



Mustansiriyah University

OPEN-SOURCE SOFTWARE



Chapter One **Open-Source Software:** **Definitions and History**



Zied O. Ahmed

Open-Source Software: Definitions and History

1.1 Definition of Terms

Open source software is software that must be distributed with source code included or easily available, such as by free download from the Internet.

The source code should be in the same form that a programmer would actually use to maintain it—not, for instance, a generated, obfuscated, or intermediate code form.

The license of this software will not restrict others from distributing the code or modifications and derived works under the same terms. It will not discriminate against people or fields of endeavor.

The Open-Source Definition (OSD) was originally written by Bruce Perens for Debian Linux and was completed in 1997. It is an established definition of open source that is simple without being too simple, and it includes several licenses that are acceptable.

Some people call software like this, distributed under licenses such as the GPL, Apache, or Mozilla licenses, “Open-Source Software.”

Others call it “Free Software.” Yet others, particularly in Europe, call it “Free/Libre or Open Source Software” (FLOSS). Capitalization of these terms varies. There are other expressions in use also, but the most widely used term seems to be open source.

The alternatives to open source can be called *proprietary software*, *commercial software*, or, alternatively, *nonopen*, *nonfree*, or *closed software*.

We will refer to software that does not completely meet the OSD as “*closed code*.”

If a piece of software (a product) combines open and closed elements, such as Apple OS X, it will be defined as closed. So, any software product, as licensed, will be defined as either open or closed.

There is a **third** term that we will use in two special cases. Some software is offered under different licenses. It may be offered to some groups as open source and to others as closed. Note that to meet the

terms of the **OSD** (to be considered open source), this must be at the user's discretion. This is a *hybrid licensing model*.

We can also apply the term *hybrid* when looking at a group of products. If a group of products is purchased or installed together, some of which are open source and some closed code, we will call the result a hybrid system.

An example would be IBM WebSphere, which includes Apache (open source) and some other products that are closed code.

First, we will deal with a few frequently asked questions.

1.1.1 What Is Free Software?

Many people prefer the term *free software* to *open-source software*. The term free software dates from 1984, when the idea and arguments for it were first published by Richard Stallman. The idea is not that software should be free or available at no charge, but that it should be free so you can review it and change it as you need to.

The term *open-source* dates from 1997, when a group of people, decided that the term *free software* and some of the arguments employed in support of it were making the idea less attractive to many businesses.

They decided, as a marketing decision, to emphasize technical and practical advantages of open source software rather than arguments from principle.

1.1.2 What Are Good Examples of Open Source?

Flagship products of open-source software include:

- **The Apache Web server**, with a share of 65 percent of installed worldwide Web servers and still growing
- **The Linux operating system**, used on millions of servers, which demonstrates that no system is too large and complex to be developed as open source
- **The GNU C/C++ language suite**, used to build Linux and Apache and thousands of programs on almost every operating system

1.1.3 Is It Necessary to Adopt Open-Source Wholesale?

It is possible to assemble a complete “platform” from open source software, and many of the most popular open source programs work well together.

Some small startup companies have done this. For most people, open-source solutions will be adopted one product at a time. Most adopters will not be able to, or desire to, discard all closed code

software at this time, and there is no reason they should need to. Many new solutions, such as IBM WebSphere, contain a mixture of open software and closed code.

1.1.4 Does “Open Source” Mean Linux?

It is possible to adopt open source without using Linux at all. Many developers, for example, may use open source on Windows. On servers, for example, Apache is an open-source Web server that runs on Windows and UNIX servers and IBM mainframes as well as Linux.

On workstations, OpenOffice rivals Microsoft Office in functionality and can be installed from CD or the Internet in less than five minutes for no cost on Windows desktops. The Mozilla open-source browser can be chosen, for instance, to standardize on a single browser for Windows and the Mac.

1.1.5 Does Open Source Require Different Business Methods?

It is not necessary to adopt an open-source approach to development or distribution in order to use open-source software, and most organizations do not.

