is one-sided change of conditions whenever vendor wants. This is achieved through a combination of software features and licensing terms. License usually allows vendor to change conditions of a license, or terminate users right to use the program. A more used method of forcing users to accept one-sided license change is to drop support for older operating system version and have a new license for the new version. As the code is proprietary, there are no third party vendors to provide critical security patches, which makes it impossible to use the program in business environment. This kind of forced change was used to get people over to Windows XP, whose license has changed a lot from previous versions of Windows. Reverse engineering is often not allowed by the license, but this kind of limitation does not necessarily apply to Finland. Sometimes licenses contain impossible terms to reduce vendor responsibility.

Licenses are not just a cost by itself. To follow license agreements with vendor, IT support must be continuously counting licenses. To avoid buying unused licenses, IT support is forced

to create many configurations for workstations. When workstations are managed by imaging (as in Helia), this creates a lot of unneeded manual work.

Multiple license agreements create a hard to manage legal portfolio. In my personal experience consulting companies, even large firms often solve this problem by forgetting it. Little effort is made to read agreements, maybe because client companies feel they could not choose vendors or negotiate the terms anyway. Many proprietary software licenses are very long, and purposefully made hard to read. Some software vendors go as far as using a tiny font for printing the agreement and using a matchbox sized window to display a ten-page agreement on computer screen.

Many of the problems of proprietary licenses don't exist with Free licenses. A lot of Free software uses compatible licenses, and most of the software use the same license, namely the GPL. This reduces costs of managing the legal portfolio. Not only there are less licenses for company lawyers to be familiar with, but also there is an ongoing public review of the widely used licenses, especially the GPL. This results in less costs and fewer risks.

Free software gives huge benefits to users of software. What is their downside? More equal stand to negotiate price and service terms between customers and vendors makes it very hard to have artificially high profits based on customer lock-in or dominating position on the market. What some vendors may find threatening, most customers and users find beneficial.

2.4 Recommended licenses

As we have laid out the desirable qualities of licenses in previous chapters, coming up with a recommendation should be easy.

Following the recommendations of the Free Software Foundation, we can achieve license

popularity, backing of a huge community and throughout testing of licenses. Free Software Foundation recommended licenses are

- GNU General Public License (GPL) for software
- GNU Free Documentation License (FDL) for guides and other texts
- Lesser GPL for software libraries, which are rarely produced on Helia
- GNU recommends Creative Commons license for non-software intellectual property, when FDL is not suitable. Creative Commons license has been used with music.

(Free Software Foundation 2003).

Helia should use GNU General Public License (GPL) as preferred license for choosing and publishing software, and FDL or GPL for publishing course materials and student works. For other works, Helia should follow Free Software Foundation's recommendation for licensing. The choice of a license should be volunteer for each student and teacher.

3 Software and Operating System for Workstations

Linux is usually installed from a distribution, so that everything a working system needs comes from a single DVD. An installed distribution contains applications, such as word processors and web browsers, and all that is needed under the hood from them to work such as operating system and drivers (Illustration 3). In this chapter we'll examine the choice of distributions and applications for common purposes.

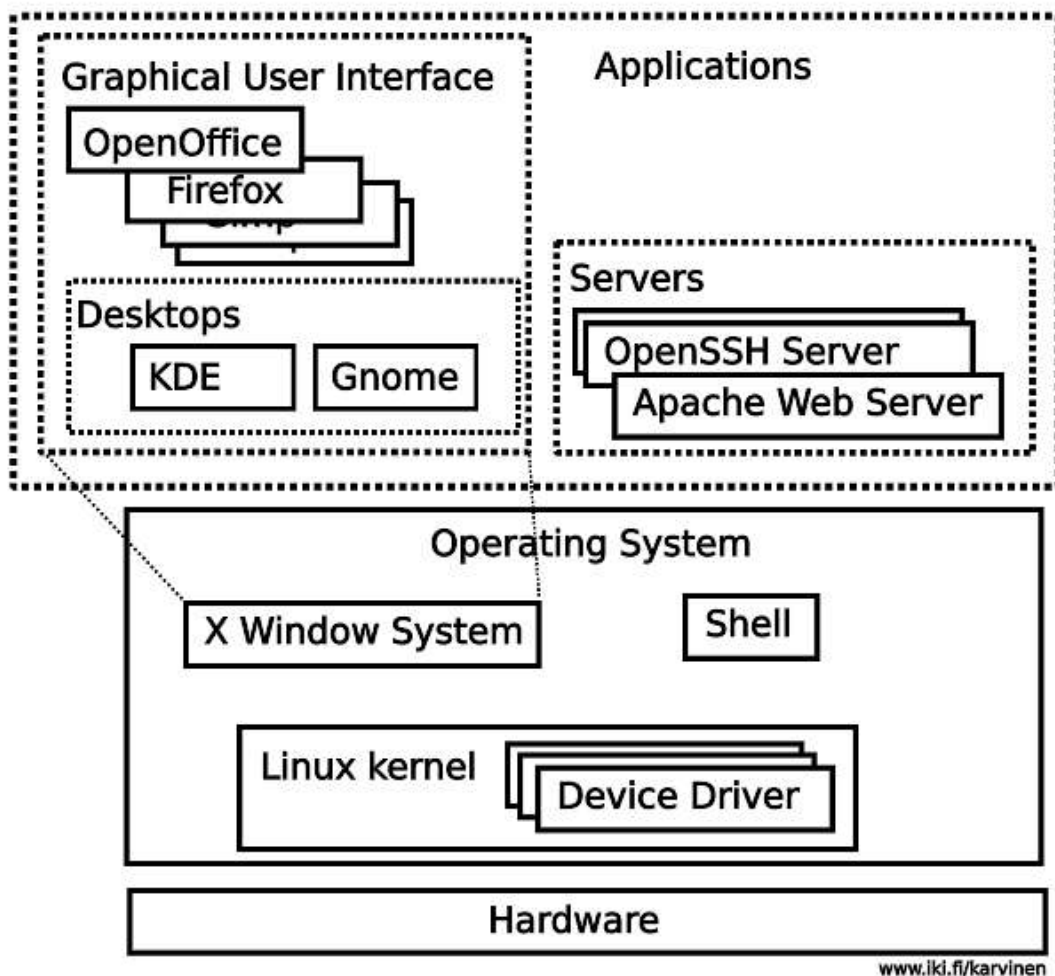


Illustration 3. A working Linux system consists of an operating system and applications.

3.1 Distributions

Linux distribution is an installable Linux system. A distribution usually contains the Linux kernel and operating system, an installer and software. There are more than a hundred Linux distributions being actively developed. Because of the Free license, anyone can create a new distribution and publish it. However, creating and maintaining a quality distribution is a huge effort. (Wikipedia 2003)

We are looking for a distribution for workstations in Helia. If the same distribution is suitable for servers, we could consider it a benefit. Thus, we can exclude all embedded, floppy, router and other special distributions. As Helia is running almost exclusively on "normal PC" Intel x86 architecture computers, we can exclude distributions that do not work on Intel architecture, such as those made for Alpha-processors and Mackintoshes. Most, maybe all students of Helia speak Finnish or English, so we can exclude distributions aimed for other language groups. To reach all the benefits of open source (peer review, fast development, lot of contributed software), we want to select a distribution that is both popular and has an open development and distribution policy. As we aim to save licensing costs, there would be no point selecting a distribution that required license fees.

Selecting a distribution is a strategic decision, as it is a long term economically significant decision that affects many daily choices. Because of the level of commitment made here, it is important to select a distribution that is both well supported and expected to stand the test of time. On the other hand, despite a different logo, all distributions are just Linux in another package. They can be made to run the same software, and produce the same document formats. Open Source programs are completely documented by their source code (by definition), which prevents lock-in caused by secret protocols and file formats.

Similarities are unimportant to comparison, so to lay down a selection criteria we must concentrate to what is different in the distributions. When we have found the best distribution for Helia, decided possible modifications, settings and additional software, we can then compare it to the existing closed source (Windows based) solution. Based on experience in teaching and administering most of the popular distributions, the main differences are easily pointed out.

On the technical side, distributions differ in software installation tools and its back end, package format. Different software is installed by default and provided by vendor. Availability of contributed software differs. Technical stability, the ability to run long and under load without crashes and errors, is highly different. In addition to design and quality

assurance, stability is greatly affected by how new the software is – the latest and greatest "bleeding edge" software is less stable than older, more tested software. Hardware support differs, even though all distributions run the same kernel. Most distributions only run on intel architecture (i386, "normal PC"), but for example Debian supports 11 processor architectures. On device support, newer distributions usually support newer peripheral devices, and distributions with more loose licensing policy sometimes support devices that don't have Free drivers available.

Immaterial differences in distributions are at least as important as technical ones. Even smaller users can immediately benefit from popularity, especially if the larger installed base is also active. Popularity usually results in more Free support material and contributed software being available. Openness of development makes it easier to forecast future changes in distribution, and also help integrating own changes upstream to distribution. Because platform choice, such as distribution, is a strategical one for organization, the software should be available five years from now too. In practice, this is affected by vendor reliability, distribution history and popularity. Distribution image should be good to help harvesting all the image benefits available to those who move to Free software. As the goal of moving to Linux is usually to have all the benefits of Free software, the licenses used by the distribution are highly important. Even though all distributions are mostly compiled of Free software, some distributions are not really useful without non-free parts. Thus, distributions that avoid non-free parts make it a lot easier to get all the benefits of Free software. For companies, the availability of third party support is at least said to be very important, even though evaluation of the value of third party support should be based on how much it has really be used before. Previous experiences in organization can immediately save money by reducing the need for education, but it is also likely to reduce resistance to change when rolling out new distribution.

Currently, Helia is using Red Hat in most of its servers, including myy.helia.fi. It handles shell accounts, some of Helia's email and web access to email with IMP. Helia's proxy server is also running Linux. Helia's two Linux courses (which I am responsible of) are using mostly

Red Hat, even though one of them used Debian in the past. Slackware is used on some test servers (Pakkanen 2003). Knoppix, a live-cd Debian derivative, has been used in some demonstrations.

There is no way to exactly tell which distribution is the most popular, because there is no way of even counting how many Linux machines exist (Linux Counter Project 2003). However, there are many estimates on distribution popularity, based on page load statistics and user registrations. The most popular Linux distributions (in descending order) are Mandrake, Yoper, Red Hat, Gentoo, Debian, Knoppix, SuSe, Slackware, Lycoris, Morphix (Distrowatch 2003) Red Hat, Debian, Mandrake, SuSe, Slackware, Gentoo, Conectiva (Linux Counter Project) Linux Counter Project's list is much nearer of my assumptions. Yoper's ranking is artificially high because Distrowatch ran a huge advertisement for it before and during the time this paper was written, so it will be excluded. Yoper is also in too early development phase for production use.

To sort the list, we can divide the distributions to Red Hat based (rpm) and Debian based (deb) distributions. Gentoo (emerge) and Slackware (.tar.gz based) are independent distributions (Illustration 4).

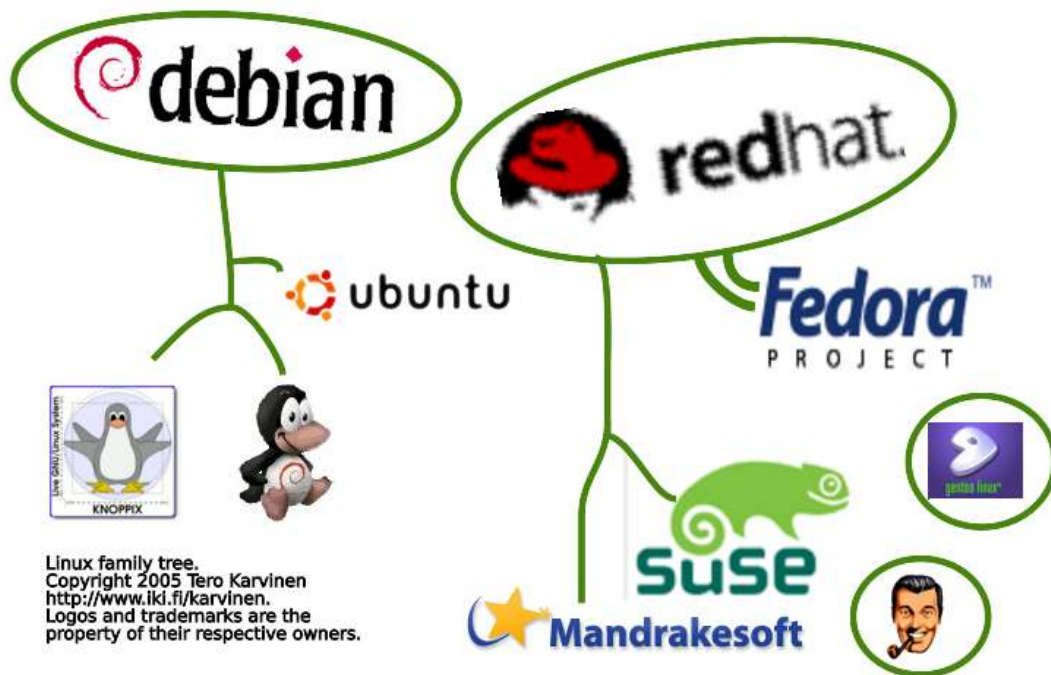


Illustration 4. Family tree of some Linux distributions. Distributions with their own package management system have been circled.

