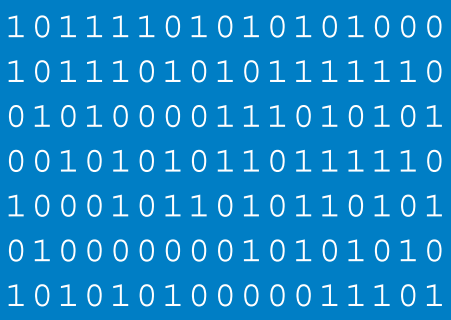
International
Open
Source
Network

An Initiative of the
UNDP's Asia-Pacific
Development Information
Programme

Free/Open Source Software

A General Introduction

Kenneth Wong and Phet Sayo



Asia-Pacific Development Information Programme
e-Primers for the Free/Open Source Software series

Free/Open Source Software

A General Introduction

Kenneth Wong and Phet Sayo

e-Primers for the Information Economy, Society and Policy

Published by
the United Nations Development Programme's
Asia-Pacific Development Information Programme (UNDP-APDIP)
Kuala Lumpur, Malaysia

www.apdip.net
Email: info@apdip.net

ISBN: 983-3094-00-7

Design, layout and cover illustrations by:

Rezonanze
www.rezonanze.com

PREFACE

The world of Information and Communications Technology (ICT) changes rapidly. New technologies and with them, new opportunities, come and go at an ever increasing speed. The Free and Open Source Software (FOSS) movement is one such development that is playing out before us today. It is many things - revolutionary development process, disruptive technology, ideological movement, new knowledge and standards, and more. It offers many opportunities for governmental, private sector, and educational institutions. Organizations, as well as developing nations, that take advantage of FOSS and implement them appropriately stand to gain much, while those that fail to take advantage of this opportunity may find their ICT development lagging behind that of comparable organizations.

This primer is the first in a series of primers focused on the FOSS movement. Intended for policy- and decision-makers in general, the primer gives an overview of the issues and technologies involved. Although geared more for developing countries, the points discussed and the resources contained in this primer are relevant to a broad range of individuals around the world.

The remaining primers in the series will focus in greater detail on particular aspects of the Free/Open Source Software movement, such as issues, technologies, and experiences in FOSS in government, education, network infrastructure, licensing and localization.

Finally, despite the prominence of “software” in its name, the Free/Open Source Software movement is based on three “Open” pillars - Open Source, Open Standards and Open Content. In the spirit of the movement, this primer is released as Open Content, allowing redistribution and usage under very broad conditions. Readers are encouraged to use, distribute, and contribute back to this resource as much as possible. The updated version of the primer will be available from the International Open Source Network website at:

http://www.iosn.net/downloads/foss_primer_current.pdf

The primers are brought to you by the International Open Source Network (IOSN), an initiative of the UNDP’s Asia-Pacific Development Information Programme (APDIP). We would like to thank all those who have been involved in creating this primer, including the researchers, peer reviewers and production team. In particular, we would like to thank APDIP and the International Development Research Centre (IDRC) of Canada for their generous financial support, without which this primer would never have been written.

It is our hope that this document will become a valuable resource for many in the years to come.

[1] Please refer to Annex III for full details of licensing.

INTRODUCTION

What is Free/Open Source Software?

"Briefly, OSS/FS programs are programs whose licenses give users the freedom to run the program for any purpose, to study and modify the program, and to redistribute copies of either the original or modified program (without having to pay royalties to previous developers)." *David Wheeler*[1]

Free and Open Source Software (FOSS) has become an international phenomenon, moving from relative obscurity to being the latest buzzword in a few short years. However, there is still a lack of understanding about what really constitutes FOSS and the ramifications of this new concept. To better explain this phenomenon, we will examine the philosophy and development methods behind FOSS.

The FOSS philosophy

There are two major philosophies in the FOSS world: the Free Software Foundation (FSF) philosophy and the Open Source Initiative (OSI) philosophy. We begin with the FSF philosophy, due to its historical precedence (see the following section, "A Brief History of FOSS") and pioneering position in the movement.

According to the FSF, free software is about protecting four user freedoms:

- The freedom to run a program, for any purpose;
- The freedom to study how a program works and adapt it to a person's needs. Access to the source code is a precondition for this;
- The freedom to redistribute copies so that you can help your neighbour; and
- The freedom to improve a program and release your improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.[2]

At the heart of FSF is the freedom to cooperate. Because non-free (free as in freedom, not price) software restricts the freedom to cooperate, FSF considers non-free software unethical. FSF is also opposed to software patents and additional restrictions to existing copyright laws. All of these restrict the four user freedoms listed above. For a more detailed explanation of why software needs to be free, please refer to the FSF explanation, "Why Software Should Be Free", found at http://www.fsf.org/philosophy/shouldbefree.html

The OSI philosophy is somewhat different:

> The **basic idea behind open source** is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.[3]

The OSI is focused on the technical values of making powerful, reliable software, and is more business-friendly than the FSF. It is less focused on the moral issues of Free Software and more on the practical advantages of the FOSS distributed development method.

While the fundamental philosophy of the two movements are different, both FSF and OSI share the same space and cooperate on practical grounds like software development, efforts against proprietary software, software patents, and the like. As Richard Stallman says, the Free Software Movement and the Open Source Movement are two political parties in the same community.

The FOSS development method

The FOSS development model is unique and became possible only with the advent of the Internet and the communication boom caused by it. The cathedral and bazaar analogies[4] are used to contrast the FOSS development model with traditional software development methods.

Traditional software development is likened to the way cathedrals were built in ancient times. Small groups of skilled artisans carefully planned out the design in isolation and everything was built in a single effort. Once built, the cathedrals were complete and little further modification was made. Software was traditionally built in a similar fashion. Groups of programmers worked in isolation, with careful planning and management, until their work was completed and the program released to the world. Once released, the program was considered finished and limited work was subsequently done on it.

In contrast, FOSS development is more akin to a bazaar, which grows organically. Initial traders come, establish their structures, and begin business. Later traders come and establish their own structures, and the bazaar grows in what appears to be a very chaotic fashion. Traders are concerned primarily with building a minimally functional structure so that they can begin trading. Later additions are added as circumstances dictate. Likewise, FOSS development starts off highly unstructured. Developers release early minimally functional code to the general public and then modify their programs based on feedback. Other developers may come along and modify or build upon the existing code. Over time, an entire operating system and suite of applications develops and evolves continuously.

The bazaar method of development has been proven over time to have several advantages:

1) **Reduced duplication of effort**

By releasing programs early and granting users the right to modify and redistribute the source code, FOSS developers reuse the work produced by compatriots. The economies of scale can be enormous. Instead of five software developers in 10 companies writing a single networking application, there is the potential for the combined efforts of 50 developers. The reduced duplication of effort allows FOSS development to scale to massive, unheard of levels involving thousands of developers around the world.

2) **Building upon the work of others**
With the availability of existing source code to build on, development times are reduced. Many FOSS projects rely on software built by other projects to supply needed functionality. For example, instead of writing their own cryptographic code, the Apache web server project uses the OpenSSL project's implementation, thereby saving thousands of hours of coding and testing. Even in cases where source code cannot be directly integrated, the availability of existing source code allows developers to learn how another project has solved a similar problem.

3) **Better quality control**
"Given enough eyeballs, all bugs are shallow"[5] is an oft-cited quotation in the FOSS world. It means with enough qualified developers using the application and examining the source code, errors are spotted and fixed faster. Proprietary applications may accept error reports but because their users are denied access to the source code, users are limited to reporting symptoms. FOSS developers often find that users with access to the source code not only report problems but also pinpoint the exact cause and, in some cases, supply the fixes. This greatly reduces development and quality control time.

4) **Reduced maintenance costs**
Maintenance of any software package can often equal or exceed the cost of initial software development.[6] When a single organization has to maintain software, this can be an extremely expensive task. However, with the FOSS development model, maintenance costs can be shared among the thousands of potential users of a software application, reducing per-organization costs. Likewise, enhancements can be made by the organization/individual with the best expertise in the matter, which results in a more efficient use of resources.

What is the history of FOSS?

"The free/open source software movement began in the "hacker" culture of U.S. computer science laboratories (Stanford, Berkeley, Carnegie Mellon, and MIT) in the 1960's and 1970's.

The community of programmers was small, and close-knit. Code passed back and forth between the members of the community—if you made an improvement you were expected to submit your code to the community of developers. To withhold code was considered gauché—after all, you benefited from the work of your friends, you should return the favor."

A Brief History of Free/Open Source Software Movement[7]

The FOSS movement dates back to almost the very beginning of the computer industry, although it was not then formally defined or conceptualized. It was only in the late 1970s and early 1980s that the sharing of software began to really come in conflict with proprietary software. One of the earlier references to proprietary software was made by William H. Gates III in his now-famous "An Open Letter to Hobbyists."[8] In this letter, dated 3rd February 1976, he rails against the prevailing culture of software sharing:

> *Why is this? As the majority of hobbyists must be aware, most of you steal your software. Hardware must be paid for, but software is something to share. Who cares if the people who worked on it get paid?*

Proprietary software would gain momentum over the years. At the pioneering MIT Artificial Intelligence Lab in the early 1980s, a company called Symbolics was formed and took what was freely available code (the LISP programming language) and made it proprietary. In the process, it wiped out the software-sharing culture of the MIT lab at the time[9]. This destruction, however, would eventually result in the creation of the FSF and the FOSS culture today.

Richard Stallman, one of the MIT lab members at the time, was appalled at the turn of events. It would shape his view of proprietary software and instill in him the resolve to create a free operating system. The GNU (recursive acronym for GNU is Not Unix) project was born in January 1984 and over the next decade, it created a variety of critical tools that formed a portion of the operating system. The FSF was created a year later to promote Free Software and the GNU project. However, up until 1991, the GNU project had yet to produce a totally free software system due to a missing critical piece: the kernel.

The kernel is the heart of the operating system. In 1991, Linus Torvalds, who at the time was a second year graduate student at the University of Helsinki, wrote and distributed a Unix-like kernel. In the manner of FOSS development, it was distributed widely, improved upon and soon adapted to become the core of the GNU/Linux operating system.

There were other FOSS projects in progress at the time, including BIND, Perl and the BSD operating systems. All of these projects eventually ended up merging or cross-pollinating.

The GNU/Linux operating system would continue to grow steadily in features and capabilities. In 1997 Linux exploded into the press limelight, with International Data Corp (IDC) noting that GNU/Linux already owned 25 percent of the server market[10] and was growing at an annual compound growth rate of 25 percent.

In 1998, in response to Netscape's release of its Netscape Navigator code as FOSS, a group of FOSS developers came together and the label "Open Source" was created. This led to the formation of the Open Source Initiative and the Open Source Definition. The primary purpose of this initiative was to get the corporate world to pay attention to the FOSS development process and steer a path away from the "confrontational" attitude of the Free Software movement[11].

In 1999, the massively successful IPO of GNU/Linux distributor Red Hat gave it a market capitalization of US$4.8 billion. Other successful IPOs that year were VA Linux (US$ 7 billion), Cobalt Networks ($3.1 billion) and Andover.net ($712 million)[12]. As the poster child of FOSS, GNU/Linux's success meant that FOSS had truly arrived.

WHY FOSS?

"Open-source software has been called many things: a movement, a fad, a virus, a Communist conspiracy, even the heart and soul of the Internet. But one point is often overlooked: Open-source software is also a highly effective vehicle for the transfer of wealth from the industrialized world to developing countries."

Andrew Leonard[13]

Is FOSS free?

The popular myth surrounding Free/Open Source Software is that it is always "free"—that is, "free of charge." To a certain degree this is true. No true FOSS application charges a licensing fee for usage. Most FOSS distributions (Red Hat, SuSE, Debian, etc.) can be obtained at no charge off the Internet. On a licensing cost basis, FOSS applications are almost always cheaper than proprietary software.

However, licensing costs are not the only costs of a software package or infrastructure. It is also necessary to consider personnel costs, hardware requirements, opportunity costs and training costs. Often referred to as the Total Cost of Ownership (TCO), these costs give the clearest picture of the savings from using FOSS[2].

How large are the savings from FOSS?

