

Unix (/ˈjuːnɪks/, *YOO-niks*; trademarked as **UNIX**) is a family of multitasking, multi-user computer operating systems that derive from the original AT&T Unix, whose development started in 1969^[1] at the Bell Labs research center by Ken Thompson, Dennis Ritchie, and others.^[4]

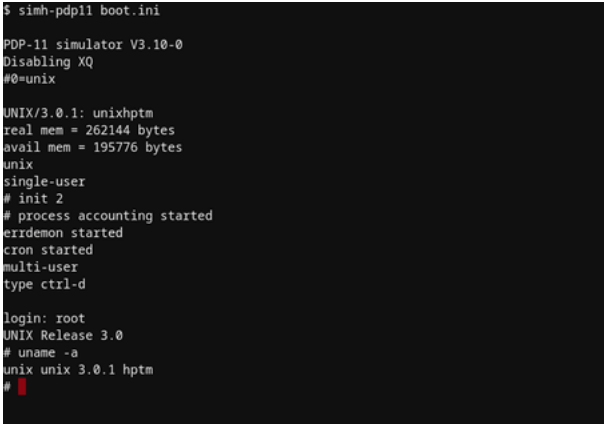
Initially intended for use inside the Bell System, AT&T licensed Unix to outside parties in the late 1970s, leading to a variety of both academic and commercial Unix variants from vendors including University of California, Berkeley (BSD), Microsoft (Xenix), Sun Microsystems (SunOS/Solaris), HP/HPE (HP-UX), and IBM (AIX). In the early 1990s, AT&T sold its rights in Unix to Novell, which then sold the UNIX trademark to The Open Group, an industry consortium founded in 1996. The Open Group allows the use of the mark for certified operating systems that comply with the Single UNIX Specification (SUS).

Unix systems are characterized by a modular design that is sometimes called the "Unix philosophy". According to this philosophy, the operating system should provide a set of simple tools, each of which performs a limited, well-defined function.^[5] A unified and inode-based filesystem (the Unix filesystem) and an inter-process communication mechanism known as "pipes" serve as the main means of communication,^[4] and a shell scripting and command language (the Unix shell) is used to combine the tools to perform complex workflows.

Unix distinguishes itself from its predecessors as the first portable operating system: almost the entire operating system is written in the C programming language, which allows Unix to operate on numerous platforms.^[6]

Overview

Unix was originally meant to be a convenient platform for programmers developing software to be run on it and on other systems, rather than for

<div><div>Unix</div><div><div>UNIX[®]</div><div>An Open Group Standard</div></div></div>	
	
Unix System III running on a PDP-11 simulator	
Developer	Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, and Joe Ossanna at Bell Labs
Written in	C and assembly language
OS family	Unix
Source model	Historically proprietary software, while some Unix projects (including BSD family and illumos) are open-source
Initial release	Development started in 1969 First manual published internally in November 1971 ^[1] Announced outside Bell Labs in October 1973 ^[2]
Available in	English
Kernel type	Varies; monolithic, microkernel, hybrid
Influenced by	CTSS, ^[3] Multics

non-programmers.^{[7][8][9]} The system grew larger as the operating system started spreading in academic circles, and as users added their own tools to the system and shared them with colleagues.^[10]

At first, Unix was not designed to be portable^[6] or for multi-tasking.^[11] Later, Unix gradually gained portability, multi-tasking and multi-user capabilities in a time-sharing configuration. Unix systems are characterized by various concepts: the use of plain text for storing data; a hierarchical file system; treating devices and certain types of inter-process communication (IPC) as files; and the use of a large number of software tools, small programs that can be strung together through a command-line interpreter using pipes, as opposed to using a single monolithic program that includes all of the same functionality. These concepts are collectively known as the "Unix philosophy". Brian Kernighan and Rob Pike summarize this in *The Unix Programming Environment* as "the idea that the power of a system comes more from the relationships among programs than from the programs themselves".^[12]

By the early 1980s, users began seeing Unix as a potential universal operating system, suitable for computers of all sizes.^{[13][14]} The Unix environment and the client–server program model were essential elements in the development of the Internet and the reshaping of computing as centered in networks rather than in individual computers.

