

Chapter

16

Open Source Software

Learning Objectives

After completing this chapter, you will be able to:

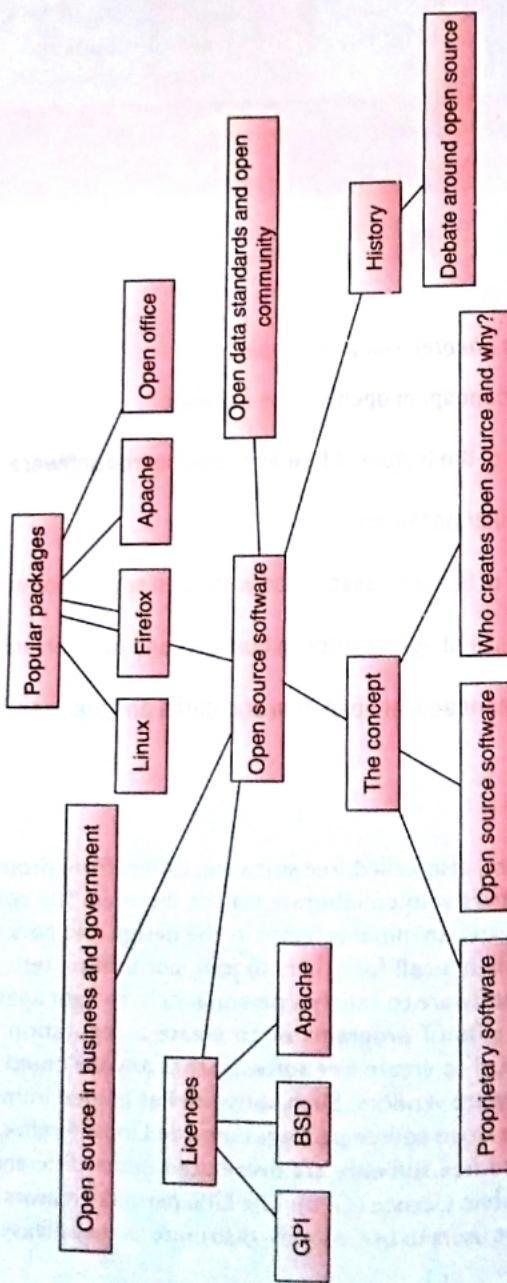
- Understand the concept of open source software
- Get an overview of the history of free and open source software
- Learn about popular packages
- Get an overview of licences associated with open source software
- Learn about the use of open source in business and government
- Understand the concepts of open data standards and the open community

Open source software, also called free software, differs from proprietary software as it is made by a community of programmers who collaborate via the Internet. The collaboration effort starts with an identified need for a software, an initial attempt at the design and conception of the software along with some working code, and then a call for others to join, contribute, test, re-design and distribute. Programmers create open source software to satisfy a personal itch, to fight against the control of proprietary software, to give and receive gifts of programs or to create a reputation for themselves. Open source software originated from a need to create free software that anyone could use, modify and distribute without any hindrance from software vendors. Such software has gained immense popularity across the world and is widely used. Popular open source packages include Linux, Firefox, Apache, GIMP and Open Office.

Open source and free software are distributed under different licences, the most popular of which is the GNU General Public Licence (GPL). The GPL permits creators of the software to retain copyright over their work, but allows users to use, modify, distribute as they please, while retaining the GPL. Other licences,

MIND MAP OF CHAPTER TOPICS

such as BSD and Apache, allow different facilities. Business firms have evolved various models by which to draw revenues from open source software and services. Many governments have adopted open source as it enables them to reduce costs of IT and dependence on vendors; it allows them to experiment and also reduces piracy. Open source software also enables use of open data standards and promotes the culture of open collaboration and sharing, thus helping create the digital commons.



JustDial describes itself as a local search engine. It is a text- and speech-based service that allows users to call in and ask for a business or service in their locality. This could be a place where they can order a pizza or get in touch with a nearby dry-cleaning shop. JustDial operators can answer on the phone, after looking up the caller's location and searching for the business in their databases, or they can text the answer to the caller, which is usually the preferred mode. And, they need just 60 s to do all this.

The search engine market in India was valued at USD 100 million in 2008 and is expected to reach USD 600 million by 2015. Local web search constitutes 30% of the overall search. Local search has evolved from word-of-mouth and print directories to phone-based information services and online web directories. The prerequisites for an effective local search engine include comprehensiveness and updated information, the relevance of the search result in the context of local information, and the ease and speed of accessing information.

JustDial is one of the leading players providing services across multiple channels including a print directory (Yellow pages), a phone/voice search, web search (www.JustDial.com) and mobile phone search (SMS and WAP services). The firm has very impressive statistics to boast, including a database comprising of 4 million companies, 125,000 small and medium enterprise (SME) advertisers, 25 million unique users and a spread across 240 cities in late 2010. The business operation is supported by eight call centres and 3500 staff.

The business model for JustDial is based on providing customers with people-assisted search data, which is information on local SMEs. The customers for JustDial are the end-users who are the audience for its data and the SMEs, whose data it stores. It obtains revenues from the listing fee local businesses pay for being on its database, the advertisements it includes on the website, and the banners it includes in text messages. JustDial also obtains revenues from sharing its database with partners, and also by participating in search queries of global database engines.

JustDial is in a highly competitive market as their business requires low capital investments, has very low switching costs for customers (as there is very little lock-in for them), and also has low lock-in costs for their clients (the SMEs). Its competition includes India-specific websites such as sulekha.com, guruji.com, OnYoMo.com, askme.com, khoj.com and asklaila.com; local players such as discoverbangalore.com and dialgurgaon.com; and global search engines such as Google and Yahoo. The one thing that gives JustDial strength is its existing database, which is hard to replicate, as the cost of generating local information on SMEs across hundreds of cities is prohibitive. However, for JustDial it is also essential that its database is constantly updated and provides relevant and current information. This is a strategic necessity.