There have been recent reports about the tremendous savings from FOSS, most noticeably from giant corporations that have migrated their internal systems to GNU/Linux. Intel reportedly saved US$200 million from a move to GNU/Linux from Unix, and Amazon reported a savings of US$17 million[14] from switching their servers to GNU/Linux. Major financial institutions such as Credit Suisse First Boston, Morgan Stanley, Goldman Sachs and Charles Schwab are moving a significant portion of their infrastructure to FOSS systems to reap these cost savings[15].

There are few TCO studies showing the total cost of running FOSS systems versus proprietary systems. These studies analyze multiple cost factors other than software licensing costs, including maintenance, personnel and opportunity costs from service disruptions. Several have been very positive towards FOSS:

- A TCO study performed by the Robert Frances Group showed that GNU/Linux costs roughly 40 percent of Microsoft Windows and as low as 14 percent of Sun Microsystem's Solaris[16].

- NetProject reported that the TCO of GNU/Linux was 35 percent of Microsoft Window's TCO[17]. Even more interesting was that the savings was due not just to licensing costs but also to various other costs, including reduction in the number of support staff and software updates that results from using GNU/Linux.

- Gartner reported that using GNU/Linux in a "locked" configuration resulted in a roughly 15 percent lower TCO compared to Windows XP[18].

---

[2] There is some argument that a better measure would be Return On Investment (ROI). However, there are very few studies on the ROI of FOSS systems and it is just as difficult to measure as TCO, if not more. One article on ROI vs. TCO can be found at: http://www.infoworld.com/infoworld/article/03/08/29/34FElinux_1.html

Merrill Lynch, a major financial management company, recently reported that using GNU/Linux could reduce costs dramatically. The unusual part of their TCO study was that the largest costs savings was not from software licensing costs but from personnel and hardware costs[19].

Direct Cost Savings – An Example

Cybersource[20] of Australia has done an analysis of FOSS savings based on a comparison between Microsoft products and FOSS-based software that provide similar functionalities. The study, "Linux vs. Windows: The Bottom Line", looked at potential savings for three hypothetical companies (A: 50 users; B: 100 users; and C: 250 users). All numbers are in US dollars:

	Microsoft Solution	Linux/FOSS Solution	Savings
Company A: 50 Users	$87,988	$80	$87,908
Company B: 100 Users	$136,734	$80	$136,654
Company C: 250 Users	$282,974	$80	$282,894

Note: The savings achieved from implementing the FOSS solution instead of the Microsoft solution actually increases with the number of users—the bigger the outfit, the greater the savings. The financial incentives for migrating to FOSS increase with the size of the organization.

The Cybersource study is straightforward, comparing nothing more than the costs of software packages. The following two tables list the prices of two software solutions, Microsoft and FOSS, for a company of 50 users.

Microsoft Solution Software Cost		
Software	Copies	Cost
Norton Antivirus 2002	50	$2,498
MS Internet Information Server	2	$0
MS Windows 2000 Advanced Server	5	$19,995
MS Commerce Server	1	$12,333
MS ISA Standard Server 2000	1	$1,499
MS SQL Server 2000	1	$4,999
MS Exchange Standard Server 2000	1	$1,299
Windows XP Professional	50	$14,950
MS Visual Studio 6.0	3	$3,237
MS Office Standard	50	$23,950
Adobe Photoshop 6	2	$1,218
Additional Client Access Licenses	30	$2,010
Total		$87,988

FOSS Solution Software Cost		
Software	Copies	Cost
Linux Distribution (eg Red Hat 9.0)	1	$80
Apache (Web server)		$0
Squid (Proxy server)		$0
PostgreSQL (Database)		$0
iptables (Firewall)		$0
Sendmail/Postfix (Mail servers)		$0
KDevelop (IDE)		$0
GIMP (Graphics)		$0
Open Office (Productivity suite)		$0
OSCommerce (e-Commerce suite)		$0
Total		**$80**

Note: The cost of the GNU/Linux software solution remains fixed even when the number of users increases. This is because the licensing for GNU/Linux is not limited, whereas there are additional costs per user in licensing Microsoft and other proprietary software.

Public sector organizations often have far more users, which means even more dramatic savings. For example, the government of Sweden has identified savings of $1 billion a year while the government of Denmark has identified savings of between $480 million to $730 million[21].

What are the benefits of using FOSS?

Besides the low cost of FOSS, there are many other reasons why public/private organizations are aggressively adopting FOSS. These include:

- Security
- Reliability/Stability
- Open standards and vendor independence
- Reduced reliance on imports
- Developing local software capacity
- Piracy, IPR, and WTO
- Localization

Of particular importance to governments are the last four points as they are government-specific. Corporations and end users usually do not deal with these issues.

Security

While there is no perfectly secure operating system or platform, factors such as development method, program architecture and target market can greatly affect the security of a system and consequently make it easier or more difficult to breach. There are some indications that FOSS systems are superior to proprietary systems in this respect:

1. The Gartner Group recommends that businesses switch from Microsoft Internet Information Server (IIS) to Apache or another web server, due to IIS's poor security track record. The Gartner Group noted that by July 2001 US enterprises had spent US$1.2 billion simply fixing Code Red (IIS-related) vulnerabilities[22].

2. "Hacker Insurance" issued by J.S. Wurzler Underwriting Managers costs five to 15 percent more if Windows is used instead of GNU/Linux or Unix systems. Walter Kopf, senior vice president of underwriting at J.S. Wurzler Underwriting Managers, says, "We have found out that the possibility for loss is greater using the NT system."[23]

The security aspect has already encouraged many public organizations to switch or to consider switching to FOSS solutions. The French Customs and Indirect Taxation authority migrated to Red Hat Linux 6.2 largely because of security concerns[24].

Three reasons are often cited for FOSS's better security record:

- **Availability of source code**: The availability of the source code for FOSS systems has made it easier for developers and users to discover and fix vulnerabilities, often before a flaw can be exploited. Many of the vulnerabilities of FOSS listed in Bugtraq were errors discovered during periodic audits and fixed without any known exploits. FOSS systems normally employ proactive rather than reactive audits.

- **Security focus, instead of user-friendliness**: FOSS can be said to run a large part of the Internet[25] and is therefore more focused on robustness and functionality, rather than ease of use. Before features are added to any major FOSS application, its security considerations are considered and the feature is added only if it is determined not to compromise system security.

- **Roots**: FOSS systems are mostly based on the multi-user, network-ready Unix model. Because of this, they come with a strong security and permission structure. Such models were critical when multiple users shared a single powerful server—that is, if security was weak, a single user could crash the server, steal private data from other users or deprive other users of computing resources. Consequently, vulnerabilities in most applications result in only a limited security breach.

Reliability/Stability

FOSS systems are well known for their stability and reliability. There are many anecdotal stories of FOSS servers functioning for years without requiring maintenance. However, quantitative studies are more difficult to come by. Here are two of the studies conducted to date:

- **In 1999 Zdnet ran a 10-month reliability test** between Red Hat Linux, Caldera Systems OpenLinux and Microsoft's Windows NT Server 4.0 with Service Pack 3. All three ran on identical hardware systems and performed printing, web serving and file serving functions. The result was that NT crashed once every six weeks but none of the FOSS systems crashed at all during the entire 10 months[26].

- **A stress test using random testing** stressed seven commercial systems and the GNU/Linux system in 1995. Random characters were fed to these systems, to simulate garbage from bad data or users. The result was that the commercial systems had an average failure rate of 23 percent while Linux as a whole failed nine percent of the time. GNU utilities (software produced by the FSF under the GNU project) failed only six percent of the time. A follow-up study years later found that the flaws identified by the study were all fixed in the FOSS system, but were generally untouched in proprietary software[27].

Open standards and vendor independence

Open standards give users, whether individuals or governments, flexibility and the freedom to change between different software packages, platforms and vendors. Proprietary, secret standards lock users into using software only from one vendor and leave them at the mercy of the vendor at a later stage, when all their data is in the vendor's proprietary format and the costs of converting them to an open standard is prohibitive.

The authors of the paper "Free/Libre and Open Source Software: Survey and Study" produced by the International Institute of Infonomics in the Netherlands also argue against use of proprietary software in government. They say:

> …Consequently one major argument against the implementation of proprietary software in the public sector is the subsequent dependency on proprietary software vendors. Whenever the proprietary standards are established the necessity to follow them is given. Even in an open tender acquisition system, this requirement for compatibility with proprietary standards makes the system biased towards specific software vendors, perpetuating a dependency.

Another advantage of FOSS is that they almost always use open standards. This is due to two primary reasons:

- **Availability of the source code**: With the source code, it is always possible to reverse-engineer and document the standard used by an application. All possible variations are plainly visible in the source code, making hiding a proprietary standard in FOSS systems impossible. Proprietary software, however, are much harder to reverse-engineer and in some cases are deliberately obfuscated.

- **Active standards compliance**: When established standards exist, such as HyperText Markup Language (HTML), which controls how web pages are displayed, FOSS projects actively work to follow the standards faithfully. The Mozilla web browser, a FOSS effort, is fully compliant with many standards from the World Wide Web Consortium (W3C). Webstandards.org notes that Mozilla is one of the most compliant browsers available today[28]. Compliance with standards is due to the FOSS development culture, where sharing and working together with other applications are the norm. It is also much easier to work with a globally dispersed group of developers when there is a published standard to adhere to.

Using FOSS systems as a means of gaining vendor independence has been raised in several areas. A report to the UK Government concludes that "the existence of an OSS reference implementation of a data standard has often accelerated the adoption of such standards, and recommends that the Government consider selective sponsorship of OSS reference implementations."[29]

Reduced reliance on imports

A major incentive for developing countries to adopt FOSS systems is the enormous cost of proprietary software licenses. Because virtually all proprietary software in developing countries is imported, their purchase consumes precious hard currency and foreign reserves. These reserves could be better spent on other development goals.

The European study, "Free/Libre and Open Source Software: Survey and Study", also notes that, "The costs of this more service-oriented model of open source are then also normally spent within the economy of the governmental organization, and not necessary to large multinational companies. This has a positive feedback regarding employment, local investment base, tax revenue, etc."[30]

Developing local software capacity

It has been noted that there is a positive correlation between the growth of a FOSS developer base and the innovative capacities (software) of an economy. A report from the International Institute of Infonomics lists three reasons for this[31]:

- **Low barriers to entry**: FOSS, which encourages free modification and redistribution, is easy to obtain, use and learn from. Proprietary software tends to be much more restrictive, not just in the limited availability of source code, but due to licensing, patent and copyright limitations. FOSS allows developers to build on existing knowledge and pre-built components, much like basic research.

- **FOSS as an excellent training system**: The open and collaborative nature of FOSS allows a student to examine and experiment with software concepts at virtually no direct cost to society. Likewise, a student can tap into the global collaborative FOSS development network that includes massive archives of technical information and interactive discussion tools.

- **FOSS as a source of standards**: FOSS often becomes a *de facto* standard by virtue of its dominance in a particular sector of an industry. By being involved in setting the standards in a particular FOSS application, a region can ensure that the standard produced takes into account regional needs and cultural considerations.

The FOSS developmental approach greatly facilitates not only innovation but also its dissemination. A Microsoft internal memo noted, "Research/teaching projects on top of Linux are easily 'disseminated' due to the wide availability of Linux source. In particular, this often means that new research ideas are first implemented and available on Linux before they are available / incorporated into other platforms."[32]

Piracy, IPR, and the WTO

Software piracy is a problem in almost every country around the world. The Business Software Alliance estimates that software piracy in 2002 alone cost US$13.08 billion. Even in developed nations where software is affordable in theory, piracy rates were as high as 24 percent in the United States and 35 percent in Europe. Piracy rates in developing countries, where lower incomes make software far more expensive, are upwards of 90 percent[33].

Software piracy and lax laws against it can and does hurt a country in many ways. A country with poor protection for Intellectual Property Rights (IPR) is not as attractive to foreign investors. Membership in the World Trade Organization (WTO) and access to its benefits are strongly affected by the level of protection given to IPR in a country. Finally, a culture of software piracy hurts local software development, as there is less incentive for local software developers to create a local product.

Localization

"Localization involves taking a product and making it linguistically and culturally appropriate to the target locale (country/region and language) where it will be used and sold."

