Open Source Software: A discussion of the issues surrounding its applicability and usage in education

An extended essay for the *International Baccalaureate Organization*

Rajesh Swaminathan

Candidate Code : 001188 – 034

Subject : Information Technology in a Global Society

Supervisor : Mr. J. Kamimura

Word Count: 3940

February 24, 2006

Abstract

Free and Open Source Software is having an immense impact on the way we use computers for day-to-day computing. In the existing model of computing, pay-for-use licensing by copyright holders has hindered collaborative development. Open Source Software is changing the way we see and utilize computers and software, and this change has had a direct impact on education. Open source software is enabling students in developing and under-developed nations gain access to high-quality applications, code-libraries and tools, for no monetary expense whatsoever. Schools and other educational institutions, once hesitant to set up computer labs due to high costs of software licensing, are now able to do so while keeping within the constraints of their IT budgets. Today in our global world, open source software is characterizing a new kind of cooperative learning known as *e-learning* and has come to affect almost every realm of education including library management, administration, intranet maintenance, content development, in-school communication, teaching, and several other pedagogical domains. This essay aims to address the myriad areas in education where open source can be applied, underline specific important advantages of its usage and provide useful information on the issues involved. Finally, it will form a cohesive argument, with the help of supportive case histories, that open source systems are a better way to facilitate IT literacy in classrooms.

Contents

1 Research Question

What are the major advantages of using Free and Open Source Software (FOSS) in an educational setting such as a school? Discuss the issues involved and outline how an ideal classroom (or computer laboratory) using open source software might function.

2 Introduction

Today in this global world, education has grown to become one of the most important aspects of modern living. So much so that besides food, clothing and shelter, education is also increasingly becoming a standard necessity in life. In a world where education bares so much importance in our society, it doesn't come as a surprise that educators have been continuously inventing novel methods to educate a wider population in a more efficient manner. Free and Open Source Software (FOSS) is one such invention, or discovery rather, that has been helping students around the world come together and learn more — thanks to the uniting nature of the Internet.

To define Open Source in simplistic terms is a daunting task, and one that has been attempted by many in the past who have met with varying degrees of *failure*. The world of open source, quite problematically, is as diverse as our own globe. Since open source has come to be more of an ideology rather than a concrete idea or model, it makes it all-the-more tougher to describe its importance in education.

> "It is too much of a task to speak about or document authoritatively the open source phenomenon. The field is simply too vast and it affects too many aspects of economic, cultural, social, and political sciences to be fully grasped by one individual." [OSDevCVS]

Information Technology is an intrinsic part of today's modern schools. Computers are found almost everywhere, and the basis for running a computer smoothly is the computer's operating system and the numerous applications and tools that help perform a user's specific task. However, people around the world seem to be running into constant problems with licensed software, known

as *proprietary* software, powering their systems. High licensing costs, inability to modify, copy or re-distribute the program, extended delays between releases, numerous security concerns characterized by uninvited Internet viruses, dependency problems, lack of reliability, standards noncompliance, bouncy performance and unfixed bugs are just a soupçon of problems users of proprietary softwares experience on a daily basis.

Open Source Software promises to not only put an end to these hair-pulling problems, but also to provide a firm infrastructural foundation for future growth. Today, open source software is being deployed globally, especially across developing countries. Many schools are recognizing the benefits of putting FOSS to use in the building of their Information and Communication Technologies (ICTs). ICTs are computer-utilizing technologies, such as the Internet and email, used for rapid communication and information dissemination. Their presence in educational institutions such as schools is indispensable for an all-rounded advancement of a student.

2.1 A word about Terminology

In today's terminology, the phrase *Free Software* refers to the *free* and *open* nature of software and has no connection whatsoever with the cost of the software, whether they be purchase costs, licensing costs or maintenance costs. The 'free' thus denotes freedom rather than the absence of a price tag. Not surprisingly, the term 'free software' is vague and ambiguous, and has often been mis-interpreted by the media. As a result, the community has decided to reference free software as *Open Source Software* (OSS), which in short refers to the bundling of source code along with the software. OSS takes a more pragmatic approach to software development and distribution, but essentially, the two terms are synonymous and will be used interchangeably in this essay.[1]

[1] See Richard Stallman's excellent essay, *Why 'Free Software' is better than 'Open Source'* at www.gnu.org/philosophy/free-software-for-freedom.html for a well-written presentation of the case that the two terms are *not* interchangeable.