Many companies develop closed code software products that run on open-source software and sell them for a profit; this includes almost every large software company today—IBM, HP, Oracle, SAP, and Apple.

Other companies develop applications that run on open source for their own business purposes—for instance, Google, Amazon, or eBay—without feeling in any way restricted by open-source licensing or development models.

If you change or extend the open-source software and redistribute it, you may have responsibilities under the license. In that situation, it is necessary to look carefully at the license and choose one that works for your business model.

1.1.6 Will All Systems Be Open Source One Day?

Given current installed systems and sales trends, we can be certain that over the next few years *open source* and *closed code* software will coexist. In the long run, there is reason to expect some balance. There are advantages to open source but also limits.

1.2 A Brief History of Software

In this section, we will look at the closely entwined histories of open-source software, **UNIX**-like operating systems, **Linux**, the **Internet**, and the **World Wide Web**.

1.2.1 Early Years

In the 1950s, the first modern business computers were introduced, based on work done during World War II. These were practically one-of-a-kind machines, running custom programs written in binary code, later assembler language. They were extremely expensive and it was thought by some (such as a famous IBM market estimate) that only a few might ever exist.

In the 1960s, commercial mainframes were introduced by several companies. These were custom programmed for applications, usually in *COBOL* or *FORTRAN*. System software was developed in assembler language.

Prior to the 1970s, software was, essentially by default, “open” or “free.” It had grown out of the scientific research community, where information was generally shared, and nobody had thought of an alternative. Software was developed by vendors, or by user companies, to meet a particular user problem, and was then freely

distributed by user groups or computer vendors to other users who might have the same problem. Since software only ran on systems from a single vendor, a good software product could help to sell the platform.

1.2.2 UNIX

At the beginning of the minicomputer period, the C programming language and UNIX operating system were developed together at *AT&T* and distributed through noncommercial channels.

The history of UNIX code dates back to 1969, when it was first developed. It is a long history and an involved one, since the code has had several corporate owners and has forked and even been reunited on more than one occasion.

UNIX was written in C and constructed as a set of small programs that worked well together.

By 1974, the UNIX system, including the C language and tools, had been ported to several platforms.

AT&T was not at the time in the computer business, and versions of UNIX were distributed freely, particularly throughout the academic and scientific communities, until Version 7 in 1978.

AT&T stopped publishing UNIX source code in 1976, and after 1978 the distributions of UNIX from Bell Labs ceased.

Earlier versions of UNIX have been placed in the public domain and are often used in teaching operating system principles.

1.2.3 Linux

In 1991, a Helsinki University student named Linus Torvalds began developing a free UNIX kernel for Intel-based PC systems. It was initially based on Andrew Tanenbaum's Minix system, which was a teaching system based on early UNIX source code and published as a book.

From early on, Linus began using the "hacker" model of software development, and this was surprisingly successful.

As early as 1992, by combining Linux with GNU, a complete operating system was available.

1.2.4 The Internet

The Internet is surprisingly old. The first ARPANET paper was published at an ACM conference in 1967.

By 1969, there were Internet nodes at UCLA and Stanford.

By 1972, the file transfer protocol (FTP) and Internet mail addressing had been invented and the system was publicly demonstrated in Paris.

The open-source mail routing program Send mail was originally written by Eric Allman at Berkeley in 1975. It is probably the oldest open-source product still in widespread use, since it predates BSD and GNU efforts by years.

The Internet switched to the full TCP/IP protocol in 1981. At that time, Berkeley Internet Name Domain (BIND), the open-source program that implements DNS and maps names such as “news.google.com” to Internet addresses such as “10.1.203.45,” was written.

This program runs on 95 percent of Internet name servers and all the root DNS servers. Those servers have been accessed by every

transaction between machines on the Internet from 1981 through today, so it is probably the most widely used program on the Internet.

From 1981 through the 1990s, the Internet, as measured by connected systems, users, and information transfer, was approximately doubling every year.

Zied O. Ahmed