With the increasing availability of the Internet, more and more consumers are opting for web searches instead of accessing data through a phone. Global search engines, such as Google and Yahoo, which act as default search sites, are often used for local information search as well. Competition is also emerging from websites such as OnYoMo and AskLaila, which are tying up with mobile network providers; and from mobile alerts service being provided by Metromela. (AskLaila is tying up with online map services Bing maps and social networking site Facebook.)

In light of the competition, it is imperative for JustDial to maintain a high quality of service, which would differentiate it from its competitors, and also provide search services that are superior to engines such as Google and Yahoo.

CASE STUDY: JustDial and Open Source

CASE STUDY: JustDial and Open Source

JustDial's choice to compete on the basis of technology meant that the IT systems it deployed had to be positioned at the cutting edge of technology. The technical requirements included:

1. Need for high reliability.
2. Accurate information that is constantly updated and relevant.
3. Fast response time (internal target of 95% of the calls to be addressed in less than 60 s). This needs the ability to digest the customer query, identify the information and retrieve it, communicate it to the customer, and do all of this in less than 60 s!
4. Almost instantaneous and guaranteed SMS response requiring a strong integration with the telecommunications service providers.
5. Instantaneous data extraction needing superior search engine capabilities.
6. Simple user interface on top of a superior technology at the back-end.

Given the diverse and critical needs for technology, JustDial chose open source software as their platform of choice for developing various IT systems. Vendor independence, flexibility, superior quality and scalability, in addition to cost advantage, were cited as the main reasons for adopting open source software.

Considering that there was a strong need for constantly evolving the system, JustDial could not afford any sort of vendor lock-in and sought the ability to change at will without being tied down to any external vendor. The independence provided by open source went beyond just cost and time implications, and allowed JustDial the freedom to innovate; flexibility to change and address their stringent time-to-market considerations.

Open source provided JustDial the ability to choose the hardware platform and build the software by mixing and matching several open source components that delivered flexibility and superior performance. In addition, the choice of these open source components provided JustDial the ability to scale up and support their aggressive growth targets.

The ability to view the source code that had contributions from a vast community of developers also provided the JustDial team a source of new ideas and the required impetus to constantly innovate.

The cost advantage inherent in open source also gave them the ability to do vast in-house development and build a superior technology platform to support the numerous business demands without the worry of astronomical IT development costs.

The open source platform deployed at JustDial was based on a CLAMP stack (C, Linux, Apache, MySQL and PHP). Red Hat Enterprise Linux was chosen for the servers and Ubuntu Linux was deployed on the workstations. The ability to customise Ubuntu with Mozilla Firefox integration so as to enhance the data extraction performance on the IRO's (Information Retrieval Officer) workstation was a major factor in support of the open source decision. The system was built by carefully evaluating and selecting software components based on specific business and technological needs. One example was the use of Sphinx, a full-text search engine that integrates well with an SQL database and is able to provide fast, relevant full-text search.

JustDial belongs to the elite list of rare companies who ventured to develop and use in-house IT systems and establish a market leadership in India based on the strength of their technical innovation. Based on the success of their model in India, JustDial is looking at expanding their global footprint, having already launched their operations in the USA. This is perhaps a unique case where an Indian company is looking at capturing the global market and competing on the basis of superior in-house built IT systems. That open source software is a central piece of this story underlies its growing importance as a strategic tool for organisations.

CASE STUDY: JustDial and Open Source

16.1 THE CONCEPT OF OPEN SOURCE SOFTWARE

16.1.1 Creation of Proprietary Software

Proprietary software that comes from large corporations is usually created by a team. Team members first create a design of the software that includes specifying its component parts, based on requirements they have obtained from a client or from the design of an existing software. These parts are then allocated to individual programmers who create the code by writing it from scratch, or by re-using existing modules if such modules are available. Programmers also test the portion of the software they are working on. The code modules are then handed over to testers who have specialised skills in creating a varied set of conditions under which the modules are tested, individually and in a combined manner. The tested code may be sent back to its programmer for further enhancements and fixes, or may be moved on and merged with other modules that have been created by other programmers in the team.

The team functions as a unit and usually works on the software entirely on its own without relying on support from other teams or outsiders. The idea is to create a unique software product that can bring revenues to the creator firm from sale or licencing.

Many argue that manufacturing software is more of a craft than an assembly line operation. Software even today cannot be made by creating component parts independently according to a design, and then assembling them. For, although it is technically possible, it still requires considerable effort on the part of the software creators to ensure that all the parts fit together and function coherently. During the writing of software, errors also creep in the code that is being created. For ensuring smooth working of the software, it has to be rid of bugs or debugged. This is a detailed process that requires skill and experience, and borders on being an art as much as it is a science.

The software created in the above manner is strictly controlled by licencing arrangements that ensure that only the firm that has created the software has rights to distribute, modify and extend it. Users are permitted to use the software but not change, distribute or share it. They do not get to see the source code that was used to create the binary version of the software. Since software is very easy to copy, the licencing the firm employs is designed to protect its interests and restrict use of the software to only those who have paid-up licences. In many cases, the licences too are temporary, and users have to renew them to continue using the software.

An important aspect of the software that is created and sold or distributed in this proprietary manner is that it is given away in compiled form. The detailed source code is compiled into a comprehensive package and then given to users and customers. The source code is not given away. Users may have the ability to customise and configure the software to their own needs but they cannot change what is compiled into the software.

Open source software differs from commercial or proprietary software in one essential aspect – the software source code is available to users. Open source software is given out in compiled form, and may be sold or commercially licenced, but the source code is always provided.

16.1.2 Creation of Open Source Software

Open source software is created by a large community of people who collaborate via the Internet. These people are programmers or designers or testers who have a common interest in a particular type of software, either an application software, such as a word processor, or a system software such as a driver for a digital camera. These people collaborate through online websites such as Sourceforge (www.sourceforge.net).

A typical open source project is initiated by some person or group creating a project page on a website such as Sourceforge and then asking their contacts or others to join in. The design is settled upon by exchanging e-mails and messages, and by jointly working on the specifications. Coding for the design also begins and people submit their code to others to review and test. As the design and coding progress, others may join the project to provide assistance with testing, documentation, revisions and extensions.

