



# A History and Politics Lesson

**L**inux is more than just software. It's an entire community of users, and as such, there's a detailed social history behind it. In this chapter, we'll look at the origins of Linux, both in terms of where it came from and the people who make it.

You might be tempted to skip this chapter and move on to the information about installing Ubuntu. To be fair, nothing of vital technical importance is mentioned here. But it's important that you read this chapter at some stage, because Linux is more than simply the sum of its parts. It's far more than simply a set of computer programs.

If nothing else, this chapter explains the fundamental philosophies behind Linux and attempts to answer some of the often-baffling questions that arise when Linux is considered as a whole.

## In the Beginning

Linux was created 16 years ago, in 1991. A period of 16 years is considered a lifetime in the world of computing, but the origin of Linux actually harks back even further, into the early days of modern computing in the mid-1970s.

Linux was created by a Finnish national named Linus Torvalds. At the time, he was studying in Helsinki and had bought a desktop PC. His new computer needed an operating system. Torvalds's operating system choices were limited: there were various versions of DOS and something called Minix. It was the latter that Torvalds decided to use.

Minix was a clone of the popular Unix operating system. Unix was used on huge computers in businesses and universities, including those at Torvalds's university. Unix was created in the early 1970s and has evolved since then to become what many considered the cutting edge of computing. Unix brought to fruition a large number of computing concepts in use today and, many agree, got almost everything just right in terms of features and usability.

Versions of Unix were available for smaller computers like Torvalds's PC, but they were considered professional tools and were very expensive. This was in the early days of the home computer craze, and the only people who used IBM PCs were businesspeople and hobbyists.

**Note** Linux is a pretty faithful clone of Unix. If you were to travel back in time 20 or 30 years, you would find that using Unix on those old mainframe computers, complete with their teletype interfaces, would be similar to using Linux on your home PC. Many of the fundamental concepts of Linux, such as the file system hierarchy and user permissions, are taken directly from Unix.

---

Torvalds liked Unix because of its power, and he liked Minix because it ran on his computer. Minix was created by Andrew Tanenbaum, a professor of computing, to demonstrate the principles of operating system design to his students. Because Minix was also a learning tool, people could also view the *source code* of the program—the original listings that Tanenbaum had entered to create the software.

But Torvalds had a number of issues with Minix. Although it's now available free of charge, at the time Minix was only available for a fee, although in many universities, it was possible to obtain copies free of charge from professors who paid a group licensing fee. Nevertheless, the copyright issue meant that using Minix in the wider world was difficult, and this, along with a handful of technical issues, inspired Torvalds to create from scratch his own version of Unix, just as Tanenbaum had done with Minix.

---

**Note** Most clones or implementations of Unix are named so that they end in an “x.” One story has it that Torvalds wanted to call his creation Freax, but a containing directory was accidentally renamed Linux on an Internet server. The name stuck.

---

From day one, Torvalds intended his creation to be shared among everyone who wanted to use it. He encouraged people to copy it and give it to friends. He didn't charge any money for it, and he also made the source code freely available. The idea was that people could take the code and improve it.

This was a master stroke. Many people contacted Torvalds, offering to help out. Because they could see the program code, they realized he was onto a good thing. Soon, Torvalds wasn't the only person developing Linux. He became the leader of a team that used the fledgling Internet to communicate and share improvements.

---

**Note** The popular conception of Linux is that it is created by a few hobbyists who work on it in their spare time. This might have been true in the very early days. Nowadays, in addition to these “bedroom programmers,” Linux is programmed by hundreds of professionals around the world, many of whom are employed specifically for the task. Torvalds adds to the effort himself and also coordinates the work.

---

It's important to note that when we talk here about Linux, we're actually talking about the kernel—the central program that runs the PC hardware and keeps the computer ticking. This is all that Torvalds initially produced back in 1991. It was an impressive achievement, but needed a lot of extra add-on programs to take care of even the most basic tasks. Torvalds's kernel needed additional software so that users could enter data, for example. It needed a way for users to be able to enter commands so they could manipulate files, such as deleting or copying them. And that's before you even consider more complicated stuff like displaying graphics on the screen or printing documents.

Linux itself didn't offer these functions. It simply ran the computer's hardware. Once it booted up, it expected to find other programs. If they weren't present, then all you saw was a blank screen.