3 Advantages of Open Source Software in Education

Recalcitrant schools not implementing an open source strategy in their organizations frequently pose the question *Why Open Source for Education?* This section explores the various advantages of Free and Open Source Software (FOSS) in classroom environments and provides compelling reasons for schools, colleges and universities to consider switching to FOSS for their software requirements.

3.1 Cost Benefits

The first and biggest constraint when educational institutions decide to equip their classrooms with computers is cost. A simple cost analysis reveals that the hefty costs involved in setting up labs is primarily due to the high costs of software licensing. Open Source Software (OSS) comes up with an elegant solution to this problem: it provides all software for free. The cost barrier is thus lowered by potentially reducing the cost of software by nullifying the initial cost of acquisition. All OSS can be easily downloaded via the Internet[2], and so any user with a modest Internet connection has access to thousands of freely available software titles.

Gary Glasscock demonstrates how a school system could save $10,000 on each computer lab or network segment [GaryG]. The money saved by cutting down on licensing fees, which once formed a major portion of the budget, can be put to better use elsewhere — training for example. Money can be invested in invaluable training services to enhance student productivity and improve their skills.

Public schools around the world work under a very tight budget and are continuously under financial pressure. With OSS, schools are now able to build better laboratories with powerful back-end servers — all running on open-source technologies. OSS is efficient; it requires minimal computing power. Consequently, old (but working) machines can be recycled rather than used for landfill. Sometimes, the money saved by cutting licensing costs can be put into better hardware or more

[2] One can download an entire operating system, the Linux kernel, for free from ftp.kernel.org/pub/linux/

workstations, thus allowing a greater number of students access to the ICT infrastructure simultaneously.

CASE STUDY: KERALA, INDIA

In 2002, a project to introduce computer facilities in schools in Kannur, Kerala was initiated by the local Member of Parliament and the district administrators. To cut cost it was decided that a Linux Terminal Server Project (LTSP) solution would be used to set up the facilities.

Using the LTSP to set up the computer facilities resulted in substantial savings as more computer facilities could be set up in more schools. Forty-three government schools now have access to their own computer facilities running GNU/Linux with numerous FOSS. The teachers were given basic training on GNU/Linux systems and the computer facilities have been well received by both teachers and students.

More information about this project is available at the following URL: http://s2s2net.netfirms.com/project.html

This case study adopted from [FossPrimer]

A very prominent example arises when high schools attempt to teach database rudiments to their students. Most commercial databases are extremely expensive, and schools can therefore not afford to purchase them. However, many schools across the globe have deployed the open source MySQL[3] and PostGreSQL[4] databases, both of which are reliable, robust and most importantly, free. High schools can therefore afford to offer a head-start to database fundamentals even before students enter a specialized tech school.

In a nutshell, tax-payers' monies are no longer spent on unnecessary purchases of proprietary or commercial software. For instance, Microsoft™ Windows™ is a common operating system found on school computers. In a lab of 30 workstations, 30 unique licenses need to be purchased, and the total can amount to a tremendous amount of expenses for the school board. Licenses for newer versions will have to be re-purchased, and this usually discourages schools from upgrading their software, thus compelling them to use old, incompatible and bug-ridden operating systems and

[3] http://www.mysql.com
[4] http://www.postgresql.org

applications.

> "Companies recognize the importance of the education market because the students of today are tomorrow's employees in the ICT sector. They will also be the users of technologies either on a personal basis or in the workplace. Hence, if they are exposed to certain products during their education, they will tend to continue to use them in the future. For this reason, companies will go out of their way to provide incentives, such as hefty discounts, to capture the education market." [FossPrimer]

OSS is replicable, which means that it can be installed on any number of computers, and can also be distributed to students who may then proceed to install OSS on their home computers or laptops. Since students find it convenient to install the same operating system that is used in their schools, cost benefits are seen not only within the educational institution itself but in the homes of students as well.