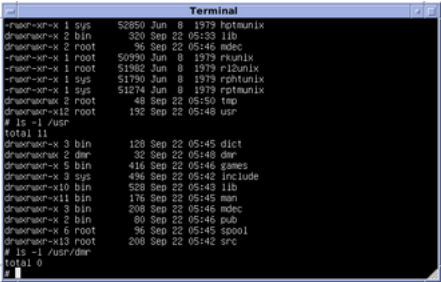
Both Unix and the C programming language were developed by AT&T and distributed to government and academic institutions, which led to both being ported to a wider variety of machine families than any other operating system.

The Unix operating system consists of many libraries and utilities along with the master control program, the kernel. The kernel provides services to start and stop programs, handles the file system and other common "low-level" tasks that most programs share, and schedules access to avoid conflicts when programs try to access the same resource or device simultaneously. To mediate such access, the kernel has special rights, reflected in the distinction of *kernel space* from user space, the latter being a lower priority realm where most application programs operate.

History

The origins of Unix date back to the mid-1960s when the Massachusetts Institute of Technology, Bell Labs, and General Electric were developing Multics, a time-sharing operating system for the GE-645 mainframe computer.^[15] Multics featured several innovations, but also presented severe problems. Frustrated by the size and complexity of Multics, but not by its goals, individual researchers at Bell Labs started withdrawing from the project. The last to leave were Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna,^[11] who decided to reimplement their experiences in a new project of smaller scale. This new operating system was initially without organizational backing, and also without a name.

Default user interface	<u>Command-line interface</u> and <u>Graphical</u> (<u>Wayland</u> and <u>X Window System</u> ; <u>Android SurfaceFlinger</u> ; <u>macOS Quartz</u>)
License	Varies; some versions are <u>proprietary</u> , others are <u>free/open-source software</u>
Official website	<u>opengroup.org/unix</u> (<u>http://opengroup.org/unix</u>)



Version 7 Unix, the Research Unix ancestor of all modern Unix systems

The new operating system was a single-tasking system.^[11] In 1970, the group coined the name *Unics* for *Uniplexed Information and Computing Service* as a pun on *Multics*, which stood for *Multiplexed Information and Computer Services*. Brian Kernighan takes credit for the idea, but adds that "no one can remember" the origin of the final spelling *Unix*.^[16] Dennis Ritchie,^[11] Doug McIlroy,^[1] and Peter G. Neumann^[17] also credit Kernighan.

The operating system was originally written in assembly language, but in 1973, Version 4 Unix was rewritten in C.^[11] Version 4 Unix, however, still had many PDP-11 dependent codes, and was not suitable for porting. The first port to another platform was a port of Version 6, made four years later (1977) at the University of Wollongong for the Interdata 7/32,^[18] followed by a Bell Labs port of Version 7 to the Interdata 8/32 during 1977 and 1978.^[19]

Bell Labs produced several versions of Unix that are collectively referred to as *Research Unix*. In 1975, the first source license for *UNIX* was sold to Donald B. Gillies at the University of Illinois Urbana–Champaign Department of Computer Science (UIUC).^[20]

During the late 1970s and early 1980s, the influence of Unix in academic circles led to large-scale adoption of Unix (BSD and System V) by commercial startups, which in turn led to Unix fragmenting into multiple, similar — but often slightly and mutually incompatible — systems including DYNIX, HP-UX, SunOS/Solaris, AIX, and Xenix. In the late 1980s, AT&T Unix System Laboratories and Sun Microsystems developed System V Release 4 (SVR4), which was subsequently adopted by many commercial Unix vendors.

