

Linux Kernel

8.1 Introduction: The **Linux kernel** is a Unix-like operating system kernel used by a variety of operating systems based on it, which are usually in the form of Linux distributions. The Linux kernel is a prominent example of free and open source software.

The Linux kernel is released under the GNU General Public License version 2 (GPLv2) (plus some firmware images with various non-free licenses), and is developed by contributors worldwide. Day-to-day development discussions take place on the Linux kernel mailing list.

The Linux kernel was initially conceived and created in 1991 by Finnish computer science student Linus Torvalds. Linux rapidly accumulated developers and users who adapted code from other free software projects for use with the new operating system. The Linux kernel has received contributions from thousands of programmers.

8.2 History: History

In April 1991, Linus Torvalds, a 21-year-old student at the University of Helsinki, Finland started working on some simple ideas for an operating system. He started with a task switcher in Intel 80386 assembly language and a terminal driver. On 25 August 1991, Torvalds posted the following to *comp.os.minix*, a newsgroup on Usenet:

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since April, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months [...] Yes - it's free of any minix code, and it has a multi-threaded fs. It is NOT portable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have.

[...] It's mostly in C, but most people wouldn't call what I write C. It uses every conceivable feature of the 386 I could find, as it was also a project to teach me

about the 386. As already mentioned, it uses a MMU, for both paging (not to disk yet) and segmentation. It's the segmentation that makes it REALLY 386 dependent (every task has a 64Mb segment for code & data - max 64 tasks in 4Gb. Anybody who needs more than 64Mb/task - tough cookies). [...] Some of my "C"-files (specifically mm.c) are almost as much assembler as C. [...] Unlike minix, I also happen to LIKE interrupts, so interrupts are handled without trying to hide the reason behind them.

After that, many people contributed code to the project. Early on, the MINIX community contributed code and ideas to the Linux kernel. At the time, the GNU Project had created many of the components required for a free operating system, but its own kernel, GNU Hurd, was incomplete and unavailable. The BSD operating system had not yet freed itself from legal encumbrances. Despite the limited functionality of the early versions, Linux rapidly accumulated developers and users.

By September 1991, Linux version 0.01 was released on the FTP server (ftp.funet.fi) of the Finnish University and Research Network (FUNET). It had 10,239 lines of code. In October 1991, Linux version 0.02 was released.^[14]

In December 1991, Linux 0.11 was released. This version was the first to be self-hosted - Linux 0.11 could be compiled by a computer running Linux 0.11. When he released version 0.12 in February 1992, Torvalds adopted the GNU General Public License (GPL) over his previous self-drafted license, which had not permitted commercial redistribution.

A newsgroup known as *alt.os.linux* was started, and on 19 January 1992, the first post to alt.os.linux was made. On 31 March 1992, alt.os.linux became *comp.os.linux*.

The X Window System was soon ported to Linux. In March 1992, Linux version 0.95 was the first to be capable of running X. This large version number jump (from 0.1x to 0.9x) was due to a feeling that a version 1.0 with no major missing pieces was imminent. However, this proved to be somewhat overoptimistic, and from 1993 to early 1994, 15 development versions of version 0.99 appeared.

On 14 March 1994, Linux 1.0.0 was released, with 176,250 lines of code. In March 1995, Linux 1.2.0 was released (310,950 lines of code).

Version 2 of Linux, released on 9 June 1996, was followed by additional major versions under the version 2 header:

- 25 January 1999 - Linux 2.2.0 was released (1,800,847 lines of code).
- 18 December 1999 - IBM mainframe patches for 2.2.13 were published, allowing Linux to be used on enterprise-class machines.
- 4 January 2001 - Linux 2.4.0 was released (3,377,902 lines of code).
- 17 December 2003 - Linux 2.6.0 was released (5,929,913 lines of code).

Starting in 2004, the release process changed and new kernels started coming out on a regular schedule every 2–3 months, numbered 2.6.0, 2.6.1, up through 2.6.39.

On 21 July 2011 Linus Torvalds announced the release of Linux 3.0: "Gone are the 2.6.<bignum> days". The version bump is not about major technological changes when compared to Linux 2.6.39; it marks the kernel's 20th anniversary. The time-based release process remained the same.

As of 2013, the Linux 3.10 release had 15,803,499 lines of code.