Localisation Industry Standards Association[34]

Localization is one of the areas where FOSS shines because of its open nature. Users are able to modify FOSS to suit the unique requirements of a particular cultural region, regardless of economic size. All that is necessary is the technical capability within a small number of individuals to create a minimally localized version of any FOSS. While the construction of a completely localized software platform is no small feat, it is at least possible. Microsoft's decision in 1998 against producing an Icelandic version of Windows 98[35] would have had serious implications if it were not for the emergence of FOSS alternatives.

Most initial FOSS initiatives in the Asia-Pacific region have dealt with localizing FOSS. More details on localization can be found in the "Localization and Internationalization" section of this primer.

What are the shortcomings of FOSS?

For all the benefits FOSS brings, it is not suitable for every situation. There are areas where FOSS needs improvement.

Lack of business applications

While there are many FOSS projects out there today, there are still many areas that lack a full-featured product, especially in the business world. The recent porting of Enterprise Resource Planning platforms such as SAP and Peoplesoft[36] have helped cover the high-end application market, but the Small and Medium Enterprise (SME) market is still poorly served. Basic, polished accounting applications such as Quickbooks, Peachtree or Great Plains do not have FOSS equivalents at this time.

This problem has come about in part due to the scarcity of people competent in both technical and business subjects. Technical developers who encountered problems and wrote software to "scratch an itch" started most of the existing FOSS projects today. These projects are usually fairly technical in nature, such as the creation of web servers, programming languages/environments and networking tools. It is rare for a software developer to encounter accounting problems, for example, and have the business knowledge to create a technical solution.

Interoperability with proprietary systems

FOSS systems, especially on the desktop, are not completely compatible with proprietary systems. For organizations that have already invested massive amounts of capital into proprietary applications and data storage formats, attempting to integrate FOSS solutions can prove to be prohibitively expensive. Changing proprietary standards, which is often aimed at preventing the integration of alternate solutions, exacerbates this problem.

In time, as organizations shift from proprietary to open standards, this problem should be reduced.

Documentation and “polish”

Established FOSS lacks the extensive documentation and user-friendliness found in commercial software[37]. The primary focus of early FOSS developers was functionality. Creating a program that worked well was far more important than ease of use.

Besides the dearth of high-quality documentation, there are also user interface issues with FOSS Graphical User Interfaces (GUI). Because the GUI element in most FOSS systems is not a single element but a collection of different projects glued together, the behaviour of the GUI elements differ greatly. Command-to-save data differ from one program to another, quite unlike proprietary desktop operating systems such as the Mac OS X or Microsoft Windows. Cutting and pasting between different programs can be wildly inconsistent or even impossible. While there is significant ongoing work to unify the desktop, the desktop is likely to remain inconsistent for some time to come.

FOSS Success Stories

What are governments doing with FOSS?

Various governments around the world have begun to take notice of FOSS and launch initiatives to reap the benefits it poses. Many of these initiatives are still in the early stages, but there is a significant trend towards incorporating FOSS into procurement and development policies. Besides the large numbers of reports and white papers recommending FOSS solutions, there are reportedly about 70 proposed laws mandating or encouraging FOSS around the world[38]. A few are at the national level while most are at much lower (state or city) levels. The following are highlights of some of the more noteworthy efforts from around the world.

Europe

Besides being home to a significant number of FOSS developers, Europe is also an area with strong government interests in FOSS. In particular, the European Union, Germany, France and the United Kingdom are leading the way in FOSS development.

European Union

The European Union has produced a working paper that stresses Open Standards and encourages Free/Open Source Software where appropriate. The paper, titled “Linking Up Europe: the Importance of Interoperability for E-government Services”, focuses on connecting the different national e-government systems together. It is also critical of past developments that “resulted in closed, vertical, un-scalable and frequently proprietary information systems”[39]. This paper was produced as part of the eEurope initiative. The European Union is also creating FOSS competency centers and funding the development of certain health-related applications[40].

Germany

Germany has many different initiatives underway. The German Bundestag uses Linux on its 150 servers[41], while the city of Munich plans to switch over 14,000 desktops to Linux, despite Microsoft’s last minute price cuts[42]. The police force is also transitioning 11,000 clients to Linux. It is interesting to note that price is not always the factor cited for the large number of migrations to Linux. Germany’s Interior Minister, Otto Schilly, said, “We are raising computer security by avoiding a monoculture, and we are lowering dependence on a single supplier”[43]. The German Parliament decided in 2001 that FOSS products should be used wherever costs could be decreased by their usage[44]. The Ministry of Finance has an Apache/Linux-based intranet system that supports 15,000 users[45].

France

The officially sanctioned Agency for Technologies of Information and Communication in Administration (ATICA) counts as part of its mission, "to encourage administrations to use free software and open standards."[46] The Authority for Customs and Indirect Taxation has also migrated to Linux, citing security reasons[47]. The French agency for e-government has made open standards mandatory for all public administrations to guarantee full interoperability[48].

United Kingdom

The United Kingdom (UK) has only recently started formulating policy regarding FOSS procurement but the policies that have been produced to date have been favorable towards FOSS. The UK is primarily interested in avoiding the proprietary lock-in problem and has produced a policy to "only use products for interoperability that support open standards and specifications in all future IT developments"[49]. One of the most active proponents of FOSS is the National Health Service, in part due to the insolvency of a proprietary software vendor that forced hospitals to migrate to Linux[50].

Finland

It is only fitting that the homeland of Linux's creator is also active in the FOSS arena. One of the more public initiatives is the gradual migration of the city of Turku to Linux and Open Office. All desktop systems will be migrated, with the first pilot project of 200 computers in progress.

The Finnish State Administration is also reportedly considering replacing all of its desktops with Linux, affecting as many as 147,000 computers[51].

Americas

United States

Although there is no official FOSS policy in the US federal government, there have been a number of attempts to pass pro-FOSS legislation at the state level. The states include California[52], Texas[53] and Oregon[54]. To date none of the bills have been passed but the momentum is not expected to slow down anytime soon.

Finding detailed information about FOSS usage in the US government is difficult, but a survey from MITRE Corporation shows that the US Department of Defense used a total of 115 different FOSS applications, with 251 examples of their use[55]. Additionally, multiple reports recommending the use of FOSS in the US Federal government have appeared, including one by the (US) President's Information Technology Advisory Committee (PITAC) which recommended that the US "Federal

government should encourage the development of open source software as an alternate path for software development for high end computing"[56].

A few smaller public institutions have shifted over to FOSS platforms. The most well known is the City of Largo, Florida. They have transitioned 900 city employees over to GNU/Linux, saving over $1 million in both hardware and software costs[57]. The City of Largo did more than just use Linux as an operating system; they changed their entire computing model to a thin-client system (something which Microsoft Windows currently cannot do) and as a result saved a huge amount in hardware costs. The city of Houston, Texas has also shifted systems over to a FOSS platform after Microsoft demanded that the city change to a $12 million dollar, multi-year licensing plan[58].

Peru

Peru is well known within the FOSS community for being one of the first countries in the world to have introduced legislation favoring FOSS in government procurement. The ensuing publicity, Microsoft's response and Dr. Edgar David Villanueva Nuñez's (legislation sponsor) powerful reply would occupy IT news media for quite a while. Among the choice quotes from Dr. Nunez's response are:

> *To guarantee the free access of citizens to public information, it is indespensable (sic) that the encoding of data is not tied to a single provider. The use of standard and open formats gives a guarantee of this free access, if necessary through the creation of compatible free software.*
>
> *To guarantee the permanence of public data, it is necessary that the usability and maintenance of the software does not depend on the goodwill of the suppliers, or on the monopoly conditions imposed by them. For this reason the State needs systems the development of which can be guaranteed due to the availability of the source code.*[59]

Although the bill remains stalled (after a US$550,000 donation by Microsoft and pressure from the US Embassy), the reasoning behind the Peruvian bill is something all governments concerned with public data should consider.

Brazil

The Brazilian government plans to migrate 80 percent of all computers in state and state-owned institutions to Linux over the next three years. Pilot programs are already underway and a slow, gradual migration is planned. A "Chamber for the Implementation of Software Libre" has been set up by the government to smooth this transition. Among the reasons cited for this move are lower costs, increased production of local software and "democratiz(ing) access to knowledge"[60].

Asia Pacific

Regional

The Asia-Pacific region, with its mix of developed and developing nations, is a very active region in FOSS usage and development. Three of the major nations in the area—Japan, South Korea and China—have recently announced an initiative to create a FOSS operating system adapted to their regional needs[61].

China

China is set to be a major stronghold for FOSS over the next few years. FOSS usage in the country is growing rapidly, with Linux growth alone expected to be 175 percent in 2003[62].

A primary driver of this massive growth is the Chinese government itself. One of its goals is to create both a hardware and software industry that "will not fall into the foreign intellectual property rights trap"[63]. Rather than becoming dependent on foreign hardware and software vendors, China is trying to develop its local technology industry, and FOSS fits well into its software needs. Recently, the Chinese government announced that government departments would be barred from purchasing foreign produced software, effectively eliminating most proprietary software vendors such as Microsoft and Oracle[64].

Beyond sponsoring the creation of localized versions of GNU/Linux (RedFlag Linux, Blue Point Linux, etc.), China is also implementing FOSS solutions at the government level. The city of Beijing has had a project to convert 2,000 desktops to Red Flag Linux since 2001[65]. China Post Office signed a deal with IBM to run GNU/Linux at 1,200 branch offices[66]. While these projects cover only a small fraction of the Chinese government, they also serve as capacity-building projects for future transitions.

The Yangfan and Qihang projects launched in January 2002 are part of the Beijing Municipal Government's computerization project. The goal of these projects is to produce a localized GNU/Linux with functionality, consistency and ease of use matching that of Microsoft Windows 98. Over 150 engineers have completed their initial target of a basic operating system, office suite, web browser and email client. The latest iteration of the project includes font development and experimental transition of government applications to GNU/Linux[67].

China is also one of three countries (the other two are Japan and South Korea) that are forming a joint FOSS project that would cover the entire spectrum of software, from operating systems to middle ware and desktop applications[68].

India
While the Indian federal government currently has no official position on the FOSS/proprietary software issue[69], India represents a hotbed of FOSS development. There are many department level initiatives:

- The Central Excise Department has moved 1,000 desktops to Linux.
- The government supercomputer arm, the C-DAC, has moved over entirely to Linux[70].
- The Supreme Court has several pilot projects under way.

At the state level, there have been several FOSS initiatives. The most prominent is the Madhya Pradesh state government's plan to use Linux in its e-governance and Headstart programs, according to Chief Minister Digvijay Singh[71]. Red Hat has installed its version of Linux on over 6,000 desktops in schools[72], with more likely to come. The state of Kerala has also several initiatives underway, including e-government and educational initiatives.

Other state level initiatives have been announced, but little has been heard about these initiatives since Microsoft's much-publicized investment in 2002[73]

Taiwan
In 2003, Taiwan launched its "National Open Source Plan", a two-year plan to build a software industry that could replace all of the proprietary software on government and educational systems. The primary drivers for this plan are the existing dependence on a monopoly supplier and the expected cost savings. The National Computer Center is drafting the basic plan while the national education system will be switched to FOSS "to provide a diverse IT education environment and ensure the people's rights to the freedom of information."[74] Expected savings from the plan are about NT$2 billion for the government and NT$10 billion for the society as a whole.

Thailand
An article in the *Bangkok Post* on 23rd June 2003 reported that the Thailand ICT Ministry had set a target of five percent Linux installations on government systems by the end of 2003. A 10 million baht budget has been allocated. The ultimate goal is to have 50 percent of all government systems on Linux. No time frame has been set for the more ambitious target but pilot projects are already underway.

Thailand's low-cost PC program is also credited with forcing Microsoft to offer both the Windows XP operating system and Microsoft Office for a mere US$40, the lowest price available in the world at present (3rd quarter 2003)[75].

Malaysia

The government has expressed support for FOSS solutions since November 2001. In April 2002, the Association of Computer and Multimedia Industry of Malaysia (PIKOM) produced a paper suggesting that Malaysia "officially embrace OSS" in April 2002[76]. Initial deployment will start on servers and then gradually shift to desktops to minimize disruptions in operations.

