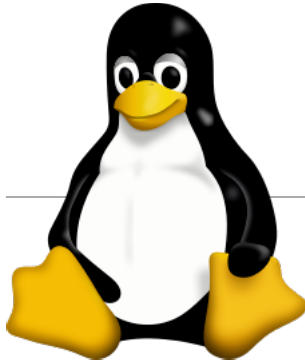


Linux (/ˈlɪnʊks/ *LIN-uuks*)^[11] is a family of open-source Unix-like operating systems based on the Linux kernel,^[12] an operating system kernel first released on September 17, 1991, by Linus Torvalds.^{[13][14][15]} Linux is typically packaged as a Linux distribution (distro), which includes the kernel and supporting system software and libraries, many of which are provided by the GNU Project. Many Linux distributions use the word "Linux" in their name, but the Free Software Foundation uses and recommends the name "**GNU/Linux**" to emphasize the use and importance of GNU software in many distributions, causing some controversy.^{[16][17]}

Popular Linux distributions^{[18][19][20]} include Debian, Fedora Linux, Arch Linux, and Ubuntu. Commercial distributions include Red Hat Enterprise Linux and SUSE Linux Enterprise. Desktop Linux distributions include a windowing system such as X11 or Wayland and a desktop environment such as GNOME or KDE Plasma. Distributions intended for servers may not have a graphical user interface at all, or include a solution stack such as LAMP. Because Linux is freely redistributable, anyone may create a distribution for any purpose.^[21]

Linux was originally developed for personal computers based on the Intel x86 architecture, but has since been ported to more platforms than any other operating system.^[22] Because of the dominance of Linux-based Android on smartphones, Linux, including Android, has the largest installed base of all general-purpose operating systems as of May 2022.^{[23][24][25]} Linux is, as of March 2024, used by around 4 percent of desktop computers,^[26] the Chromebook, which runs the Linux kernel-based ChromeOS, dominates the US K–12 education market and represents nearly 20 percent of sub-\$300 notebook sales in the US.^[27] Linux is the leading operating system on servers (over 96.4% of the top one million web servers' operating systems are Linux),^[28] leads other big iron systems such as mainframe computers, and is used on all of the world's 500 fastest supercomputers^[d] (as of November 2017, having gradually displaced all competitors).^{[29][30][31]}

Linux



Tux the penguin, the mascot of Linux^[1]

| | |
|-------------------------|---|
| Developer | Community contributors, <u>Linus Torvalds</u> |
| Written in | <u>C</u> , <u>assembly languages</u> and others |
| OS family | <u>Unix-like</u> |
| Working state | Current |
| Source model | <u>Open source</u> |
| Initial release | September 17, 1991 |
| Repository | <u>git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/</u> (<u>https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/</u>) |
| Marketing target | <u>Cloud computing</u> , <u>embedded devices</u> , <u>mainframe computers</u> , <u>mobile devices</u> , <u>personal</u> |

Linux also runs on embedded systems, i.e., devices whose operating system is typically built into the firmware and is highly tailored to the system. This includes routers, automation controls, smart home devices, video game consoles,^[32] televisions (Samsung and LG Smart TVs),^[33]^[34]^[35] automobiles (Tesla, Audi, Mercedes-Benz, Hyundai, and Toyota),^[36] and spacecraft (Falcon 9 rocket, Dragon crew capsule, and the *Perseverance* rover).^[37]^[38]

Linux is one of the most prominent examples of free and open-source software collaboration. The source code may be used, modified, and distributed commercially or non-commercially by anyone under the terms of its respective licenses, such as the GNU General Public License (GPL). The Linux kernel, for example, is licensed under the GPLv2, with an exception for system calls that allows code that calls the kernel via system calls not to be licensed under the GPL.^[39]^[40]^[21]

History

Precursors

The Unix operating system was conceived and implemented in 1969, at AT&T's Bell Labs, in the United States by Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna.^[41] First released in 1971, Unix was written entirely in assembly language, as was common practice at the time. In 1973, in a key pioneering approach, it was rewritten in the C programming language by Dennis Ritchie (except for some hardware and I/O routines). The availability of a high-level language implementation of Unix made its porting to different computer platforms easier.^[42]

Due to an earlier antitrust case forbidding it from entering the computer business, AT&T licensed the operating system's source code as a trade secret to anyone who asked. As a result, Unix grew quickly and became widely adopted by academic institutions and businesses. In 1984, AT&T divested itself of its regional operating companies, and was released from its obligation not to enter the computer business; freed of that obligation, Bell Labs began selling Unix as a proprietary product, where users were not legally allowed to modify it.^[43]^[44]