The cost benefits don't end with just lab set-up costs. Economists state that an institution will need to taken into account the Total Cost of Ownership (TCO) of the entire application suite before deciding if the change is feasible. Figure 1 illustrates the relative difference in TCO between FOSS and proprietary software. It is evident that although the 'other costs' incurred are almost the same, the TCO of FOSS is considerably lowered due to the absence of licensing costs.

In addition, most OSS come with excellent documentation which has a potential to reduce training costs. Maintenance costs can also be lowered since OSS is an open platform: the institution setting up the ICT can itself develop tools to assist in this process.

The flip-side to these advantages is that OSS comes with no guarantee whatsoever. Popular applications are often bug free, but the program does not necessarily have to deliver what it promises. However, this limitation can be circumvented by ensuring regular backup-taking and by updating the software periodically, which serves to eliminate any known bugs.

Due to the enormous number of domains where cost factors can come into play, it is not possible to evaluate the cost benefits of Open Source Software to the fullest extent. Holistically speaking

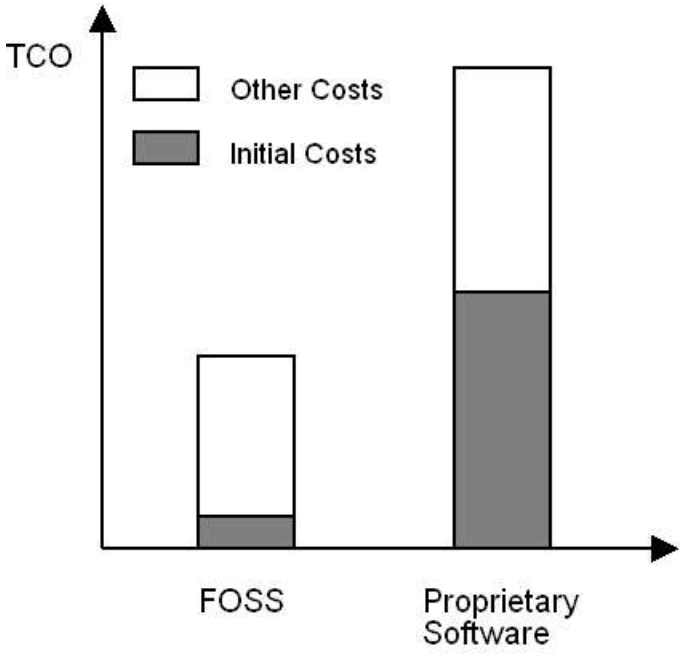

Figure 1: *Relative comparison between TCO of FOSS and proprietary software (not based on actual figures).* Adapted from [FossPrimer]

however, OSS tends to scale better, and public schools that are stranded for money—such as in under-developed nations—are able to deploy an OSS strategy to educate their students in the same way and to the same degree as their counter-parts in other developed nations. This is absolutely critical to achieving educational equality in our present technological world.

3.2 High Availability

Another defining characteristic of Free and Open Source Software (FOSS) is its high availability. Many proprietary softwares are not available in poor countries because companies find it infeasible to distribute their product to a small market. In contrast, FOSS is readily available for free download directly from the Internet, a global network. Popular software also have "mirror sites" that duplicate packages so that FOSS can be downloaded from multiple sources if one server happens to go down. A workstation in a classroom even on a remote island can therefore complete a full installation without hassle — the only requirement being an Internet connection.

In cases where bandwidth is limited (as in schools in developing countries), FOSS is easily obtain-

able on a CD-ROM for a very nominal fee from a variety of online stores. An added luxury is that software can be copied onto multiple CDs and installed on several computers, or can be distributed freely to students without having to purchase additional licenses, as is the case with proprietary software. Some open source operating systems such as Linux can be run directly off a CD[5] or from a terminal server[6], making OSS a snap to install and use in schools. Upgrades to software can be obtained in a similar fashion via the Internet.