In the 1990s, Unix and Unix-like systems grew in popularity and became the operating system of choice for over 90% of the world's top 500 fastest supercomputers,^[21] as BSD and Linux distributions were developed through collaboration by a worldwide network of programmers. In 2000, Apple released Darwin, also a Unix system, which became the core of the Mac OS X operating system, later renamed macOS.^[22]

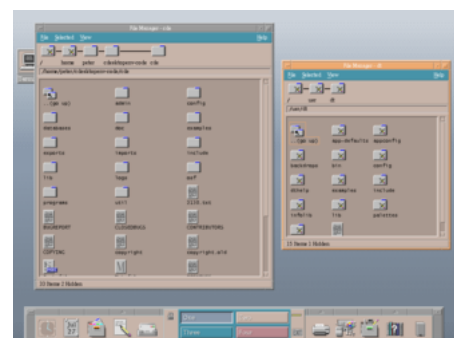
Unix-like operating systems are widely used in modern servers, workstations, and mobile devices.^[23]

Standards

In the late 1980s, an open operating system standardization effort now known as POSIX provided a common baseline for all operating systems; IEEE based POSIX around the common structure of the major competing variants of the Unix system, publishing the first POSIX standard in 1988. In the early 1990s, a separate but very similar effort was started by an industry consortium, the Common Open Software Environment (COSE) initiative, which eventually became the Single UNIX Specification (SUS) administered by The Open Group. Starting in 1998, the Open Group and IEEE started the Austin Group, to provide a common definition of POSIX and the Single UNIX Specification, which, by 2008, had become the Open Group Base Specification.



Ken Thompson (sitting) and Dennis Ritchie working together at a PDP-11



The Common Desktop Environment (CDE), part of the COSE initiative

In 1999, in an effort towards compatibility, several Unix system vendors agreed on SVR4's Executable and Linkable Format (ELF) as the standard for binary and object code files. The common format allows substantial binary compatibility among different Unix systems operating on the same CPU architecture.

The Filesystem Hierarchy Standard was created to provide a reference directory layout for Unix-like operating systems; it has mainly been used in Linux.

Components

The Unix system is composed of several components that were originally packaged together. By including the development environment, libraries, documents and the portable, modifiable source code for all of these components, in addition to the kernel of an operating system, Unix was a self-contained software system. This was one of the key reasons it emerged as an important teaching and learning tool and has had such a broad influence.

The inclusion of these components did not make the system large – the original V7 UNIX distribution, consisting of copies of all of the compiled binaries plus all of the source code and documentation occupied less than 10 MB and arrived on a single nine-track magnetic tape, earning its reputation as a portable system.^[24] The printed documentation, typeset from the online sources, was contained in two volumes.

The names and filesystem locations of the Unix components have changed substantially across the history of the system. Nonetheless, the V7 implementation is considered by many to have the canonical early structure:

- *Kernel* – source code in `/usr/sys`, composed of several sub-components:
 - *conf* – configuration and machine-dependent parts, including boot code
 - *dev* – device drivers for control of hardware (and some pseudo-hardware)
 - *sys* – operating system "kernel", handling memory management, process scheduling, system calls, etc.
 - *h* – header files, defining key structures within the system and important system-specific invariables
- *Development environment* – early versions of Unix contained a development environment sufficient to recreate the entire system from source code:
 - *ed* – text editor, for creating source code files
 - *cc* – C language compiler (first appeared in V3 Unix)
 - *as* – machine-language assembler for the machine
 - *ld* – linker, for combining object files
 - *lib* – object-code libraries (installed in `/lib` or `/usr/lib`). *libc*, the system library with C run-time support, was the primary library, but there have always been additional libraries for things such as mathematical functions (*libm*) or database access. V7 Unix introduced the first version of the modern "Standard I/O" library *stdio* as part of the system library. Later implementations increased the number of libraries significantly.
 - *make* – build manager (introduced in PWB/UNIX), for effectively automating the build process
 - *include* – header files for software development, defining standard interfaces and system invariants
 - *Other languages* – V7 Unix contained a Fortran-77 compiler, a programmable arbitrary-

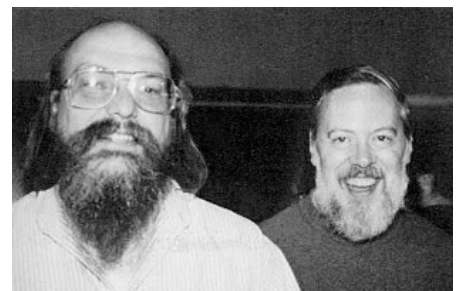
precision calculator (*bc*, *dc*), and the *awk* scripting language; later versions and implementations contain many other language compilers and toolsets. Early BSD releases included Pascal tools, and many modern Unix systems also include the GNU Compiler Collection as well as or instead of a proprietary compiler system.