Onyx Systems began selling early microcomputer-based Unix workstations in 1980. Later, Sun Microsystems, founded as a spin-off of a student project at Stanford University, also began selling Unix-based desktop workstations in 1982. While Sun workstations did not use commodity PC hardware, for which Linux was later originally developed, it represented the first successful commercial attempt at distributing a primarily single-user microcomputer that ran a Unix operating system.^[45]^[46]

| | |
|--|--|
| | <u>computers</u> , <u>servers</u> , <u>supercomputers</u> |
| Available in | Multilingual |
| Platforms | <u>Alpha</u> , <u>ARC</u> , <u>ARM</u> , <u>C-Sky</u> , <u>Hexagon</u> , <u>LoongArch</u> , <u>m68k</u> , <u>Microblaze</u> , <u>MIPS</u> , <u>Nios II</u> , <u>OpenRISC</u> , <u>PA-RISC</u> , <u>PowerPC</u> , <u>RISC-V</u> , <u>s390</u> , <u>SuperH</u> , <u>SPARC</u> , <u>x86</u> , <u>Xtensa</u> |
| Kernel type | <u>Monolithic</u> |
| Userland | <u>GNU</u> , ^[a] <u>BusyBox</u> ^[b] |
| Default user interface | <u>Unix shell</u> (<u>CLI</u>) Most <u>distributions</u> include a <u>desktop environment</u> (<u>GUI</u>). |
| License | <u>GPLv2</u> ^[9] ^[c] |
| Official website | <u>kernel.org</u> (<u>https://kernel.org/</u>) |
| Articles in the series | |
| <u>Linux kernel</u> <u>Linux distribution</u> | |

With Unix increasingly "locked in" as a proprietary product, the GNU Project, started in 1983 by Richard Stallman, had the goal of creating a "complete Unix-compatible software system" composed entirely of free software. Work began in 1984.^[47] Later, in 1985, Stallman started the Free Software Foundation and wrote the GNU General Public License (GNU GPL) in 1989. By the early 1990s, many of the programs required in an operating system (such as libraries, compilers, text editors, a command-line shell, and a windowing system) were completed, although low-level elements such as device drivers, daemons, and the kernel, called GNU Hurd, were stalled and incomplete.^[48]



Linus Torvalds, principal author of the Linux kernel

Minix was created by Andrew S. Tanenbaum, a computer science professor, and released in 1987 as a minimal Unix-like operating system targeted at students and others who wanted to learn operating system principles. Although the complete source code of Minix was freely available, the licensing terms prevented it from being free software until the licensing changed in April 2000.^[49]

Although not released until 1992, due to legal complications, the development of 386BSD, from which NetBSD, OpenBSD and FreeBSD descended, predated that of Linux. Linus Torvalds has stated on separate occasions that if the GNU kernel or 386BSD had been available at the time (1991), he probably would not have created Linux.^{[50][51]}

Creation

While attending the University of Helsinki in the fall of 1990, Torvalds enrolled in a Unix course.^[52] The course used a MicroVAX minicomputer running Ultrix, and one of the required texts was *Operating Systems: Design and Implementation* by Andrew S. Tanenbaum. This textbook included a copy of Tanenbaum's Minix operating system. It was with this course that Torvalds first became exposed to Unix. In 1991, he became curious about operating systems.^[53] Frustrated by the licensing of Minix, which at the time limited it to educational use only,^[49] he began to work on his operating system kernel, which eventually became the Linux kernel.

On July 3, 1991, to implement Unix system calls, Linus Torvalds attempted unsuccessfully to obtain a digital copy of the POSIX standards documentation with a request to the *comp.os.minix* newsgroup.^[54] After not finding the POSIX documentation, Torvalds initially resorted to determining system calls from SunOS documentation owned by the university for use in operating its Sun Microsystems server. He also learned some system calls from Tanenbaum's Minix text.

Torvalds began the development of the Linux kernel on Minix and applications written for Minix were also used on Linux. Later, Linux matured and further Linux kernel development took place on Linux systems.^[55] GNU applications also replaced all Minix components, because it was advantageous to use the freely available code from the GNU Project with the fledgling operating system; code licensed under the GNU GPL can be reused in other computer programs as long as they also are released under the same or a compatible license. Torvalds initiated a switch from his original license, which prohibited commercial redistribution, to the GNU GPL.^[56] Developers worked to integrate GNU components with the Linux kernel, creating a fully

functional and free operating system.^[57]

Naming

Linus Torvalds had wanted to call his invention "**Freax**", a portmanteau of "free", "freak", and "x" (as an allusion to Unix). During the start of his work on the system, some of the project's makefiles included the name "Freax" for about half a year. Initially, Torvalds considered the name "Linux" but dismissed it as too egotistical.^[58]

To facilitate development, the files were uploaded to the FTP server (`ftp.funet.fi`) of FUNET in September 1991. Ari Lemmke, Torvalds' coworker at the Helsinki University of Technology (HUT) who was one of the volunteer administrators for the FTP server at the time, did not think that "Freax" was a good name, so he named the project "Linux" on the server without consulting Torvalds.^[58] Later, however, Torvalds consented to "Linux".