3.1.1 Gentoo and Slackware

Gentoo and Slackware are aimed at advanced, single users (Gentoo FAQ). They are both made to be easily customized and optimized. Gentoo requires compiling the entire system from source code, even though great tools for this are provided. Gentoo lacks software to manage updates of several workstations, and Slackware's tools for updates (and tar.gz package format) are very limited. Helia is using Slackware on a course about routing (Pakkanen 2003), but the teacher is considering moving to Red Hat. As both the implementation and development goals are very different to Helia's need for stability, ease of use, popularity, minimization of support need and easy updates, Gentoo and Slackware are not suitable for Helia. Logo for Gentoo and the de-facto logo for Slackware are shown in illustration.

3.1.2 Debian

Debian has been one of the most popular distributions for years. Main goals for Debian are stability, quality, Free licenses in all software and easy updates. Except for finalizing of software installation packages, it meets these goals quite well. It is developed not by a single company, but by its users around the world, mostly Europe and the USA. Many commercial companies have chosen Debian as the bases for their distribution. I have taught one course using Debian 3.0 and been one of the administrators of a commercial server running an older version of Debian. Debian has huge benefits: versions are supported for a very long time, update system is great and the system is stable. Version support means that security updates for a version published years ago are still available, and updating from major version to another can be done with standard update tools. Neither Red Hat nor Microsoft Windows support for old versions come even close to Debian. Old versions end up getting a lot of real life testing, and are thus very secure and stable. Backwards incompatible changes in system foundations are rare. Debian update system, apt-get, enables updating operating system, vendor released and contributed software with a single command. Apt-get has been an ported to other distributions, and has been an example to other update systems.

The worst downsides of Debian are lousy packages, difficult installation and hardware configuration. Even though most Debian users consider the package format an advantage, it has many problems. Interactive installation, where the installer asks questions from user, is annoying when each of the three hundred programs to be installed wants to ask two questions. Also, packages don't contain all the settings needed to get things running. For example, installing PHP dynamic scripting to Apache web server requires reading manuals and adding lines to web server configuration files. Manual tweaking is error prone, and may lower security. Installation is widely agreed to be the weakest part of Debian. Easier installation for this great distribution has created a need answered by more commercial Debian derivatives. None of these commercial derivatives is currently very popular. Installer asks questions that are very hard to answer, such as selecting suitable kernel modules from hundreds of possibilities. This question is also unnecessary, because Debian has a great hardware detection system called discover. If you are not careful, system is installed in an insecure way with firewall open, just like Microsoft Windows. Official Debian manuals are not of very high

quality, but a lot of other good support is available for free. Despite some downsides, Debian has many unique benefits and qualifies for further testing. (Karvinen 2003)

3.1.3 Knoppix and derivatives

Knoppix is a live-CD - you just boot from the single cdrom, and two minutes later you have a graphical desktop, with network and other hardware working. Pre-installed software includes OpenOffice, kOffice, web browsers, games, c++ compiler, Java, Flash... Knoppix has the best automated hardware detection of any platform, architecture or operating system that I have ever seen. Knoppix and its derivatives are probably the best live-CDs at the moment. It has worked in every Intel-based computer I have tried, including an ancient 166 MHz Pentium, five years old laptop and a state of the art desktop workstation. As it does not install anything to a hard disk, it works well as a Linux demo or a repair disk. Morphix is a more easily modifiable Knoppix derivative, but in my experience it is not as polished. Forensic Incident Response Environment (F.I.R.E) is a more security and repair oriented live-CD, which has a great deal of security related tools, but showed out to be completely unsuitable for normal workstation use. While Knoppix is more a demo CD than a workstation distribution, it could be useful in Helia as Installation tool as a Debian Demo CD to be given in presentations, as backup system for computer crashes, as a repair tool, for hardware support testing when buying new hardware and an example for own modifications for another distribution.

Turning a live-CD like Knoppix into a workstation distribution installer was tested during this project. Many technical choices in Knoppix are not suitable for normal workstation, such as running X Window System for single user only. The standard installer removed most hardware detection, which makes it pointless to duplicate installed Knoppix images.

Remastering the installer to contain new software required using alpha and beta quality tools that were badly documented and required tweaking a host system to work. Currently, turning Knoppix into a Debian installer is currently too big effort, but outside efforts in this direction should be followed. This kind of advances are likely reported on Knoppix community website knoppix.net (2003).

3.1.4 Red Hat / Fedora Core

Red Hat has pioneered ease of Linux installation and hardware detection. It has steadily gained market share as an easily installed company server, but with Red Hat 9 and Bluecurve desktop theme it officially started competing as a company desktop. It is estimated to be the worlds most popular Linux distribution (Linux Counter Project 2003) and the most popular distribution in USA and Finland. Its installer is very similar to MS Windows installer, and in every course I have given it's user friendliness has surprised students. Being the most popular means wide availability of third party software, support by commercial software and hardware vendors (such as IBM and Sun) and availability of commercial support. Companies that I have taught Linux preferred some version of Red Hat.

Lately there have been efforts to bring the many benefits of Debian into Red Hat. The great apt-get update and installation tool has been ported to Red Hat and other Red Hat Package Manger, RPM based distributions. Free software repositories for automatic install and update have emerged, including freshrpms.net (the first big and popular one), my own (<http://iki.fi/karvinen/apt>) and fedora.us (new, but has big developer base). Fedora attempts to bring Debian like policies, development status definitions and quality assurances to contributed RPM packages.

Downsides of Red Hat Linux are the just the other side of its benefits. Being up to date with the latest and greatest software means shorter testing periods for stability and security. Being backed up by a profitable public company means there has to be a more closed part that costs something. Red Hat has its own non-free update system, up2date, but luckily there are better Free alternative updaters (apt, yum). Being the leading distribution, Red Hat has kept up surprisingly high ethics. Sometimes it tries to use its position to push open standards to replace older, proprietary ones. Despite a good purpose, moving to Unicode utf-8 charset caused many compatibility problems, as did dropping default mp3 support in favor of technically more advanced Ogg Vorbis. Despite these minor annoyances, Red Hat is definitely a strong candidate for a distribution of choice.

During the writing of this paper, in the late 2003, Red Hat published a Linux distribution called Fedora Core, and discontinued free Red Hat Linux distribution. In practice, Fedora Core is just like a newer, improved version of Red Hat. Some third party software to be added by this project, such as yum (Yellowdog Updater, modified) were officially added to the distribution. The trademark policy was made more liberal and more clear. It seems that this change of name and project organization is beneficial, in case Red Hat / Fedora Core becomes the distribution of choice.

3.1.5 Mandrake

Mandrake is a French Red Hat based distribution. It has wide user base (Linux Counter Project 2003, Distrowatch 2003) in Europe. It aims to be an easy home desktop distribution. The company developing Mandrake Linux has had serious financial problems, and as I forecasted in a Linux course on 2002, is now near to bankruptcy. Even though the Free license makes it possible for Mandrake Linuxes development to continue even after Mandrake software, there are more reliable options available. Mandrake has tried to profile as "the Macintosh of Linux", the easy system with funny animations. This means that it may have too few advanced users so that it's development could continue without MandrakeSoft. I tested Mandrake in 2001-2002. At the time, it was very similar to Red Hat, with all the changes just glued on. This has partially changed during 2002-2003. It promotes graphical tools for managing system settings, which may seem nice at first, but quickly becomes annoying for an advanced user. Mandrake is very well featured, for example, it is already (in 2004) running the latest 2.6 kernel, and supports easy loop encryption of hard disks. Keeping up with the latest features (in the expense of stability) seems to be a new direction for Mandrake. Many published versions of Mandrake are only available for cost, or their downloading has been made somehow difficult. Mandrake has very little more to offer than Red Hat, and it's future seems uncertain.

3.1.6 SuSe

SuSe is German based Linux. It is the most popular Linux in Central Europe. The German government is very involved with Linux and its support to Linux projects may prove valuable in the future. SuSe has had many advanced features before Red Hat, such as decent font smoothing (xft) and support for some hardware. SuSe is not very open. Cd-rom images ready for burning are not available from SuSe homepage, even though they do provide free installation through network. Official SuSe distribution is bundled with non-free software. Even though Helia has enough IT resources so it could build a SuSe installer of its own, closedness may reduce the benefits of free software image, reduce free support from advanced users and limit Helia's possibilities for tailoring software. It has been tested in a minor scale in Helia IT services along with Red Hat and Mandrake.

3.1.7 SOT Linux

SOT Linux is a Finnish Red Hat derivative. Its share of the world market is minimal, versions are often older than official Red Hat releases, and development is quite closed. Website does not provide cd-rom iso images. System installation will be done in English, so SOT's specific benefits do not match with Helia's needs. SOT is thus excluded from further testing.

3.1.8 Recommended Distribution

Red Hat / Fedora Core has the best popularity among companies near Helia, IT staff has prior experience on Red Hat, it has a suitable combination of programs and their development model is very open. Fedora Core is Free, except for the trademark, which can be removed if Helia wishes to redistribute the distribution. Red Hat / Fedora Core has the best administrative tools of all distributions. Helia should use Red Hat / Fedora Core as its distribution.

3.2 Software needs

Software is not an end to itself, but rather a means to an end. Because of this, it was suggested in a FreeHelia meeting that there should be a committee defining needs for software. From personal experience consulting many computer projects I know that defining abstract needs in a committee with people from many organizational levels and business units ends up being expensive and time consuming. If this kind of committee would be necessary, it should be founded to research software needs for current closed source, Microsoft based solution too. Workstations used by hundreds of persons for their daily chores is not a place for wild experimentation. Because of this, the software selection tends to be on the safe side. Most software on a typical workstation contains programs that have become the cost of doing business in any school or firm, like web browser or office suite.

3.2.1 Course Experiences

In six courses teaching Linux, I asked what programs (or program types) the class considered most important. Those same programs were then the ones taught in the class. Two of these classes were for external companies, four were normal courses for Helia's students. In addition to this, 15 persons were shortly interviewed in very free form around the question “what software is most important for your daily work”.

It was found out that most users expect certain basic software from a workstation, most important being a word processor. Everyone implied that a workstation has a window manager. Many also implied a browser and considered it very important when asked. The most important programs in descending order

- Window manager
- A web browser
- Word processor
- Spreadsheet

- Winha, a proprietary tailored course information software
- Presentation slides (such as Impress or PowerPoint)
- Programmers editors and an IDE

Many random programs were mentioned as being important, but not related directly to work, such as music player, video (dvd, divX...) player, image manipulation program, instant messaging program.

3.2.2 Currently Installed Software and Availability of Free Alternatives

Unlike the tedious task of finding hidden or unmet software needs, list of filled software needs is rather easy to find. The list of currently installed software provides a practical view of filled software needs. Currently installed software as listed by Helia IT services (2003) and observed by writer is listed in detail in appendix "Currently Installed Software in Helia".

Even though many programs differ on Linux and Windows, it is usually easy to find a program that performs the same function. I have added examples of Linux alternatives to programs based on my experience as a Linux user, administrator and teacher.

Linux distributions usually have all drivers with them. Exotic hardware can be difficult to find. Red Hat, Debian and Knoppix have been tested in many workstations in Helia and hardware support has been good. This text is partly written on Helia's laptop running Linux.

Using Windows file sharing (Server Message Block, SMB) from Linux requires manual tweaking, but this only needs to be done once for the network. Linux has many native file sharing systems too, such as Network File System (NFS). Linux also has filesystems that implement more advanced concepts in the areas of encryption, key exchange, disconnected operation and scalability, such as FUSE based systems, AFS and Intermezzo. Due to maturity, easier installation and bigger installed base, NFS and SMB remain more popular than those advanced systems. (Karvinen 2003).

Web browser and document processor are likely the most important applications in a workstation. Web browser is probably the most important piece of application software on a modern workstation. After publishing an earlier version of this paper, Helia installed Free Mozilla browser in many workstations. Both Mozilla Suite and Mozilla Firefox installed by default in many distributions. Web browsers are discussed in detail later in this paper. Office software includes document processor, spreadsheet and slides based presentation programs. Like most organizations, Helia uses Microsoft Office package. Free office suites on Linux are discussed in detail later in this paper.

Multimedia players play music and videos in many file formats, such as mp3, dvd, real, quicktime, flash and shockwave. Currently, workstations have Windows Media Player, Flash player, Winamp, and the notorious Realplayer installed. Mplayer can play most of these formats, even Realmedia and Windows media proprietary formats if the required codecs are installed. However, it is not clear if those extra codecs have licensing issues sorted out. Other generic media players for Linux include Xine and VideoLan. Proprietary versions of Flash and Realplayer are available. Realplayer has had allegations of violating its users privacy (See for example Gibson 2003), so its exclusion from the distribution should be considered.

Utilities for compression, viewing some document formats, remote logins, file transfer and advanced text editing are used, and well supported in Linux. During the course of writing this paper, Helia IT Center started to officially support multiple Free utilities on Windows platform as recommended in an earlier version of this paper. For example, Helia now uses free software for secure remote control and file transfer.

Web pages are currently created with Microsoft Frontpage and text editors. Frontpage is quite inadequate for its purpose, as it usually fails to follow any standards when saving documents. Some workstations have Adobe PageMaker desktop publishing software installed. Free desktop publishing software Scribus exists, but is not yet as good as PageMaker. Some

workstations have Photoshop image manipulation program. GIMP, the GNU image manipulation program is a Free alternative for that.