- *Other tools* – including an object-code archive manager (*ar*), symbol-table lister (*nm*), compiler-development tools (e.g. *lex* & *yacc*), and debugging tools.
- *Commands* – Unix makes little distinction between commands (user-level programs) for system operation and maintenance (e.g. *cron*), commands of general utility (e.g. *grep*), and more general-purpose applications such as the text formatting and typesetting package. Nonetheless, some major categories are:
 - *sh* – the "shell" programmable command-line interpreter, the primary user interface on Unix before window systems appeared, and even afterward (within a "command window").
 - *Utilities* – the core toolkit of the Unix command set, including *cp*, *ls*, *grep*, *find* and many others. Subcategories include:
 - *System utilities* – administrative tools such as *mkfs*, *fsck*, and many others.
 - *User utilities* – environment management tools such as *passwd*, *kill*, and others.
 - *Document formatting* – Unix systems were used from the outset for document preparation and typesetting systems, and included many related programs such as *nroff*, *troff*, *tbl*, *eqn*, *refer*, and *pic*. Some modern Unix systems also include packages such as TeX and Ghostsript.
 - *Graphics* – the *plot* subsystem provided facilities for producing simple vector plots in a device-independent format, with device-specific interpreters to display such files. Modern Unix systems also generally include X11 as a standard windowing system and GUI, and many support OpenGL.
 - *Communications* – early Unix systems contained no inter-system communication, but did include the inter-user communication programs *mail* and *write*. V7 introduced the early inter-system communication system UUCP, and systems beginning with BSD release 4.1c included TCP/IP utilities.
- *Documentation* – Unix was one of the first operating systems to include all of its documentation online in machine-readable form.^[25] The documentation included:
 - *man* – manual pages for each command, library component, system call, header file, etc.
 - *doc* – longer documents detailing major subsystems, such as the C language and *troff*

Impact

The Unix system had a significant impact on other operating systems. It achieved its reputation by its interactivity, by providing the software at a nominal fee for educational use, by running on inexpensive hardware, and by being easy to adapt and move to different machines. Unix was originally written in assembly language, but was soon rewritten in C, a high-level programming language.^[26] Although this followed the lead of CTSS, Multics and Burroughs MCP, it was Unix that popularized the idea.

Unix had a drastically simplified file model compared to many contemporary operating systems: treating all kinds of



Ken Thompson and Dennis Ritchie, principal developers of Research Unix

files as simple byte arrays. The file system hierarchy contained machine services and devices (such as printers, terminals, or disk drives), providing a uniform interface, but at the expense of occasionally requiring additional mechanisms such as ioctl and mode flags to access features of the hardware that did not fit the simple "stream of bytes" model. The Plan 9 operating system pushed this model even further and eliminated the need for additional mechanisms.



Photo from USENIX 1984, including Dennis Ritchie (center)

Unix also popularized the hierarchical file system with arbitrarily nested subdirectories, originally introduced by Multics. Other common operating systems of the era had ways to divide a storage device into multiple directories or sections, but they had a fixed number of levels, often only one level. Several major proprietary operating systems eventually added recursive subdirectory capabilities also patterned after Multics. DEC's RSX-11M's "group, user" hierarchy evolved into OpenVMS directories, CP/M's volumes evolved into MS-DOS 2.0+ subdirectories, and HP's MPE group.account hierarchy and IBM's SSP and OS/400 library systems were folded into broader POSIX file systems.