Malaysia also launched in July 2002 Komnas, a low-cost computer based on FOSS[77]. Komnas carries a localized version of Linux, including office suite, web browser and various utilities.

Japan

Japan is considering moving its e-government projects over to FOSS platforms due to security problems in Microsoft Windows software[78]. Authorities are reportedly putting together a panel of experts to study FOSS deployment. In the meantime, the Japanese government is moving its entire payroll system over to a GNU/Linux platform. The switch is expected to cut operating costs by half, especially maintenance costs from hardware[79].

Other Regions

Africa

The South African government has a policy preferring FOSS systems unless there are compelling reasons otherwise[80]. Among the reasons cited for this preference is that with the traditional proprietary software model, South Africa ends up primarily being an importer of software, with little influence over how software is developed. It is hoped that using FOSS systems will change this.

Tanzania is also implementing FOSS systems in its government for cost reasons, while Uganda, Ghana and Zambia are also reportedly moving towards FOSS[81].

What are some successful FOSS projects?

While FOSS may seem a relatively new concept, it has actually been around since long before the Internet came into existence. FOSS has more than proven that it is ready for prime time, mission-critical usage. In some cases, it is the critical linchpin that makes the Internet possible. The following is a small sample of successful FOSS projects.

BIND (DNS Server)

Internet addresses such as yahoo.com or microsoft.com would not function if not for Domain Name Servers (DNS). These servers take these human-friendly names and convert them into the computer-friendly numeric addresses and vice-versa. Without these servers, users would have to memorize numbers such as 202.187.94.12 in order to use a website.

The Berkeley Internet Name Domain (BIND) server runs 95 percent of all DNS servers[82], including most of the DNS root servers. These servers hold the master record of all domain names on the Internet. BIND is a FOSS program licensed under a BSD-style license by the Internet Software Consortium.

Apache (Web Server)

Responsible for receiving and fulfilling requests from web browsers, the Apache web server is one of the foundations of the World Wide Web (WWW) as we know it today. Apache has been the number one web server since April 1996 and currently commands 62.53 percent of the total web server market[83] “. That is more than double the market share (27.17 percent) of its closest competitor, Microsoft’s IIS server.

These figures fluctuate monthly of course. The latest figures can be found at Netcraft’s Web Server Survey site, at: http://news.netcraft.com/archives/web_server_survey.html.

Sendmail (Email Server)

The Internet as we know it would not exist without email and once again, FOSS is one of the primary drivers. An email server’s (sometimes called a Mail Transport Agent or MTA) function is to deliver user email to its destination. Complex functionality, such as email forwarding and redirection, junk email rejection and routing, makes email servers rather complex systems. The problem of junk email (sometimes referred to as spam) makes security a critical feature, as spammers sending their unsolicited email to unsuspecting users would otherwise hijack an email server and render it useless to legitimate users.

A 2001 survey by D.J. Bernstein found that Unix Sendmail had the largest market share, at 42 percent of all email servers. This was larger than the share of its next two competitors combined, Microsoft Exchange and Unix qmail, with 18 percent and 17 percent, respectively[84]. Note that qmail is a Unix-based email server but is not FOSS as its licensing terms are too restrictive.

OpenSSH (Secure Network Administration Tool)

Because Internet traffic can pass through multiple networks when a user connects into a remote server, security is a major concern. The Secure Shell (SSH) protocol allows system administrators to control their servers from a distance, safe in the knowledge that it is almost impossible to intercept and decipher the information that they may be transmitting.

OpenSSH, a FOSS implementation of the SSH protocol, has grown from a mere five percent of the market in 2000 to 66.8 percent of the market in April 2002. OpenSSH came into existence as a result of a restrictive licensing change in the standard SSH implementation at that time.

Open Office (Office Productivity Suite)

While FOSS products have been strong on the server side, FOSS desktop applications are relatively new. Open Office, which is based on the source code of the formerly proprietary StarOffice, is a FOSS equivalent of Microsoft Office, with most of its features. It includes a full-featured word processor, spreadsheet and presentation software. One of the advantages for many considering the shift from a Windows desktop environment to Open Office is that it reads most Microsoft Office documents without problems. This makes the transition relatively painless and Open Office has been used in recent high profile switches from Windows to Linux. While it does not have a very large market share as yet, its usage is expected to grow dramatically over time as more organizations use this full-featured, low-cost application.

Linux

What is Linux?

Linux is the most frequently heard FOSS buzzword in the mass media today. However, because of its common usage, the term Linux has been used to refer to broader and broader definitions. It is important to understand the different definitions of Linux to be able to follow the discussions on FOSS.

Linux as the kernel

Linux was originally the name of the kernel created by Linus Torvalds. A kernel is the critical center point of an operating system that controls CPU usage, memory management and hardware devices. It also mediates communication between the different programs running within the operating system. There are other FOSS kernels, including the Mach kernel that is the core of some of the BSD distributions.

Kernels are to a certain extent interchangeable. Most FOSS applications will run on a Mach kernel, Linux kernel or even the experimental GNU Hurd kernel, without too much difficulty. However, the kernel type greatly influences performance and the hardware platforms that the FOSS system can run on. For instance, the less mature GNU Hurd kernel can run only on the x86 (PC) architecture. The Linux kernel has been ported to run on almost any computing architecture, including the Playstation 2[85], mainframes and embedded devices.

Linux as a distribution

The more common usage of Linux today refers to a Linux distribution, which includes far more than the kernel. The Linux distribution (sometimes referred to as the GNU/Linux distribution in recognition of the GNU Project's significant contribution) contains the Linux kernel at its heart and all the FOSS components required to produce full operating system functionality. This includes the system libraries, GUI, various databases, web servers, email utilities, and others. These same components are also often found on other FOSS and even on proprietary operating systems. For instance, XFree86 is the default GUI foundation in Linux and BSD. XFree86 is also used on proprietary Unix systems such as Solaris, HP-UX and IBM's AIX system.

Reports that say "Munich May Opt for Linux After All"[86] refer to the Linux distribution, including word processing, printing and email software. Even though the Linux kernel forms less than 0.25 percent (binary file size) of a Linux distribution, its functionality is critical enough to allow the entire distribution to be called Linux.

There is no single Linux distribution. While all distributions contain the Linux kernel at its heart, the FOSS applications included and the configurations supported vary. There are multiple commercial distributions, several freely available, and numerous customized distributions that are targeted to the unique needs of different users. While the FOSS contents of different Linux distributions are mostly identical,

they are optimized for different uses such as for high-end servers, user-friendly desktops or even embedded systems. Localized distributions at a minimum include the fonts, input methods and menu translations necessary to use the software in the regional language.

Is Linux FOSS?

The Linux kernel is FOSS, licensed under the GNU General Public License. However, different distributions of Linux contain different components, some of which are not FOSS. For example, the German SuSE Linux distribution includes the non-FOSS YaST (Yet another Setup Tool) installation program.

The Debian GNU/Linux[87] distribution is one of the few distributions that are committed to including only FOSS components (as defined by the Open Source Initiative) in its core distribution.

Where can one obtain Linux?

FOSS, in binary and source code format, is free and downloadable from the Internet. The Linux kernel itself can be downloaded from http://www.kernel.org and other applications from their respective websites. However, most users tend to obtain distributions of Linux. The following is a table of some of the most popular distributors of Linux:

Popular Linux Distributors

Debian	www.debian.org
Redhat	www.redhat.com
SuSe	www.suse.com
Mandrake	www.mandrakelinux.com
SlackWare	www.slackware.com
TurboLinux	www.turbolinux.com

The advantages of going with distributions of Linux are many. The single most important advantage of vendor Linux over “stock” Linux is that it saves users time:

- **Download time**: The Linux operating system and complementary software involve large files and long download times. A standard 56kbps modem would take at least 45 days to download a standard 3 CD distribution. Vendors also provide bundled software—browsers, server applications, office suite, etc.—that saves the users from the tedious work of hunting and downloading individual software packages.

- **Installation and compiling time**: Many FOSS packages are downloadable only as source code. Users are required to compile and install the software themselves, assuming they are competent enough to do so. On a slow computer, compilation of source code may take days or even weeks. Vendor distributions of Linux often come precompiled and packaged with an easy installation system that takes less than an hour to install on most modern systems.

- **Quality assurance**: Vendors typically perform extensive testing to ensure that all of the components work well together. Since FOSS projects are developed independently, there is always the chance that changes in one package have outdated another package. Vendors resolve these dependencies for the user, supplying an integrated package that works “out of the box”.

- **Learning time**: Vendors provide manuals and publish reference material (for sale) for their products, making Linux much easier for the average user to learn.

Intellectual Property Rights and Licensing

What are the licensing arrangements for FOSS?

FOSS is released under a variety of different licenses. There are two primary types of licenses and countless variants. The two main licenses are the GNU (recursive acronym for GNU's not Unix) General Public License and the BSD-style licenses. A more detailed listing of licenses can be found on the FSF's website at http://www.fsf.org/licenses/license-list.html.

The GNU General Public License (GPL)

This is designed to ensure that user freedoms under this license are protected in perpetuity. Users are allowed to do pretty much anything they want to a GPL program, including copying, distributing and modifying. The conditions of the license primarily affect the user when it is distributed to another user.

Among the key provisions of distributing GPL software are:

- The distributor of a GPL program must also make available the source code to the recipient.
- Any changes made to a GPL program by the distributor must also be licensed under the GPL.
- Distributors may not place any non-GPL restrictions upon the users they distribute the GPL program to.
- Recipients of GPL software are granted the same rights to copy, modify and distribute the software as the original distributor.

GPL software forms a significant majority of FOSS: as much as 73 percent of FOSS projects[88]. One of the main motivations for the usage of the GPL in FOSS is assurance that once something is released as FOSS, it will remain so permanently. It is not possible to add additional licensing to strip away a user's right to redistribute or modify the program. A commercial software company cannot take a GPL program, modify it and then sell it under a different, proprietary license.

The full text of the GPL can be found at http://www.fsf.org/licenses/gpl.html.

BSD-style Licenses

BSD-style (Berkeley System Distribution) licenses are so named because they are identical in spirit to the original license issued by the University of California, Berkeley. These are among the most permissive licenses possible, because they basically permit users to do anything they wish with the software as long as:

- Attribution is given to the original licensor by including the original copyright notice in source code files; and
- No attempt is made to sue or hold the original licensor liable for damages.

Earlier versions required the acknowledgement of the University of California, Berkeley (or whichever organization released the original software) in all promotional material but this requirement has been dropped in most recent versions.

A large number of FOSS projects, including several critical components, are licensed under BSD-style licenses. Examples include:

- **The Apache Web Server** – the #1 web server used on the Internet today[89]
- **The XFree86 Window System** - the foundation of almost all graphical user interfaces in FOSS systems
- **FreeBSD, NetBSD and OpenBSD** – all variants based on the original Berkeley System Distribution (BSD) version of Unix; all are widely used on the Internet, especially FreeBSD, which runs Yahoo and Microsoft's Hotmail services[90]

It is fairly easy to incorporate BSD-style licensed code into commercial applications. Even Microsoft has used some BSD code in the networking portions of its Windows code[91] in the past. Many companies include the Apache web server as part of their commercial software offerings. Unlike the GPL, BSD-style licenses do not require the distribution of source code, allowing a company to hide its modifications to the original code. Nor are companies required to grant users the right to view, modify or distribute the company's modifications to the code.

A more detailed listing of different software licenses can be found in Annex II.

Can FOSS be combined with proprietary software?

Combining FOSS with proprietary software is possible, depending on the manner of "combination" and on the specific license of the software. Of all the common FOSS licenses, the GNU GPL license is the one that requires the most care. It defines "combination" as follows:

> *Mere aggregation of two programs means putting them side by side on the same CD-ROM or hard disk. We use this term (…) where they are separate programs, not parts of a single program. In this case, if one of the programs is covered by the GPL, it has no effect on the other program. Combining two modules means connecting them together so that they form a single larger program. If either part is covered by the GPL, the whole combination must also be released under the GPL—if you can't, or won't, do that, you may not combine them.*[92]

In this case, if one uses a proprietary application in a FOSS operating system environment, the proprietary application is unaffected by the licensing of the FOSS system. An example of this is running an Oracle database on a GNU/Linux operating system.

