

A Brief History of Linux

Unix is one of the most popular operating systems worldwide because of its large support base and distribution. It was originally developed as a multitasking system for minicomputers and mainframes in the mid-1970s. It has since grown to become one of the most widely used operating systems anywhere, despite its sometimes confusing interface and lack of central standardization.

The real reason for Unix's popularity? Many hackers feel that Unix is the Right Thing—the One True Operating System. Hence, the development of Linux by an expanding group of Unix hackers who want to get their hands dirty with their own system.

Versions of Unix exist for many systems, ranging from personal computers to supercomputers such as the Cray Y-MP. Most versions of Unix for personal computers are quite expensive and cumbersome. At the time of this writing, a one-machine version of AT&T's System V for the 386 runs at about \$US1500.

Linux is a freely distributable version of Unix, originally developed by Linus Torvalds, who began work on Linux in 1991 as a student at the University of Helsinki in Finland. Linus now works for Transmeta Corporation, a start-up in Santa Clara, California, and continues to maintain the Linux *kernel*, that is, the lowest-level core component of the operating system.

Linus released the initial version of Linux for free on the Internet, inadvertently spawning one of the largest software-development phenomena of all time. Today, Linux is authored and maintained by a group of several thousand (if not more) developers loosely collaborating across the Internet. Companies have sprung up to provide Linux support, to package it into easy-to-install distributions, and to sell workstations pre-installed with the Linux software. In March 1999, the first Linux World Expo trade show was held in San Jose, California, with reportedly well over 12,000 people in attendance. Most estimates place the number of Linux users worldwide somewhere around the 10 million mark (and we expect this number will look small by the time you read this).

Inspired by Andrew Tanenbaum's Minix operating system (another free Unix for PCs—albeit a very simple one), Linux began as a class project in which Linus wanted to build a simple Unix system that could run on a 386-based PC. The first discussions about Linux were on the Usenet newsgroup *comp.os.minix*. These discussions were concerned mostly with the development of a small, academic Unix system for Minix users who wanted more.

The very early development of Linux dealt mostly with the task-switching features of the 80386 protected-mode interface, all written in assembly code. Linus writes:

After that it was plain sailing: hairy coding still, but I had some devices, and debugging was easier. I started using C at this stage, and it certainly speeds up development. This is also when I start to get serious about my megalomaniac ideas to make “a better Minix than Minix.” I was hoping I’d be able to recompile **gcc** under Linux some day ...

Two months for basic setup, but then only slightly longer until I had a disk driver (seriously buggy, but it happened to work on my machine) and a small filesystem. That was about when I made 0.01 available [around late August of 1991]: it wasn’t pretty, it had no floppy driver, and it couldn’t do much anything. I don’t think anybody ever compiled that version. But by then I was hooked, and didn’t want to stop until I could chuck out Minix.

No announcement was ever made for Linux Version 0.01. The 0.01 sources weren’t even executable: they contained only the bare rudiments of the kernel source and assumed that you had access to a Minix machine to compile and play with them.

On October 5, 1991, Linus announced the first “official” version of Linux, version 0.02. At this point, Linus was able to run **bash** (the GNU Bourne Again Shell) and **gcc** (the GNU C compiler), but not much else was working. Again, this was intended as a hacker’s system. The primary focus was kernel development; none of the issues of user support, documentation, distribution, and so on had even been addressed. Today, the situation is quite different—the real excitement in the Linux world deals with graphical user environments, easy-to-install distribution packages, and high-level applications such as graphics utilities and productivity suites.

Linus wrote in *comp.os.minix*:

Do you pine for the nice days of Minix-1.1, when men were men and wrote their own device drivers? Are you without a nice project and just dying to cut your teeth on an OS you can try to modify for your needs? Are you finding it frustrating when everything works on Minix? No more all-nighters to get a nifty program working? Then this post might be just for you.

As I mentioned a month ago, I’m working on a free version of a Minix-lookalike for AT-386 computers. It has finally reached the stage where it’s even usable (though may not be depending on what you want), and I am willing to put out the sources for wider distribution. It’s just version 0.02 ... but I’ve successfully run **bash**, **gcc**, GNU **make**, GNU **sed**, **compress**, etc. under it.

After Version 0.03, Linus bumped the version number up to 0.10, as more people started to work on the system. After several further revisions, Linus increased the version number to 0.95, to reflect his expectation that the system was ready for an “official” release very soon. (Generally, software is not assigned the version number 1.0 until it’s theoretically complete or bug-free.) This was in March 1992. Almost a year and a half later, in late December 1993, the Linux kernel was still at Version 0.99.pl14—asymptotically approaching 1.0. Version 1.0 appeared in March 1994. As of the time of this writing (March 1999), the current kernel version is 2.2.6, while the 2.3 kernel versions are being concurrently developed. (We’ll explain the Linux versioning conventions in detail later.)

Linux could not have come into being without the GNU tools created by the Free Software Foundation. Their **gcc** compiler, which we’ll discuss in [Chapter 13](#) gave life to Linus Torvalds’s code. GNU tools have been intertwined with the development of Linux from the beginning. Because of the critical contributions of these tools, the Free Software Foundation even requests that distributions of Linux with accompanying utilities be called GNU/Linux.

Berkeley Unix (BSD) has also played an important role in Linux—not so much in its creation, but in providing the tools that make it popular. Most of the utilities that come with Linux distributions are ported from BSD. Networking daemons and utilities are particularly important. The kernel networking code for Linux was developed from the ground up (two or three times, in fact), but the daemons and utilities are vintage BSD.