At the time of initiation, the persons who originally started the project usually decide the language or platform on which the code will be created, the manner in which it will be distributed and what licencing policies they will follow. This allows those who join later to participate in a project whose ultimate goals are specified in advance. Within these specifications and constraints, the project team may experiment and create the software according to their desires and wishes.

Many open source projects do not succeed. People start the projects with some goals, and then find that they cannot proceed with it, or others do not join them and there is no progress in building the software. Sometimes a project reaches a certain level of maturity, that is, a lot of software is created, tested and found to be functional. But then the project may be abandoned as the creators lose interest and stop working on it or the project is superseded by another that has a better design and potential.

In this sense, the creation of open source software is often compared to the bazaar where many players come to participate, some stay on and succeed while others leave. From this bazaar model, many powerful software packages have emerged that rival and sometimes outperform their proprietary counterparts. For example, many argue that the Apache web server, an open source package, is as good as or better than any proprietary web server.

16.1.3 Who Creates Open Source Software and Why?

Open source software is created by software enthusiasts, professionals, students, professors and hobbyists from all around the world. Many work part-time on projects, contributing to the code or testing or documentation as and when they can. Many students work on projects as part of their assignments in colleges or as part of their summer projects.

Many professional software programmers in large software companies such as Sun Microsystems and IBM also work on open source projects. These are highly paid professionals and experts in their craft. They work on some open source projects as their companies have an interest in those particular projects.

However, the vast majority of developers and programmers who participate in open source projects work part time. They may be employed as software or IT professionals in their day jobs, but find time to contribute to and work on projects as they are very much interested in a particular software. Many economists and psychologists have asked the question as to why so many software programmers, who are highly trained and skilled professionals, give so much of their time to create highly complex software knowing fully well that it will be given away to the community and they will not make any salary or profits from it. Some of the answers that have emerged are discussed below.

1. Programmers and developers work on open source to satisfy a personal *itch*. They may have a desire for a particular kind of software that works in a particular manner, and since they cannot find what they want in the vast amount of software already created, they set out to build their own. This itch is highly personal and individualised, and the developers seek to involve others, via the open source websites, to participate in their project, hoping they will have a similar itch.
2. Many work on open source software to create an alternative to proprietary software. Their objection is to the closed and un-free manner in which proprietary software is controlled and the restrictions placed on its manipulation and re-distribution. Open source software is a reaction to industry control. Programmers create high-quality software to rival the one produced by software companies and distribute it freely, thus diminishing the control of proprietary software.
3. Some argue that open source developers give away software as a gift. Their expectation is that others, who take their gift, will return the favour in kind. In this *gift economy*, the utility of creating and giving away complex software is derived from the return gifts obtained from the community. Open source programmers and developers use only open source products and platforms to build their own code that they give away. They maintain a reciprocity – that of taking and giving gifts.
4. For many young developers, students and starting professionals, participating in open source projects is a way of showing off their programming skills. It is well known that for famous projects, such as the Linux open source operating system, it is very difficult to write code that is finally included in the product. Programmers have to have very strong skills to be able to write code that survives the scrutiny of thousands of collaborators around the world. For those who are involved with projects, it is an achievement that they can boast of and include in their resumes, and this helps them professionally to obtain better and high-paying jobs.

16.1.4 History of Free and Open Source Software

When the first commercial computers were sold by companies such as IBM in the early 1950s, the hardware was accompanied by the instructions to run them, which was the software. The source code for the software was provided as sophisticated compilers had not been invented as yet, and code had to be carefully arranged to be run on the hardware. The computers were very expensive, and only a handful of firms

and organisations could afford them. The main cost was for the hardware, and software was literally given for free to ensure that the hardware would run.

This trend continued, and later as the computer industry grew, more firms began making and selling hardware packaged with software. Hardware gained in sophistication with larger memories, greater speed, and better and faster secondary storage. Software too gained in sophistication, with the emergence of distinct categories of application software and systems software. Specialised departments were created within computer manufacturing firms that dealt with creating software alone. The discipline of software engineering evolved and an entire industry began to grow that focused exclusively on creating software that could be run on the available hardware. At this stage too, the revenues obtained from software were mainly linked to the hardware sale. Software was still not separately licenced and sold.

By the late 1970s, several companies, such as Commodore and Apple, had started to market personal computers, which came bundled with software, and were targeted mainly at individual users, not at large corporations. These computers were cheaper and were affordable by ordinary people interested in computing. In the early 1980s, IBM began marketing the first versions of desktop computers for commercial organisations that could use these in offices for all workers. IBM made these personal computers or PCs, as they are called, as highly modular component-based units that could be easily assembled and also upgraded. A large number of manufacturers were encouraged to make the components for PCs, which reduced their prices and made their market competitive. Software manufacturing too was given to vendors who could build and compete to sell their products. This gave rise to the PC software industry with a large number of players making software for the PCs. This initiated the idea of proprietary software, where software manufacturers would protect their source code by licences or by simply not giving them out.

The non-PC manufacturers, the mainframe manufacturers as they were now called, too had switched to the proprietary form of software for distribution by this time. When hardware was sold, the operating system and applications were given, with the users in some cases having to sign single-user agreements, preventing them from sharing the software or source code with others.

The idea of proprietary software was exploited extensively by Microsoft Corporation that created the most popular operating system for the PCs. Microsoft's strategy was to give away only the compiled code that was bundled with the PC hardware. Microsoft used software licences that permitted only use of the software by its owner, thus disallowing re-distribution or sharing. They also prevented changing the software code by not letting users have access to the source code.

As a reaction to this move in the industry, a pioneer of computing technology, Richard Stallman, started the Free Software Movement in 1983. Stallman was a graduate of the Massachusetts Institute of Technology (MIT) where he had also worked as a software programmer. He created a software licence known as the GNU General Public License (also known as the GPL) that explicitly allowed users of software to use, inspect, modify and re-distribute software. He used the term *free software* to identify with freedom. GPL gave users freedom to use and work with software as they wished. Stallman also created a host of applications, mainly consisting of utilities to build software that were given under GPL, thus allowing anybody to freely use the software.