Miscellaneous software for education and legacy purposes is installed. Most users read their email using web mail client IMP. However, some teachers still use Tiimiposti. Databases are used in teaching. Some courses use databases, such as Solid. Database support in Linux excellent, and MySQL databases are already used in Helia (Karvinen 2005). There are various educational programs, such as typing tutors and dictionaries. Some of the must be run under emulation, but for some, a Free alternative exist. Some courses use Rational Rose for drawing UML models. Free Dia could be used instead, and during the writing of this paper some courses started to use Dia. Software for network analyses, such as network sniffers, have been already changed to Free alternatives. SAP has for cost Linux client, which could be used instead of Windows client. Virus protection programs are not needed in Linux, because more generic security methods to protect from virii (Bartolich 2002). Course management is done with legacy software Winha. Current version of Winha is very difficult to run in emulation on Linux. Winha will likely have a web interface soon, so that it can be used from Free workstations.

Workstations management is done by installing images with Ghost, remote controlling by desktop sharing with a VNC based proprietary program, and updates with a logon script. Workstations management could be improved a lot when moving to a Free solution, as described in chapter "Managing workstations".

To give a compact view of how a Free workstation could fill the software needs that Helia has currently filled, the above list of software can be listed by the level of support on Linux. Some software is readily available for free on Linux, but not for Windows: servers, compilers, distributed compilers, network security tools. Such software is not discussed here in detail.

<i>Good Support on Linux</i>	<i>No native support on Linux</i>
<ul style="list-style-type: none"> – Office Suite: Word processor, spreadsheet, slides – Web browser, email – Multimedia players – Device drivers – Ssh and secure file transfer – Compression (zip, tar.gz...) – Text and code editing – Relational databases (but with different software) – Image manipulation – Visual modeling – Workstations management 	<p>Requires emulation, changes in back end systems or search for an alternative</p> <ul style="list-style-type: none"> – Desktop publishing (PageMaker, Scribus) – Winha student and course management, Otso – Language learning <p>Should be dropped</p> <ul style="list-style-type: none"> – TiimiPosti, move users to IMP.

3.3 Office suites

The most popular office suites for Linux are OpenOffice.org and kOffice, OpenOffice.org being the most popular free Office suite in the world. Other office suites include desktop environment Gnome's office suite consisting of Abiword document processor and Gnumeric spreadsheet.

3.3.1 Word processor

In my opinion, a good document processor would let user concentrate on writing content, instead of fiddling with layout. Some editors, especially LaTeX based, share my view. They explicitly force user to handle layout separately from the writing process. Many word processors are filled with useless features, making them complex and slow to use, unstable, slow and expensive to computing resources. However, when teaching people to move over from one office suite to another, the question always seems to be: does it have this or that feature, such as label printing, funny bullets, pie charts with gradients... To buy user satisfaction, this feature addiction must be taken into account when selecting an office suite.

I will first prescreen suitable office candidates with a criteria based on my experiences as a teacher, user and administrator of systems using word processors. Another project will do a more detailed analyses of office suite. Later, these suitability of these office suites is put on review by Helia's office suite teachers and users.

Requirements for Office Suite

- Stable, not crashing or losing data
- Structured: headings h1, h2, h3..., paragraphs p, numbered and bulleted lists with nesting, program code preformatted. Styles should be easy to modify.
- Fast
- Multilingual. Support for charsets to display national characters, such as Finnish åäö. UTF-8 is an international standard for this. Spell checking for Finnish, English, other languages used in Helia are a plus
- Possibility to embed bitmap images
- Save formats are fully documented and standardized, including the native format of the program. Saving as html and plain text.
- Printing
- Selected distribution is an officially supported platform by program developers
- Microsoft Office compatibility, mostly ability to open MS Word doc. This is required for interoperability with other organizations and own legacy systems. In practice, this requires the office suite to have a huge number of features.

Nice to have features would include large user base, so that its worth learning and teaching, ability to format text without styles, possibility to define new structured styles, embedded object support, PDF and XHTML export, document templates and outline view. The license should be free for the program to be part of the Free workstation. As Helia will likely be running at least Linux and Windows in the near future, the program should be multiplatform. Support for BSD and Apple OSX would be nice.

3.3.2 Comparison table

Based on the criteria set above, I will first limit the scope of office suites to be compared, and then compare a few free solutions to currently used office suite, Microsoft Office 2000. The office suites to be tested were chosen by user base and inclusion to major distributions. Especially, the chosen office suite should be officially supported by chosen Linux distribution. At this point, office suite should be part of Red Hat / Fedora Core, or otherwise the selection of distribution should be reconsidered. Despite their quality and popularity among older Linux and unix users, Vi, Emacs and Latex based systems were not considered because of steep learning curve.

The results were obtained by using each office suite for two days to edit a document of 10-50 pages. In addition to this, I have used (an earlier version before this project) these office suites for at least four months daily in my work. The assumed native environment for each office suite was used: Red Hat Linux and KDE for Kword, Red Hat Linux and Gnome for OpenOffice.org Writer and Windows 2000 for MS Word. For some questions, such as multiplatform compatibility, direct testing was not practical within available time. In these questions I have used information provided by program vendors' web site.

	<i>Kword 3.1</i>	<i>OpenOffice.org Writer 1.0.3</i>	<i>MS Word 2000</i>
<i>Stability</i>	Bad (crashed during testing losing data)	Good (no crashes during testing)	Bad (crashed during testing losing data)
<i>Speed</i>	Fast (when running K Desktop Environment. Medium to slow when KDE is not running.)	Slow (very slow start, otherwise fast. Start is much faster in 1.1)	Fast (preloaded, not tested without)

	<i>Kword 3.1</i>	<i>OpenOffice.org Writer 1.0.3</i>	<i>MS Word 2000</i>
<i>Structured</i>	Yes	Yes (but default template does not automatically list styles in drop menu, fix with a document template)	Yes (but sometimes breaks styles, maybe because internal representation does not emphasize structure)
<i>Spelling (English and Finnish)</i>	Yes: English (but spelling system requires manual tweaking for Finnish, sometimes crashed)	Yes: English, Finnish (soikko, non-Free)	Yes (must be bought separately. Hard to get Finnish spelling with English user interface)
<i>International charset</i>	Yes (utf-8)	Yes (utf-8)	Yes (proprietary)
<i>Bitmap images</i>	Yes	Yes (easy layout)	Yes (layout breaks easily when document is edited)
<i>Export text</i>	Yes	Yes	Yes
<i>Save as HTML</i>	Average (valid HTML, but loses markup, such as list depth)	Good.	Bad (“Save as HTML” feature exists, but produces non-valid, broken html that is very hard to clean)
<i>Printing</i>	Yes	Yes	Yes
<i>Linux support in major distributions (Redhat/Fedora, Debian...)</i>	Yes	Yes	No (emulation possible with Wine, but not tested)
<i>Windows support (win32)</i>	No	Yes	Yes

	<i>Kword 3.1</i>	<i>OpenOffice.org Writer 1.0.3</i>	<i>MS Word 2000</i>
<i>Other platforms (not Linux or Windows)</i>	Yes (BSD)	Yes (Versions for: BSD, Solaris, MacOS X)	Yes (Versions for: MacOS 9, MacOS X)
<i>Public and Documented file format</i>	Average (source available)	Good (compressed XML)	Bad (purposefully undocumented and obscure, often has non-backwards compatible changes)
<i>User Base</i>	Small	Large	Huge (Counting all versions together the most popular Office suite in the world)
<i>MS Word Compatibility</i>	Bad	Very good (both save and open)	Good (All previous versions open perfectly)
<i>Embedded object support (charts, drawings, spreadsheets)</i>	Yes (kOffice suite)	Yes (OpenOffice.org suite)	Yes (MS Office suite, some required programs are sold separately)
<i>PDF Creation</i>	Yes (two step, not intuitive)	Yes (buggy, fixed in 1.1)	No (sold separately with Adobe Acrobat. A no-cost option with a custom printer driver might be possible)
<i>Document Templates</i>	Yes	Yes (can use MS Word templates too)	Yes
<i>Free License</i>	Yes (GPL)	Yes (GPL)	No (MS EULA)

	<i>Kword 3.1</i>	<i>OpenOffice.org Writer 1.0.3</i>	<i>MS Word 2000</i>
<i>Price</i>	0 EUR	0 EUR	494 EUR (Office 2000 Pro fi, volume licensing is available and used by Helia)

Interoperability from kWord showed out to be the worst of the three, as it was not easy to exchange data between kWord and any other office suite.

OpenOffice.org interoperability showed out to be amazingly good, as it made very few mistakes and even supported templates. However, it was noticed that in some cases Scandinavian special characters were sometimes broken as a result of conversation. Also, different fonts in Windows and Linux sometimes caused layout to change. Font problems could be fixed by installing freely available fonts from Linux to Windows or vice versa. OpenOffice could be installed to Windows machines too without any licensing costs.

Comparing major word processors, I recommend OpenOffice writer is the primary choice. Kword should be tested further, if K desktop environment becomes the chosen desktop environment.

3.3.3 Spreadsheet

In testing, kSpread showed out to have very limited features compared to OpenOffice.org Calc. The version used in testing had some serious bugs, even though it was announced to be a stable release version.

Gnumeric is light, working fast in slower computers than the other mentioned spreadsheets. The version tested required a lot of basic settings made manually. For example, fonts were not

readable by default. Some of these bugs seem to be corrected in the version of Gnumeric included in Fedora Core 1, published after the tests were done.

OpenOffice.org Calc seems to be an obvious replacement for MS Excel. In the six courses tested, none of the student could point out important features lacking. The only downsides compared to MS Excel that were found out were different macro language and minor differences in user interface and some obscure features. MS Excel Visual Basic Macros are not supported. This protects against viruses, but macros are currently in use in some companies. Some buttons are not on same places (this was seen as a problem by less than five persons). Solver, the linear optimization add-on, is not available in OpenOffice.org Calc.

OpenOffice.org Calc is the recommended spreadsheet application.

3.3.4 Other Office Tools

Having observed people using office suites, I have recognized that the problem of simple maths remains unsolved by office suites. Typically, calculations are made in a spreadsheet application, which is overkill if you want to know how much is $26 * 1,7$ or $\sin(30)$, and not practical if you have to do multiple simple calculations. Calculator utilities with pictures of buttons on them are even slower to use. Would you use a word processor by clicking picture of a keyboard? Usually this problem is solved by having a good old pocket calculator on the table. I have found that calc "C-like arbitrary precision calculator" is good enough to replace a pocket calculator. (Karvinen 2002) Mathomatic also does symbolic maths ($x * x + 3 = y \Leftrightarrow x = ?$), but I have tested it only briefly. One of these lighter calculators should be included in Helia's workstation installation.

3.4 Web Browsers

Currently, Helia is using Microsoft Internet Explorer and Netscape Navigator on Windows

workstations, Linux course uses Mozilla and Galeon. Browser is already one of the most important programs. In addition to browsing information on the web, it can act as a user interface for platform independent web programs. For example, most users read their mail with a web browser.

If Windows and Linux platforms are co-existing in Helia for a long time, using same or similar software on those might reduce support costs. On the other hand, the main reason for supporting multiple operating systems might be to make wide palette of software available. Browsers working in both Windows and Linux are those derived from Mozilla: Mozilla Communicator, Mozilla Firebird and Netscape Navigator. Proprietary Opera Browser is multiplatform too, but the availability of Free browsers makes it very hard to justify Opera's licensing costs.

Most common Linux (graphical user interface) browsers are Mozilla, Firebird, Galeon and Konqueror. Mozilla is the most popular Free browser, but this popularity will probably later transfer to Mozilla Firebird as it reaches 1.0 (stable) version. Mozilla has a lot of extra features, such as web page editor and an email client. Galeon is a browser for Gnome desktop environment. Galeon is highly suitable for advanced users with fast computers, as it has the best tabbed browsing. Konqueror is the default browser in K Desktop Environment. It has the least web related features of these major browsers, but it also includes the most full featured file manager in Linux. Konqueror (KHTML engine) was chosen as the basis for MacOS X Safari browser, because it was said to have the cleanest code base of Free browsers. Also light browsers exist, but they render pages incorrectly and have limited features. For example, links (g-links, not elinks that is included with many distributions) opens instantaneously, and can be run graphically on computers with out X Window System.

Helia should use Mozilla Firebird in its workstations because it is simple, multiplatform and works in multiple desktop environments. The extension system in Firebird makes it possible to choose between features and stability.

3.5 Running Windows Software in Linux

In this chapter, I will look at possible methods for running Windows Software in Linux. Reasons for running Windows software are legacy software, special software and supporting users. Usually, tailor made software that is used widely in whole organization is very hard to change. Even if there were now better alternatives, it would not be cost effective to push a new system through organization. There are some areas of special software where Linux alternatives don't match the best Windows software. For example, Illustrator and Freehand artistic vector drawing applications are still a head of Linux alternatives such as Sodipodi. User support may require trying the same programs that user has, and to avoid requiring two computers, it might be necessary to use software from other operating systems. In practice, it is often just one or two Windows programs that are keeping users from changing. Thus, running Windows software in Linux is important for helping the change to Linux.