Making the command interpreter an ordinary user-level program, with additional commands provided as separate programs, was another Multics innovation popularized by Unix. The Unix shell used the same language for interactive commands as for scripting (shell scripts – there was no separate job control language like IBM's JCL). Since the shell and OS commands were "just another program", the user could choose (or even write) their own shell. New commands could be added without changing the shell itself. Unix's innovative command-line syntax for creating modular chains of producer-consumer processes (pipelines) made a powerful programming paradigm (coroutines) widely available. Many later command-line interpreters have been inspired by the Unix shell.

A fundamental simplifying assumption of Unix was its focus on newline-delimited text for nearly all file formats. There were no "binary" editors in the original version of Unix – the entire system was configured using textual shell command scripts. The common denominator in the I/O system was the byte – unlike "record-based" file systems. The focus on text for representing nearly everything made Unix pipes especially useful and encouraged the development of simple, general tools that could easily be combined to perform more complicated *ad hoc* tasks. The focus on text and bytes made the system far more scalable and portable than other systems. Over time, text-based applications have also proven popular in application areas, such as printing languages (PostScript, ODF), and at the application layer of the Internet protocols, e.g., FTP, SMTP, HTTP, SOAP, and SIP.

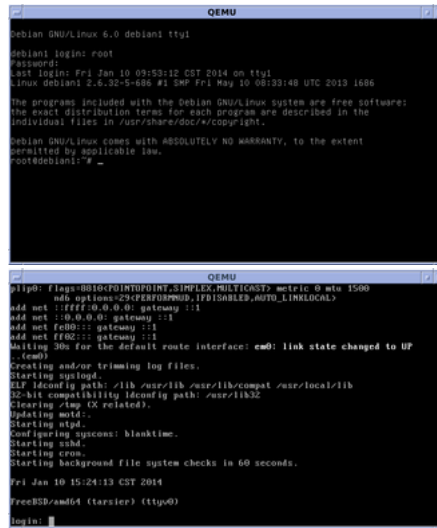
Unix popularized a syntax for regular expressions that found widespread use. The Unix programming interface became the basis for a widely implemented operating system interface standard (POSIX, see above). The C programming language soon spread beyond Unix, and is now ubiquitous in systems and applications programming.

Early Unix developers were important in bringing the concepts of modularity and reusability into software engineering practice, spawning a "software tools" movement. Over time, the leading developers of Unix (and programs that ran on it) established a set of cultural norms for developing software, norms which became as important and influential as the technology of Unix itself; this has been termed the Unix philosophy.

The TCP/IP networking protocols were quickly implemented on the Unix versions widely used on relatively inexpensive computers, which contributed to the Internet explosion of worldwide, real-time connectivity and formed the basis for implementations on many other platforms.

The Unix policy of extensive on-line documentation and (for many years) ready access to all system source code raised programmer expectations, and contributed to the launch of the free software movement in 1983.

Free Unix and Unix-like variants



```
Debian GNU/Linux 6.0 debian tty1
debian login: root
Password:
Last login: Fri Jan 10 09:53:12 CST 2014 on tty1
Linux debian 2.6.32-5-686 #1 SMP Fri May 10 08:33:48 UTC 2013 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@debian:~# _

QEMU
qtp0: flags=0x0<POINTOPOINT,SIMPLEX,MULTICAST> metric 0 wta 1500
add options<24/FERDUMP,IFDINBLK,AUTO_LOCAL>
add net :ffff:0.0.0.0: gateway :1
add net :0.0.0.0: gateway :1
add net fe80:: gateway :1
add net ff02:: gateway :1
Waiting 30s for the default route interface: em0: link state changed to UP
(em0)
Creating and/or trimming log files.
Starting syslogd.
ELF ldconfig path: /lib /usr/lib /usr/lib/compat /usr/local/lib
32-bit compatibility ldconfig path: /usr/lib32
Clearing /tmp (X related).
Updating world.
Starting atd.
Configuring systems: blanktime.
Starting lsmd.
Starting cron.
Starting background file system checks in 60 seconds.
Fri Jan 10 15:24:13 CST 2014
FreeBSD/amd64 (tarsier) (tty0)
login: _
```