Initially the free software created by Stallman and his collaborators was available to only those who had access to them physically, on media such as floppies or tapes, as those days the Internet was restricted to a few universities, commercial organisations and government departments. Only employees of these organisations could get the free software off the Internet. The growth of the free software movement and the Internet are intertwined. In the 1990s, as the Internet became commercial and available to people around the world, so too the free software movement grew and gained strength. Computer programmers from around the world could now access the free software created by others and build on these.

The operating system Linux originated on a call given by a Finnish graduate student to build such a system using tools already available under GPL. His call attracted a lot of developers to work on the system. Linux grew exponentially and further attracted a lot of users and developers. It started a trend of releasing updated software versions frequently, about every 6 months, and created a culture where users expected new versions frequently and looked forward to them.

By the mid-1990s, there was immense commercial interest in Linux as it was a very stable operating system and rivalled any of its proprietary competitors. Some entrepreneurs decided to create a business out of distributing and servicing Linux but decided to use a new term to describe the manner in which Linux was created and distributed. They chose the term *open source* to signify that Linux is always provided with the source code. Many industry analysts agreed that this provided a better brand for this kind of software and promoted open source actively.

Open source software, thus, is not very different from free software, and in fact most open source software use GPL as the user licence. Nowadays there are many licences that are used to distribute open source software and they are variants of the original GPL (some differ significantly). The idea of freedom of use, modification and re-distribution is, however, retained by most open source software, making them highly competitive in the commercial market.

By late 2010, open source software had gained immense popularity and was used around the world in a large number of organisations. Packages such as Apache and Firefox are very popular. Open source and free software are phrases that organisations and individual users around the world recognise as signifying both high quality of and low (or often zero) cost of software.

16.1.5 Debate around Open Source Software

Open source software poses a competitive threat to proprietary software. It works on practically all types of hardware. Organisations can easily download open source software, distribute it freely to all employees and make modifications or changes to suit their needs. Faced with a threat to their business from open source software, some rival proprietary software firms have engendered a debate around the merits of open source software, arguing that the software may be free to acquire but its *total cost of ownership* is higher. The phrase 'total cost of ownership' refers to the final costs of acquiring, installing, customising, maintaining and upgrading software.

Those favouring proprietary software argue that open source software only reduces the initial cost of software acquisition, but finally organisations have to pay

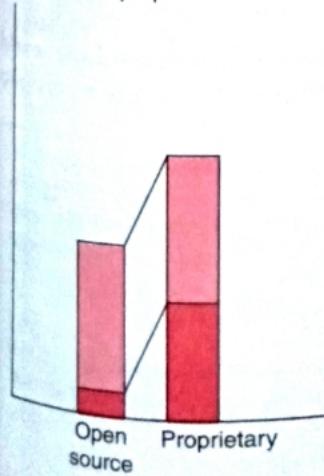
out more to use the software than they would if they used proprietary software. The genesis of this argument is in the fact that many organisations, which already use proprietary software, have a *lock-in* to it. Lock-in implies that the organisation has invested money in acquiring the software, trained its employees to use the software and created routines and processes that use the capabilities of the software, and may have extended contracts with service firms that help in maintaining the software. This investment constitutes a lock-in that is quite high in monetary value. The *switching* cost of moving to another software type, like open source, would be quite high as the organisation would have to re-learn the software, re-create its processes and re-train its employees.

Furthermore, the proponents of proprietary software argue, since proprietary software is used more on the desktop, more people are familiar with using and supporting it. For example, proprietary software such as Microsoft's Windows operating system has about 90% of the market share for desktop and laptop operating systems. There are a large number of service firms who can help with using, installing, upgrading and maintaining Windows. This helps organisations reduce their costs of using the Windows software as opposed to open source software.

Proponents of open source software argue that the low initial costs of acquisition for the software, which is zero in most cases as the software can be downloaded free of cost, actually offsets the cost of getting support for and training users on the new software (see Fig. 16.1). Furthermore, the long-term costs of upgrades and maintenance are lower as support is available for free, albeit not immediately (support for open source software is available online on websites and mailing lists).

Possibly, the strongest argument for using open source for a desktop operating system is that of security. It is well known that the Windows operating system is susceptible to virus attacks. In 2008, there were more than a million known viruses for

According to open source proponents



According to proprietary software firms

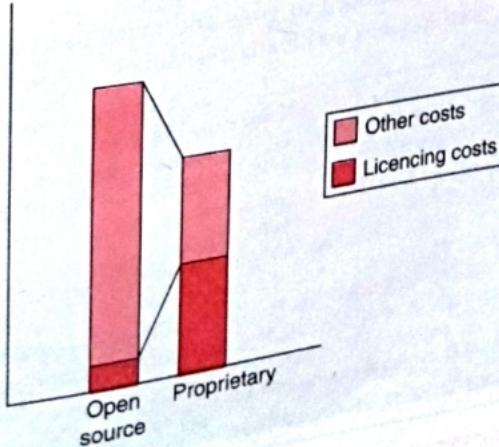


FIGURE 16.1 Costs of owning open source versus proprietary software.

Windows. Comparably, for the Linux operating system, there are hardly any viruses and there is no need to use any anti-virus software either. For Windows users, virus-protection software costs a lot to acquire and maintain, but the most damaging aspect is the losses incurred by organisations that have been struck by viruses and lose hours and days of productivity.

Over the decades of its existence, the free software and open source movement has produced many highly stable and efficient programs that are widely used. Some of these packages are described in the sections below.

16.1.6 Popular Packages

16.1.6.1 Linux

The Linux operating system was initiated by Linus Torvalds, a Finnish graduate student, in 1991. Torvalds was keen to build an operating system of his own as the ones available at the time, to run on desktop computers, were either proprietary, such as the one sold by Microsoft, or were not powerful enough (such as an operating system called Minix). Torvalds wrote the basic structure of an operating system using free software tools built by Richard Stallman. He then sent out an e-mail message on the Minix user group (a community of enthusiasts working with the Minix operating system) asking if anyone was interested in the operating system he was developing. A portion of the text he sent out is quoted below in Fig. 16.2.

The response to the message sent out was immense. Enthusiasts around the world responded to this call by first downloading the system Torvalds put out on a website and then by adding, commenting, testing and providing bug fixes to the code. The system was called Linux and became very popular.