In teaching Linux, I have noticed a tendency of users converting from Windows to Linux to look for emulating their favorite programs. This was even more common in 2002, when Linux was not as popular as now. For example, there even exist a product to run Microsoft Office on Linux, CrossOver Office. Looking for native Linux alternatives for programs should always be the priority for choosing programs. Benefits of finding native programs are

- Vendor support (for example, Microsoft does not support running Word on Linux)
- Speed (especially virtual machine emulation is slow)
- Stability
- Distribution support
- Price (most of popular Linux software is Free)

3.5.1 Emulating Operating System

Operating system emulators are the fastest class of emulators. They pretend to be the

Operating System being emulated, implementing all operating system calls. In practice, the speed is achieved at cost of requiring more manual setup for many programs.

The most popular operating system emulator is Wine, “Wine Is Not an Emulator”. It is Free, and included in most popular distributions. It can run simple programs, such as cd-rom multimedia interfaces very well. However, any complex programs that were probably not tested by its authors required manual setup and had various problems running.

WineX is a version of Wine that emulates also DirectX, the library for 3D games. The version freely available from vendor website did not work for any of the games tried. Emulating 3D games is considered to be the hardest form of emulation.

Win4Lin is a closed source emulator that requires a paid license. It was not tested, as it has not outperformed Wine in published tests in magazines.

3.5.2 Remote Controlling Windows

Remote controlling Windows is mostly used for remote administration. Common programs used for this are Virtual Network Computer VNC and Remote Desktop Protocol clients. A less used but interesting tool that has earned a reputation as black hat tool is BO2k.

Currently, Helia uses VNC to remotely control Windows. VNC is truly multiplatform, as Windows, Linux and MacOS can all be both servers and clients. The protocol is open, well documented and many commented reference implementations and free tools are available. All popular distributions have VNC clients. VNC does not have any security features by itself, but the connection can be easily put to a secure tunnel with SSH. As noted before in this paper, the VNC protocol is most inefficient, as it is based on sending compressed screenshots.

All computers running X Window System (the Linux window system) can be controlled remotely, and the protocol is very efficient. There are Free X Window System servers for MS Windows, such as the one included in Cygwin. During testing, this solution worked, but seemed unpolished, and left many questions. Some unreliable hacks were required. I could not find any large organizations using this method.

The Remote Desktop Protocol is a proprietary Microsoft Windows protocol very similar in principle to X Window System protocol. Windows Terminal Server is a Microsoft Windows tool for allowing multiple users to log in with graphical user interface. Linux has a Free client for Windows Terminal Server, called rdesktop. Some students administering both Windows and Linux boxes at work have been very satisfied with rdesktop. Using rdesktop to remotely connect to Windows Terminal Server would allow running any Windows programs, except some 3D applications, such as games. Compatibility is obvious, because the programs are actually run on Windows. Linux users login to a Windows server through using rdesktop, and see a Windows desktop in a Window on their Linux desktop. Even though this requires a Windows server, this solution has many benefits over others

- No manual setup for each program, full compatibility
- Minimal startup time, as Windows is already running on server
- Better compatibility than operating system emulators, faster than hardware emulators, cheaper than VMware

3.5.3 Emulating Hardware

Hardware emulator pretends to be a whole x86 computer hardware. An operating system is running inside a hardware emulator. For emulating Windows, this system is expensive. Each computer requires a Windows license in addition to the programs being run. Emulating hardware places huge load on memory and CPU, and true hardware emulation usually makes programs too slow for real use.

VMware is a popular closed source multiplatform hardware emulator. Helia has 10 licenses for VMware, that I have been used for teaching companies. VMware is faster than real hardware emulators, but it takes some shortcuts that make it something between a hardware emulator and an operating system emulator. It is still slower than Wine. In my experience, VMware works very well for testing user problems while giving user support through phone. Its cost, almost 300 USD for download version and the cost of an operating system license make it unsuitable for general emulation.

3.5.4 Altering original software

If source code is available, programs can often be compiled again to work on another platform. The effort required varies greatly by the program. For closed source programs, porting for just one user organization is often not available or too expensive.

A web interface to original software can be written or bought, making it usable from practically any platform, including mobile phones and PDAs. If it is wanted, this makes the software usable from anywhere. Web interfaces are available for many closed source programs too. In this case, web interface should be the first option to consider. It is possible to write web interfaces to databases, if database descriptions are available and license permits this. Writing own code is usually more expensive and error prone than using a tested solution.

3.5.5 Recommended Method for Running Windows Programs

Helia should first try to find Free Linux alternatives to all Windows programs. If web interfaces for Windows programs are available, these should be used. Helia should install a Windows Server with Terminal Server and required legacy programs. Linux users could use these programs by logging to Windows with rdesktop. Remote use from outside Helia should be allowed by using an SSH tunnel to secure connection until it reaches internal network.

3.6 Other Information Systems

Even though our focus here is mainly the workstations, the whole of information systems must be looked at as a whole. The same need can usually be filled with a mainly server side or mainly workstation solution. For example, email can be read with a workstation mail user agent (Mozilla Mail, mutt, pine, Outlook, Sylheed...) or with a web mail client (IMP, Teromail, Squirrelmail...). Moving a software user interface to web often solves cross platform usage problems.

Helia has a huge mix of partly overlapping information systems, bought and rolled out in different times. Some of the systems are the latest production quality technology, some are clearly legacy systems. According to Weill and Broadbent (1998), this is not much different to a typical company of this size. They suggest putting the existing systems to health grid according to each systems technical and managerial value (1998:224). Even though this kind of strategic analysis is out of the scope of this paper, it might provide interesting insight to the value and purpose of systems in use.

In addition to moving to free, open source solutions, Helia could achieve additional cost savings in support costs by dropping duplicate systems and concentrating to support a single system for each operational purpose. Obviously, excess diversity is only bad for operational systems, but teaching many systems, programs and platforms is mostly a benefit.

3.7 File Formats

File format defines the order of data in storage. For example, this thesis has been can be saved as OpenOffice.org sxw, MS Word doc or XHTML web page – all different file formats. If information is of high value to an organization, it is of paramount importance that organization itself has unlimited access to its own data.

File formats are a significant cause of vendor lock-in. Vendor lock-in means that even though a customer is not satisfied with quality or price provided by vendor, the cost of change has become too big. (McHugh 1999) One of the biggest lock-in scheme used by file formats is the Microsoft Word .doc format.

Some file formats are specifically made to hinder users ability to access data on his own hard disk. Realmedia file formats are a good example to this. Digital Restrictions Management (often called DRM or Digital Rights Management) is getting more common, and more vendors are likely to implement it. Using DRM reduces software interoperability, adds a single point of failure to the chain of information from author to user and wastes some bits that could be used to store the data with better quality. As all DRM can be bypassed, often easily, it does not do much to prevent illegal copying.

Patented file formats add expenses and legal risks to companies. An example of this is Fraunhofer attempt to start charging for mp3 music players.

Badly documented and uselessly complex file formats may make use of historical data expensive, often so expensive that it is discarded completely.

3.7.1 Criteria for Format Choice

Selection of file formats can be seen as an infrastructural choice, as it decides and selects which software can be used to author, view and index data. To avoid the problems listed above, the suggested file formats should be completely documented, patent free and widely used. To maintain access to information in the future, after possible vendor and platform changes, formats should be extractable by standard tools, preferably viewable as plain text (such as XML). If possible, chosen formats should also be efficient and promoted by visible organizations.

Suggestions for file formats does not mean that it should be forbidden to use other formats. Receiving a memo written in MS Word and not opening that would be ridiculous. Rather, it forms the bases for new software selection and guidance for planning courses.

For what kind of data do we need format suggestions? At least we need formats for the most important software we use, such as word processor, spreadsheet, music and video.

3.7.2 Recommended File Formats

Helia should use open formats meeting the criteria set above. Obvious choices are XML, XHTML+CSS, unencrypted PDF for documents layed out for printing, PNG and JPG for images, Ogg Vorbis for music, Vorbis Speex for speech, ASCII or UTF-8 encoded text for plain text, gzip, bzip2 and zip (with suitable compression methods) for packaging.

As OpenOffice becomes more popular, OpenOffice.org formats should be used for storing office documents. Their format is fully documented, and opens with free standard tools. OpenOffice.org Documents are just compressed XML.

3.7.3 Format Support

In addition to suggested formats above, chosen distribution should support most common file formats. These formats are formats used by currently installed software, Linux related formats (rpm, tar.gz...) and other popular formats. To test support for these formats, a test kit should be built. The test kit could be a web page with sample file for each document, with suggested formats and legacy formats separated. Non-standard and proprietary format samples should be made with original program. The most important formats to support can be deduced from the list of currently installed software (Helia IT Services 2003, Appendix "Currently Installed Software in Helia and Free Alternatives"), and from the discussion of software needs. The table below lists the most important document formats by category. As it is based on the needs of a workstations end user, it does not include packaging formats, data interchange formats or source code formats.

<i>Basic formats</i>	<i>Common formats</i>
<ul style="list-style-type: none"> –Text ascii .txt ALLCAPS –HTML, XHTML .htm, .html –Text encoded (unicode utf-8) .txt –PDF –Ogg Vorbis .ogg –DivX;) .avi –Mpeg .mpg, .mpeg 	<ul style="list-style-type: none"> –MS Word .doc –MS Excel –RTF –PDF encrypted –OpenOffice.org Writer .sxw
	<i>Rare or obsolete formats</i>
	<ul style="list-style-type: none"> –Windows help .chm –Realmedia .rm, .ram –Quicktime

4 Administration of Workstations

Managing workstations means automatically handling security patches, software updates, settings and software installation. When automatic handling fails or there are other problems, remote control is required. Many organizations still do some of these administrative tasks by walking to computers and accessing them locally. However, ability to do these tasks remotely over the network or completely automating parts of these tasks can provide huge benefits.

In addition to administrative remote control, naive users sometimes need advice on how to use a program, and this can be done using shared sessions. In my view, shared sessions are not really needed in administration, and some administrations are used to them just because their platform lacks decent method of remote control without interfering normal usage.

4.1 Current solution

Currently, Helia is using Microsoft Windows platform in all of its workstations, and software from various vendors. Operating system, updates and programs are installed by two methods: major updates are done by copying images with Norton Ghost, minor updates are installed with network login script (Helia IT Services 2003). While feature updates are just nice to have, security updates are mandatory to keep network safe. In Windows, virus database updates are nearly as important as operating system security updates.

Copying image means that an example workstation is built with all software installed and settings made. Then an image of hard disk is saved, that is, every byte on the hard disk is saved into a huge file, the image. This image is then copied to the hard disks of target computers. Helia handles images with Norton Ghost. Imaging Windows has many downsides: computers must be almost identical, all previous information on target computers is lost and thus updates end up being once or twice a year. Computers are rarely identical, so any

difference in workstation configuration means software has to be installed manually after imaging. Installing different software configurations is sometimes mandated by different needs (eg. Winha), but more often by licensing reasons (eg. Creative CD Writer).

Network login script can install minor updates. Virus protection databases are installed this way. Installation in a login script cannot install big software, because in Windows this often requires answering many questions and rebooting computer. Helia IT Services is researching alternatives to automate software install without overwriting the whole computer.

Windows operating system has had some form of automatic installer since Windows 98, but automatic installation only installs operating system but practically no programs. I have created automatic installation script for Windows 98, but it was not taken into production as Ghost was used instead.

4.2 Methods of Software Installation and Update

Software installation and update is a basic task in computer administration. Even though larger organizations can probably see the needs for most of the software before rolling out workstations, continuous updating is required for security reasons. Different methods for software installation differ in their level of automation, compatibility between programs and possibility of uninstallation. Different methods for installing software are copying static binary with an installation wizard, compiling from source, using package manager and using an automated package manager.

4.2.1 Installation Wizard

Installation Wizard is the most popular method for installing programs in Windows. Some Linux programs use it too, such as the original OpenOffice binary installer. Installer is an executable program, containing the installer and software in compressed form. When run, software is uncompressed and copied to the system according to users answers to installers questions. Even though there are user interface guidelines provided by Microsoft, the implementation of installers is varies a lot. Different third-party vendors have created software to create installers, for example InstallShield and NullSoft installers. Installation wizard has the benefit that it is intuitively usefull for naive users, who most likely double-click a file to execute it after download. Some find that easy questions can make user feel in control. Downsides are many: installing is slow, it cannot be automated, most questions are unnecessary and show lack of standardization and uninstall does not really work as system is not returned to previous state after uninstall. Usually, despite many questions, the level of actual user control is still minimal.

4.2.2 Compile from Source

Some software, especially development versions, are distributed as source code. User has maximum control over the software, and the resulting runnable binary program is optimized for users hardware. Even though GNU make system makes it possible for anyone computer literate user to compile software, some programs are harder to compile than others. Basically, a program is compiled by uncompressing the package, then typing './configure && make'. Compiling software is not efficient method if software is to be distributed to many computers. Some benefits of compilation can be reached by compiling multiple binaries to different architectures, and thus source code is usually just an addition to other installation methods.

4.2.3 Package Manager

The problems of Installation Wizards lead to creation of specific programs to install software, package managers. Package manager install software packaged according to guidelines specific to the package manager. Strict guidelines make non-interactive installation possible, thus making it practically possible to install many packages at once. For example, installing 50 packages interactively, each asking 10 questions would not be practical. Package managers also handle dependencies, typically by aborting installation if required software libraries are missing.