The ubiquity of OSS is thus an important factor to be taken into consideration while evaluating its usefulness and interoperability.

3.3 Bridging the Digital Divide

Open Source Software gains are more than monetary. People have recently come to question "the potential of Free and Open Source Software (FOSS) in bridging the so-called *digital divide*." [SchoolDisc] This 'digital divide' is an important cause for concern among educational institutions and also businesses as students enter the work force. Organizations in developing and under-developed countries are at a disadvantage since they cannot take advantage of computers and software to solve everyday problems in say Mathematics, Biology or Astronomy.

A very prominent example related to this discussion is that of cluster software. Because of the so-called 'digital divide,' only certain countries that excel in the software industry are able to build super-computer clusters to indulge in hardware intensive processes (HIPs) and solve complicated problems. In the modern field of Bioinformatics for instance, proteins can be modelled by iterating over the various permutations and selecting the best structure. Countries with poorly developed software infrastructures can neither afford to buy cluster software nor do they have the resources to build this software themselves. In order to combat this inequality so that everyone around the world can participate in research at an equal footing (software democracy), OpenMosix[7], an OSS application, has developed a Cluster Management System, which can be downloaded by anyone for

[5] Knoppix at http://www.knoppix.net/
[6] See http://www.ltsp.org/ for more information on the Linux Terminal Server Project.
[7] http://openmosix.sf.net/

free. This is a fine example of how OSS has helped bridge the growing digital divide in our world, and one that can be extended to within the realms of a classroom.

The open philosophy of FOSS can have significant implications especially in educational institutions where FOSS aesthetics are "consistent with academic freedom and the open dissemination of knowledge and information." [FossPrimer] The importance of openness has been summarized by Vessels:

> "The advances in all of the arts and sciences, indeed the sum total of human knowledge, is the result of the open sharing of ideas, theories, studies and research. Yet throughout many school systems, the software in use on computers is closed and locked, making educators partners in the censorship of the foundational information of this new age." [Vessels01]

Students must therefore ensure that the software they use is open to scrutiny and verifiability, especially if it is being used in research work and/or computations.

PERSONAL EXPERIENCE

In building a web application framework, I ran into a dead-end because I didn't possess the technical competency to implement a specific feature. I was programming using the open source PHP language, and the very fact that PHP was open source software generated an ingenious idea within me. I did a bit of searching and I finally found what I was looking for:

http://cvs.php.net/co.php/pear/DB_Sqlite_Tools/Tools.php?r=1.12

The open nature of PHP code thus allowed me to solve a crucial framework problem, which would not have otherwise been possible.

3.4 Development of a Global Community

> "Using open-source software, you teach co-operation and community, you teach [students] that everyone is a participant." [Kaivo]

Proprietary software is usually packaged in a *binary* format which is hard or even impossible to reverse-engineer or modify. If proprietary software is used for academic research purposes, the al-

gorithms involved in analyzing and manipulating the data are not exposed to public scrutiny. One of the defining features of scientific experiments is that the research procedure is made available to other scientists working in the same domain. If the algorithms aren't published openly, science cannot progress and proceed at a normal pace. Each team would be competing rather than cooperating with one another. This would be a bleak scenario. *There would eventually be an urgent need to uncover, unobstruct and thus unimpede the spread of Information and Technology in our Global Society.* Knowledge must be open. Knowledge must be free. This is a serious ethical concern that today's educators need to address immediately.

Open Source Software, on the other hand, dictates openness wherever put to use. Furthermore, OSS can be licensed using the special GNU GPL[8] license, which means that the software author needn't necessarily have to give up ownership of his code.

Academic environments where FOSS is prevalent will encourage students to tinker with the source. This might pique students' curiosities and might eventually result in novel and innovative solutions to common problems in the software industry. FOSS also encourages cooperative development among student programmers around the world. All this helps in building a solid global community upon which our ever-growing software industry will be based.