An example of combining programs would be writing a GUI application using the Gnome application framework. The Gnome application framework speeds up the development of any GUI program by supplying functionality developers who would otherwise have to write from scratch. Gnome is licensed under the GPL. Because the completed application program (after a compiler has been through it) would contain source code from the Gnome application framework, the entire application would have to be licensed under the GPL.

Other licenses are usually far less strict. Writing the same application above using a BSD-style license would only require keeping the attributions to the original licensor within the source code. The matrix below highlights the differences when distributing software combined with GPL or BSD-style licensed software:

	GPL Licensed	BSD License
Must distribute original source code	Yes	No
Must distribute user-created source code	Yes	No
User-created source code must be GPL'ed	Yes	No

Localization and Internationalization

What is localization? What is internationalization?

According to the Localization Institute,

> **Localization** is the process of creating or adapting a product to a specific locale, i.e., to the language, cultural context, conventions and market requirements of a specific target market. With a properly localized product, a user can interact with this product using his/her own language and cultural conventions. It also means that all user-visible messages and all user documentation (printed and electronic) use the language and cultural conventions of the user. Finally, the properly localized product meets all regulatory and other requirements of the user's country/region.
>
> **Internationalization** is a way of designing and producing products that can be easily adapted to different locales. This requires extracting all language, country/region and culturally dependent elements from a product. In other words, the process of developing an application whose feature design and code design do not make assumptions based on a single locale, and whose source code simplifies the creation of different local editions of a program, is called internationalization.[93]

What is an example of localization and internationalization?

Localization and internationalization are often used interchangeably. The definitions provided above with reference to software development clearly show the distinction. In terms of FOSS development, an excellent example of both 'internationalization' and 'localization' is the Mozilla Project. Mozilla is the most well known and widely used of the FOSS web browsers available. Mozilla is internationalized because the community of developers behind the Mozilla Project have designed and developed their software to function in multiple locales. Mozilla is localized when local developers, using guidelines and localization toolkits provided by the Mozilla Project, modify or adapt the product to suit a particular locale. This modification often involves translating user interfaces, documentation and packaging, as well as changing and customizing features to match the usage patterns of that locale.

Internationalization and localization of Mozilla by anyone is possible because it is a FOSS project. The Mozilla source code is distributed under the Mozilla Public License (MPL), which is a license that is based on and approved by the Open Source Initiative. The Mozilla Project aims to serve the greater Internet community, which it recognizes as a global community made up of users belonging to a great array of language groups. One of the goals of the Mozilla Project is to "advocate the localization of mozilla.org products into any world language".

Fully localized versions of Mozilla cover 34 different languages. Localization efforts are still continuing for most of the other languages[94].

What are the methods of localizing GNU/Linux?

"The localisation of Linux to Indian languages can spark off a revolution that reaches down to the grassroots levels of the country,"[95]

Prof. Venkatesh Hariharan

For each different locale or country, the challenges involved in localizing GNU/Linux vary. Some locales may find that localization requires minimal effort. Other locales may find that localization requires extensive modification and customized programming. This depends largely on the similarity between the locale's requirements and the requirements already localized in GNU/Linux.

There are many different methods used to localize GNU/Linux, using different encoding, input and display systems. At present, the most technically effective method is localization via the Linux-Unicode-OpenType model. A brief explanation of the different technologies follows.

Unicode (www.unicode.org)

The Unicode encoding system, the latest version being Unicode 4.0, is an industry standard for encoding characters and symbols. It is closely related to the ISO Universal Character Set standard 10646. Additions to either standard are coordinated between the ISO and the Unicode Consortium. The Unicode Consortium, co-founded by Apple and Xerox in 1991, now has more than 100 members, including Adobe, IBM, Microsoft, Sybase, Compaq, Hewlett Packard, Oracle, Sun Microsystems, Netscape and Ericsson.

The aim of Unicode and ISO 10646 is to encompass all of the languages of the world, with each character code corresponding to a 'glyph'. Combinations of character codes produce combined glyphs for complex characters (particularly in the Asian languages). The initial Unicode standard specified an encoding for 16-bit characters, which allows for a total of 65,535 possible characters/symbols. Later versions of the standard have expanded the encoding to a 32-bit range, allowing over one million different characters and symbols to be encoded.

The Unicode standard is more and more relevant in light of accelerated globalization. It is the most relevant encoding system for the Internet. As Internet penetration continues to increase in both developing and developed countries, the benefits of integrating Unicode in software and content development cannot be ignored.

OpenType (www.adobe.com/type/opentype/main.html)

Fonts are at the 'front end' of localization and often receive the most attention from non-technical observers. Thus, font development is very often seen as the be-all and end-all of localization. However, font

development is only one crucial component of the entire localization process, although it is the most visible.

Just as we are advocating the Unicode encoding system, we are advocating OpenType font file formats as the appropriate standard for font development in localization efforts.

OpenType is a cross-platform font file format jointly developed by Microsoft and Adobe. It is based on the Unicode encoding standard and offers multiple language character sets in one font file. Whereas traditional Western Postscript fonts are limited to 256 glyphs, an OpenType font may contain more than 65,000 glyphs, allowing multiple languages to be displayed using a single font.

Using the Linux-Unicode-OpenType model, most localization efforts involve the following steps:

1) Unicode standard corrections/enhancements
2) Font development
3) Input methods
4) Modifying applications to handle local language characteristics
5) Translating application messages
6) Ensuring that changes are accepted by the global FOSS community

Unicode standard corrections/enhancements

Creating encoding that adequately handles the needs of the countless languages throughout the world is highly complex. The immensity of this task has resulted in errors and inadequacies in the specification of certain languages, particularly languages from countries that have low levels of ICT development. Additionally, while Unicode may have included encoding for all of the major languages in the world, encoding for the other languages and dialects (India alone has over 1,000 languages and dialects) is either incomplete or non-existent. In countries where the existing Unicode standard is lacking, a review of the existing Unicode standard and recommendation of changes to the Unicode Consortium will be necessary.

Font development

Once a satisfactory Unicode standard has been developed, the next challenge is ensuring that there is a freely available, cross-platform font. Without fonts, it is impossible to display, use and manipulate any language electronically. Modern fonts, particularly OpenType fonts, are more than just the visual representation of a language. OpenType fonts contain the logic behind the display of the words, how glyphs interact with and change surrounding glyphs. Languages that differ greatly from the western alphabet (Arabic, Laotian, Dzongkha, etc) often do not have a commonly available, non-proprietary font.

Font development is no small task. A high-quality, professional font can take several years to develop.

Input methods

The next step involves standardizing and implementing a system for input in that language. The most common input method in computing is via the keyboard and many countries have created mappings between the standard keys to characters in their local language. These are often ad hoc adoptions and several are used within a country. For example, there are several keyboard layouts in use regularly in Bangladesh. The lack of a single standard is a result of and contributes further to incompatible implementations of character sets/encoding, keyboard mappings, fonts, and the like. Addressing and standardizing input methods from the outset provides developers with a common starting point.

Once an input method has been standardized, software has to be written to implement the standard under GNU/Linux. If the number of characters is less than the possible key combinations, this becomes a simple task of remapping the keys on a keyboard. It is when the number of characters far outnumber the keys on a keyboard (e.g., Chinese with its 30,000 characters) that more advanced techniques become necessary.

Modify applications to handle local language characteristics

While most major FOSS applications have been internationalized, some modification may still be necessary to adapt to local language characteristics. For example, most word processors break words on a space but in languages that do not use spaces, special rules must be created to specify breaking order. Similar problems exist with word sorting, text flow and other issues. Most languages will require minimal modification but certain languages may require extensive modification to applications.

Additionally, locale-specific information such as date format, currency symbols and other issues has to be specified. This is normally a simple task involving editing text files.

Translating application messages

The next step in localizing GNU/Linux involves the translation of messages that the application passes to the user. Messages such as “File Not Found” or “Operation Complete” have to be translated to the local language. This task involves very little technical skill as the messages are normally stored in text files for easy viewing and editing. However, translating the thousands of messages and help files is an undertaking that can take several years to complete and is often the slowest part of the localization process. Even if the task is limited to the most commonly used applications (web browser, office productivity suite), significant effort has to be expended.

Ensuring that changes are accepted by the global FOSS community

One of the major advantages of the FOSS development method is that maintenance costs are often shared among the various users of the software. However, this is possible only if the changes made are accepted by the global community. Localization may involve changes in many different software components, each maintained by different project teams. Therefore, there should be a focused effort to ensure that all changes made are accepted by each of the teams, often by ensuring that the changes are made in a manner compatible with the future direction of the project team. In essence, one must be a player in the global team effort from the very start or risk being the only one left maintaining an isolated version of GNU/Linux.

CASE STUDIES

Case Study: FOSS in Government

Introduction

The city of Largo is one of the earliest high-profile cases of a government administration migrating over to Linux. The IT system of this small city in the state of Florida, USA, supports 800 city workers, including local safety and health services. Implementation began in 2000 and their experience with Linux in the years since then have been nothing but positive.

Motivation for migrating to Linux

In 2000, the IT department of Largo was evaluating upgrade options as problems were being encountered with existing OpenServer and UnixWare products from the Santa Cruz Operation. Various options were evaluated, including Microsoft Windows on personal computers. However, since they were already on a Unix-based thin-client infrastructure, the combination of hardware and software costs involved in such a migration was deemed prohibitively expensive. Additionally, the IT team did not want to be locked into a 2-3-year upgrade cycle, where they would be forced to pay upgrade costs even when upgrades were not necessary.

Ultimately, the decision was made to keep the existing thin-client infrastructure but migrate systems to a Linux system based on Red Hat's distribution.

Implementation approach

A solution was tested and implemented starting in 2000 and completed by mid-2001. Two powerful (for that time) dual-processor Compaq servers delivered the services needed by most users. A variety of FOSS and non-FOSS applications were combined, including Netscape (web browser), Evolution (email client) and WordPerfect 8 (word processor). Heavy-duty database needs were run on a proprietary Oracle database while Microsoft's Excel and PowerPoint were made available to Linux users via a combination of Windows NT and the Citrix Metaframe server. In total, there were about 20 different servers working together, running a mix of Linux, Windows and Unix operating systems.

On the desktop side, things were simpler. The thin-client model requires only the barest minimum from desktop units. Hence, desktop units could be obtained at a relatively low cost. In some cases, the IT team managed to obtain desktop systems for as little as US$5 per unit. With 10-year lifespans and few moving parts, these desktop units rarely broke down and had a longer useful lifespan than normal PC desktops.

Results

The migration to Linux was estimated to have saved the city as much as US$1 million in the first year alone. Largo currently has an IT budget that is only about 40 percent the size of comparable cities. Where cities of a comparable size normally spend 3-4 percent of their city budget on IT, the Largo team gets along quite comfortably with only 1.3 percent of the city budget. The efficiency with which Linux uses hardware has also reaped enormous savings. The IT team estimates that they will not need to upgrade their desktops until 2007.

The reduction in number of personnel required is also significant. The end-user help desk requires only two to three people to support a user base of 800 workers. This low ratio is attributed to the reliability, stability and predictability of the system. The remaining staff members of the city's 10-member IT department are then freed for other tasks, including making additional improvements to the IT infrastructure.

For further reading:

1. Haber, Lynn, "City saves with Linux, thin clients", 10 April 2002, ZDNet; available from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860180,00.html; Internet; accessed on November 7, 2003.
2. Harris, Stephen E., "City of Largo Completes Desktop Transition", 27 August 2001, ConsultingTimes; available from http://www.consultingtimes.com/articles/desktop/largo.html; Internet; accessed on November 7, 2003.
3. Miller, Robin, "Largo loves Linux more than ever", 9 December 2002, Newsforge.com; available from http://newsforge.com/article.pl?sid=02/12/04/2346215&mode=thread&tid=19; Internet; accessed on November 7, 2003.
4. The Dravis Group, "Open Source Software: Case Studies Examining Its Use", April 2003; available from http://www.dravis.net/reports.html; Internet; accessed on November 7, 2003.

Case Study: FOSS in Education

Introduction

The Goa Schools Computer Project (GSCP) was launched in the Indian state of Goa to provide affordable computer labs to secondary and higher secondary schools in the state. The first pilot projects were launched in 2000 and after evaluation a second, larger project was launched in 2002.