Some of the most famous package managers include the RPM Package Manager (formerly Red Hat Package Manager) and Debian package manager dpkg. The underlying package management format is often the biggest different between distributions. Packages created for one distribution can easily be recompiled to work in another distribution using the same packaging format. Distributions categorized by packaging format to RPM (Red Hat) based, DEB (Debian) based and to those using distributions own format. This categorization was discussed in detail in chapter "3.1 Distributions" and in Illustration 4.

Package managers handle the installation of a software or a software library for a single piece of software. Typically, if a required software library is missing, a warning is given and installation is aborted. After this, user must find the missing library and start the installation again. This is commonly known as “dependency hell”. Because dependencies are not handled automatically, system updates could not be fully automated.

Programs could almost always be compiled so that they don't rely on external software. This requires that they don't use other programs as commands (system calls) and that they have all software libraries included as static. Not using other programs is against the toolbox principle (combining small programs to achieve big tasks) and results duplicate coding efforts and too little specialization in software. Compiling all binaries as static wastes RAM memory as same code is loaded once for each statically compiled program. Static binaries require program vendor (or packager) to recompile the program each time any of the libraries needs updating.

4.2.4 Automated Package Managers

Fully automated package management software handles also dependencies, and thus fully automates installation and updates of software. This means that a computer can nightly update all programs without user intervention. Automated package managers use normal package managers (such as RPM) to actually install packages. Well defined package format (and thus a working basic package manager) is a requirement for automatic package manager.

Automated package managers replace user in all or some of these routine tasks:

- Installing requirements
- Removing conflicting software
- Verifying package author
- Listing what software is available
- Locating latest versions of software
- Downloading software before installation

To know what software is available and to receive installation packages, automated package managers use client server model. A server is a repository of packages offered for automatic installation, and client is the computer being updated.

4.2.5 Proprietary Automated Package Managers

Proprietary package managers are typically specific to a single distribution, use non-documented protocols and are somehow controlled by a single entity. Most installation systems have a small number of predefined repositories, and don't allow users to create their own repositories.

Even though proprietary package managers solve many technical problems, the closed nature

creates new risks. They create a single point of failure (vendor controlled repositories), they cause vendor lock-in and they can only provide a limited selection of software.

Red Hat up2date used to be very popular, because it was bundled with the most popular distribution, and probably because it had a graphical user interface. It is quickly getting obsolete by more advanced and more open yum. Even though up2date is still included in Fedora Core / Red Hat, the vendor officially also supports yum.

Microsoft Windows Update could be considered an automatic package manager too. However, its scope is very limited compared to update systems in Linux, as Windows Update can only update some software from Microsoft. Thus, it is not adequate solution for updating a typical Windows workstation. Microsoft controls the update repositories, and Blaster worm used this fact to launch a denial of service attack against Windows Update repository.

4.2.6 Free Automated Package Managers

Free automatic package managers make it possible for anyone to create repositories (servers) for automated updates. For users, a lot more software is available and single point of failure is eliminated. For companies with many workstations, own repository makes it possible to test updates before rolling them out to all workstations.

Advanced packaging tool apt was the first free automated package manager. Some years ago the ability to upgrade every piece of software, core system and install security updates by just typing “apt-get update” made Debian distribution and apt unique. Even though most of apts working has been ported to Red Hat, there are still some special cases where apt works best in its native system Debian.

Yum, “Yellow Dog Updater, modified”, was originally used for updating Linux made for

Macintosh hardware. It was ported to Red Hat 9.0, and became so popular that it is now officially included in Fedora Core 1. Even though yum and apt have very similar user interface and basic idea of operation, their inner workings are different. Having taught many courses with both package managers and peeked through their code, I have noticed that yum fixes some shortcomings of apt on Red Hat / Fedora platform. The code base of apt is a lot larger, because it is C and has some code for Debian only. Unlike apt, yum uses rpm directly through application programming interface (API), which leads to greater stability. Apt checks the authenticity of repositories, yum checks also packages. It is a lot easier to create a software repository with yum (yum-arch) than with apt. Finally, the RPM (installation package) of yum requires less work from user.

Many other automated package managers exist. In my experience, they require user interaction more often than yum and apt. This makes them unsuitable for updating large number of machines. Probably that has kept popularity of these tools quite low. There are many minor package managers mentioned in Freshmeat.net, and Red Hat Linux RPM Guide has reviewed some of them.

4.2.7 Recommended Method of Software Installation

Based on the analysis above and many years of experience with the most popular package management systems, and having administered repositories for both apt and yum, I recommend yum as the package management system for Helia. Main benefits of yum are good integration with Red Hat / Fedora (recommended distribution), stability, popularity (large and growing user base), security, ease of use (especially server side) and use of standard web servers.

Helia should begin by using public yum repositories, and later build its own repository. Packages tailored in Helia should be submitted to Fedora Extra or other popular repository for quality assurance. Later, if there are resources and need to test updates, Helia could start its

own repository. A suitable list of repositories is included as an appendix.

4.3 User authentication

To use computer, users must be authenticated. Because most medium to large organizations have thousands of users, it would be rather inconvenient to handle each workstation separately.

In the case organization, Helia, users are authenticated using Windows Active Directory. There are nearly 10 000 users. Also the Linux server myy.helia.fi (running Red Hat Advanced Server) uses Windows Active Directory to authenticate users.

Benefits of Helia's current solution:

- Using a single user database is an obvious choice for a network of this size
- Solution seems quite stable (having used it for a year and a half)

Shortcomings of Helia's current solution:

- Password cannot be changed from main linux server myy, so it cannot be changed by login in remotely
- Using Windows Active Directory lowers the security of the whole system to Windows level
- Active directory uses proprietary extensions to standards. Microsoft often changes the specifications, probably to keep the system incompatible with competing solutions.

Samba could connect Linux workstations directly to Helia's current solution, Windows Active Directory. Samba is a Linux server that provides native-like file shares and other services to Windows workstations. New features in Samba 3 include Active Directory support. Samba 3.0 is now able to join a ADS realm as a member server and authenticate users using LDAP/Kerberos (Samba Team 2003). However, there is not much experience anywhere on

authenticating Linux workstations against Active Directory, and as Microsoft can change its implementation according to its whim, this solution could prove risky in the long run.

4.3.1 Selection criteria

User authentication method must provide

- Password obfuscation (not send passwords clear text over network)
- Built-in support in selected distribution
- Linux and Windows support

The chosen authentication system should also meet the generic criteria for software choice laid out earlier. The system and its protocols should be fully documented and accepted as a standard. Previous in-house experience and easy of installation for IT-support would make roll out cheaper. The system should be widely interoperable to avoid multiple passwords. At least Windows and Linux workstation login should be supported, but authentication for web servers and proxies could immediately be useful too. Strong encryption with no known faults in implementation would be a requirement in an ideal world, but as the network has many other weaker points, any encryption that is relatively difficult to break is good enough at the moment.

4.3.2 NIS Yellow Pages

The Network Information Service or NIS is Sun Microsystems' "Yellow Pages" (YP) client-server protocol for distributing system configuration data such as user and host names between computers on a computer network. (Wikipedia 2003)

Even though NIS was once a popular method for centralized authentication, it has now been replaced because of security concerns (Wikipedia 2003). Many distributions, including Red

Hat, no longer give advice on setting up NIS in basic documentation.

4.3.3 LDAP

Lightweight Directory Access Protocol (LDAP) allows data to be stored on a central server, and accessed by clients through encrypted SSL tunnels. LDAP could be used to store any data that is often read and written rarely, but we are interested in using LDAP just to store user authentication data and contact information.

LDAP is in production use in many large educational organizations. Helsinki University is using LDAP to authenticate users in multiple systems, including Mappi webmail, WebOodi course evaluation and gym reservations. Some systems already use authentication tickets to allow access to all services with single login. In the future, there will be a single place for authentication. (Harjuniemi 2003)

Funet, Finnish IT network for science, which provides Internet connectivity for Helia, is researching possibilities for providing LDAP services for its members (CSC 2003, Kanner 2002). Helia should follow Funet's development and attempt to build interoperable systems. System interoperability could improve service by letting users work with less passwords and access more systems. Costs could be saved by combining development efforts, and using the development efforts already done under CSC funding.

Helia should consider implementing LDAP too. Following the CSC recommended LDAP schema "FunetEduPerson" (CSC 2003) would be an obvious first step towards compatible systems. Implementing LDAP on Linux is explained in distributions own documentation (For example Red Hat Inc 2003), and in more detail but distribution independent way in LDAP-HOWTO (Malère 2003).

Even though Microsoft Active Directory supports access with the LDAP protocol, it might be dangerous to create a system where Microsoft components are in a critical role. Microsoft has a reputation of making surprising, obscure changes in protocols that make them incompatible with other vendors products. On the other hand, Active Directory LDAP implementation is quite standards compliant at the moment. This makes combining the systems easier.

4.3.4 Recommendation for Authenticating Users

Helia should use LDAP for authenticating users. Funet recommendations should be followed when they are compatible with existing systems. Practical implementation of LDAP in Helia requires more research.

4.4 Remote Control

Currently, remote control is done with a Virtual Network Computer (VNC) protocol based closed source system with unknown level of encryption. Remote control initiates a shared session to target computer, so it cannot be used for other purposes while on remote control. Shared sessions might make it easier to use remote control for giving users advice.

Most widely used remote control methods for Linux are ssh command line connection and graphical X Window System and Virtual Network Computer (VNC) connection. All methods above should be secured. Ssh has encryption and two way authentication built in, and the graphical remote control tools can be protected with an ssh encrypted tunnel. It is obvious that only secure, encrypted communication methods are suitable for remote control. Otherwise any attacker sniffing (eavesdropping) traffic could gain administrative root access to machines.

4.4.1 Virtual Network Computer VNC

Virtual Network Computer (VNC) is truly multiplatform, with both client and server for Linux, Windows and Macintosh. All clients and servers are interoperable. VNC works by taking screenshots of target computer, and sending them compressed through the network. This makes it very slow for anything but local area network. Interoperability between Linux and Windows could be useful in Helia. Graphical remote control might be easier than command line when beginning administration.

4.4.2 X Window System

X window system, the foundation of graphical interfaces in Linux and other POSIX systems, is made for multiple users. Any computer running X window system could serve X terminals and allow many clients to log in graphically simultaneously. Protocol is very efficient compared to VNC, as instead of screenshots it sends descriptions of windows to draw and text to write on screen. Still, it is not useful for slow lines. It is trivial to create a secure tunnel for X window system traffic to allow secure graphical remote control to any computer that allows ssh access and has X Window System installed. In fact, this tunnel is created automatically in most setups, and graphical user interface programs can be run by typing their name to ssh command line. Used this way, X window system does not draw target computers desktop, but opens the remotely run programs window only. Some see this as a benefit, some would use another program, such as Xnest, to draw the desktop.

4.4.3 SSH Secure Shell

In my experience, command line is the most useful method of remote control. In Linux, anything can be changed from the command line. Mass execution of commands in multiple computers is only possible on command line. If remote access is done from outside the network, scarce bandwidth makes graphical remote control impractical, but text mode works still. Command line ssh remote control works with tiny devices too, such as mobile phones

and PDA's. The simplest method for mass execution is printing target machines names separated by whitespace, then using ssh command mode in bash for-loop. Also specialized tools for this exist.

A free ssh server, OpenSSH, is installed to Red Hat / Fedora Core by default. It is also available for all POSIX (Unix and Linux like) platforms. Free ssh client is part of practically all Linux distributions, and all Helia's workstations already have a closed source ssh client. As OpenSSH server is installed to Fedora Core by default, enabling SSH requires only opening ssh port and possibly improving OpenSSH configuration. Some configuration could be disabling now obsolete ssh-1 protocol and allowing only remote control user to log in trough ssh.

SSH allows remote logins by knowing password or owning the secret key of installed public key. As OpenSSH is integrated with Pluggable Authentication Modules (PAM), it can use any method to authenticate users. For example, a one time password system was briefly tested. To allow remote logins to workstations, a public key could be installed to allow remote root access to only the owner of the secret key. A secret key is a lot harder to be mistakenly communicated to outside parties, as it is a text file full of non-pronounceable gibberish. Also, it is more resistant to brute force attack than normal password, in case a workstation is captured.

4.4.4 Remote Control Recommendation

Based on the comparison above, I suggest that workstations are remote controlled by SSH in command line mode by default, and X Window System trough SSH tunnel, without desktop, is used for graphical remote control. Authentication should be done with a pre-installed public key. Commands could be mass executed by command line ssh.

5 Practical Recommendation for Case Organization

Free software was defined as software that meets Free Software Foundations or Open Source Initiatives criteria for free software. Some main qualities of Free software are that it does one does not have to pay for using or distributing it, and it can be modified and studied. For Helia and similar organizations, this paper recommends following Free Software Foundations guidelines for licenses, especially using GNU General Public License for software.

Linux distribution is a combination of operating system, software, installer and documentation. Distributions were compared with multiple criteria, such as prior experience in Helia, popularity in Finland and in the world, administrative automation and licensing. Using this criteria, Red Hat / Fedora Core and Debian distributions were found to be the most suitable. Helia should use Red Hat / Fedora Core distribution.