Console screenshots of Debian (top, a popular Linux distribution) and FreeBSD (bottom, a popular Unix-like operating system)

In 1983, Richard Stallman announced the GNU (short for "GNU's Not Unix") project, an ambitious effort to create a free software Unix-like system—"free" in the sense that everyone who received a copy would be free to use, study, modify, and redistribute it. The GNU project's own kernel development project, GNU Hurd, had not yet produced a working kernel, but in 1991 Linus Torvalds released the Linux kernel as free software under the GNU General Public License. In addition to their use in the GNU operating system, many GNU packages – such as the GNU Compiler Collection (and the rest of the GNU toolchain), the GNU C library and the GNU Core Utilities – have gone on to play central roles in other free Unix systems as well.

Linux distributions, consisting of the Linux kernel and large collections of compatible software have become popular both with individual users and in business. Popular distributions include Red Hat Enterprise Linux, Fedora, SUSE Linux Enterprise, openSUSE, Debian, Ubuntu, Linux Mint, OpenMandriva, Slackware Linux, Arch Linux and Gentoo.^[27]

A free derivative of BSD Unix, 386BSD, was released in 1992 and led to the NetBSD and FreeBSD projects. With the 1994 settlement of a lawsuit brought against the University of California and Berkeley Software Design Inc. (*USL v. BSDi*) by Unix System Laboratories, it was clarified that Berkeley had the right to distribute BSD Unix for free if it so desired. Since then, BSD Unix has been developed in several different product branches, including OpenBSD and DragonFly BSD.

Linux and BSD Unix are increasingly filling the market needs traditionally served by proprietary Unix operating systems, as well as expanding into new markets such as the consumer desktop and mobile and embedded devices. Because of the modular design of the Unix model, sharing components is relatively common: most or all Unix and Unix-like systems include at least some BSD code, while some include GNU utilities in their distributions.

In a 1999 interview, Dennis Ritchie voiced his opinion that Linux and BSD Unix operating systems are a continuation of the basis of the Unix design and are derivatives of Unix:^[28]

I think the Linux phenomenon is quite delightful, because it draws so strongly on the basis that Unix provided. Linux seems to be among the healthiest of the direct Unix derivatives, though there are also the various BSD systems as well as the more official offerings from the workstation and mainframe manufacturers.

In the same interview, he states that he views both Unix and Linux as "the continuation of ideas that were started by Ken and me and many others, many years ago".^[28]

OpenSolaris was the free software counterpart to Solaris developed by Sun Microsystems, which included a CDDL-licensed kernel and a primarily GNU userland. However, Oracle discontinued the project upon their acquisition of Sun, which prompted a group of former Sun employees and members of the OpenSolaris community to fork OpenSolaris into the illumos kernel. As of 2014, illumos remains the only active, open-source System V derivative.

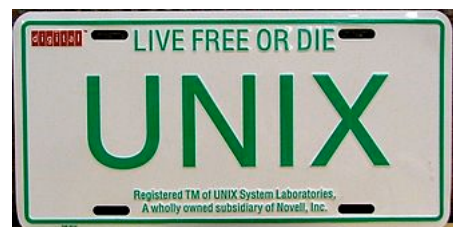
ARPANET

In May 1975, RFC 681 described the development of *Network Unix* by the Center for Advanced Computation at the University of Illinois Urbana-Champaign.^[29] The Unix system was said to "present several interesting capabilities as an ARPANET mini-host". At the time, Unix required a license from Bell Telephone Laboratories that cost US\$20,000 for non-university institutions, while universities could obtain a license for a nominal fee of \$150. It was noted that Bell was "open to suggestions" for an ARPANET-wide license.