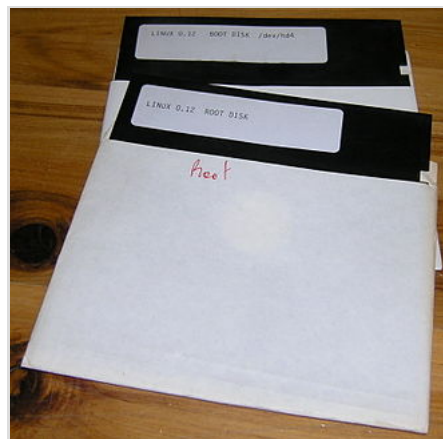
According to a newsgroup post by Torvalds,^[11] the word "Linux" should be pronounced (/ˈlɪnʊks/ *LIN-uuks*) with a short 'i' as in 'print' and 'u' as in 'put'. To further demonstrate how the word "Linux" should be pronounced, he included an audio guide with the kernel source code.^[59] However, in this recording, he pronounces Linux as /ˈlinʊks/ (*LEEN-uuks*) with a short but close front unrounded vowel, instead of a near-close near-front unrounded vowel as in his newsgroup post.

Commercial and popular uptake

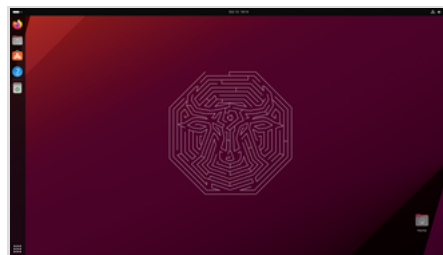
The adoption of Linux in production environments, rather than being used only by hobbyists, started to take off first in the mid-1990s in the supercomputing community, where organizations such as NASA started to replace their increasingly expensive machines with clusters of inexpensive commodity computers running Linux. Commercial use began when Dell and IBM, followed by Hewlett-Packard, started offering Linux support to escape Microsoft's monopoly in the desktop operating system market.^[60]

Today, Linux systems are used throughout computing, from embedded systems to virtually all supercomputers,^{[31][61]} and have secured a place in server installations such as the popular LAMP application stack. The use of Linux distributions in home and enterprise desktops has been growing.^{[62][63][64][65][66][67][68]} Linux distributions have also become popular in the netbook market, with many devices shipping with customized Linux distributions installed, and Google releasing their own ChromeOS designed for netbooks.

Linux's greatest success in the consumer market is perhaps the mobile device market, with Android being the dominant operating system on smartphones and very popular on tablets and,



5.25-inch floppy disks holding a very early version of Linux



Ubuntu, a popular Linux distribution

more recently, on wearables. Linux gaming is also on the rise with Valve showing its support for Linux and rolling out SteamOS, its own gaming-oriented Linux distribution, which was later implemented in their Steam Deck platform. Linux distributions have also gained popularity with various local and national governments, such as the federal government of Brazil.^[69]

Current development

Linus Torvalds is the lead maintainer for the Linux kernel and guides its development, while Greg Kroah-Hartman is the lead maintainer for the stable branch.^[70] Zoë Kooyman is the executive director of the Free Software Foundation,^[71] which in turn supports the GNU components.^[72] Finally, individuals and corporations develop third-party non-GNU components. These third-party components comprise a vast body of work and may include both kernel modules and user applications and libraries.

Linux vendors and communities combine and distribute the kernel, GNU components, and non-GNU components, with additional package management software in the form of Linux distributions.

Design

Many developers of open-source software agree that the Linux kernel was not designed but rather evolved through natural selection. Torvalds considers that although the design of Unix served as a scaffolding, "Linux grew with a lot of mutations – and because the mutations were less than random, they were faster and more directed than alpha-particles in DNA."^[73] Eric S. Raymond considers Linux's revolutionary aspects to be social, not technical: before Linux, complex software was designed carefully by small groups, but "Linux evolved in a completely different way. From nearly the beginning, it was rather casually hacked on by huge numbers of volunteers coordinating only through the Internet. Quality was maintained not by rigid standards or autocracy but by the naively simple strategy of releasing every week and getting feedback from hundreds of users within days, creating a sort of rapid Darwinian selection on the mutations introduced by developers."^[74] Bryan Cantrill, an engineer of a competing OS, agrees that "Linux wasn't designed, it evolved", but considers this to be a limitation, proposing that some features, especially those related to security,^[75] cannot be evolved into, "this is not a biological system at the end of the day, it's a software system."^[76]

A Linux-based system is a modular Unix-like operating system, deriving much of its basic design from principles established in Unix during the 1970s and 1980s. Such a system uses a monolithic kernel, the Linux kernel, which handles process control, networking, access to the peripherals, and file systems. Device drivers are either integrated directly with the kernel or added as modules that are loaded while the system is running.^[77]

The GNU userland is a key part of most systems based on the Linux kernel, with Android being



Nexus 5X running Android



In-flight entertainment system booting up displaying the Linux logo

the notable exception. The GNU C library, an implementation of the C standard library, works as a wrapper for the system calls of the Linux kernel necessary to the kernel-userspace interface, the toolchain is a broad collection of programming tools vital to Linux development (including the compilers used to build the Linux kernel itself), and the coreutils implement many basic Unix tools. The GNU Project also develops Bash, a popular CLI shell. The graphical user interface (or GUI) used by most Linux systems is built on top of an implementation of the X Window System.^[78] More recently, the Linux community has sought to advance to Wayland as the new display server protocol, in place of X11. Many other open-source software projects contribute to Linux systems.