The GSCP is a collaboration involving public, private and NGO organizations. The Goa Department of Education, Red Hat Linux, the Goa Computers in Schools Project NGO and the Goa Sudharop Community Development Charity all contributed to making this project a success. Using recycled computers and the FOSS GNU/Linux system, a total of 125 schools received computers that otherwise would not have been available to them.

Motivation

Cost was a primary motive for using the GNU/Linux system, particularly the licensing cost of proprietary software. Because the project decided from the start to recycle computers (also for cost reasons), finding software to place on these systems became a major issue. These systems were typically received with blank hard drives, due to concerns over security of the organizations donating the computers. Purchasing software to run on these systems would have multiplied the costs of using these computers manyfold.

By going with the recycled computer/GNU/Linux combination, the GSCP was able to install systems for as little as US$35 per system, with full computer labs, including networking, costing less than US$500. Proprietary software for a single computer would have cost at least US$400–500, many times the cost of the computer itself.

A comprehensive costing was performed for this project. Based on data from previous projects in other Indian states (Andhra Pradesh, Karnataka, Tamil Nadu and Kerala), it was estimated that the GNU/Linux/recyled hardware model would save as much as 77 percent of a traditional solution (proprietary software, new hardware). Combining GNU/Linux with new hardware would have saved 64 percent of the costs of the proprietary software/new hardware model.

Implementation approach

The GSCP used refurbished computers imported from wealthier, more developed nations. These computers were typically outdated models, replaced in regular corporate upgrade cycles. After testing and refitting as necessary, the computers were installed with the GNU/Linux operating system. The larger installations (labs with more than four computers) used GNU/Linux in a thin-client configuration.

Each computer lab was typically a cooperative effort between GSCP and the local school. GSCP would supply the equipment and teacher training while the school would supply the UPS, wiring and furniture for the computer lab. Once set up, the computer lab would be used by the schools during school hours and by the community at large after hours.

Results

A survey carried out one year after the computers were shipped found that 90 percent of the PCs had been installed and 76 percent of the PCs were operational. Schools using the thin-client model, which were also the schools that received four or more PCs, fared best. Urban schools fared better than rural schools for a variety of reasons, including better support and a larger number of available computers (due to larger student populations).

The schools are now charging 20 cents per student to pay for maintenance and Internet access costs. Pilot experiments are also underway to test the sustainability of charging the community for after-hours access to the computing facilities and the Internet. Experiences from projects in other countries have shown this to be feasible and it is hoped that it will be just as successful in Goa.

For further reading:

1. The Goa Schools Computer Project website; available from http://www.gscp.org/; Internet; accessed on November 7, 2003.
2. Martyris, Daryl, 'Community – Government partnerships and open source technology for low cost IT access in India – A case study', July 2003; available from http://www.developmentgateway.com/node/133831/sdm/blob?pid=5474; Internet; accessed on November 7, 2003.
3. Noronha, Frederick, 'Linux provides cheaper alternatives for schools in India', 9 January 2002, Newsforge.com; available from http://newsforge.com/article.pl?sid=02/01/09/1252220&mode=thread&tid=23; Internet; accessed on November 7, 2003.

Annex I: Glossary

Application

Applications are software written to supply functionality to the user. Functionality can range from end-user functions such as word processing and email reading, to server functions like databases and web servers.

Bug

An error in software that causes the program to malfunction, fail or not meet specifications. Modern bugs are usually introduced by programmer error and almost all major applications have some bugs in the system.

FTP

File Transfer Protocol – the protocol used to transfer files, both text and data. The standard dates back to the early years of the Internet and is still one of the more commonly used methods for transferring data between computer systems.

HTML

HyperText Markup Language – the language in which all web pages on the World Wide Web are encoded. HTML contains both data and instructions on how to format the data properly in a web browser. It also contains instructions on accessing related data.

HTTP

HyperText Transfer Protocol – a protocol for controlling the transfer of data between different machines. HTTP is the most commonly used transfer method between web servers and web browsers, although it has been used to transfer other types of data and traffic. It has support for encryption and compression.

LAN

Local Area Network – a data network of computers, typically covering a small geographic region, such as an office building or house. A LAN may be connected to the Internet or be a separate, distinct network that communicates only within itself. Common uses for LANs include sharing printing resources and data between computers.

Operating System

The Operating System (OS) is the collection of software that controls the hardware (disk drives, displays, keyboard, mouse, etc.) and software applications on a computer. The OS manages and allocates the physical resources (CPU processing time, hard disk space, inputs from the keyboard, etc.) among the different applications that run within it. The OS supplies minimal user functionality. User functionality is typically supplied by applications, while the OS serves as an intermediary between hardware and application. Examples of an OS are Microsoft Windows, GNU/Linux, Sun Microsystem's Solaris and Mac OS X.

Proprietary Software

Typically refers to software produced by commercial companies and licensed to users under very restrictive licenses. Any software released under licenses other than the FSF and/or OSI approved licenses are considered proprietary software for the purpose of this primer. Most proprietary software typically cannot be redistributed by the user; nor is access provided to the source code, though there are exceptions. Public domain software is not considered proprietary software.

Public Domain Software

Software that is not owned by anyone and is available to all, with no restrictions.

Source Code

Source code represents the human readable instructions that form the heart of any program, be they operating systems such as Linux or Microsoft Windows, or accounting, database and graphical applications such as Oracle, MS SQL Server and Photoshop. Although it is not easily readable to lay people, software engineers can easily understand, correct and modify programs using the original source code. For example, a portion of code may look like this[96]:

```
  Float distance (p0, p1)
  struct point p0, p1;
   {
     float xdist = p1.x - p0.x;
float ydist = p1.y - p0.y;
  return sqrt (xdist * xdist + ydist * ydist);
  }
```

Once software engineers are finished with source code, it is converted to machine-readable code that looks like this:

```
1314258944     -232267772     -231844864     1634862
1411907592      -231844736     2159150        1420296208
-234880989     -234879837     -234879966     -232295424
1644167167     -3214848       1090581031     1962942495
572518958      -803143692     1314803317
```

Few engineers are able to understand, much less modify, a program once it has been converted to a machine-readable format. Because of this, most proprietary software is distributed in machine-readable form only and the source code is a jealously guarded secret.

TCP/IP

Transmission Control Protocol over Internet Protocol – the protocol that underlies most of the Internet today, as well as most Ethernet LANs. TCP/IP was developed by the US agency DARPA. It supplies a reliable communication protocol at a very low level. Most Internet protocols (HTTP, FTP, telnet) are built on top of the TCP/IP protocol.

Thin Client

A thin-client infrastructure is one where most of the computational and data storage tasks are done on powerful server systems. The end-user system sitting on desktops are low-powered, displaying only the data returned by the servers. Standard desktop systems today running Microsoft Windows are fat-client systems, where most of the processing and data storage work is done by the desktop system and only a limited amount of work is done by the server.

Various advantages can be gained from using a thin-client infrastructure, mostly centering around lower maintenance and hardware costs. Because all data is stored on the server, including the applications, it is easier for administrators to manage and update the systems. A single change on the main server will immediately be reflected on all user systems. Likewise, the minimal data storage and processing requirements result in cheap desktop systems that do not need to be updated as regularly as today's fat-client desktops.

Annex II: Software Licenses

The following is a listing of common software licenses, in comparison to GPL:

Table derived from: http://www.fsf.org/licenses/license-list.html

GPL-Compatible, Free Software Licenses		
1	The GNU General Public License, or GNU GPL for short	http://www.fsf.org/licenses/gpl.html
2	The GNU Lesser General Public License, or GNU LGPL for short	http://www.fsf.org/copyleft/lesser.html
3	The license of Guile	-
4	The license of the run-time units of the GNU Ada compiler	-
5	The X11 license	http://www.x.org/terms.htm
6	Expat license	http://www.jclark.com/xml/copying.txt
7	Standard ML of New Jersey Copyright License	http://cm.bell-labs.com/cm/cs/what/smlnj/license.html
8	Public Domain	-
9	The Cryptix General License	http://www.cryptix.org/docs/license.html
10	The modified BSD license	http://www.xfree86.org/3.3.6/COPYRIGHT2.html#5
11	The license of ZLib	http://www.gzip.org/zlib/zlib_license.html
12	The license of the iMatix Standard Function Library	-
13	The W3C Software Notice and License	http://www.w3.org/Consortium/Legal/copyright-software.html
14	The Berkeley Database License	http://www.sleepycat.com/license.net
15	The OpenLDAP License, Version 2.7	http://www.openldap.org/software/release/license.html
16	The License of Python 1.6a2 and earlier versions	http://www.python.org/doc/Copyright.html
17	The License of Python 2.0.1, 2.1.1, and newer versions	http://www.python.org/2.0.1/license.html
18	The Perl License	-
19	The Clarified Artistic License	http://www.statistica.unimib.it/utenti/dellavedova/software/artistic2.html
20	The Artistic License, 2.0	
21	The Zope Public License version 2.0	http://www.zope.org/Resources/ZPL
22	The Intel Open Source License (as published by OSI)	http://www.opensource.org/licenses/intel-open source-license.html

GPL-Incompatible, Free Software Licenses		
23	The Arphic Public License	ftp://ftp.gnu.org/non-gnu/chinese-fonts-truetype/LICENSE
24	The original BSD license	http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6
25	The Apache License, Version 1.0	http://www.apache.org/LICENSE-1.0
26	The Apache License, Version 1.1	http://www.apache.org/LICENSE-1.1
27	The Zope Public License version 1	http://www.zope.org/Resources/ZPL
28	The license of xinetd	http://www.xinetd.org/license
29	The License of Python 1.6b1 and later versions, through 2.0 and 2.1	http://www.handle.net/python_licenses/python1.6_9-5-00.html
30	The old OpenLDAP License, Version 2.3	-
31	The license of Vim, Version 5.7	-
32	IBM Public License, Version 1.0	http://oss.software.ibm.com/developerworks/opensource/license10.html
33	Common Public License Version 0.5	http://www.eclipse.org/legal/cpl-v05.html
34	The Phorum License, Version 1.2	http://phorum.org/license.txt
35	The LaTeX Project Public License	-
36	The Mozilla Public License (MPL)	http://www.mozilla.org/MPL/MPL-1.1.html
37	The Netizen Open Source License (NOSL), Version 1.0	http://bits.netizen.com.au/licenses/NOSL/nosl.txt
38	The Interbase Public License, Version 1.0	http://www.borland.com/devsupport/interbase/opensource/IPL.html
39	The Sun Public License	http://www.netbeans.org/spl.html
40	The Netscape Public License (NPL)	http://www.mozilla.org/NPL/NPL-1.0.html
41	The Jabber Open Source License, Version 1.0	http://www.jabber.com/license/index.shtml
42	The Sun Industry Standards Source License 1.0	http://www.openoffice.org/licenses/sissl_license.html
43	The Q Public License (QPL), Version 1.0	http://www.trolltech.com/developer/licensing/qpl.html
44	The FreeType license	-
45	The PHP License, Version 2.02	http://www.php.net/license/2_02.txt

Non-Free Software Licenses		
46	The (Original) Artistic License	http://www.perl.com/language/misc/Artistic.html
47	The Apple Public Source License (APSL)	http://www.publicsource.apple.com/apsl/
48	The Sun Community Source License	-
49	The Plan 9 License	-
50	Open Public License	http://koala.ilog.fr/jackaroo/OPL_1_0.TXT
51	The Utah Public License	-
52	eCos Public License	-
53	The Sun Solaris Source Code (Foundation Release) License, Version 1.1	-
54	The YaST License	-
55	Daniel Bernstein's licenses	-
56	The "Aladdin Free Public License"	-
57	The Scilab license	-
58	The AT&T Public License	-
59	The Jahia Community Source License	-

Annex III: Primer Licensing

Summary

This document is released under the Creative Commons Attribution 1.0 license. In short, you are free to:

- copy, distribute, display, and perform the work
- make derivative works
- make commercial use of the work

Under the following conditions:

- Attribution: You must give the original authors credit.
- For any reuse or distribution, you must make clear to others the license terms of this work.