The RFC specifically mentions that Unix "offers powerful local processing facilities in terms of user programs, several compilers, an editor based on QED, a versatile document preparation system, and an efficient file system featuring sophisticated access control, mountable and de-mountable volumes, and a unified treatment of peripherals as special files." The latter permitted the Network Control Program (NCP) to be integrated within the Unix file system, treating network connections as special files that could be accessed through standard Unix I/O calls, which included the added benefit of closing all connections on program exit, should the user neglect to do so. In order "to minimize the amount of code added to the basic Unix kernel", much of the NCP code ran in a swappable user process, running only when needed.^[29]

Branding

In October 1993, Novell, the company that owned the rights to the Unix System V source at the time, transferred the trademarks of Unix to the X/Open Company (now The Open Group),^[30] and in 1995 sold the related business operations to Santa Cruz Operation (SCO).^{[31][32]} Whether Novell also sold the copyrights to the actual software was the subject of a federal lawsuit in 2006, SCO v. Novell, which Novell won. The case was appealed, but on August 30, 2011, the United States Court of Appeals for the Tenth Circuit affirmed the trial decisions, closing the case.^[33] Unix vendor SCO Group Inc. accused Novell of slander of title.



Promotional license plate by Digital Equipment Corporation

The present owner of the trademark *UNIX* is The Open Group, an industry standards consortium. Only systems fully compliant with and certified to the Single UNIX Specification qualify as "UNIX" (others are called "Unix-like").

By decree of The Open Group, the term "UNIX" refers more to a class of operating systems than to a specific implementation of an operating system; those operating systems which meet The Open Group's Single UNIX Specification should be able to bear the UNIX 98 or UNIX 03 trademarks today, after the operating system's vendor pays a substantial certification fee and annual trademark royalties to The Open Group.^[34] Systems that have been licensed to use the UNIX trademark include AIX,^[35] EulerOS,^[36] HP-UX,^[37] Inspur K-UX,^[38] IRIX,^[39] macOS,^[40] Solaris,^[41] Tru64 UNIX (formerly "Digital UNIX", or OSF/1),^[42] and z/OS.^[43] Notably, EulerOS and Inspur K-UX are Linux distributions certified as UNIX 03 compliant.^{[44][45]}



HP 9000 workstation running HP-UX, a certified Unix operating system

Sometimes a representation like *Un*x*, **NIX*, or **N?X* is used to indicate all operating systems similar to Unix. This comes from the use of the asterisk (*) and the question mark characters as wildcard indicators in many utilities. This notation is also used to describe other Unix-like systems that have not met the requirements for UNIX branding from the Open Group.

The Open Group requests that *UNIX* always be used as an adjective followed by a generic term such as *system* to help avoid the creation of a genericized trademark.

Unix was the original formatting, but the usage of *UNIX* remains widespread because it was once typeset in small caps (*Unix*). According to Dennis Ritchie, when presenting the original Unix paper to the third Operating Systems Symposium of the American Association for Computing Machinery (ACM), "we had a new typesetter and *troff* had just been invented and we were intoxicated by being able to produce small caps".^[46] Many of the operating system's predecessors and contemporaries used all-uppercase lettering, so many people wrote the name in upper case due to force of habit. It is not an acronym.^[47]

Trademark names can be registered by different entities in different countries and trademark laws in some countries allow the same trademark name to be controlled by two different entities if each entity uses the trademark in easily distinguishable categories. The result is that Unix has been used as a brand name for various products including bookshelves, ink pens, bottled glue, diapers, hair driers and food containers.^[48]

Several plural forms of Unix are used casually to refer to multiple brands of Unix and Unix-like systems. Most common is the conventional *Unices*, but *Unices*, treating Unix as a Latin noun of the third declension, is also popular. The pseudo-Anglo-Saxon plural form *Unixen* is not common, although occasionally seen. Sun Microsystems, developer of the Solaris variant, has asserted that the term *Unix* is itself plural, referencing its many implementations.^[49]

See also

- Comparison of operating systems and free and proprietary software
- List of operating systems, Unix systems, and Unix commands
- Plan 9 from Bell Labs
- Timeline of operating systems
- Unix time
- Market share of operating systems