Various layers within Linux, also showing separation between the userland and kernel space

| | | | | | | |
|---|---------------------------|--|---|--|---|--|
| User mode | User applications | bash, LibreOffice, GIMP, Blender, 0 A.D., Mozilla Firefox, ... | | | | |
| | System components | <u>init daemon:</u> OpenRC, runit, systemd... | <u>System daemons:</u> polkitd, smbd, sshd, udevd... | <u>Window manager:</u> X11, Wayland, SurfaceFlinger (Android) | <u>Graphics:</u> Mesa, AMD Catalyst, ... | <u>Other libraries:</u> GTK, Qt, EFL, SDL, SFML, FLTK, GNUstep, ... |
| | <u>C standard library</u> | fopen, execv, malloc, memcpy, localtime, pthread_create... (up to 2000 subroutines) glibc aims to be fast, musl aims to be lightweight, uClibc targets embedded systems, bionic was written for Android, etc. All aim to be POSIX/SUS-compatible. | | | | |
| Kernel mode | <u>Linux kernel</u> | <u>stat, splice, dup, read, open, ioctl, write, mmap, close, exit, etc.</u> (about 380 system calls) The Linux kernel <u>System Call Interface (SCI)</u> , aims to be <u>POSIX/SUS-compatible</u> ^[79] | | | | |
| | | <u>Process scheduling subsystem</u> | <u>IPC subsystem</u> | <u>Memory management subsystem</u> | <u>Virtual files subsystem</u> | <u>Networking subsystem</u> |
| | | Other components: <u>ALSA, DRI, evdev, klibc, LVM, device mapper, Linux Network Scheduler, Netfilter</u> <u>Linux Security Modules: SELinux, TOMOYO, AppArmor, Smack</u> | | | | |
| Hardware (CPU, main memory, data storage devices, etc.) | | | | | | |

Installed components of a Linux system include the following:^{[78][80]}

- A bootloader, for example GNU GRUB, LILO, SYSLINUX or systemd-boot. This is a program that loads the Linux kernel into the computer's main memory, by being executed by the computer when it is turned on and after the firmware initialization is performed.
- An init program, such as the traditional sysvinit and the newer systemd, OpenRC and Upstart. This is the first process launched by the Linux kernel, and is at the root of the process tree. It starts processes such as system services and login prompts (whether graphical or in terminal mode).

- Software libraries, which contain code that can be used by running processes. On Linux systems using ELF-format executable files, the dynamic linker that manages the use of dynamic libraries is known as ld-linux.so. If the system is set up for the user to compile software themselves, header files will also be included to describe the programming interface of installed libraries. Besides the most commonly used software library on Linux systems, the GNU C Library (glibc), there are numerous other libraries, such as SDL and Mesa.
 - The C standard library is the library necessary to run programs written in C on a computer system, with the GNU C Library being the standard. It provides an implementation of the POSIX API, as well as extensions to that API. For embedded systems, alternatives such as musl, EGLIBC (a glibc fork once used by Debian) and uClibc (which was designed for uClinux) have been developed, although the last two are no longer maintained. Android uses its own C library, Bionic. However, musl can additionally be used as a replacement for glibc on desktop and laptop systems, as seen on certain Linux distributions like Void Linux.
- Basic Unix commands, with GNU coreutils being the standard implementation. Alternatives exist for embedded systems, such as the copyleft BusyBox, and the BSD-licensed Toybox.
- Widget toolkits are the libraries used to build graphical user interfaces (GUIs) for software applications. Numerous widget toolkits are available, including GTK and Clutter developed by the GNOME Project, Qt developed by the Qt Project and led by The Qt Company, and Enlightenment Foundation Libraries (EFL) developed primarily by the Enlightenment team.
- A package management system, such as dpkg and RPM. Alternatively packages can be compiled from binary or source tarballs.
- User interface programs such as command shells or windowing environments.

User interface

The user interface, also known as the shell, is either a command-line interface (CLI), a graphical user interface (GUI), or controls attached to the associated hardware, which is common for embedded systems. For desktop systems, the default user interface is usually graphical, although the CLI is commonly available through terminal emulator windows or on a separate virtual console.

CLI shells are text-based user interfaces, which use text for both input and output. The dominant shell used in Linux is the Bourne-Again Shell (bash), originally developed for the GNU Project. Most low-level Linux components, including various parts of the userland, use the CLI exclusively. The CLI is particularly suited for automation of repetitive or delayed tasks and provides very simple inter-process communication.