By the late 1990s, many commercial firms starting showing interest in Linux, and companies such as RedHat and Caldera were formed that sold a commercial version of Linux. Although, Linux was still open source, and anybody could access all the code for free, these companies built some extensions on their own and also specialised them for certain kinds of applications for which customers were willing to pay their commercial price.

Linux continued to grow and many users and organisations around the world adopted it. Linux was made available in many languages and an extensive portfolio

```
Hello everybody out there using minix - I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things). I've currently ported bash (1.08) and gcc (1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)
Linus (torvalds@kruuna.helsinki.fi)
Linus Torvalds, 25 August 1991.
```

FIGURE 16.2 Part of the message sent out by Linus Torvalds calling for participation in the Linux project.

of applications was developed for it. Linux also grew into many distributions. These distributions essentially work on the same core kernel of the operating system, but they have built their own specialised user interfaces and applications depending on the needs of the community they are serving. Some details about Linux are provided in Box 16.1. Linux is now used in all variety of hardware: Servers, desktops, laptops, mobile phones, personal digital assistants and portable audio players.

It is used to create massive parallel processing supercomputers as also small devices that have embedded operating systems. Linux development continues at a rapid pace, with a new version of a kernel being released about every 6 months. The number of applications and packages that are available for Linux continue to multiply. A screenshot of a typical Linux user interface is shown in Fig. 16.3.

BOX 16.1 Some facts about Linux in 2009

1. The 4.0 version of Debian Linux had 283 million software lines of code in 2009.
2. The Linux core kernel had 300 active developers in early 2009. There are thousands of supporting developers.
3. A website dedicated to linux distributions, www.linux.org, lists 220 distinct distributions.

Source: Wikipedia and The Linux Foundation, www.linuxfoundation.org (accessed on July 2009).



FIGURE 16.3 Image of a Linux desktop.

16.1.6.2 Mozilla Firefox

Firefox is an open source browser that is used to view pages and multimedia content on the World Wide Web. Firefox is the second most popular browser in the world today (July 2011) with a market share of 27% (the most popular browser is Internet Explorer, a proprietary source browser made by Microsoft Corporation). Firefox runs on all the major desktop operating systems, including Windows, Linux and Mac. Firefox is also available in 75 languages.

The Mozilla Firefox browser derives from the erstwhile Netscape browser that was one of the first browsers created after the World Wide Web was launched. Netscape was very popular in the 1990s and was eventually replaced almost entirely by Microsoft's Internet Explorer. Netscape was one of the first projects, which started as a proprietary project and then was made open. After Netscape's code was made open source, it was sponsored and further developed by the Mozilla Foundation (a non-profit organisation), leading to the current version of the Firefox browser.

16.1.6.3 Apache

Apache is a web server, a software that allows websites to serve pages to visitors. Apache is arguably one of the most successful open source projects ever, as it is unrivalled in the proprietary software world. The Apache server is responsible for the initial rapid growth of the World Wide Web.

As such, the Apache server is an HTTP server, where HTTP is the hypertext transfer protocol, the protocol used to transmit data on the World Wide Web. An Apache server can be installed on any computer connected to the Internet, and with an appropriate IP address, the machine can begin to act as a web server. The server is very easy to install and configure and can run on a wide variety of operating systems. It is maintained and updated by the Apache Software Foundation.

Apache, in March 2011, hosted close to 180 million websites and has 60% of the total market share of the Web.

16.1.6.4 Open Office

OpenOffice.org is the name of an *office suite* software application. An office suite software consists of applications that are typically used in the office environment and include a word processor, a spreadsheet, a presentation software, a database, and some others such as a graphics creator and calendar. OpenOffice.org being an office suite contains most of these applications.

OpenOffice.org is derived from a product called StarOffice that was owned by the firm Sun Microsystems. Sun spun off OpenOffice in 2000 to create an open source product as a direct competitor to proprietary office packages. (The name OpenOffice.org was used for the package as the phrase Open Office was already being used by another product.) Later, Sun Microsystems was acquired by Oracle Corporation, so now Open Office licenses are held by the parent firm.

In 2009, OpenOffice.org had about 9 million lines of code in about 30,000 source files. It is available in 110 languages and had been downloaded 98 million times, with an estimate of 15–20% of the market share for desktop office applications.

One of OpenOffice.org's greatest strengths is that it is compatible with a number of other office applications, including Microsoft Office. It can open and create files in formats of other office products. On its own, it relies on open standards and formats for data files. For example, its word processor, Writer, saves files in the Open Document Text format (files with the extension .odt) which is a universal open standard for text files.

Due to the availability of the source code, a number of variations of the basic office suite have also been released. These products extend the basic functionality of OpenOffice.org in some manner, either by speeding up the programs, or by adding features that make them attractive to special kinds of users. One example is the IBM Lotus Symphony, which is released by IBM, and is based on the OpenOffice.org source. Symphony is packaged by IBM with other products from their portfolio, giving their customers a better bundle of packages.

16.1.6.5 GIMP

GIMP stands for the GNU Image Manipulation Program. It is an image manipulation and raster graphics program that is entirely open source and is used widely by professional graphic artists and amateurs alike. GIMP can be run on many popular operating systems and is currently available in about 50 languages. GIMP is widely perceived to be a free substitute for the professional quality proprietary program called Adobe Photoshop. Many use GIMP as such, as it can manipulate and touch up digital photographs. It uses advanced techniques of image manipulation. Many businesses have used the source code to modify GIMP for their own particular needs.

16.2 OPEN SOURCE LICENCES

16.2.1 GNU General Public Licence (GPL)

The GNU licence is one of the oldest and most used licences for software in the open source community. It stands as a guide for other licences, many of which have been inspired by it. GNU stands for Gnu is Not Unix, which is a recursive name given by Richard Stallman who created the licence.

The philosophy of the GPL is to allow free and unhindered usage of software, and to ensure that this is perpetuated. When any software is given the GPL, it ensures that anybody is free to use the software, see its source code, distribute to as many users as desired, and change the source code to suit one's needs. Seen differently, the GPL 'protects' users from anyone taking away these freedoms to use, distribute and modify the software.