3.5 Elimination of Piracy

As mentioned previously, students often find it convenient to install an operating system that is used by all. Running the same operating system and applications as the school does makes it easy to transfer homework assignments back and forth. In such a scenario, students often find themselves forced to use a particular brand of software because failing to do so may result in incompatibilities. If this software turns out to be commercial or proprietary, most students, instead of purchasing a legal copy of the software from an authorized vendor, relegate themselves to downloading a pirated copy from the Internet. This is equivalent to stealing someone else's intellectual property. In today's

[8] General Public License. www.opensource.org/licenses/gpl-license.php

terminology, this is known as *software piracy*, and is a serious crime punishable by law.[9]

> "Software piracy–or 'software theft'–includes counterfeit products and fake licences. It also includes the unlicensed copying of software."[10]

It therefore becomes the foremost duty of every student to ensure the absence of pirated softwares on their home computers. "If proprietary softwares were used as the basis for teaching, students would have no choice but to use pirated copies of software to allow them to do homework and assignments at home or on their laptop computers." [FossPrimer] If students are unable to purchase the appropriate licenses, they must resort to finding other alternatives to complete the task at hand. Open Source Software is one such alternative.

With more and more schools adopting the "open source" ideology, students become more comfortable in installing free software at home. The open source movement will thus help reduce the need for students to break the law and unrighteously steal others' intellectual property.

There is also the question of ethics involved here. One of the primary goals of public school education is to inculcate a set of ethical and social mores in the student. Not stealing someone else's intellectual property is one such moral responsibility, and OSS helps achieve this goal by abolishing all motives for software piracy. In developing countries especially, OSS can have a huge impact as a staggering percentage of all installed software is pirated. Open Source Software encourages students to share and share alike, and eminent open source licenses such as the GPL encourage this paradigm.

3.6 Numerous Educational Applications

The Debian GNU/Linux operating system, for example, has approximately 8710 packages that can be downloaded and installed for free.[11] There are a plethora of tools available for a variety of education-related tasks such as writing papers, performing complex calculations, preparing graphs,

[9] Further information can be found at http://www.bsa.org/usa/antipiracy/Piracy-and-the-Law.cfm. Retrieved Sep. 11, 2004.
[10] http://www.microsoft.com/canada/piracy/basics/what/default.mspx. Retrieved Sep. 11, 2004.
[11] See http://packages.debian.org/stable/

modelling proteins and chemical molecules, studying planetary motions, word processing, analyzing data through spreadsheets, making presentation slides, writing scripts, programming, and a myriad other programs, applications and tools most likely to be used by students.

Volunteers around the world have even collaborated on a customized Linux operating system just for schools. This project, known as SkoleLinux, has its home at http://www.skolelinux.org. Such a tailored system of software applications and components make it incredibly easy for schools and students to install OSS on their computers.

3.7 Reliability, Performance and Security

In an educational situation, it is imperative that students' information and data be kept secure. Also, the computers that they use must be reliable and must not have to deal with application crashes frequently. OSS is considered to have better reliability, performance and security due to its very versatile developmental methodology. Since OSS is openly available, anyone can identify vulnerabilities, fix bugs and send patches to the maintainers. In a classroom setting, competent students can fix bug themselves and can modify software to suit the institution's specific needs. The betterments that these students make will ensure superior performance of the lab for years to come.

For instance, if a spreadsheet application does not have a specific feature required by the student, the generation of histograms for example, students can look at the code and add this feature themselves. The histogram feature will then be made available to all, including future students, and others would be able to suggest improvements or file bug reports on the feature in question. This kind of developmental model can empower students tremendously as they are now bound only by their own capabilities, and can uplift the software's reliability, performance and security substantially. Labs would become more "organic" as students are both users as well as contributors.

Mature open source softwares are extremely reliable and are geared towards exceptional performance. For example, in a quantitative analysis of database software carried out by Reasoning Inc., it was found that the OSS database MySQL has six times fewer defects than other proprietary

databases [MysqlReas].

3.8 Software Localization

This factor of “software localization” is crucial in determining if software can be used in non-English speaking countries, or in countries where English may not be the language of choice. For example, students in China, Malaysia, Norway or Brazil may find it much more convenient to use software designed to have menus, messages and text in their preferred locale.