On desktop systems, the most popular user interfaces are the GUI shells, packaged together with extensive desktop environments, such as KDE Plasma, GNOME, MATE, Cinnamon, LXDE, Pantheon, and Xfce, though a variety of additional user interfaces exist. Most popular user interfaces are based on the X Window System, often simply called "X". It provides network transparency and permits a graphical application running on one system to be displayed on

another where a user may interact with the application; however, certain extensions of the X Window System are not capable of working over the network.^[81] Several X display servers exist, with the reference implementation, X.Org Server, being the most popular.

Server distributions might provide a command-line interface for developers and administrators, but provide a custom interface for end-users, designed for the use case of the system. This custom interface is accessed through a client that resides on another system, not necessarily Linux-based.

Several types of window managers exist for X11, including tiling, dynamic, stacking, and compositing. Window managers provide means to control the placement and appearance of individual application windows, and interact with the X Window System. Simpler X window managers such as dwm, ratpoison, or i3wm provide a minimalist functionality, while more elaborate window managers such as FVWM, Enlightenment, or Window Maker provide more features such as a built-in taskbar and themes, but are still lightweight when compared to desktop environments. Desktop environments include window managers as part of their standard installations, such as Mutter (GNOME), KWin (KDE), or Xfwm (xfce), although users may choose to use a different window manager if preferred.

Wayland is a display server protocol intended as a replacement for the X11 protocol; as of 2022, it has received relatively wide adoption.^[82] Unlike X11, Wayland does not need an external window manager and compositing manager. Therefore, a Wayland compositor takes the role of the display server, window manager, and compositing manager. Weston is the reference implementation of Wayland, while GNOME's Mutter and KDE's KWin are being ported to Wayland as standalone display servers. Enlightenment has already been successfully ported since version 19.^[83] Additionally, many window managers have been made for Wayland, such as Sway or Hyprland, as well as other graphical utilities such as Waybar or Rofi.

Video input infrastructure

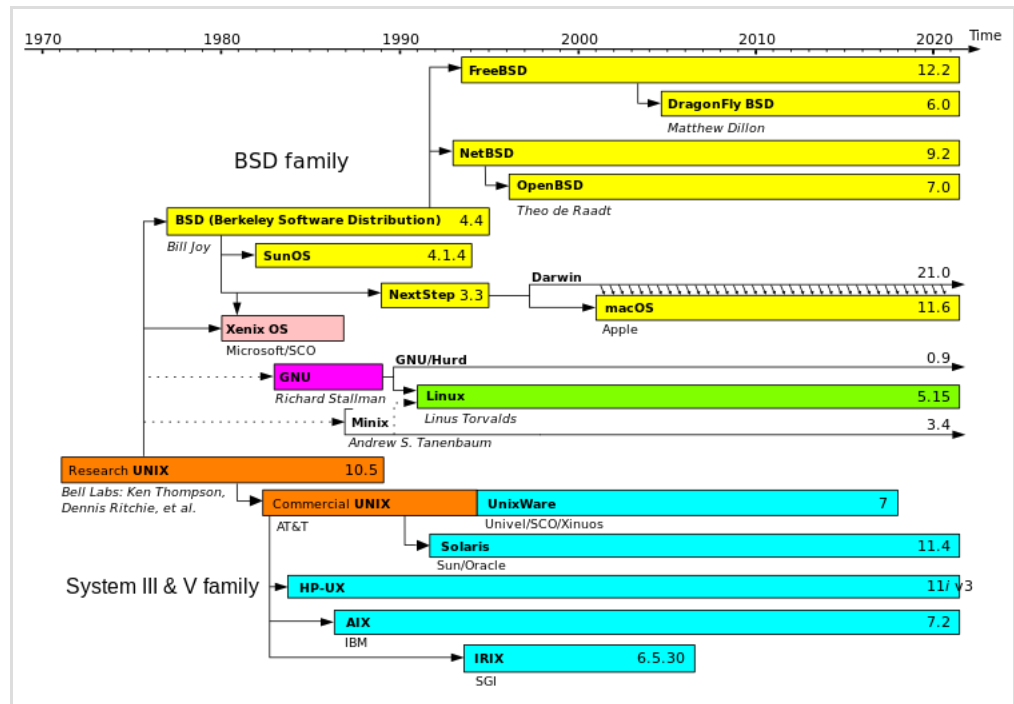
Linux currently has two modern kernel-userspace APIs for handling video input devices: V4L2 API for video streams and radio, and DVB API for digital TV reception.^[84]

Due to the complexity and diversity of different devices, and due to the large number of formats and standards handled by those APIs, this infrastructure needs to evolve to better fit other devices. Also, a good userspace device library is the key to the success of having userspace applications to be able to work with all formats supported by those devices.^{[85][86]}