One particular aspect of the GPL is regarding distributing modified software. When a software that is already under the GPL is modified, the new modified version is known as the *derived work*. The GPL states that if the derived software is distributed, it has to be distributed under the same licence. It cannot assume a new licence that violates GPL's essential freedoms. It is important to understand that if a person or organisation uses and then modifies GPL software, then they can continue to use the modified version as long as they like. If they choose to distribute the derived software then the GPL ensures that the source code has to be distributed also. This protects the freedom of the receiving entity to see, use and modify the source code.

The GPL does not impose any restrictions on sale of software. Software created under the GPL licence can be sold, provided the source code is distributed also. Furthermore, the author of the software can retain the *copyright* to the software under the GPL. This implies that organisations or individuals who create GPL software and distribute it retain possession of the intellectual content of the software (thereby preventing others from claiming the software as theirs).

The GPL allows copyright but prevents patenting of software. A *patent* is a special protection under the legal framework of most countries, where a patent holder has complete rights over the idea patented; anyone who may later want to use a similar or the same idea has to pay royalties to the original patent holder. Patents in software are highly restrictive and curb innovation. Many developers around the world are opposed to software patents. To understand why, consider a popular application that many people use – the word processor. When word processors were first developed, the idea of word processing was not patented. This allowed many firms and individuals to design and develop various kinds of word processors and either sell or distribute them freely. Had the first word processor been patented, none of the later developers would have had any incentive to create new and original word processors, as they would all have to pay royalties to the patent holder for any revenues they obtained for their own work.

16.2.2 BSD Licence

BSD stands for the Berkeley Software Distribution. This licence is one of the most permissive licences, which allows free distribution of the software and derived works with or without the source code. The licence specifies two main conditions for software users – the software may be distributed freely in source or binary form, but the copyright notice of the original copyright holder has to be retained, and the name of the copyright holder cannot be used to promote any derived works. Furthermore, the licence clarifies that there are no guarantees of performance or warranties associated with the software under the licence.

The BSD licence prompted many open source developers to adopt it as it allowed them to create software that they could distribute as they pleased. There are many variants of the BSD licence, which borrow the main ideas and then add some further clauses of their own. These variants include – the MIT licence, the NetBSD licence, the FreeBSD licence and so on.

16.2.3 Apache Licence

The Apache software licence is somewhat different from the above licences. This licence also allows free distribution, use and modification of the software to create derivative works. It further allows licencing of software along the lines of any other licence. Thus, those who use Apache licenced software can modify the code, create a new application, and then licence it differently. This extends further to create patent restrictions.

The main difference between Apache and the GPL is that the GPL explicitly forbids patenting of software derived from its licence, whereas Apache allows this. This has made Apache particularly attractive to many developers who want to take free software and extend it with their own work, but want to make it proprietary to extract revenues from it.

As with the others, the Apache licence does not warranty the performance of the software licenced by it, and also limits the liability arising from the use of the software. The latter provision means that if people download and use Apache licenced software and suffer some financial damages because of that (or related to the software), they cannot sue the software maker for the damages. This aspect of the licence is common for all the open source licences.

There are many other open source licences that are used by the community, including the GNU Affero General Public licence, the Mozilla Public Licence and the Common Public Attribution licence.

John M. Gossman

16.3 OPEN SOURCE IN BUSINESS AND GOVERNMENT

16.3.1 Open Source in Business

Open source software has been widely adopted in business and government. Many prominent information technology businesses such as IBM, Sun Microsystems (later a part of Oracle Corporation), Hewlett-Packard, Oracle and many others saw the value of open source quite early and adjusted their business models and products to accommodate the value and power of open source. IBM, for instance, actively supports the development of many open source products, and has contributed extensively to their growth and popularity. Some businesses such as Google and Yahoo that are very large Internet-based businesses have developed a significant part of their infrastructure using open source software components.

Businesses not in the information technology industry use open source products quite widely. The OpenOffice software is used by businesses for the routine desktop activities, along with packages such as the Firefox browser and the Linux operating systems. There are also a large number of specialised open source packages that businesses use as low cost, high value and quality alternatives to proprietary software. Packages such as SugarCRM (a customer relationship management software), Compiere Open Source ERP (an enterprise resource planning software), GNU Cash (an accounting software) and MySQL (a database software) are quite popular.

Many firms have now set up business models based on open source software. Some examples of how firms are doing this are as follows:

1. **The Dual-Licencing Model:** In this business model firms create open source software that they release to the community under an appropriate licence such as GPL. Community members use and modify and extend the software, thereby contributing to its growth. The creators, however, retain the copyright and also sell the software on a second, proprietary, licence, which is a closed source (see Fig. 16.4). Businesses that want the open source software customised to their needs have the copyright holders modify the software and give it to them on a proprietary licence, thus protecting their business interests. MySQL, the open source database software, is managed by its creators on such a dual licence. Businesses that buy the proprietary version have an advantage of a customised software (which is of lower cost than other proprietary softwares); the firm selling the software has revenues; and the community benefits as it has a free version that it can use.

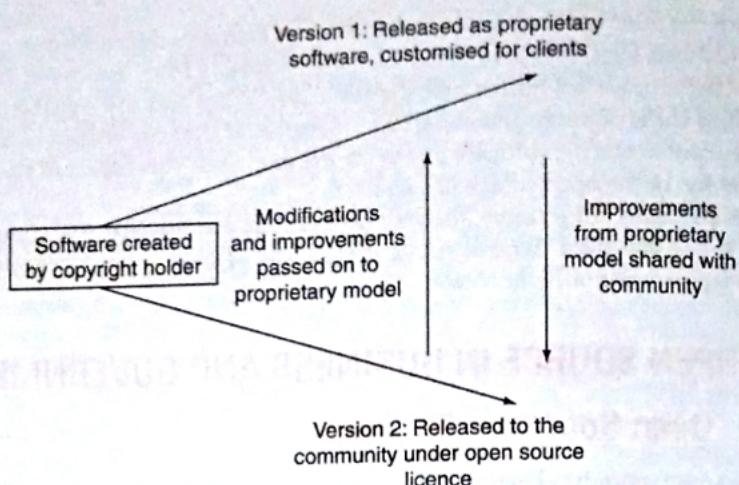


FIGURE 16.4 Dual-licencing business model for open source software.