## LINUS TORVALDS

Linus Benedict Torvalds was born in Helsinki, Finland, in 1969. A member of the minority Swedish-speaking population, he attended the University of Helsinki from 1988 to 1996, graduating with a Masters degree in Computer Science.

He started Linux not through a desire to give the world a first-class operating system but with other goals in mind. Its inspiration is in part due to Helsinki winters being so cold. Rather than leave his warm flat and trudge through the snow to the university's campus in order to use its powerful minicomputer, he wanted to be able to connect to it from home! He also wanted to have a platform to use to experiment with the properties of the Intel 386, but that's another story. Torvalds needed an operating system capable of such tasks. Linux was born.

It took Torvalds the better part of a year to come up with the very first version of Linux, during which he worked alone in a darkened room. In 1991, he announced his creation to the world, describing Linux as "just a hobby," and saying it would never be big. It wouldn't be until 1994 that it reached version 1.0.

In the early days, Torvalds's creation was fairly primitive. He was passionate that it should be free for everyone to use, and so he released it under a software license that said that no one could ever sell it. However, he quickly changed his mind, adopting the GNU Public License.

Torvalds was made wealthy by his creation, courtesy of the dot.com boom of the late 1990s, even though this was never his intention; he was driven by altruism. Nowadays, he lives in Portland, Oregon, with his wife and children, having moved to the United States from Finland in the late 1990s.

Initially, Torvalds worked for Transmeta, developing CPU architectures as well as overseeing kernel development, although this wasn't part of his official work. He still programs the kernel, but currently he oversees the Open Source Development Lab, an organization created to encourage open source adoption in industry and which is also referred to as the home of Linux.

## The GNU Project

Around the time Linus created Linux, another project, called GNU, also existed. This project team also hoped to create an operating system that used Unix as its inspiration, while avoiding some of the pitfalls that had blighted that operating system, both technically and in terms of its licensing. GNU is a so-called recursive acronym that stands for “GNU’s Not Unix,” a play on words favored by computer programmers.

GNU’s parent organization, the Free Software Foundation (FSF), had been formed eight years prior to Torvalds’s effort, and since that time, had produced the majority of the core software that Linux desperately needed. However, as luck would have it, FSF lacked the essential functionality of the kernel. The developers were in the process of creating their own kernel, but it had not come to fruition.

The GNU software was distributed for free to anyone who wanted it. The source code was also made available, so users could adapt and change the programs to meet their own needs (in fact, Torvalds had used the GNU model when deciding how to distribute Linux).

Richard Stallman is the man behind GNU and, along with Linus Torvalds, is the second accidental hero in our story. Stallman had been around since the Dark Ages of computing, back when wardrobe-sized computers were “time-shared” among users who used small desktop terminals to access them. Like Torvalds, Stallman started GNU as a personal project, but then found others who were more than willing to join his cause.

---

**Note** Stallman created the Emacs text editor and the GNU C Compiler (GCC). Together, they allow the creation of yet more software, so it’s no surprise that one of the very first programs Torvalds used in the early days to create Linux was Stallman’s GCC.

---

Back in Stallman’s day at the legendary Massachusetts Institute of Technology (MIT), computer software was shared. If you created a program to perform a particular task, you offered it to practically anyone who wanted it. Alternatively, if you found an existing program wasn’t adequate or had a bug, you improved it yourself, and then made the resulting program available to others. People might use your improved version, or they might not; it was up to them.

This way of sharing software was disorganized and done on an ad hoc basis, but came about of its own accord. Nobody questioned it, and it seemed the best way of doing things. There certainly wasn’t any money involved, any more than there would be money involved in one friend explaining an idea to another.

## RICHARD STALLMAN

Richard Matthew Stallman, usually referred to as RMS, was born in 1953 in Manhattan. He comes from the old school of computing forged during the 1970s and was a member of MIT's legendary Artificial Intelligence Lab.

Seemingly destined for a life in academia, Stallman left MIT in 1984 to found the GNU Project. This was as a reaction to the increasing commercialization of computer software. Whereas once all hackers (that is, programmers) had shared ideas and program code, the trend in the 1980s was toward proprietary, nonshared code, as well as legal contracts, which forced programmers to keep secrets from one another.