Software needs were estimated by looking at currently used software, and by random interviews in different parts of Helia. Recommended software was OpenOffice.org document processor, spreadsheets and slides, Mozilla Firebird web browser and Gnome Desktop environment. A lot of other software was recommended too. User authentication should be done with Lightweight Directory Access Protocol, LDAP. Legacy Windows applications should be replaced by Linux alternatives, turned to web applications or, during a transition period, run remotely on a Windows Terminal Server.

Administering workstations could become a lot easier with Linux-based solution. Installation can be automated with Kickstart installation script, that makes it possible to automatically install a complete workstation with software. Software updates should be done with yum (Yellowdog Updater , modified), which became a standard part of Fedora Core while writing this paper. Remote control should be done with OpenSSH, in graphical mode if needed.

User testing should start immediately by installing Linux to computers accessible to all

students in Helia, and including Linux in more courses.

Moving workstations to Linux is likely save licensing costs, even though producing an exact forecast of this was left for further research. By creating using and publishing its own package of Linux software and experiences, Helia can distribute these benefits to its partner companies.

Helia should continue installing and using Free software on proprietary platforms too. As Helia is a Polytechnic teaching about computers, it should use all common environments in teaching. Because of that, proprietary platforms will likely co-exist with Free platforms in the foreseeable future.

Helia should start using Free software in workstations, preferring GPL licensed software. A solution based on Fedora Core distribution could be installed using network boot (magic packet and PXE) and kickstart. Machines could update operating system and all software using yum, with a local repository to reduce network load. Remote administration should be done with ssh, using mass command execution and public key authentication. Central authentication of normal users should be done with LDAP. Workstations should all contain the same end-user applications, including Mozilla Firefox web browser, OpenOffice.org office suite, GIMP image manipulation program and OpenSSH client tools. A terse and simplified technical diagram on most of these aspects is presented in illustration 5.

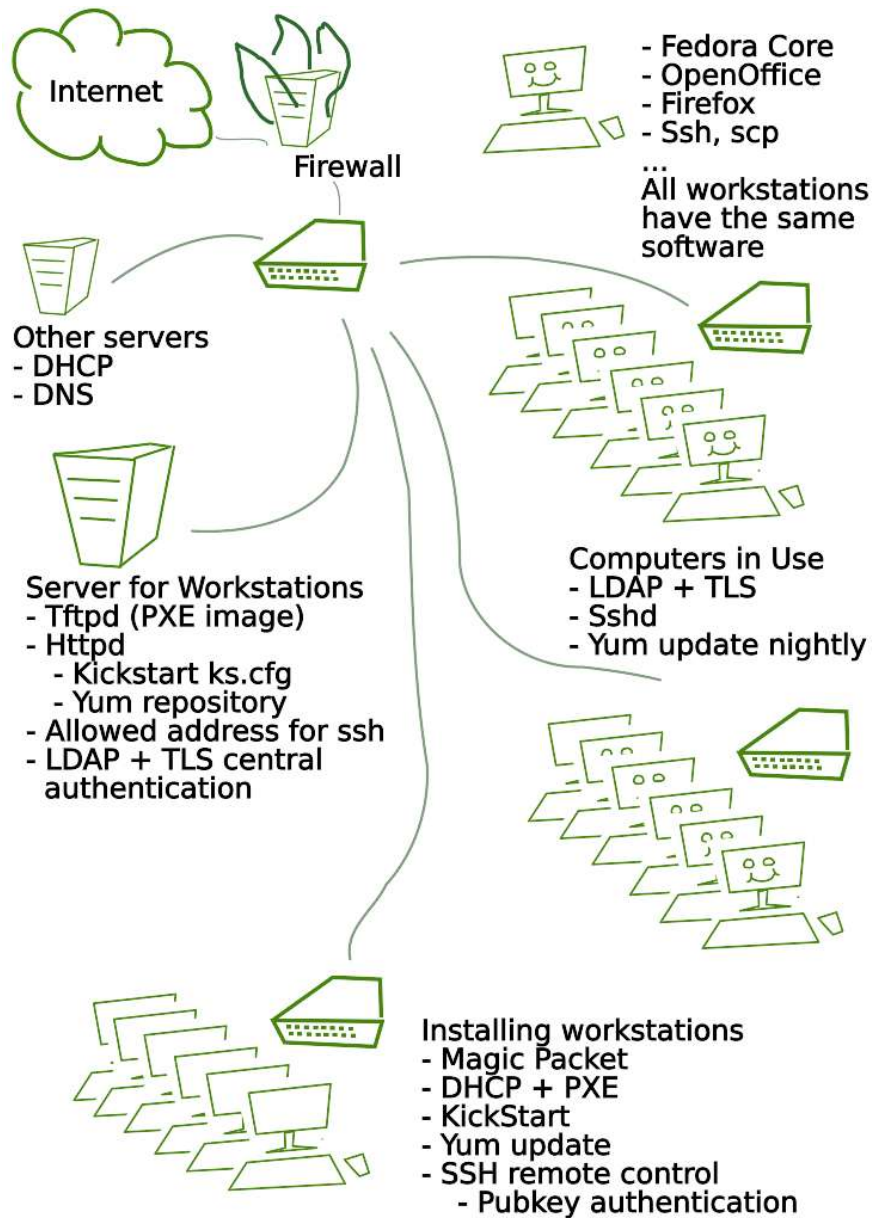


Illustration 5. A network of Linux workstations. Simplified technical diagram of a practical solution.

5.1 Costs

Cost effects are very hard to predict. Both the technical quality of chosen systems and organizational issues can affect costs greatly. However, some estimates can be given. The costs that are likely to fall after transition to Linux systems are software licenses and

repetitive administration tasks. More liberal licenses allow uses that were impossible before.

Basic software for a workstation is free. For example, operating system, document processor, spreadsheet, web browser and compilers are contained in the chosen distribution free of charge. However, Helia has bought some licenses where the cost is based on the count of workstations in use (Ivonen 2003). More savings can be achieved when these deals are renegotiated. Piloting a free solution creates a risk for software vendors by lessening lock-in, and thus gives Helia a better position negotiations.

Some applications are not yet available with free licenses. Some examples of this at the beginning of 2004 are CAD software and artistic vector drawing such as Illustrator or Freehand. These are not in wide use in Helia. Also some accounting software may be part of this category.

Some software is widely adopted in Helia, and their implementation requires big organizational effort. Winha might be an example of this. Helia might also have some accounting software that is hard to change. Even though these needs could be met by Free software, more costs can be saved by using the software until its end of life.

Special software and legacy software will not be replaced by free options, and no costs are saved in the short run. Using these programs emulated can create minor costs. In the long run, if the Free software is developed as fast as now, there will be Free alternatives for more specialized needs. Legacy applications are replaced by Free alternatives when their useful lifespan is over. Thus, some cost savings are possible in the long run.

5.1.1 Technical Support Costs

Currently, installation and update of workstation requires huge manual effort. Licensing

requirements force different software setups for each classroom. Major software updates are not possible without complete reinstall. Computers can be remote controlled, but command mass execution is limited to login script.

Administration tools described in the first part of this document can create savings in technical support. Software installation and update can be automated with yum, ssh enables remote control without interrupting current users, and installation scripts allow full installation of both software and operating system to varying hardware. Linux is also considered more stable, but in my experience stability benefit is lot bigger in servers than workstations. Viruses have caused both licensing costs and required a lot of administrative work last year. This effort would be a lot smaller in Linux, where viruses are almost non-existent. After the initial cost of education, Linux could be cheaper for technical support than Windows.

Moving to another system will require some training. As basic user tasks don't change much, most education should be put for teaching technical support. Training costs are bigger at first and going lower all the time.

5.1.2 Difficulties of Cost Measurement

Moving a large share of workstations to another operating system is a big technical and organizational change, and has an amount of risk in it. Before Helia has completely moved to Linux, environment is very heterogeneous and puts bigger requirements for support. On the other hand, this can improve learning as many workplaces have multiple operating systems and students can get experience from two important environments.

Having an alternatives instead of locking in few software vendors improves negotiation power as a customer. Licensing risk is greatly reduced by using fewer licenses that have better terms. Implementing Linux may improve Helia's image, as Linux creates many positive associations . Market share of Linux is growing fast, and by using Linux Helia can provide well educated workers to a growing market.

Due to lack of financial data, exact cost estimations could not be done. By pointing out the areas where costs will change, it seems obvious that costs can be saved by moving workstations to Linux. Removal of licensing costs and simpler administration will be the largest areas of savings. Simpler licensing removes risks and makes risk management easier.

6 Conclusions

The purpose of this paper was to point out benefits of Free software and realize them by implementing a workstation based on Free software. First, we looked into the definition of Free licensing and Free software. Using the Free software definition, we pointed out various material and immaterial benefits of Free software. To realize these benefits, we searched for a complete package of Free software to form a complete workstation. To try the criteria against real life demands, Helsinki Business Polytechnic Helia was used as a case example. Based on the evaluation of demands, a system was described and implemented, resulting in a meta-distribution. Finally, possible cost effects of rolling out this distribution to production workstations was briefly discussed.

The work resulted in a meta-distribution that was implemented and installed into several computers in non-production environment in Helia. It was shown that Free software solution can provide Free alternatives to programs that were in use in the case organization. Only a couple of legacy programs did not have Free alternatives, but for those, other solutions were described to allow usage from Free workstations. Details of implementing central authentication were left for further research. As a result of wide availability of Free software, nearly all licensing costs can be eliminated from workstations. Administration tools tested show great promise of reducing manual work compared to current situation in case organization. Administration will also be made easier by the lack of virii and simplified license handling.

Previous research has handled the development process of Free software well. This area is evolving fast, and some papers on Free software from end-user organization point of view was published during the writing of this paper. Still, there has been little academic research on implementing Free software solution on workstations before.

It seems that the license should be given a high priority on software choice, preferring

software that uses the licenses recommended in this paper, or at least meets the Free software definition. In addition to removing licensing costs, a Free license can help combining the benefits of tailor-made and shrink wrap software. It can also improve security both in technical and legal sense. Many organizations could benefit from Free workstations, even though there are transition costs that are difficult to forecast. Based on the experiences in the case organization, the possible move to Free workstations can be a slow, gradual process, where Free software is first used together with proprietary software.

The main limitation of this thesis was the lack of a large scale market test. The best way to test the solutions presented here would be installing the proposed solution in at least 500 production workstations, but the time frame of a thesis does not allow the luxury of waiting until such a big move is decided and implemented. However, many limited tests were done in both test and production environment. All systems recommended here have been tested by both writer and other users in Helia computer laboratories, producing hundreds of pages of documented experiences. New Free software was installed in Helia's production workstation during (and probably because of) this research, even though Linux is not yet officially supported in production environment in Helia. Draft versions of this paper and related documents have produced a lot of feedback from Finland and abroad.

Possible further studies should concentrate on a larger scale market test, and detailed cost analysis based those experiences. On the technical side, centralized authentication with LDAP with integration to other systems, even over organizational borders, could simplify user authentication and help sharing resources. New advanced methods of file sharing, such as AFS, could prove interesting in improving security, availability and better support for moving users. Technical research reaching further could look into resource sharing, new methods for software installation and lighter workstations. Resource sharing, such as harnessing the processing power of idle workstations, has frameworks in the low level in Linux, but not yet too many practical implementations.

7 References

Amazon 1997: United States Patent 5,960,411: Method and system for placing a purchase order via a communications network. <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/search-bool.html&r=1&f=G&l=50&co1=AND&d=ptxt&s1=5,960,411.WKU.&OS=PN/5,960,411&RS=PN/5,960,411>

Bartolich, Alexander (ed) 2002-03-14: The Linux Virus Writing HOWTO: Introduction. Visited 2003-07-01.

http://wizard.ae.krakow.pl/~wasylisp/txt/Virus_Writing_HOWTO/index.html#INTRO

BBC News 2003: UK tests open source waters. Visited 2005-02-09.

<http://news.bbc.co.uk/1/hi/technology/3181108.stm>

Benkler, Yochai 2002: Coase's Penguin, or, Linux and The Nature of the Firm: Conclusion. The Yale Law Journal. New Haven: Dec 2002. Vol. 112, Iss. 3; pg. 369, 78 pgs.

<http://proquest.umi.com/pqdweb?RQT=309&VInst=PROD&VName=PQD&VType=PQD&sid=3&index=17&SrchMode=1&Fmt=3&did=000000275540231&clientId=10156>

Bretthauer 2002: Open source software: A history. David Bretthauer. Information Technology and Libraries. Chicago: Mar 2002. Vol. 21, Iss. 1; pg. 3, 8 pgs

Brooks, Frederick, Jr 1995: Mythical Man-Month - essays on software engineering. 20ed. ISBN 0-201-83595-9.

Castelluccio, Michael 2000: Can the Enterprise Run on Free Software? Strategic Finance. Montvale: Mar 2000. Vol. 81, Iss. 9; pg. 50, 6 pgs. ISSN/ISBN: 1524833X.

<http://proquest.umi.com/pqdweb?RQT=309&VInst=PROD&VName=PQD&VType=PQD&sid=3&index=35&SrchMode=1&Fmt=3&did=000000051242424&clientId=10156>

Cole, Robert & Lee, Gwendolyn 2003: From a Firm-Based to a Community-Based Model of Knowledge Creation: The Case of the Linux Kernel Development. Organization Science.