Open Source Software scores on this one because it can be localized for free. This localization is possible primarily due to the open nature of the software itself. The good news is that such software localizations need not involve the original developer, and can be done by anyone.

In contrast, the proprietary development model is such that localization is done only if it is feasible commercially. This poses a problem in countries where certain language-speaking populations are small, and there is almost no incentive to localize software for that narrow market.

4 The Ideal Classroom

It may not be immediately obvious, but modern schools depend a lot on computers for their every-day tasks. Teachers use computers to organize their courses and record students' marks; students use computers to perform simple spreadsheet work, build web applications, conduct presentations, transmit information, communicate with other students, and so on. In a setting where computers are required for so many functions, surveyors are often skeptic as to whether there are enough software in the Open Source world for all these common tasks. The answer is yes. This section attempts to provide a brief run-through of how an OSS-equipped classroom might run.

There are 30 workstations in the classroom, just adequate for the 30 students in the classroom. Students need not share computers with their peers, which usually results in only one out of the two or three learning. The students power on what is known as a *thin client*. Thin clients have

only a processor, network card and other essential peripherals, but no hard-drive, CD-ROM drive or floppy drive. This reduces costs noticeably. The thin client obtains the *Linux* operating system from a master server through the network and takes advantage of the master server's resources for its purposes. Three competent students along with their lab supervisor manage the entire set-up themselves (since it was they who installed it in the first place), thus having no need to keep an IT staff on call. This, again, has lowered costs. Furthermore, all software being used is accompanied by excellent documentation and when in doubt, students find it much more quicker to have their queries answered through mailing lists, forums and IRC channels. The installation of this simple server-client framework has forced them to know their systems in and out and can therefore use their own wise judgement to clarify issues rather than having to rely on technical support. When they come across a bug or a missing feature, they simply code it in themselves and re-compile. They are good denizens of the open source community and so immediately make this fix/feature available online as a patch for other students around the world to freely use.

This may all sound hypothetical and unrealistic, but is indeed happening in some countries around the world, for example at Yorktown High School in Arlington, Virginia. [YorkTown]

The Open Source phenomenon addresses software requirements by providing a variety of software capable of dealing with routine tasks. For example, one can browse the Internet with Mozilla Firefox, check email with Mozilla Thunderbird, word process with Abiword, maintain library databases with Koha and so on. There exists an OSS application for almost every educational task, and all of these are freely available with source code and easily obtainable from public repositories. There are many active student developers around the world who still continue to improve pre-existing software and also develop new software that has the potential to enhance the present educational standard of our world.

Even if software for a specific task is unavailable or unsatisfactory, competent and determined students may collaborate with other students, both within the school as well as internationally, to build their own applications. All tools necessary for such development are available and free.

5 Conclusions

This essay has approached the advantages of open source software from several angles. It has looked at the major issues surrounding the use of Open Source Software both in the classroom as well as the student's computer. However, the essay highlights only the positive aspects of OSS and hardly identifies any limitations or inadequacies. For example, OSS can be detrimental to students' ambitions of becoming software developers, as anyone with sufficient patience and determination can learn to do so himself. Second, OSS comes with absolutely no guarantees and victims cannot hold anyone responsible for any damages incurred. In this essay, I have deliberately steered away from discussing the "viral" nature of open source licenses, such as the GNU GPL. Another obstacle is that OSS can come with a rather steep learning curve since software sophistication is rarely obscured and requires the user to have a general understanding of the behind-the-scene workings of the application. Nevertheless, case histories have proven that OSS is extremely cost-effective, secure and delivers a more solid performance as compared to their proprietary counter-parts.

From the standpoint of a computer science curriculum, learning on an open source platform is definitely advantageous, economic considerations aside. When looked at from a pedagogical point of view, students learn to share their knowledge by using OSS. Software piracy is abolished and students are taught to respect an individual's intellectual rights by properly crediting them for their efforts. Use of proprietary systems in schools can hinder software development and may force students into belief that the development of anything useful must always come attached with a hefty price-tag.