Stallman is a very talented programmer and is considered a genius by many observers. He single-handedly created many essential programming tools in his initial efforts to get GNU off the ground. Many of these find a home in Linux.

Stallman is also widely applauded for the creation of the GNU Public License. This is a legal document that lets people share software. It introduces the concept of *copyleft* and is opposed to the legal concept of copyright, which attempts to limit the freedom of individuals when using a piece of software (or any other creative work). Nowadays, the concept of copyleft has been applied to literature, music, and other arts in an attempt to avoid restricting who can and cannot access various items, as well as to encourage a collaborative working environment.

## Proprietary Software and the GPL

In the 1980s, everything changed. The world became more corporate, and with the rise of the desktop PC, the concept of proprietary software became prevalent. More and more companies started to sell software. They reasoned that this was impossible to do if they shared it with everybody else, so they kept it secret. Microsoft led this charge and did very well with its proprietary software.

To Stallman, this “trade secrets” approach to software was anathema. He had nothing against software being sold for a profit, but he hated the fundamental ideas behind software being kept secret. He felt passionately that sharing software and being able to understand how it worked was akin to free speech—necessary and vital for the furthering of technology, and therefore society itself. How could the new generation of programmers improve on the previous generation's work if they were unable to see how it worked? It was absurd to need to create software from scratch each time, rather than taking something that already existed and making it better.

Because of his beliefs, Stallman resigned from his job in the MIT Artificial Intelligence Lab and founded GNU. His aim initially was to produce a complete clone of Unix that would be shared in the ways he knew from the early days of computing. This software

would be available for everyone to use, to study, and to adapt. It would be free, in the same sense as free speech—shared and unrestricted. This gave rise to the vital concept of “free software” and soon GNU, and the FSF, became not just a programming venture, but also a political movement.

---

**Note** A very common misconception of “free software” is that it is always free of charge. This isn’t correct. The word *free* is used here in its political sense, as in “free speech.” Many companies and individuals make a healthy profit from selling free software and, in fact, selling free software is encouraged by the GNU Project.

---

To protect the rights of people to share and adapt the GNU software, Stallman came up with the GNU Public License (GPL). Various drafts of this license were produced over time, until it became a completely watertight legal contract, which furthered the concept of free software.

Most software you buy comes with a license agreement—that big chunk of text you must agree to when installing software (in the case of Windows desktop software, it’s frequently referred to as the End-User License Agreement, or EULA). The license agreement usually says that you cannot copy the software or share it with friends. If others want to use the software, they must buy their own version.

The GPL turns this on its head. Rather than restricting what people can do with the software, it gives them permission to share the software with whomever they wish. However, if they modify the program in any way, and then distribute it to others, the program they come up with must also be licensed under the GPL. In other words, people cannot make changes to a program that has a GPL, and then sell the modified program, keeping their improvements secret.

---

**Note** An interesting side note is that the actual wording of the GPL says that any changes you make should be shared with others *only if the software is redistributed*. This means that if you modify some GPL software and don’t give it to anyone else, there’s no need for you to publish your changes or make others aware of those changes.

---

## GNU and Linux Together

The Linux kernel, developed by Torvalds, and the GNU software, developed by Stallman, were a perfect match. It’s important to note that this doesn’t mean the two projects joined forces. It simply means that the Linux project took some of the GNU software and gave it a good home. This was done with Stallman’s blessing, but there wasn’t any official union between the two groups. Remember that Stallman had intended everyone to freely share

and use the GNU tools. Linux represented a set of people doing just that. GNU is still working on its own kernel, called Hurd, which may provide an alternative to using Torvalds's Linux kernel.

---

**Note** Hurd was first planned back in the 1980s and, at the time of writing, still has yet to see the light of day (although testing versions are available). Hurd is a hugely ambitious project and will set a gold standard when it is released.

---

GNU and Linux together formed a complete operating system, which mimicked the way Unix operated. Other projects and individuals spotted the success of Linux and came onboard, and it wasn't long before Linux realized the potential for a graphical user interface (GUI), the fundamentals of which were provided by the XFree86 Project. A lot of additional software was also provided by individuals and organizations, all using the same "share and share alike" example set by Stallman, with the GNU tools, and Torvalds, with his kernel.