Linthicum: Nov/Dec 2003. Vol. 14, Iss. 6; pg. 633.

<http://proquest.umi.com/pqdweb?RQT=309&VInst=PROD&VName=PQD&VType=PQD&sid=3&index=9&SrchMode=1&Fmt=3&did=000000523481721&clientId=10156>

CSC 2003-06-16: Recommendation for the Schema of the Funet Directories in Finland v 1.0.
Visited 2003-08-18.

http://www.csc.fi/suomi/funet/middleware/valinen/funetEduPerson_1_0.pdf

CSC 2003-06-18 Käyttäjähallinto korkeakouluissa. Visited 2003-08-18.

<http://www.csc.fi/suomi/funet/middleware/>

CSC 2003-07-28 Middleware in Finnish higher education. Visited 2003-08-18.

<http://www.csc.fi/suomi/funet/middleware/english/index.phtml>

Distrowatch 2003-06-08: Linux Distributions - Facts and Figures: How independent is your distribution? Visited 2003-06-30.

<http://www.distrowatch.com/stats.php?section=independence>

Distrowatch 2003-06-08: Linux Distributions - Facts and Figures: What is your distribution's package management? <http://www.distrowatch.com/stats.php?section=packagemangement>

Electronic Frontier Finland: Software patents: Finland's position. Visited 2003-06-30.

<http://effi.org/patentit/index.en.html>

Free Software Foundation 2003: "GNU General Public License": "How to Apply These Terms to Your New Programs". Visited 2003-06-30. <http://www.gnu.org>

Free Software Foundation 2003: Licenses. Visited 2003.

<http://www.gnu.org/licenses/licenses.html#LicenseList>

FreeBSD Team 1994-2004: The FreeBSD Copyright. Visited 2003-06-30.

<http://www.freebsd.org/copyright/freebsd-license.html>

freshmeat.net 2003: "Smbldap-tools tools to manager users and group using a Samba+LDAP Domain Controler". Visited 2003-08-18. http://freshmeat.net/projects/smbldap-tools/?topic_id=253

Gibson 2002-08-02: The Anatomy of File Download Spyware. Visited 2003-07-01.

<http://grc.com/downloaders.htm>

Harjuniemi, Minna 2003-03-13: LDAP-hakemistoa kehitetään. Visited 2003-08-18.

<http://www.helsinki.fi/atk/lehdet/103/LDAP-hakemistoa%20kehitetaan.html> (Helsingin yliopiston atk-osaston tiedotuslehti 1/2003 2003-03-14)

Heise Online 2005-02-06: Linux migration makes progress at Deutsche Bahn.

<http://www.heise.de/english/newsticker/news/56021>

Helia IT Services 2003: Mama Ghost. Unpublished.

IBM: "Try It: Free Rational Rose evaluation". Visited 2003-07-01.

<http://www.rational.com/tryit/rose/index.jsp?SMSESSION=NO#e2>

Johnson, Gary: "Gary Johnson's Mutt Page". Visited 2003-08-17.

<http://www.spocom.com/users/gjohnson/mutt/>

Kanner, Janne 2001-11-09: Esitys korkeakoulujen hajautetusta hakemistoinfrastrukturista.

Visited 2003-08-18.

<http://www.csc.fi/suomi/funet/middleware/projektit/haka/raportit/esitys/esitys.html>

Kanner, Janne 2002-03-15: LDAP-hakemistot käyttäjähallinnossa

<http://216.239.51.104/search?q=cache:www.csc.fi/suomi/funet/middleware/projektit/hstya/muut/loppuseminaari/LDAP-hakemistot%2520k%25E4ytt%25E4j%25E4hallinnossa.ppt&hl=en&ie=UTF-8>

Karvinen 2003: "un - extract archive in a new directory" Visited 2003-08-13.

<http://iki.fi/karvinen/linux/doc/un>

Karvinen, Tero 2002: "My favourite software: Calc replaced my real TI-85 calculator".

<http://iki.fi/karvinen/>

Karvinen, Tero 2003: Samba quickstart - File Sharing Between Linux and Windows. Visited

2003-07-01. <http://www.iki.fi/karvinen/samba-quickstart.html>

Karvinen, Tero 2005: "Build Web Interface to Database - LAMP Linux Apache MySQL

PHP". Visited 2005-04-27. <http://iki.fi/karvinen/lamp-linux-apache-mysql-php.html>

Knoppix User Community: Knoppix Community website. Visited 2003. <http://knoppix.net>

Liljeblad, Oskar: atool. Visited 2003-08-13. <http://www.student.lu.se/~nbi98oli/atool.html>

Macromedia, Inc: Macromedia Web Players. Visited 2003-07-01

<http://www.macromedia.com/shockwave/download/alternates/#linux>

Malère, Luiz Ernesto Pinheiro 2003-04-02: LDAP Linux HOWTO. Visited 2003-08-18.

<http://www.tldp.org/HOWTO/LDAP-HOWTO/>

McHugh 1999: Making It Big in Software. Rubic Publishing. ISBN: 0953548708

Open Source Initiative 2005: The Open Source Definition. Version 1.9. Visited 2005-04-27.

<http://www.opensource.org/docs/definition.php>

McKusick, Marshall Kirk: Twenty Years of Berkeley Unix - From AT&T-Owned to Freely Redistributable: In Open Sources: Voices from the Open Source Revolution. O'Reilly 1999.

Visited 2003-06-30. <http://www.oreilly.com/catalog/opensources/book/kirkmck.html>

Microsoft 2004: Shared Source Initiative: Licensing Overview. Visited 2005-02-09.

<http://www.microsoft.com/resources/sharedsource/Articles/LicensingOverview.msp>

Mikrobitti 2003: "MBnet Hintaseuranta – Microsoft Office 2000 Pro, suomenkielinen".

Visited 2003-07-31. <http://www.mbnet.fi/hintaseuranta/tuote.asp?TID=1135>

MySQL AB Trademark Policy (April 2003): "C. Trademarks and the GPL License", visited

2003-06-30. <http://www.mysql.com/company/trademark.html>

National Board of Patents and Registration of Finland 2003-02-14: Tietokoneohjelmat ja

patentointi. Visited 2003-06-30. <http://www.prh.fi/fi/uutiset/111.html>

Netcraft Ltd 2005: Web Server Survey. Visited 2005-02-06.

http://news.netcraft.com/archives/web_server_survey.html

Onnela, Eija: Varteentottava vaihtoehto - avoimen lähdekoodin käyttö julkishallinnossa. (English translation of the name: Considerable Option – Using open source in the public sector).

OSDN (Open Source Developer Network) 2003: Statistics and Top 20. Visited 2003-06-30.

<http://freshmeat.net/stats/>

Pakkanen, Atte: Spoken information, 2003-05-01.

Raymond, Eric 2000a : The Cathedral and the Bazaar: Release Early, Release Often. Version

3.0. <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/>

Raymond, Eric 2000b : The Cathedral and the Bazaar: How Many Eyeballs Tame

Complexity. Version 3.0. <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral->

[bazaar/](#)

Realnetworks, Inc: Community Supported RealPlayer Download Page. Visited 2003-07-01.

<http://forms.real.com/real/player/unix/unix.html>

Red Hat Inc: "Red Hat Linux 9: Red Hat Linux Reference Guide": "Chapter 13. Lightweight Directory Access Protocol (LDAP)". Visited 2003-08-18.

<http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/ref-guide/ch-ldap.html>

Redhat.com Trademark Guidelines: "Copyright". visited 2003-06-30.

<http://www.redhat.com/about/corporate/trademark/guidelines/index.html>

Reuters 2003-06-16: Reuters Latest Financial News / Full News Coverage: SCO shares slump after IBM lawsuit deadline passes. Visited 2003-07-01.

Samba Team 2003-08-15: The Samba Team announces Samba 3.0.0 RC1. Visited 2003-08-

18. <http://samba.org/samba/whatsnew/samba-3.0.0rc1.html>

SAP AG: mySAP Business Suite on Linux. Visited 2003-07-01. <http://www.sap.com/linux/>

Sargent 1980: United States Patent 4,873,662: Information handling system and terminal apparatus therefor <http://patft.uspto.gov/netacgi/nph->

[Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/search-](http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/search-)

[bool.html&r=1&f=G&l=50&co1=AND&d=ptxt&s1=4873662.WKU.&OS=PN/4873662&RS=PN/4873662](http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=/netahtml/search-bool.html&r=1&f=G&l=50&co1=AND&d=ptxt&s1=4873662.WKU.&OS=PN/4873662&RS=PN/4873662)

Scribus team: Scribus Desktop Publishing for Linux. Visited 2003-07-17.

<http://web2.altmuelhlnet.de/fschmid/about.html>

Shen, Xiaobai: 2005: Developing Country Perspectives on Software: Intellectual Property and Open Source - A Case Study of Microsoft and Linux in China. International Journal of IT Standards & Standardization Research. Hershey: Jan-Jun 2005. Vol. 3, Iss. 1; pg. 21, 23 pgs.

<http://proquest.umi.com/pqdweb?RQT=309&VInst=PROD&VName=PQD&VType=PQD&sid=3&index=1&SrchMode=1&Fmt=3&did=000000741552421&clientId=10156>

Taylor, Ian Lance 2003: My Visit To SCO. Published on the Linux Journal. Visited 2003-07-

01. <http://www.linuxjournal.com/article.php?sid=6956&mode=thread&order=0>

Tero Karvinen 2003: Linux perusteet ja Debian Woody 3.0. Visited 2003.

<http://iki.fi/karvinen/debian30woody.html>

The Linux Counter Project 2003: Estimating the number of Linux users.

<http://counter.li.org/estimates.php> <http://counter.li.org/reports/machines.php>

Wikipedia 2003-08-18: Lightweight Directory Access Protocol. Visited 2003-08-18.

http://www.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol

Wikipedia: Linux Distribution. Visited 2003-06-30.

http://www.wikipedia.org/wiki/Linux_distribution

Zittrain 2004: Normative Principles for Evaluating Free and Proprietary Software. Jonathan Zittrain. The University of Chicago Law Review. Chicago: Winter 2004. Vol. 71, Iss. 1; pg. 265, 23 pgs

Zoebelein, Hans 1999: The Internet Operating System Counter. Visited 2005-02-09.

<http://www.leb.net/hzo/ioscount/>

ZDNet UK 2005: Microsoft, Red Hat argue open source. Visited 2005-02-09.

<http://news.zdnet.co.uk/software/0,39020381,2092085,00.htm>

Illustration Index

- Illustration 1. License distribution of open source projects according to freshmeat.net. 12
- Illustration 2. Freedom for users versus protection of intellectual property. 13
- Illustration 3. A working Linux system consists of an operating system and applications. 18
- Illustration 4. Family tree of some Linux distributions. Distributions with their own package management system have been circled. 22
- Illustration 5. A network of Linux workstations. Simplified technical diagram of a practical solution. 64

8 Appendixes

8.1 Automated Installation

This kickstart file was demonstrated in Linux teaching session to Open Helia project. It is meant to be used with a boot cd to load it from network.

The boot cd can be created from “Fedora Core CD 1” : images/boot.iso, by modifying boot loader configuration isolinux.cfg to include the kickstart file by default, and then re-creating the cd image. A script for writing a bootable cd from this images is included in appendix “mkcd

For printing purposes, the lines ending with a backslash \ have been wrapped. These lines need to be concatenated for files to work.