2. **The Service Model:** In this model, some firms take an open source software, like a database or an operating system, and create a highly specialised service package around it that they can sell. For instance, one firm contracts with large business organisations to support and maintain one particular distribution of Linux. The advantage to the client is that it has a skilled organisation supporting its needs across its offices spread around the world, and provides support for infrastructure.
3. **The Product Model:** In this model, firms create products based on extensions of open source software and sell them as commercial software. For example, RedHat Linux sells a portfolio of Linux distributions that it has created on its own. Its model is to sell the product and along with it a service that clients find very useful. RedHat generates revenues by creating, selling and supporting operating systems specialised for mission critical needs. RedHat also maintains a basic community version of the software that it supports and extends. The benefit to the community is that it can draw on RedHat's immense contributions, and for RedHat the benefit is that it can draw on the community's extensions and features.
4. **The Hardware Model:** Some firms make or assemble specialised hardware that runs open source software. Their main revenue source is from the hardware sales, and they support users with the open source software which has been customised for their hardware. Examples of such firms include Nokia, the manufacturer of mobile phones, and Acer, the maker of laptops and desktop; both use open source operating systems for their hardware.

16.3.2 Open Source in Government

Governments in many parts of the world have sought out open source as a high-quality and low-cost alternative to proprietary software. In some cases, national governments have passed laws advocating use of open source software, wherever this is

possible, over proprietary software. And in other cases, state, city and local governments have taken steps to adopt open source software.

In developing countries, in particular, open source software has found many adherents. For example, in many states in India, including Kerala, Assam and New Delhi, local governments have sought to not only adopt but also actively promote knowledge about and value of open source software. In China, Brazil, Indonesia, Philippines, Cambodia and Malaysia, support for open source software is quite strong and the governments actively promote its use. In Spain, Germany, the USA and Sweden, several state and local governments have clearly articulated policies to adopt open source software.

Government departments have found several advantages in adopting open source. Some of these are listed below:

1. Open source software is invariably of lower cost than proprietary software. This allows departments, which always operate on budgets, to allocate limited resources to more abundantly available software. In particular, costs of upgrading software and per seat licences are low, which helps reduce costs in the long run as departments grow.
2. Open source software for developing countries has an additional advantage related to lower cost, that of reducing the dependence on foreign goods, thereby reducing the outflow of foreign exchange.
3. Open source software allows easy experimentation. Many departments around the world have 'discovered' many things they could do with computerisation simply by trying out different things with open source software at a very low cost. This allows breaking out of rigid and bureaucratic processes by seeing the value of and experimenting with information technology.
4. Open source products are mainly based on open data standards and open formats (this is discussed below). For governments this is crucial as public data can be maintained on open standards across many departments, and state and local bodies. This data can be shared and maintained in perpetuity within a format that is open, and hence not tied to any particular vendor company's fortunes.
5. Owing to weak piracy prevention in many developing countries, open source represents a way to curb piracy and encourage users in all walks of the economy to use legal software. In China, one explicit goal for promoting open source was to curb the menace of piracy.

16.4 OPEN DATA STANDARDS AND THE OPEN COMMUNITY

16.4.1 Open Data Standards

Many software programs store data in a format that is specific to them. For example, a word-processed document that is created using the Microsoft Word program is stored in the .doc format. This format is specific to Microsoft Word. Other programs may read the .doc format, but to create files in this format, they would have to obtain a commercial licence from Microsoft.

A standard in data is a common format in which the data is stored in files. Standards are created by standard setting bodies. These bodies are constituted by the firms or organisations that are responsible for creating programs that use the standard. In software, many standards are maintained by the International Organisation for Standardisation that maintains the ISO standards. The ISO creates the bodies that create the standards and uphold them. Standard setting is a detailed and drawn out process in which many parties have to agree as to what the standard will be and how it will be maintained. Another organisation that also promotes standards is the Internet Engineering Task Force (IETF) that has standards related to the Internet and its services.

Proprietary standards are those owned by some firms, and to use them others have to pay royalties. Open standards are those that are free to be used by anyone without paying royalties. Thus, many commercial firms, which create programs, are interested in creating proprietary standards to extract royalties. However, there is a risk that proprietary standards may not be widely adopted and so the firms could lose out. On the other hand, open standards may be widely adopted, but there is not much royalty income from them. Some firms work mainly with open data standards so that they are widely adopted, and the firms can then obtain revenues from complementary products.

One of the most popular data standards that is used for file sharing and exchange is the Portable Document Format or the pdf. Files that are named as 'Homework.pdf' are created with the pdf data standard. Pdf is an open standard though it was created and is owned by Adobe Incorporated, a commercial firm based in the USA. Adobe initially created the format, which was used by its own programs. Adobe also created a reader for pdf files, called the Adobe Acrobat Reader, that was a free download for all to use and that enabled the easy reading of pdf files. Pdf became a very popular method of exchanging files within organisations and also across the Internet because of the availability of the free reader. Also, now that pdf is an open format, there are many open source and proprietary programs available that can create pdf files.

Open standards are important for organisations of all types. Data files created with open data standards can be maintained in *perpetuity*. This point is important. As has been seen with software of all kinds, the evolution and changes in them is fast and sometimes abrupt. Data files created with a certain kind of, say, a word processor are sometimes unreadable a few years later as the version of the word processor has changed and does not support the earlier data format. Also, the firm that created the word processor may go out of business and may not be able to create new versions of software that can read the old files. If the data is in a proprietary format then other software makers may not want to create programs that can read the old files, or may charge a very high price to do so. However, with open standards, it is possible for many firms to create readers and provide them at reasonable prices.

For governments, in particular, open standards are important as data can be conveniently stored in perpetuity. Governments have to maintain very large volumes of data, pertaining to transactions by the entire population, and hence need to have the assurance that the digital data they have will be readable in future by programs available at reasonable prices.