Slowly but steadily, educators are reaching their goal of teaching students to *share and share alike* — thanks to the rapidly spreading open source phenomenon. Free and Open Source Software is just one contributing factor towards realizing this mammoth goal of equality of opportunity throughout the world. Thus in today's prevailing educational system, Open Source Software is definitely a *much* better way to simplify IT literacy in classrooms.

Appendices

A Walking What You're Talking: Colophon

As a tribute to the open source endeavor, only open source software has been used during the production and publication of this paper. The following free softwares and applications have been utilized:

1. The Linux Kernel http://www.kernel.org.
2. The Debian Operating System http://www.debian.org.
3. The Emacs Display Editor http://www.gnu.org/software/emacs/emacs.html.
4. The Subversion Version Control System http://subversion.tigris.org.
5. The Samba[12] File Server http://www.samba.org.
6. The LaTeX Document Preparation System http://www.latex-project.org.
7. PDFLatex for PDF Generation http://www.tug.org/applications/pdftex.

"And when all this combines and gets together, we land at an inevitable place of interest – The Open Souce Phenomenon." [OSDevCVS]

B List of Case Histories

1. http://fsedu.org/fsedu.pl?ForStudentsInstallGnuLinux
2. Yorktown High School http://www.linuxjournal.com/article/3403
3. Linux in Nicargua
   (a) http://www.linuxgazette.com/node/view/9234
   (b) http://www.linuxjournal.com/article.php?sid=7699
   (c) http://www.linuxjournal.com/article.php?sid=7701
4. Implementing Linux in Manitoba
   (a) http://www.linuxjournal.com/article.php?sid=7418
   (b) http://www.linuxjournal.com/article.php?sid=7419
5. Linux and Open Source to Hawaii http://www.desktoplinux.com/news/NS6201542989.html
6. Penguin Enrolls in US Schools http://www.wired.com/news/school/0,1383,45862,00.html

[12] Samba is an OSS file server that can run on the Linux platform and work seamlessly with workstations running Windows.

References

[ForStudents] *Wiki: ForStudents*. http://fsedu.org/fsedu.pl?ForStudents. Retrieved Sep. 19, 2004.

[FossPrimer] Tong, Tan Wooi. *Free/Open Source Software in Education*. Kuala Lumpur: UNDP-APDIP, 2004.

[GaryG] Glasscock, Gary. *Educators — Use money Wisely. Use Linux!*. http://osviews.com/modules.php?op=modload&name=News&file=article&sid=3101. Retrieved Feb. 4, 2005.

[Kaivo] Murphy, Gary. *Elementary Linux: Penguins in the School*. http://www.kaivo.org/Kaivo/Ampari/T_kolme/?page=C_elementaryLinuxPenguinsInSchool. Retrieved Feb. 4, 2005.

[MysqlReas] Reasoning Inc. Whitepaper. *How Open Source and Commercial Software Compare: MySQL 4.0.16*. 2003. Available from http://www.reasoning.com/downloads/mysql.html.

[Novell] *Why Linux is a Better Choice than Windows*. http://www.novell.com/linux/truth/better_choice.html. Retrieved Feb. 4, 2005.

[OSDevCVS] Fogel, Karl. Bar, Moshe. *Open Source Development with CVS, 3rd Edition*. Arizona: Paraglyph, 2003.

[SchoolDisc] Noronha, Frederick. *Links between FLOSS and the 'digital divide'*. http://archives.seul.org/schoolforge/discuss/Aug-2004/msg00027.html. Retrieved Sep. 12, 2004.

[Vessels01] Vessels, Terry. *Why should open source software be used in schools?* http://edge-op.org/grouch/schools.html

[WhySchool] Stallman, Richard. *Why schools should use exclusively free software*. http://www.gnu.org/philosophy/schools.html. Retrieved Sep. 19, 2004.

[YorkTown] Maurer, Justin. *Linux at Yorktown High School*. http://www.linuxjournal.com/article/3403. Retrieved Feb. 5, 2005.