Development

The primary difference between Linux and many other popular contemporary operating systems is that the Linux kernel and other components are free and open-source software. Linux is not the only such operating system, although it is by far the most widely used.^[87] Some free and open-source software licenses are based on the principle of copyleft, a kind of reciprocity: any work derived from a copyleft piece of software must also be copyleft itself. The most common free software license, the GNU General Public License (GPL), is a form of copyleft and is used for the Linux kernel and many of the components from the GNU Project.^[88]

Linux-based distributions are intended by developers for interoperability with other operating systems and established computing standards. Linux systems adhere to [POSIX](#),^[89] [SUS](#),^[90] [LSB](#), [ISO](#), and [ANSI](#) standards where possible, although to date only one Linux distribution has been POSIX.1 certified, [Linux-FT](#).^{[91][92]}



Simplified history of Unix-like operating systems. Linux shares similar architecture and concepts (as part of the [POSIX](#) standard) but does not share non-free source code with the original [Unix](#) or [Minix](#).

Free software projects, although developed through [collaboration](#), are often produced independently of each other. The fact that the software licenses explicitly permit redistribution, however, provides a basis for larger-scale projects that collect the software produced by stand-alone projects and make it available all at once in the form of a Linux distribution.

Many Linux distributions manage a remote collection of system software and application software packages available for download and installation through a network connection. This allows users to adapt the operating system to their specific needs. Distributions are maintained by individuals, loose-knit teams, volunteer organizations, and commercial entities. A distribution is responsible for the default configuration of the installed Linux kernel, general system security, and more generally integration of the different software packages into a coherent whole. Distributions typically use a package manager such as [apt](#), [yum](#), [zypper](#), [pacman](#) or [portage](#) to install, remove, and update all of a system's software from one central location.^[93]

Community

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](#).^{[94][95]}

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 of 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.^{[96][97]}

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 free software. An analysis of the Linux kernel in 2017 showed that well over 85% of the code was developed by programmers who are being paid for their work, leaving about 8.2% to unpaid developers and 4.1% unclassified.^[98] Some of the major corporations that provide contributions include Intel, Samsung, Google, AMD, Oracle, and Facebook.^[98] Several 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 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 the classic Mac OS before 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.

Programming on Linux

Most programming languages support Linux either directly or through third-party community based ports.^[99] The original development tools used for building both Linux applications and operating system programs are found within the GNU toolchain, which includes the GNU Compiler Collection (GCC) and the GNU Build System. Amongst others, GCC provides compilers for Ada, C, C++, Go and Fortran. Many programming languages have a cross-platform reference implementation that supports Linux, for example PHP, Perl, Ruby, Python, Java, Go, Rust and Haskell. First released in 2003, the LLVM project provides an alternative cross-platform open-source compiler for many languages. Proprietary compilers for Linux include the Intel C++ Compiler, Sun Studio, and IBM XL C/C++ Compiler. BASIC is available in procedural form from QB64, PureBasic, Yabasic, GLBasic, Basic4GL, XBasic, wxBasic, SdlBasic, and Basic-256, as well as object oriented through Gambas, FreeBASIC, B4X, Basic for Qt, Phoenix Object Basic, NS Basic, ProvideX, Chipmunk Basic, RapidQ and Xojo. Pascal is implemented through GNU Pascal, Free Pascal, and Virtual Pascal, as well as graphically via Lazarus, PascalABC.NET, or Delphi using FireMonkey (previously through Borland Kylix).

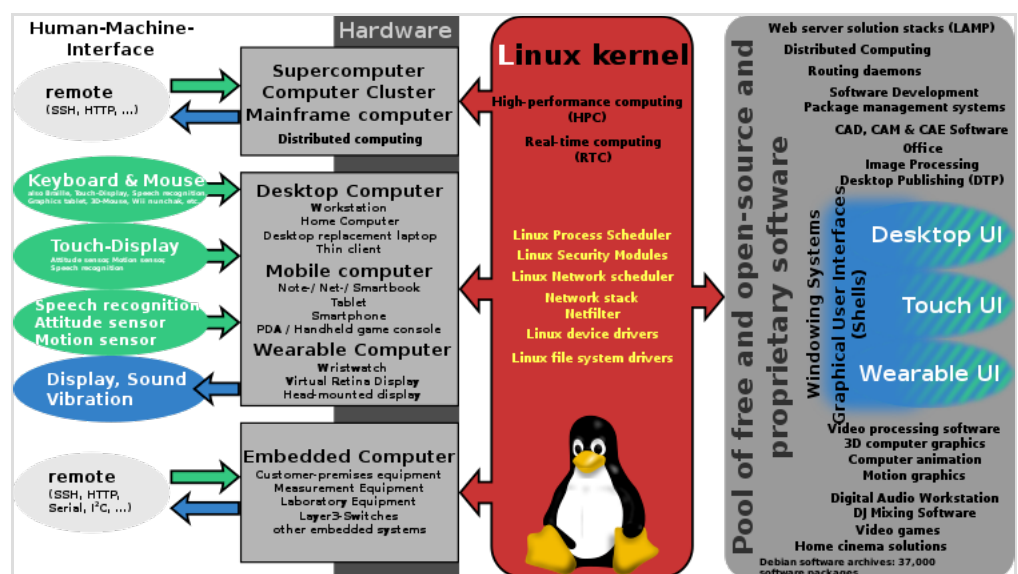
A common feature of Unix-like systems, Linux includes traditional specific-purpose