Any of these conditions can be waived if you get permission from the authors. Your fair use and other rights are in no way affected by the above.

Full License

The most recent version of this license is normally available at:
http://creativecommons.org/licenses/by/1.0/legalcode

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE IS PROHIBITED.
BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. Definitions

a. **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
b. **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License.
c. **"Licensor"** means the individual or entity that offers the Work under the terms of this License.

d. **"Original Author"** means the individual or entity who created the Work.
e. **"Work"** means the copyrightable work of authorship offered under the terms of this License.
f. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

2. Fair Use Rights. Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. License Grant. Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;
b. to create and reproduce Derivative Works;
c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;
d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works;

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. Restrictions. The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any reference to such Licensor or the Original Author, as requested.
b. If You distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to

the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. Representations, Warranties and Disclaimer

a. By offering the Work for public release under this License, Licensor represents and warrants that, to the best of Licensor's knowledge after reasonable inquiry:
    i. Licensor has secured all rights in the Work necessary to grant the license rights hereunder and to permit the lawful exercise of the rights granted hereunder without You having any obligation to pay any royalties, compulsory license fees, residuals or any other payments;
    ii. The Work does not infringe the copyright, trademark, publicity rights, common law rights or any other right of any third party or constitute defamation, invasion of privacy or other tortious injury to any third party.
b. EXCEPT AS EXPRESSLY STATED IN THIS LICENSE OR OTHERWISE AGREED IN WRITING OR REQUIRED BY APPLICABLE LAW, THE WORK IS LICENSED ON AN "AS IS" BASIS, WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES REGARDING THE CONTENTS OR ACCURACY OF THE WORK.

6. Limitation on Liability. EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, AND EXCEPT FOR DAMAGES ARISING FROM LIABILITY TO A THIRD PARTY RESULTING FROM BREACH OF THE WARRANTIES IN SECTION 5, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark “Creative Commons” or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons’ then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at http://creativecommons.org/.

Annex IV: Credits/Document History

7th November 2003

- Final copyedit by Patricia B. Arinto
- Removed GPL Annex

16th October 2003

Final copyedit version prepared incorporating additional comments from:

- Shahid Akhtar
- Tan Wooi Tong

1st October 2003

Version 0.9.2 produced incorporating comments from:

- Arun M
- Serge Marelli
- Karl O. Pinc
- Imran William Smith
- Anousak Souphavanh
- Richard Stallman
- Gaurab Raj Upadhaya

Significant changes:

- Major rewrite of the localization portion of the primer
- Added two case studies
- Added to glossary
- Added preface to define target audience of primer
- Added Open Content license to Annexes
- Modified Philosophy: FSF vs OSI definitions to correct inaccuracies about FSF

1st September 2003

Version 0.9 produced incorporating comments from:

- Shahid Akhtar
- Jethro Cramp
- Dr. Nah Soo Hoe
- Bjorn Stabell
- Tan Wooi Tong
- Raul Zambrano

1st June 2003

Original Primer material written by Kenneth Wong and Phet Sayo from the Asia-Pacific Development Information Programme.

Endnotes

[1] Wheeler, David, “Why OSS/FS? Look at the Numbers!” [home page online]; available from http://www.dwheeler.com/oss_fs_why.html; Internet; accessed on November 7, 2003.

[2] “The Free Software Definition”[home page online]; available from http://www.fsf.org/philosophy/free-sw.html; Internet; accessed on November 9, 2003.

[3] Open Source Initiative [home page online]; available from http://www.opensource.org; Internet; accessed November 8, 2003.

[4] Raymond, Eric S., “The Cathedral and the Bazaar” [home page online]; available from http://catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/; Internet; accessed on November 7, 2003.

[5] Raymond, Eric S., “The Cathedral and the Bazaar” [home page online]; available from http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/ar01s04.html; Internet; accessed on November 7, 2003.

[6] Bengtsson, Lassing, Bosch, van Vliet, “Analyzing Software Architectures for Modifiability”; available from http://www.cs.rug.nl/~bosch/papers/SAAModifiability.pdf; Internet; accessed on November 7, 2003.

[7] “A Brief History of Free/Open Source Software Movement” [home page online]; available from http://www.openknowledge.org/writing/open-source/scb/brief-open-source-history.html; Internet; accessed on November 7, 2003.

[8] “An Open Letter To Hobbyists by Bill Gates – 1976”; available from http://www.tranquileye.com/cyber/1976/gates_open_letter_to_hobbyists.html; Internet; accessed on November 7, 2003.

[9] Moody, Glyn, “Rebel Code”, Penguin Books, London, England, 2001.

[10] “A Brief History of Free/Open Source Software Movement” [home page online]; available fromhttp://www.openknowledge.org/writing/open-source/scb/brief-open-source-history.html; Internet; accessed on November 7, 2003.

[11] “History of the OSI” [home page online]; available from http://www.opensource.org/docs/history.php; Internet; accessed on November 7, 2003.

[12] Scannell, Ed. “Linux takes the operating system scene by storm”, *Infoworld.com*; available from http://archive.infoworld.com/supplements/99poy_drv/99poy_linux.html; Internet; accessed on November 7, 2003.

[13] Leonard, Andrew, "An Alternative Voice: How the Tech-Poor Can Still Be Software Rich", 28 June 2001, The International Herald Tribune Online; available from http://www.iht.com/cgi-bin/generic.cgi?template=articleprint.tmplh&ArticleId=24330; Internet; accessed on November 7, 2003.

[14] Shankland, Kane, Lemos, "How Linux saved Amazon Millions", 30 October 2001, *Cnet News.com* [home page online]; available from http://news.com.com/2100-1001-275155.html; Internet; accessed on November 7, 2003.

[15] Sisk, Michael, "Linux Woos Wall St.", August 2003, *Bank Technology News*; available from http://www.banktechnews.com/cgi-bin/readstory.pl?story=20030801BTNC617.xml; Internet; accessed on November 7, 2003.

[16] Orzech, Dan, "Linux TCO: Less Than Half The Cost of Windows", 7 October 2002, *CIO Update*; available from http://www.cioupdate.com/article.php/10493_1477911; Internet; accessed on November 7, 2003.

[17] "netproject – Cost of Ownership" [home page online]; available from http://www.netproject.com/opensource/coo.html; Internet; accessed on November 7, 2003.

[18] Maguire, James, "Windows vs. Linux: TCO Feud Rages On", 01 August 2003, *Newsfactor Network* [home page online]; available from http://www.newsfactor.com/perl/story/22012.html; Internet; accessed on November 8, 2003.

[19] Lemos, Robert, "Merrill Lynch: Linux saves money", 7 June 2003, *CNet News.com* [home page online]; available from http://news.com.com/2100-1016_3-1014287.html?tag=fd_top; Internet; accessed on November 8, 2003.

[20] "Welcome to Cybersource" [home page online]; available from http://www.cyber.com.au; Internet; accessed on November 8, 2003.

[21] Glover, Tony, "Microsoft losing market grip as rivals go on the offensive", 18 May 2002, *Scotland on Sunday*; available from http://www.scotlandonsunday.com/business.cfm?id=562032003; Internet; accessed on November 8, 2003.

[22] Pescatore, John, "Commentary: Another worm, more patches", 20 September 2001, *CNet News.com*; available from http://news.com.com/2009-1001-273288.html?legacy=cnet&tag=nbs; Internet; accessed on November 8, 2003.

[23] Luening, Eric, "Windows users pay for hacker insurance", 29 May 2001, *CNet News.com* [home page online]; available from http://news.com.com/2100-1001-258392.html?legacy=cnet; Internet; accessed on November 8, 2003.

[24] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the

European Union", June 2002; available from http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[25] Najani, Niranjan, "Free as in Education", available from http://www.maailma.kaapeli.fi/FLOSSReport1.0.html; Internet; accessed on November 8, 2003.

[26] Vaughan-Nichols, Steven J., "Can You Trust This Penguin?", 1 November, 1999, *ZDNet SmartPartner.* Article no longer available from ZDNet site but archived at http://web.archive.org/web/20010606035231/http://www.zdnet.com/sp/stories/issue/0,4537,2387282,00.html; Internet; accessed on November 8, 2003.

[27] Wheeler, David, "Why OSS/FS? Look at the Numbers!" [home page online]; available from http://www.dwheeler.com/oss_fs_why.html; Internet; accessed on November 8, 2003.

[28] "The Web Standards Project: Fighting for Standards in our Browsers" [home page online]; available from http://archive.webstandards.org/upgrade/; Internet; accessed on November 8, 2003.

[29] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the European Union", June 2002; available from http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[30] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the European Union", June 2002; available from http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[31] Ibid.

[32] "Doc1: Halloween Documents" [home page online]; available from http://www.opensource.org/halloween/halloween1.html; Internet; accessed on November 8, 2003.

[33] Roberts, Paul, "Software Piracy Declines 10 Percent", 03 June, 2003, *Infoworld.com* [home page online]; available from http://www.infoworld.com/article/03/06/03/HNpiracydecline_1.html?security; Internet; accessed on November 8, 2003.

[34] "Frequently Asked Questions" [home page online]; available from http://www.lisa.org/info/faqs.html#gil; Internet; accessed on November 8, 2003.

[35] Walsh, Mary Williams, "Microsoft in War of Words", *Los Angeles Times*; available from http://www.tungutaekni.is/ymis_frodleikur/war_of_words.html; Internet; accessed on November 8, 2003.

[36] DiCarlo, Lisa, "PeopleSoft Jumps On The Linux Train", *Forbes.com*; available from http://www.forbes.com/technology/2003/05/06/cx_ld_0506psft.html; Internet; accessed on November 8, 2003.

[37] Herrington, Jack, "Is Documentation Holding Open Source Back?" *DexX.com* [home page online]; available from http://www.devx.com/devx/editorial/11839; Internet; accessed on November 8, 2003.

[38] Miller, Robin, "Open Source: A Case For E-Government", 21 October 2002, *Newsforge* [home page online]; available from http://newsforge.com/newsforge/02/10/20/1746231.shtml?tid=4; Internet; accessed on November 8, 2003.

[39] Williams, Peter, "Europe picks Penguin to link government IT", 18 July 2003, *VNUNet.com* [home page online]; available from http://www.vnunet.com/News/1142411; Internet; accessed on November 8, 2003.

[40] "Kable Report on Open Source Software – Sponsored by Sun Microsystems", 17 March 2003, Kable Ltd,

[41] Najani, Niranjan, "Free as in Education"; available from http://www.maailma.kaapeli.fi/FLOSSReport1.0.html; Internet; accessed on November 8, 2003.

[42] "LinuxPR: Munich Goes with Open Source Software", 28 May 2003, *linuxtoday.com* [home page online]; available from http://linuxtoday.com/infrastructure/2003052802126NWDTPB; Internet; accessed on November 8, 2003.

[43] "IBM signs Linux deal with Germany", 3 June 2002, *BBC News*; available from http://news.bbc.co.uk/1/hi/business/2023127.stm; Internet; accessed on November 8, 2003.

[44] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the European Union", June 2002;available from http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[45] "Kable Report on Open Source Software – Sponsored by Sun Microsystems", 17 March 2003, Kable Ltd

[46] Najani, Niranjan, "Free as in Education"; available from http://www.maailma.kaapeli.fi/FLOSSReport1.0.html; Internet; accessed on November 8, 2003.

[47] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the European Union", June 2002, http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[48] "Kable Report on Open Source Software – Sponsored by Sun Microsystems", 17 March 2003, Kable Ltd

[49] "Open Source Software – use within UK Government", *UK Gov Talk*, 15 July 2002; available from http://www.govtalk.gov.uk/documents/oss_policydocument_2002-07-15.pdf; Internet; accessed on November 8, 2003.

[50] Ghosh, Krieger, Glott, Robles, "Free/Libre and Open Source Software: Survey and Study. Part 2B: Open Source Software in the Public Sector: Policy within the European Union", June 2002; available from http://www.infonomics.nl/FLOSS/report/FLOSSFinal_2b.pdf; Internet; accessed on November 8, 2003.

[51] Najani, Niranjan, "Free as in Education"; available from http://www.maailma.kaapeli.fi/FLOSSReport1.0.html; Internet; accessed on November 8, 2003.