Another advantage of open standards is that they promote a culture of sharing. Readers can be created for different operating systems and in different human languages that allow data in open standards to be read. Multimedia data in audio,

video, image and graphics applications can be shared easily if they are in open standard files.

It is also important to mention here that proprietary standards are also very popular. The best example is the MP3 data standard for audio files. MP3 audio files are almost universally known. Millions of MP3 files are available for sharing or purchase on the Internet, and there are thousands of software programs and hardware players (such as the famous iPod) that play MP3 files. MP3 is a patented format, and to create players or encoders for it, firms have to pay a royalty to the owners of the patent (which is the Fraunhofer Institute in Germany). Many software and hardware vendors have sought to remove MP3 as the *de facto* standard for music files and promote other and better standards. However, the MP3 popularity is so high that its usage has grown and not declined. Open standards for music files such as Vorbis are also popular and are actively used and promoted by the open source community.

Open data standards allow many commercial firms to create and sell creators and readers of data files. This competition improves the quality of the programs available and also reduces the price of the products.

Open standards also enable the continuous evolution and growth of the data formats. With a community participating, as happens in open source, the number of varied interests and requirements is very high and that contributes to the growth of standards.

16.4.2 The Open Community

The open source community represents one of the largest open communities in the world, which participates in shared goals. This community spans across nations, geographies, languages, gender and economic differences. This community collaborates to engineer some of the most complex systems ever created. And yet, it remains largely physically invisible, amorphous and transient. This is an open community that is beginning to define, in many ways, the way we live, work and share.

It is argued that the open source movement represents a *digital commons*. The commons are a shared resource whose value lies in the fact that a single individual or group's consumption of the resource does not preclude others from using it. A city park is a commons, as those who go there to walk or play do not prevent others from doing so simultaneously (a private good, like a car, in contrast, if being used by the owner, cannot be used by others). The open source community has created a digital commons as it allows anyone to use and share the resources it creates; the use by one does not block anyone else from using the same. The philosophy of the GPL is to preserve this aspect of the commons.

Open standards too help sustain the digital commons. When data is in an open standard it can be shared with all. Many universities around the world have now opened their course content and materials for sharing over the Internet. Many nations across the world, and including India, have passed laws mandating that data and information pertaining to government activities has to be made freely available to the public. All these efforts finally rely on open data formats by which data can be shared and easily read by all.

The Creative Commons (CC) Licence was established as a counterpart to a software licence for creative works. Under the CC licence, any creative work - a piece

of prose, a video, an artwork, a musical piece – can be licenced to be in the public domain. This means that anyone can read, see and listen to the work, but there are restrictions on modifying or distributing it. The CC specifies that distribution of the work is possible, but the copyright and licence notice has to be retained, and further the distributed work cannot be modified in any way. However, there are many variations that are possible within the portfolio of CC licences that may allow (or disallow) modification, permit creating derivative works, or allow commercial distribution. The CC has been widely adopted by authors and artists across the globe. This is in the spirit of the commons, enabling the public to consume the works without restricting others from doing so.

The culture of digital commons has permeated in other areas of creative collaboration such as collaborative writing of books and making of movies. The idea that digital goods can be shared is infective and has been widely adopted. Many of the activities of collaborating across cultures and sharing are driven by a motive to learn. Individuals who participate want to learn about and contribute to what interests them, as much as they want to consume what is available. This aspect of the open community is the most valuable, also the most powerful.

Chapter Glossary

Proprietary software Software that is created by commercial firms and released under a licence that restricts the software use, sharing, modification and distribution. Such software is usually released in compiled form, and source code is not provided.

Bazaar Model The open participation of many people around the world to develop open source software.

Personal itch Refers to the desire by programmers to create a software tailored to their special needs.

Gift economy A situation in which people give gifts with the expectation that others will in turn give them gifts too.

Free software Software that is licenced under the GNU General Public licence which assures that users of the software can have access to the source code, use it, modify it and distribute the modifications.

Open source Software that is released along with the source code, and usually carries a GPL or similar licence.

Total cost of ownership The costs incurred by organisations when they acquire and use software. In addition to the cost of buying the software or the

licence, these costs include expenses on maintenance, training, support, upgrades and migration.

GPL Acronym for the GNU General Public Licence that ensures that software distributed with this licence is not patentable and allows the user to use, modify and distribute without any restrictions.

BSD Acronym for the Berkeley Software Distribution licences, which are a set of permissive licences allowing creators and users of software to freely distribute and use software, with the only restrictions of retaining copyright notices and of not having any warranties.

Dual-licencing A situation where creators of a software release it under both an open source and a proprietary licence.

Data Standard An agreed upon and shared format by which data is stored in files.

Perpetuity Refers to an endless time horizon.

Creative Commons Licences A set of licences that allow works of art, such as books, poems, songs, to be released in the public domain for free viewing and consumption. The restrictions are on commercial use and modification.

Review Questions

1. What are the main differences in the manner by which proprietary and open source software are created?
2. Who are the people who participate in creating open source software?
3. Why do people create open source software?
4. What are the main arguments of the debate between the proponents of proprietary and open source software?
5. What are the main provisions of the GPL?
6. How is the BSD different from the GPL?
7. What are the provisions of the Apache licence?
8. What are the four main models of doing business with open source software?
9. Why do governments choose to use open source software?
10. What is an open data standard? Why is it preferable over a proprietary data standard?
11. What is the open community?

Research Questions

1. What are the different applications for which commercial firms have adopted open source software? Visit some local firms to do the research.
2. Read the JustDial case. What were the strategic reasons for which JustDial adopted open source?
3. The Android operating system has become very popular and is used widely on mobile phones and tablet computers. What licensing model does it follow and why has it become so popular?
4. Explore the business model of the RedHat firm that sells a Linux product by the same name. How does RedHat earn revenues? Why is it successful?

Further Reading

1. Read an article 'Computer viruses hit one million' available at: news.bbc.co.uk, 10-04-2008 (accessed on July 2009).
2. De, R. (2008) FOSS Business Models for Developing Countries in Asia, IOSN South Asia Node Report.