programming languages targeted at scripting, text processing and system configuration and management in general. Linux distributions support shell scripts, awk, sed and make. Many programs also have an embedded programming language to support configuring or programming themselves. For example, regular expressions are supported in programs like grep and locate, the traditional Unix message transfer agent Sendmail contains its own Turing complete scripting system, and the advanced text editor GNU Emacs is built around a general purpose Lisp interpreter.

Most distributions also include support for PHP, Perl, Ruby, Python and other dynamic languages. While not as common, Linux also supports C# and other CLI languages (via Mono), Vala, and Scheme. Guile Scheme acts as an extension language targeting the GNU system utilities, seeking to make the conventionally small, static, compiled C programs of Unix design rapidly and dynamically extensible via an elegant, functional high-level scripting system; many GNU programs can be compiled with optional Guile bindings to this end. A number of Java virtual machines and development kits run on Linux, including the original Sun Microsystems JVM (HotSpot), and IBM's J2SE RE, as well as many open-source projects like Kaffe and Jikes RVM; Kotlin, Scala, Groovy and other JVM languages are also available.

GNOME and KDE are popular desktop environments and provide a framework for developing applications. These projects are based on the GTK and Qt widget toolkits, respectively, which can also be used independently of the larger framework. Both support a wide variety of languages. There are a number of Integrated development environments available including Anjuta, Code::Blocks, CodeLite, Eclipse, Geany, ActiveState Komodo, KDevelop, Lazarus, MonoDevelop, NetBeans, and Qt Creator, while the long-established editors Vim, nano and Emacs remain popular.^[100]

Hardware support



Linux is ubiquitously found on various types of hardware.

The Linux kernel is a widely ported operating system kernel, available for devices ranging from mobile phones to supercomputers; it runs on a highly diverse range of computer architectures, including ARM-based Android smartphones and the IBM Z mainframes. Specialized distributions and kernel forks exist for less mainstream architectures; for example, the ELKS

kernel fork can run on Intel 8086 or Intel 80286 16-bit microprocessors, while the μClinux kernel fork may run on systems without a memory management unit. The kernel also runs on architectures that were only ever intended to use a proprietary manufacturer-created operating system, such as Macintosh computers^{[101][102]} (with PowerPC, Intel, and Apple silicon processors), PDAs, video game consoles, portable music players, and mobile phones.

Linux has a reputation for supporting old hardware very well by maintaining standardized drivers for a long time.^[103] There are several industry associations and hardware conferences devoted to maintaining and improving support for diverse hardware under Linux, such as FreedomHEC. Over time, support for different hardware has improved in Linux, resulting in any off-the-shelf purchase having a "good chance" of being compatible.^[104]

In 2014, a new initiative was launched to automatically collect a database of all tested hardware configurations.^[105]

Uses

Market share and uptake

Many quantitative studies of free/open-source software focus on topics including market share and reliability, with numerous studies specifically examining Linux.^[106] The Linux market is growing, and the Linux operating system market size is expected to see a growth of 19.2% by 2027, reaching \$15.64 billion, compared to \$3.89 billion in 2019.^[107] Analysts project a Compound Annual Growth Rate (CAGR) of 13.7% between 2024 and 2032, culminating in a market size of USD 34.90 billion by the latter year.^[108] Analysts and proponents attribute the relative success of Linux to its security, reliability, low cost, and freedom from vendor lock-in.^{[109][110]}

Desktops and laptops

According to web server statistics (that is, based on the numbers recorded from visits to websites by client devices), as of February 2024, the estimated market share of Linux on desktop computers is around 3.7%. In comparison, Microsoft Windows has a market share of around 72.9%, while macOS covers around 16.13%.^[26]

Web servers

W3Cook publishes stats that use the top 1,000,000 Alexa domains,^[111] which as of May 2015 estimate that 96.55% of web servers run Linux, 1.73% run Windows, and 1.72% run FreeBSD.^[112]

W3Techs publishes stats that use the top 10,000,000 Alexa domains and the top 1,000,000 Tranco domains, updated monthly^[113] and as of November 2020 estimate that Linux is used by 39% of the web servers, versus 21.9% being used by Microsoft Windows.^[114] 40.1% used other types of Unix.^[115]

IDC's Q1 2007 report indicated that Linux held 12.7% of the overall server market at that time;^[116] this estimate was based on the number of Linux servers sold by various companies, and did not include server hardware purchased separately that had Linux installed on it later.