8.1.1 ks.cfg Automatic Installation Script

```
# ks.cfg for FreeHelia
# Copyright 2003-2004 Tero Karvinen tero.karvinen at iki.fi
# 20040123t1000

install
#url --url http://10.0.0.1/instero/terix/
url --url \
ftp://ftp.funet.fi/pub/linux/mirrors/redhat/fedora/linux/core/1/i386/os/
lang en_US.UTF-8
langsupport --default en_US.UTF-8 en_US.UTF-8
keyboard fi
# could use 2-button ps2 here, wheel can be enabled later
# usb works instantly when plugged
mouse --emulthree
xconfig --startxonboot --defaultdesktop=kde
network --device eth0 --bootproto dhcp
rootpw piilos-ana
firewall --high --ssh
authconfig --enablesshadow --enablemd5
timezone --utc Europe/Helsinki
bootloader --location=mbr
# this will be done before script:
#zerombr yes
# reboot on pxe boot, but not with a cd in the drive
#reboot
#clearpart --linux --initlabel

#part /boot --fstype ext3 --size=100 --ondisk=hda
#part / --fstype ext3 --size=700 --grow --ondisk=hda
# Use a pre script to make swap size equal to physical memory
#part swap --size=128 --grow --maxsize=256 --ondisk=hda

%packages --resolvedeps
# Lot of stuff
@ office
```

```
@ sound-and-video
@ editors
#@ admin-tools
@ emacs
@ base-x
@ gnome-desktop
@ gnome-software-development
@ graphics
@ development-tools
@ printing
@ games
@ text-internet
@ graphical-internet
kernel
#kernel-pcmcia-cs
grub
# Text mode basics
@ base
mc
nano
openssh-clients
elinks
lynx
screen
openssh-server
redhat-config-network-tui
### X basics: kde, gnome, light window man
#@ base-x
kdebase
# icewm
redhat-config-xfree86
redhat-config-network
### minimum software
xpdf
#nedit
openoffice.org
#@ printing
gimp
epiphany
rdesktop
vnc
gaim
# g-links
### applications
@ gnome-desktop
#@ admin-tools
xmms
gaim
ethereal-gnome
nmap-frontend
#kernel-pcmcia-cs
grub
fsh
-firstboot
#-up2date
#-up2date-gnome
-rhn-applet
-rhgb



```
%pre
```


```

```
#!/bin/sh
echo "Preinstall script running"

%post
#wget http://10.0.0.1/terix/post-install.sh
#sh post-install.sh |tee >> /var/tmp/post-install.log
# reboot, have cd check rh was not just installed
wget http://my.helia.fi/~karte/helnux/yum.conf
cp yum.conf /etc/yum.conf
chkconfig yum on
wget http://my.helia.fi/~karte/helnux/installed.php
# Copyright 2003 Tero Karvinen tero.karvinen at iki.fi
```

8.1.2 Isolinux.cfg for Bootable Installer

```
# isolinux.cfg Modifications (c) Tero Karvinen
default helnux
prompt 1
timeout 60
display boot.msg
F1 boot.msg
F2 options.msg
F3 general.msg
F4 param.msg
F5 rescue.msg
F7 snake.msg
label helnux
    kernel vmlinuz
    append initrd=initrd.img ramdisk_size=8192 \
        ks=http://my.helia.fi/~karte/helnux/ks.cfg
label linux
    kernel vmlinuz
    append initrd=initrd.img ramdisk_size=8192
label text
    kernel vmlinuz
    append initrd=initrd.img text ramdisk_size=8192
label expert
    kernel vmlinuz
    append expert initrd=initrd.img ramdisk_size=8192
label ks
    kernel vmlinuz
    append ks initrd=initrd.img ramdisk_size=8192
label lowres
    kernel vmlinuz
    append initrd=initrd.img lowres ramdisk_size=8192
label memtest86
    kernel memtest
    append -
```

8.1.3 mkcd – Creating a Bootable Installer Cdrom

```
#!/bin/sh
# a script to create bootable cd from a directory with isolinux
# mount -o loop boot.iso bootmnt/ && cd bootmnt

mkisofs -v -o helnux-ksnet.iso -b isolinux/isolinux.bin -c
isolinux/boot.cat -no-emul-boot -boot-load-size 4 -boot-info-table /
home/tee/terix/minibootcd/minibootcd/
```

```
cdrecord -eject helinux-ksnet.iso
```

8.2 Estimating the Number of Users in Helia

The number of users was approximated with

```
karte@myy.helia.fi$ cat /etc/passwd|grep /bin/bash|wc -l  
10858
```

This probably includes non-active users, but we can assume that non-active users must be stored in future systems too.

The operating system version of myy was found out with

```
karte@myy.helia.fi$ cat /etc/redhat-release  
Red Hat Linux Advanced Server release 2.1AS (Pensacola)
```

8.3 Yum automated software installation and update for Red Hat Linux

Published during this study at <http://iki.fi/karvinen> and tested with more than hundred users in Helia Tiko, Helia company training and volunteers worldwide.

Yum is the easiest way to keep all programs up to date. It downloads and installs the latest version of a program. A single command can update all software installed, including third-party software, security updates and operating system. It can do the updating automatically in the night. In this howto, we install yum and make it do all the above.

Yum is similar to, but better than apt, apt4rpm, windows update, up2date, yast and many other package managers I have seen.

Yum works in a safe, standardized way. It uses rpm (Red Hat package manager) for installing programs. Authenticity of packages is checked with strong gpg encryption. Package repositories are just folders on a web server.

This tutorial is for Red Hat 9. If you are using the newer Fedora Core 1 or later, you already have yum installed. For configuring Fedora version, see [unofficial Fedora Faq](#).

We will [install yum](#), then [choose trusted packagers](#) and start [installing programs](#). We will also see some [yum tips](#).

(c) [Tero Karvinen](#)

8.3.1 Install yum

Red Hat has rpm package manager installed by default. We use rpm to install yum.

Download [yum-2.0.3-0.fdr.1.rh90.noarch.rpm](#)

Open command prompt: Main menu (the red hat menu on bottom left corner), System Tools, Terminal. Become root with `su -`. Notice how your prompt turns from `$` to `#`.

Go to the folder where you downloaded yum. Most likely `cd /home/your-user-name/`. If you can see the file with `ls`, you are on the right place.

```
rpm -Uvh yum*.noarch.rpm
```

In the command above, `rpm -Uvh` means installing just like `rpm -i` you may have used already. `-Uvh` just displays some extra info and erases old version of program if necessary. You can `rpm -q yum` to see if yum is installed. It should tell you the version number of yum.

```
# rpm -q yum
yum-2.0.3-0.fdr.1.rh90
```

8.3.2 Add trusted packagers to your keyring

In yum, it really does not matter if enemy takes over the internet and fakes to be some website offering software. All software is cryptographically checked before installation. To install some software, we must tell yum who we trust.

```
rpm --import /usr/share/doc/yum-*/Fedora-GPG-KEY
rpm --import /usr/share/doc/yum-*/RPM-GPG-KEY
```

We installed the key of Red Hat Inc, as we obviously trust the company that compiled our operating system. We installed Fedora's key, as we have already installed yum packaged by Fedora. These two keys are probably the most usefull ones.

8.3.3 Start installing software

```
yum install lynx
```

Because it is your first run, it first downloads headers that contain information about what is available. This can take as long as 20 minutes, but it only needs to be done once. Yum prints the names of headers it downloads

```

Gathering header information file(s) from server(s)
Server: Fedora Linux / stable for Red Hat Linux 9 (i386)
Server: Fedora Linux / testing for Red Hat Linux 9 (i386)
Server: Red Hat Linux 9 (i386)
Server: Red Hat Linux 9 (i386) updates
Finding updated packages
Downloading needed headers
getting /var/cache/yum/fedora-stable/headers/leafnode-0-1.9.43-
0.fdr.1.rh90.i386.hdr
getting /var/cache/yum/fedora-stable/headers/libzvt-devel-0-2.0.1-
0.fdr.5.rh90.i386.hdr
getting /var/cache/yum/fedora-stable/headers/mhash-devel-0-0.8.18-
0.fdr.1.rh90.i386.hdr
[.]
Resolving dependencies
Dependencies resolved
I will do the following:
[install: lynx 0-2.8.5-7.1.i386]
Is this ok [y/N]: y

```

Accept with `y` and press enter. Yum downloads requested packages and installs them.

If any additional programs, dependencies, are needed, yum will ask if you want to install those too. For example, lynx needs perl-CGI, so if we don't have that installed yet, yum installs it.

```

Calculating available disk space - this could take a bit
lynx 100 % done 1/1
Installed: lynx 0-2.8.5-7.1.i386
Transaction(s) Complete

```

Now all users can use lynx right away. No reboots, no changing cdroms, no nuisance. If you are still root (have a `#` on your command prompt), `exit`.

Now try `lynx`. If you can run lynx (the text mode web browser), you have succeeded.

Congratulations, you have now installed the state of the art package manager yum.

8.3.4 Yum tips

That was just too easy, wasn't it? To have some fun with yum, try

```

yum list "*ssh*"           # lists packages that have "ssh" in the name
chkconfig yum on          # make yum update all programs every night
yum remove up2date        # remove a program, dependencies handled
yum -y install pine       # -y answers "yes" to all questions
                           # pine is the package that has pico editor

```

8.3.5 Links

Duke University 2003: [Yellow dog Updater, Modified](#). Official homepage of yum. Has a list of yum [repositories](#).

Fedora Project 2003: Fedora.us. The biggest yum repository, now merging with Red Hat.

Saou, Mathias 2003: Freshrpms.net. The first big automated software repository for Red Hat.

Best packages of many programs, such as mplayer and sylpheed. Good documentation.

To use yum in some real application, try some of [my other tutorials](#). Most of them use yum to install software.

8.3.6 Copyright and Administrivia (for Appendix “Yum automated...”)

Tested with Red Hat Linux 9 Shrike

Copyright 2003-09-28, 2003-09-29, 2003-11-05 (yum list ssh), 2003-11-19 (Fedora Core advise+link) Tero Karvinen.

8.4 Legal Notice

No part of this paper should be considered legal advice. None of it has been checked by a lawyer. Licensing terms of any software referenced or included are here as reported by software vendor in question, for authoritative licenses, see legal notices in software in question.

8.5 Currently Installed Software in Helia and Free Alternatives

List of software installed in workstations of Helia on 2003, based on "Mama Ghost" (Helia IT services 2003) and observation by writer.

Multimedia players play music and videos in many file formats, such as mp3, dvd, real, quicktime, flash and shockwave. Currently installed are

- Windows Media Player (generic media player) - MPlayer or Xine
- Flash player, Shockwave player - Linux versions available from Macromedia (Macromedia, Inc 2003)
- WinAmp (mp3, ogg) - xmms, freeAmp
- Realplayer basic (rm, ram) - Linux version is available from Real.com (Realnetworks Inc 2003), but there are reports of privacy problems (See for example Gibson 2003), so excluding RealPlayer from distribution should be considered. Also Helia should avoid producing or publishing content in RealMedia formats.
- DVD playing: MPlayer, Xine, VideoLan client or Ogle
- Linux uses mime types similar to Windows (this program opens that kind of file)

Utilites

- PowerArchiver (zip, tar, gz, bz2, rar, cab, arj, arc, ace, zoo, lha) - file-roller, konqueror, or command line unarchivers called from script (I have written such a tool in 1993). Also atool, un.
- Acrobat full version (view and create pdf) - Almost all Linux programs can print pdf. Viewers include Adobe Acrobat Viewer, xpdf, kghostview, ggv. Pdf viewer plugin for Mozilla is available.
- Ssh and sftp client - openssh, scp, sftp, gftp. Also a free ssh server is available.
- Easy CD creator (data and audio cd burning) - cdrecord, kcd, k3b. Burning cd's as a non-root user must be set up manually.
- Textpad text editor - Astonishing amount of text editors are available: kate, gedit, vi, emacs...

Browsers

- Internet Explorer and Netscape, Java support - Mozilla, Netscape, Konqueror...
- Java plugin and SDK are available for Linux.

Databases

- Solid - MySQL, Postgress
- Winha and Otso, course management software - This could be challenging, as the software is tailored for Helia. Could run Windows versions in Wine, or write a web interface if sourcecode or application programming interface is available.

Office

- MS Office 2000 - kOffice, OpenOffice or Abiword and Gnumeric
- Frontpage - Mozilla has a WYSIWYG (graphical) web page editor. Many web editing tools are available: validators, preprocessors. Apache web server with scripting support could be installed to web development workstations.
- Language support fi se, uk, de, fr, ru, es, cn - some Linux distributions use utf-8 charset in all programs, so they can display practically any language with the same charset. Spelling is available in many languages including Finnish and * English. User interface language, including desktop, taskbar and programs, can often be changed easily, but the level and quality of translation varies.
- PageMaker (desktop publishing) - Scribus, but I have not tested it.
- PhotoShop (image manipulation) - Gimp, gnu image manipulation program that has functionality similar to PhotoShop, but the user interface is often considered slightly clunky. ImageMagick provides batch image handling superior to PhotoShop.
- TiimiPosti (email) - Might be able to run in an emulator like Wine. Is there really need for multiple email systems? Staff could move to IMP, which is used by students. Dropping duplicate solutions is likely to reduce support costs.

Educational

Typing Master - Makin' Bakon Typing Tutor, TuxTyping, typespeed... I have not tested any

of these typing tutorial programs. Promontor (Language learning, could be some of English, Swedish, German, French or Finnish). - Could be emulated. Many free dictionaries exist: Magic-Dic, QuickDic, StarDict...

– Venäjän Kielen Mestari (Russian language Learning) - could be emulated.

Other software

– Rational Rose (visual modeling for integrated development environments) - Linux version available (IBM 2003). UML models can also be drawn with Dia.

Sap gui (Enterprise Resource Planning) - MySAP client for Linux (SAP AG 2003).

– Virus protection - Virus protection is not necessary with Linux. Instead, Linux uses generic security methods to protect from virii (Bartolich 2002).

Workstations management

– Installation with Ghost and Windows XP sysprep. Most workstations still require manual tweaking. Licensing issues require different software combinations to be installed.

– Asset management is done with a proprietary program.

– Remote control is currently done with a proprietary Virtual Network Computer (vnc) based program, whose security level and encryption is unknown. Linux allows remote administration with very high security by installing (Free) ssh server on every workstation. Ssh server is installed by default in many popular distributions. Graphical remote control is possible with ssh, X Window System and vnc.

– Software updates are handled with ghost and network logon script. Linux has more sophisticated update systems, such as apt and yum. They are discussed below in chapter "Managing Workstations".

8.6 Glossary

License – An agreement between software vendor and user. Accepting license is usually a requirement for using software.

Distribution - A combination of operating system and software. An installer has usually an easy to use installer, documentation and support.

Linux - The Free operating system and distribution or just the Free kernel (the tiny core of an operating system), depending on context.

Operating System (OS) – software platform that lies between applications and hardware.

Some popular operating systems are Windowses, BSD, Apple OSX and Linux.

Information Technology (IT) – field of automatic processing of information, which in practice consists of computer systems and related organizations.

© 2003-2005 Tero Karvinen <http://iki.fi/karvinen>