Many people refer to Linux as GNU/Linux. This gives credit to the GNU Project that provided the majority of tools vital to making Linux into a usable operating system. However, like the majority of people in the computing world, we use the term "Linux" throughout this book to avoid confusion.

## Different Flavors of Linux

All the pieces of GNU software were available for free download and were therefore free of charge. But this brought its own problems. Not everyone had the know-how to put all the bits and pieces together into a complete operating system. Those who could do this didn't necessarily have the time for it.

Because of this, a number of companies stepped in to do the hard work. They put together versions of Linux, complete with all the software from the GNU Project, which they then sold for a fee on floppy disks, CDs, or DVDs. They also added in bits of their own software, which made it possible to install Linux easily onto a computer's hard disk, for example. They produced their own manuals and documentation, too, and did other things such as bug testing to ensure it all worked well.

What they came up with became known as *distributions* of Linux, or *distros* for short. Examples of these companies include Red Hat, SUSE, Mandrake, and many others around the world. Additionally, a number of enthusiasts got together and formed organizations to create their own distros, such as Debian and Slackware.

Modern distros are very advanced. They make it easy to install Linux on your PC, and they usually come with everything you need, so you can get started immediately. Additionally, they have their own look and feel, as well as unique ways of working and operating.

This means that Ubuntu is not the same as Red Hat Linux, for example, although they share a lot of common features and, of course, they all share the core GNU software.

## Linux Today

Nowadays, Linux is a thoroughly modern and capable operating system, considered cutting edge by many. It also runs on many different types of computer hardware, including Apple Macintosh computers, Sun SPARC machines, and the ubiquitous desktop PCs equipped with Intel or AMD processors. One of the ironies is that, although Linux was based on Unix, it has slowly come to dominate the computer operating system market. According to industry sources, Linux is on its way to making commercial varieties of Unix redundant. Companies that sell their own versions of Unix, such as Hewlett Packard and IBM, have added Linux to their traditional product range.

Recent innovations in the latest versions of the kernel mean that it finds uses on the smallest computers in the world, as well as on the biggest. Several of the top supercomputers in the world run Linux and, ironically, it can also be used on handheld PDAs or even digital watches! You'll even find it running things like digital video recorders or other household goods, where it sits invisibly in the background and makes everything work. Remember that one of the fundamental principles of Linux is that you can use it for whatever you want. You don't need to ask for permission first or tell anyone what you're doing.

Linux initially found mainstream use by software developers, and on server computers, such as those that run the Internet. However, in recent years, it has become increasingly popular on desktop computers. This is the area where experts suggest it will see massive growth over the coming years.

## Modern Linux Development

Nowadays, Linux is developed not only by Torvalds, who manages the huge project, but also by hundreds of volunteers and corporations who contribute resources. Most recently, IBM and Novell have gotten involved and contribute hundreds of people to the effort of creating Linux. Sun contributes the OpenOffice.org office suite and sells its own version of Unix. Corporations like Computer Associates contribute their own software, too.

These companies have realized that the best way of producing software is to share and share alike, rather than develop their own proprietary software and keep it secret. The proprietary ways of the 1980s are starting to seem like an ill-conceived flash in the pan.

Most recently, Novell found that by embracing Linux, it could massively enhance the functions of its aging NetWare product, without needing to return to the drawing board and start from scratch. It could just take what it wanted from the pile of Linux software. This shows the philosophy of Linux in action.



Linux has software for just about every need, ranging from simply receiving e-mail to running a huge e-mail server. There are databases, office suites, web browsers, video games, movie players, audio tools, and more, as well as thousands of pieces of specialized software used in various niches of industry (and too boring to mention here). Most of this software is available to anyone who wants it, free of charge.

What more could you want?

## Summary

This chapter has detailed the history of Linux and explained its origins. It also explained *why* Linux came into being. We looked at how Linux formed one of the building blocks of a political movement geared toward producing software that can be shared.

We discussed the creator of Linux, Linus Torvalds. We've also looked at the massive input the GNU Project has made and, in particular, that of its philosopher king, Richard Stallman.

In the next chapter, we move on to look at what you can expect from day-to-day use of Linux.