Mobile devices

Android, which is based on the Linux kernel, has become the dominant operating system for smartphones. In April 2023, 68.61% of mobile devices accessing websites using StatCounter were from Android.^[117] Android is also a popular operating system for tablets, being responsible for more than 60% of tablet sales as of 2013.^[118] According to web server statistics, as of October 2021 Android has a market share of about 71%, with iOS holding 28%, and the remaining 1% attributed to various niche platforms.^[119]

Film production

For years, Linux has been the platform of choice in the film industry. The first major film produced on Linux servers was 1997's *Titanic*.^{[120][121]} Since then major studios including DreamWorks Animation, Pixar, Weta Digital, and Industrial Light & Magic have migrated to Linux.^{[122][123][124]} According to the Linux Movies Group, more than 95% of the servers and desktops at large animation and visual effects companies use Linux.^[125]

Use in government

Linux distributions have also gained popularity with various local and national governments. News of the Russian military creating its own Linux distribution has also surfaced, and has come to fruition as the G.H.ost Project.^[126] The Indian state of Kerala has gone to the extent of mandating that all state high schools run Linux on their computers.^{[127][128]} China uses Linux exclusively as the operating system for its Loongson processor family to achieve technology independence.^[129] In Spain, some regions have developed their own Linux distributions, which are widely used in education and official institutions, like gnuLinEx in Extremadura and Guadalinex in Andalusia. France and Germany have also taken steps toward the adoption of Linux.^[130] North Korea's Red Star OS, developed as of 2002, is based on a version of Fedora Linux.^[131]

Copyright, trademark, and naming

The Linux kernel is licensed under the GNU General Public License (GPL), version 2. The GPL requires that anyone who distributes software based on source code under this license must make the originating source code (and any modifications) available to the recipient under the same terms.^[132] Other key components of a typical Linux distribution are also mainly licensed under the GPL, but they may use other licenses; many libraries use the GNU Lesser General Public License (LGPL), a more permissive variant of the GPL, and the X.Org implementation of the X Window System uses the MIT License.

Torvalds states that the Linux kernel will not move from version 2 of the GPL to version 3.^[133]^[134] He specifically dislikes some provisions in the new license which prohibit the use of the software in digital rights management.^[135] It would also be impractical to obtain permission from all the copyright holders, who number in the thousands.^[136]

A 2001 study of Red Hat Linux 7.1 found that this distribution contained 30 million source lines of code.^[137] Using the Constructive Cost Model, the study estimated that this distribution required about eight thousand person-years of development time. According to the study, if all

this software had been developed by conventional proprietary means, it would have cost about US\$1.82 billion^[138] to develop in 2023 in the United States.^[137] Most of the source code (71%) was written in the C programming language, but many other languages were used, including C+, Lisp, assembly language, Perl, Python, Fortran, and various shell scripting languages. Slightly over half of all lines of code were licensed under the GPL. The Linux kernel itself was 2.4 million lines of code, or 8% of the total.^[137]

In a later study, the same analysis was performed for Debian version 4.0 (etch, which was released in 2007).^[139] This distribution contained close to 283 million source lines of code, and the study estimated that it would have required about seventy three thousand man-years and cost US\$10.2 billion^[138] (in 2023 dollars) to develop by conventional means.

In the United States, the name *Linux* is a trademark registered to Linus Torvalds.^[10] Initially, nobody registered it. However, on August 15, 1994, William R. Della Croce Jr. filed for the trademark *Linux*, and then demanded royalties from Linux distributors. In 1996, Torvalds and some affected organizations sued him to have the trademark assigned to Torvalds, and, in 1997, the case was settled.^[141] The licensing of the trademark has since been handled by the Linux Mark Institute (LMI). Torvalds has stated that he trademarked the name only to prevent someone else from using it. LMI originally charged a nominal sublicensing fee for use of the Linux name as part of trademarks,^[142] but later changed this in favor of offering a free, perpetual worldwide sublicense.^[143]



The name "Linux" is also used for a laundry detergent made by Swiss company Rösch.^[140]

The Free Software Foundation (FSF) prefers *GNU/Linux* as the name when referring to the operating system as a whole, because it considers Linux distributions to be variants of the GNU operating system initiated in 1983 by Richard Stallman, president of the FSF.^{[16][17]} They explicitly take no issue over the name Android for the Android OS, which is also an operating system based on the Linux kernel, as GNU is not a part of it.

A minority of public figures and software projects other than Stallman and the FSF, notably distributions consisting of only free software, such as Debian (which had been sponsored by the FSF up to 1996),^[144] also use *GNU/Linux* when referring to the operating system as a whole.^{[145][146][147]} Most media and common usage, however, refers to this family of operating systems simply as *Linux*, as do many large Linux distributions (for example, SUSE Linux and Red Hat Enterprise Linux).

As of May 2011, about 8% to 13% of the lines of code of the Linux distribution Ubuntu (version "Natty") is made of GNU components (the range depending on whether GNOME is considered part of GNU); meanwhile, 6% is taken by the Linux kernel, increased to 9% when including its direct dependencies.^[148]

See also