[52] Kanellos, Shankland, "Should government mandate open source?", 12 August 2002, *CNET News.com* [home page online]; available from http://zdnet.com.com/2100-1104-949241.html; Internet; accessed on November 8, 2003.

[53] Barr, Joe, "Open source making headway in Texas government" 24 March 2003, *Linuxworld.com* [home page online]; available from http://www.linuxworld.com/2003/0324.barr.html; Internet; accessed on November 8, 2003.

[54] Duin, Steve, "Oregon is still a soft touch for Microsoft", 5 May 2003, The Oregonian, available from http://www.oregonlive.com/news/oregonian/steve_duin/index.ssf?/base/news/105377817415280.xml; Internet; accessed on November 8, 2003.

[55] "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense", 2 January 2003, *Mitre Corporation*; available from http://www.egovos.org/pdf/dodfoss.pdf; Internet; accessed on November 8, 2003.

[56] Wheeler, David, "Why OSS/FS? Look at the Numbers!" [home page online]; available from http://www.dwheeler.com/oss_fs_why.html; Internet; accessed on November 8, 2003.

[57] Haber, Lynn, "City saves with Linux, thin clients", 10 April 2003, *ZDNet* [home page online]; available from http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860180,00.html; Internet; accessed on November 8, 2003.

[58] Adelstein, Tom, "Linux Access in State and Local Government, Part II", 19 June 2003, *Linuxjournal.com*; available from http://www.linuxjournal.com/article.php?sid=6952; Internet; accessed on November 8, 2003.

[59] "Respuesta a Microsoft en idioma Ingles" [home page online]; available from http://www.gnu.org.pe/resmseng.html (English translation); Internet; accessed on November 8, 2003.

[60] "The Brazilian Public Sector to Choose Free Software", 2 June 2003, *PCLinuxOnline* [home page online]; available from http://www.pclinuxonline.com/modules.php?name=News&file=article&sid=6879; Internet; accessed on November 8, 2003.

[61] Williams, Martyn, "Japan, China, Korea plan joint open-source project", 05 September 2003, *IDG News Service*; available from http://www.idg.com.sg/idgwww.nsf/unidlookup/04B8C8F13FF8653148256D98002BC4A2?OpenDocument; Internet; accessed on November 8, 2003.

[62] Liu, Bob, "China to be stronghold for Open Source", 5 November 2002, *internetnews.com* [home page online]; available from http://www.internetnews.com/stats/article.php/1494881; Internet; accessed on November 8, 2003.

[63] Najani, Niranjan, "Free as in Education"; available from http://www.maailma.kaapeli.fi/FLOSSReport1.0.html; Internet; accessed on November 8, 2003.

[64] "China blocks foreign software use in gov't", 18 August 2003, *CNETAsia* [home page online]; available from http://asia.cnet.com/newstech/applications/0,39001094,39146335,00.htm; Internet; accessed on November 8, 2003.

[65] Chai, Winston, "Governments are latching on to Linux", 12 May 2003, *CNETAsia* [home page online]; available from http://zdnet.com.com/2100-1104-1000992.html; Internet; accessed on November 8, 2003.

[66] Berger, Matt, "ANALYSIS: Microsoft vs. open source gets political", 10 June 2002, *IDG News Service*; available from http://www.idg.net/ic_874742_1793_1-1681.html; Internet; accessed on November 8, 2003.

[67] Hu, Qing Hua, "Yangfan and Qihang Project", presented at the Asia OSS Symposium, 3-6 March 2003, Phuket, Thailand.

[68] Williams, Martyn, "Japan, China, Korea plan joint open-source project", 05 September 2003, *IDG News Service*; available from http://www.idg.com.sg/idgwww.nsf/unidlookup/04B8C8F13FF8653148256D98002BC4A2?OpenDocument; Internet; accessed on November 8, 2003.

[69] Ribeiro, John, "India official: No government edict on open source" 1 April 2002, *IDG News Services*; available from http://www.computerworld.com/softwaretopics/os/linux/story/0,10801,79918,00.html?f=x249; Internet; accessed on November 8, 2003.

[70] Basu, Indrajit, "Microsoft takes on Linux in India", 16 November 2002, *Asia Times Online*; available from http://www.atimes.com/atimes/South_Asia/DK16Df02.html; Internet; accessed on November 8, 2003.

[71] Sharma, Anil, "MP opens windows to Linux" 19 November 2003, *The Economic Times*; available from http://economictimes.indiatimes.com/cms.dll/html/uncomp/articleshow?artid=28707422; Internet; accessed on November 9, 2003.

[72] Pillai, Sanjay K., "Linux seen grabbing 10% of desktop OS segment" 26 February 2003, *Business Standard*; available from http://www.business-standard.com/today/story.asp?Menu=2&story=8930; Internet; accessed on November 9, 2003.

[73] Basu, Indrajit, "Microsoft takes on Linux in India", 16 November 2002, *Asia Times Online*;available from http://www.atimes.com/atimes/South_Asia/DK16Df02.html; Internet; accessed on November 9, 2003.

[74] Tai, Andy, "Taiwan to start national plan to push Free Software", 3 June 2002, *Kuro5hin* [home page online]; available from http://www.kuro5hin.org/story/2002/6/3/55433/41738; Internet; accessed on November 9, 2003.

[75]Lui, John, "Thailand's cheap PCs 'force Microsoft's hand'", 22 August 2003, *CNETAsia*, [home page online]; available from http://news.zdnet.co.uk/software/windows/0,39020396,39115884,00.htm; Internet; accessed on November 9, 2003.

[76] Moreira, Charles "Malaysia backs open source", 13 August 2002, *The Star Online*; available from http://asia.cnet.com/newstech/systems/0,39001153,39071821,00.htm; Internet; accessed on November 9, 2003.

[77] "DRB-HICOM'S efforts to Bridge the digital divide lauded" [home page online]; available from http://arfa.komnas.com/community/article.php?sid=5&mode=thread&order=0; Internet; accessed on November 9, 2003.

[78] Chai, Winston, "Japan mulls Windows replacement", 21 November 2002, *CNETAsia* [home page online]; available from http://zdnet.com.com/2100-1104-966700.html; Internet; accessed on November 9, 2003.

[79] "Japan Government Payroll Computer System Will Use Linux, Not Windows", 9 July 2003, *Linuxworld.com* [home page online]; available from http://www.linuxworld.com/story/33812.htm; Internet; accessed on November 9, 2003.

[80] Festa, Paul, "South Africa embraces open source", 05 Feb 2003, *CNET News* [home page online]; available from http://news.zdnet.co.uk/software/0,39020381,2129893,00.htm; Internet; accessed on November 9, 2003.

[81] Ikhemuemhe, Godfrey, "Experts Advocate Open Source for NEPAD to Realise Its ICT Objectives", 24 September 2003, *AllAfrica.com* [home page online]; available from http://allafrica.com/stories/200309240393.html; Internet; accessed on November 9, 2003.

[82] Wheeler, David, "Why OSS/FS? Look at the Numbers!"; [home page online]; available from http://www.dwheeler.com/oss_fs_why.html; Internet; accessed on November 9, 2003.

[83] "May 2003 Web Server Survey" [home page online]; available from http://news.netcraft.com/archives/2003/05/05/may_2003_web_server_survey.html; Internet; accessed on June 9, 2003.

[84] Wheeler, David, "Why OSS/FS? Look at the Numbers!" [home page online]; available from http://www.dwheeler.com/oss_fs_why.html; Internet; accessed on November 9, 2003.

[85] "Linux for Playstation 2 Community" [home page online]; available from http://playstation2-linux.com/; Internet; accessed on November 9, 2003.

[86] Proffitt, Brian, "Munich May Opt for Linux After All", 26 May 2003, *Linuxtoday.com* [home page online]; available from http://linuxtoday.com/infrastructure/2003052600126NWSWPB; Internet; accessed on November 9, 2003.

[87] "Debian GNU/Linux — The Universal Operating System" [home page online]; available from http://www.debian.org; Internet; accessed on November 9, 2003.

[88] Wheeler, David, "Make Your Open Source Software GPL-Compatible. Or Else" [home page online]; available from http://www.dwheeler.com/essays/gpl-compatible.html; Internet; accessed on November 9, 2003.

[89] Netcraft, "May 2003 Web Server Survey" [home page online]; available from http://news.netcraft.com/archives/2003/05/05/may_2003_web_server_survey.html; Internet; accessed on June 9, 2003.

[90] Somogyi, Stephan, "BSD sleight of hand", 3 April 2000, *ZDNet News* [home page online]; available from http://zdnet.com.com/2100-11-519701.html?legacy=zdnn; Internet; accessed on November 9, 2003.

[91] "Microsoft, TCP/IP, Open Source, and Licensing" *Kuro5hin* [home page online]; available from http://www.kuro5hin.org/story/2001/6/19/05641/7357; Internet; accessed on November 9, 2003.

[92] "Frequently Asked Questions about the GNU GPL" [home page online]; available from http://www.fsf.org/licenses/gpl-faq.html#MereAggregation; Internet; accessed on November 9, 2003.

[93] "The Localization Institute" [home page online]; available from http://www.localizationinstitute.com/switchboard.cfm?page=terminology; Internet; accessed on November 9, 2003.

[94] "MLP – Ongoing Localization Projects" [home page online]; available from http://www.mozilla.org/projects/l10n/mlp_status.html#contrib; Internet; accessed on November 9, 2003.

[95] Available from http://www.medialabasia.org/news/news_top2.html; Internet; accessed on May 20, 2003.

[96] Stallman, Richard M., "Why Software Should Be Free" [home page online]; available from http://www.fsf.org/philosophy/shouldbefree.html; Internet; accessed on November 9, 2003.

APDIP

The Asia-Pacific Development Information Programme (APDIP) is an initiative of the United Nations Development Programme (UNDP) that aims to promote the development and application of new Information and Communication Technologies (ICT) for poverty alleviation and sustainable human development in the Asia-Pacific region. It does so through three core programme areas, namely, Policy Development and Dialogue; Access; and Content Development and Knowledge Management.

In collaboration with National Governments, APDIP seeks to assist national and regional institutions in Asia-Pacific through activities that involve awareness raising and advocacy, building capacities, promoting ICT policies and dialogue, promoting equitable access to tools and technologies, knowledge sharing, and networking. Strategic public-private sector partnerships and opportunities for technical cooperation among developing countries (TCDC) are APDIP's key building blocks in implementing each programme activity.

http://www.apdip.net

IOSN

The International Open Source Network (IOSN) is an initiative of UNDP's Asia-Pacific Development Information Programme (APDIP). Its overall objective is to serve as a Center of Excellence and Clearinghouse for Information on Free and Open Source Software (FOSS) in the Asia-Pacific region. IOSN seeks to raise the awareness of FOSS, facilitate the networking of people involved in FOSS, strengthen capacities in FOSS and to conduct R&D on FOSS.

The beneficiaries of IOSN are governments, IT professionals, software developers, FOSS R&D community, academics, and the NGO community. IOSN serves as a resource center to help policy- and decision-makers in the public sector, educational institutions, businesses and others develop policies and plans for the use of FOSS in their respective organizations. Much of IOSN's activities are undertaken online and the IOSN portal (www.iosn.net) has been developed for this purpose and to serve as a comprehensive online resource center on FOSS. The IOSN portal also provides a means for the FOSS community in the region to contribute to its effort and to interact.

http://www.iosn.net

For e-Primers available in the Free/Open Source Software series, please visit:

www.iosn.net

Titles available in the e-Primers for the Information Economy, Society and Polity series:

1. The Information Age
2. Nets, Webs and the Information Infrastructure
3. e-Commerce and e-Business
4. Legal and Regulatory Issues in the Information Economy
5. e-Government
6. ICT in Education
7. Genes, Technology and Policy
8. Information and Communication Technologies for Poverty Alleviation

www.eprimers.org

UNDP

International Open Source Network

An Initiative of the UNDP's Asia-Pacific Development Information Programme

Asia-Pacific Development Information Programme

www.apdip.net

United Nations Development Programme
Wisma UN, Block C Kompleks Pejabat Damansara
Jalan Dungun, Damansara Heights,
50490 Kuala Lumpur, Malaysia

tel: +603 2095 9122
fax: +603 2093 9740
email: info@apdip.net