Today, Linux is a complete Unix clone, capable of running the X Window System, TCP/IP, Emacs, Web, mail and news software, you name it. Almost all major free software packages have been ported to Linux, and commercial software is becoming available. In fact, many developers start by writing applications for Linux, and port them to other Unix systems later. More hardware is supported than in original versions of the kernel. Many people have executed benchmarks on Linux systems and found them to be faster than workstations from Sun Microsystems and Compaq, and Linux performs better than or as well as Windows 98 and Windows NT on a wide range of benchmarks. Who would have ever guessed that this “little” Unix clone would have grown up to take on the entire world of personal and server computing.

Philosophy of linux

1. Small is Beautiful
2. Each Program Does One Thing Well
3. Prototype as Soon as Possible
4. Choose Portability Over Efficiency
5. Store Data in Flat Text Files
6. Use Software Leverage
7. Use Shell Scripts to Increase Leverage and Portability
8. Avoid Captive User Interfaces
9. Make Every Program a Filter

Community

See also: [Free software community](#) and [Linux User Group](#)

A distribution is largely driven by its developer and user communities. Some vendors develop and fund their distributions on a volunteer basis, [Debian](#) being a well-known example. Others maintain a community version of their commercial distributions, as [Red Hat](#) does with [Fedora](#), and [SUSE](#) does with [openSUSE](#).

In many cities and regions, local associations known as [Linux User Groups](#) (LUGs) seek to promote their preferred distribution and by extension free software. They hold meetings and provide free demonstrations, training, technical support, and operating system installation to new users. Many Internet communities also provide support to Linux users and developers. Most distributions and free software / open-source projects have [IRC](#) chatrooms or [newsgroups](#). [Online forums](#) are another means for support, with notable examples being [LinuxQuestions.org](#) and the various distribution specific support and community forums, such as ones for [Ubuntu](#), [Fedora](#), and [Gentoo](#). Linux distributions host [mailing lists](#); commonly there will be a specific topic such as usage or development for a given list.

There are several technology websites with a Linux focus. Print magazines on Linux often bundle [cover disks](#) that carry software or even complete Linux distributions.

Although Linux distributions are generally available without charge, several large corporations sell, support, and contribute to the development of the components of the system and of [free software](#). An analysis of the Linux kernel showed 75 percent of the code from December 2008 to January 2010 was developed by programmers working for corporations, leaving about 18 percent to volunteers and 7% unclassified. Major corporations that provide contributions include [Dell](#), [IBM](#), [HP](#), [Oracle](#), [Sun Microsystems](#) (now part of Oracle) and [Nokia](#). A number of corporations, notably Red Hat, [Canonical](#) and [SUSE](#), have built a significant business around Linux distributions.

The [free software licenses](#), on which the various software packages of a distribution built on the Linux kernel are based, explicitly accommodate and encourage commercialization; the relationship between a Linux distribution as a whole and individual vendors may be seen as [symbiotic](#). One common [business model](#) of commercial suppliers is charging for support, especially for business users. A number of companies also offer a specialized business version of their distribution, which adds proprietary support packages and tools to administer higher numbers of installations or to simplify administrative tasks.

Another business model is to give away the software in order to sell hardware. This used to be the norm in the computer industry, with operating systems such as [CP/M](#), [Apple DOS](#) and versions of [Mac OS](#) prior to 7.6 freely copyable (but not modifiable). As computer hardware standardized throughout the 1980s, it became more difficult for hardware manufacturers to profit from this tactic, as the OS would run on any manufacturer's computer that shared the same architecture.

Linux distribu

A **Linux distribution** (often abbreviated as **distro**) is an [operating system](#) made from a software collection, which is based upon the [Linux kernel](#) and, often, a [package management system](#). Linux users usually obtain their operating system by downloading one of the Linux distributions, which are available for a wide variety of systems ranging from [embedded devices](#) (for example, [OpenWrt](#)) and [personal computers](#) (for example, [Linux Mint](#)) to powerful [supercomputers](#) (for example, [Rocks Cluster Distribution](#)).

A typical Linux distribution comprises a Linux kernel, [GNU](#) tools and libraries, additional software, documentation, a [window system](#) (the most common being the [X Window System](#)), a [window manager](#), and a [desktop environment](#). Most of the included software is [free and open-source software](#) made available both as compiled binaries and in [source code](#) form, allowing modifications to the original software. Usually, Linux distributions optionally include some [proprietary software](#) that may not be available in source code form, such as [binary blobs](#) required for some [device drivers](#). A Linux distribution may also be described as a particular assortment of application and utility software (various GNU tools and libraries, for example), packaged together with the Linux kernel in such a way that its capabilities meet the needs of many users. The software is usually adapted to the distribution and then packaged into [software packages](#) by the distribution's maintainers. The software packages are available online in so-called [repositories](#), which are storage locations usually distributed around the world. Beside glue components, such as the distribution installers (for example, [Debian-Installer](#) and [Anaconda](#)) or the package management systems, there are only very few packages that are originally written from the ground up by the maintainers of a Linux distribution.

Almost six hundred [Linux distributions](#) exist, with close to five hundred out of those in active development. Because of the huge availability of software, distributions have taken a wide variety of forms, including those suitable for use on desktops, servers, laptops, netbooks, mobile phones and tablets, as well as minimal environments typically for use in embedded systems. There are commercially backed distributions, such as [Fedora](#) ([Red Hat](#)), [openSUSE](#) ([SUSE](#)) and [Ubuntu](#) ([Canonical Ltd.](#)), and entirely community-driven distributions, such as [Debian](#), [Slackware](#), [Gentoo](#) and [Arch Linux](#). Most distributions come ready to use and pre-compiled for a specific [instruction set](#), while some distributions (such as [Gentoo](#)) are distributed mostly in [source code](#) form and compiled locally during installation.