Tanenbaum–Torvalds debate

The fact that Linux is a monolithic kernel rather than a microkernel was the topic of a debate between Andrew S. Tanenbaum, the creator of MINIX, and Linus Torvalds. The debate, started in 1992 on the Usenet discussion group comp.os.minix, was about Linux and kernel architecture in general. Tanenbaum argued that microkernels are superior to monolithic kernels and that therefore Linux is obsolete. Unlike traditional monolithic kernels, device drivers in Linux are easily configured as loadable kernel modules and are loaded or unloaded while running the system. This subject was revisited on 9 May 2006, and on 12 May 2006 Tanenbaum wrote a position statement.

8.2 Legal aspects

Licensing terms

Initially, Torvalds released Linux under a license which forbade any commercial use. This was changed in version 0.12 to the GNU General Public License (GPL). This license allows distribution and sale of possibly modified and unmodified versions of Linux but requires that all those copies be released under the same license and be accompanied by the complete corresponding source code.

Torvalds has described licensing Linux under the GPL as the "best thing I ever did."

GPL version

Currently, Linux is licensed only under version 2 of the GPL,^[5] without offering the licensee the option to choose "any later version", and there is some debate over how easily it could be changed to use later GPL versions such as version 3 (and whether this is even desirable). Torvalds himself specifically indicated upon the release of version 2.4.0 that his own code is only under version 2. However, the terms of the GPL state that if no version is specified, then any version may be used, and Alan Cox pointed out that very few other Linux contributors have specified a particular version of the GPL. In September 2006, a survey of 29 key kernel programmers indicated 28 preferred GPLv2 to the then-current GPLv3 draft. Torvalds commented, "I think a number of outsiders... believed that I personally was just the odd man out, because I've been so publicly not a huge fan of the GPLv3."

Loadable kernel modules

It is debated whether loadable kernel modules (LKMs) are to be considered derivative works under copyright law, and thereby fall under the terms of the GPL.

Torvalds has stated his belief that LKMs using only a limited, "public" subset of the kernel interfaces can sometimes be non-derived works, thus allowing some binary-only drivers and other LKMs that are not licensed under the GPL. A very good example for this is the usage of *dma_buf* by the proprietary NVidia graphics drivers. *dma_buf* is a recent kernel feature (like the rest of the kernel, it is licensed under the GPL) that allows multiple GPUs to quickly copy data into each other's framebuffers. One possible use case would be Nvidia Optimus that pairs a fast GPU with an Intel integrated GPU, where the NVIDIA GPU writes into the Intel framebuffer when it is active. But, NVIDIA cannot use this infrastructure because it uses a technical means to enforce the rule that it can only be used by LKMs that are also GPL. Alan Cox replied on LKML, rejecting a request from one of their engineers to remove this technical enforcement from the API. Not all Linux kernel contributors agree with this interpretation, however, and even Torvalds agrees that many LKMs are clearly derived works, and indeed he writes that "kernel modules ARE derivative 'by default'".

On the other hand Torvalds has also said that "one gray area in particular is something like a driver that was originally written for another operating system (i.e. clearly not a derived work of Linux in origin). [...] THAT is a gray area, and that is the area where I personally believe that some modules may be considered

to not be derived works simply because they weren't designed for Linux and don't depend on any special Linux behaviour." Proprietary graphics drivers, in particular, are heavily discussed. Ultimately, it is likely that such questions can only be resolved by a court.

Firmware binary blobs

One point of licensing controversy is Linux's use of firmware "binary blobs" to support several hardware devices. These files are under a variety of licenses, many of them restrictive and their exact underlying source code is usually unknown. In the year 2002 Richard Stallman stated why, in his point of view, such blobs make Linux partially non-free software, and that distributing Linux "violates the GPL" (which requires "complete corresponding source code" to be available). In 2008 Free Software Foundation Latin America started a project, Linux-libre, to create a completely free kernel without proprietary objects, which is used by some completely free distributions, such as those endorsed by the Free Software Foundation. On 15 December 2010 the Debian Project announced that the next Debian stable version would come with a kernel "stripped of all non-free firmware bits". This policy was continued with the sequel release of Debian 7 "Wheezy".

Trademark[edit]

Linux is a registered trademark of Linus Torvalds in the United States and some other countries. This is the result of an incident in which William Della Croce, Jr., who was not involved in the Linux project, trademarked the name and subsequently demanded royalties for its use. Several Linux backers retained legal counsel and filed suit against Della Croce. The issue was settled in August 1997 when the trademark was assigned to Linus Torvalds.

SCO litigation

In March 2003, the SCO Group (SCO) filed a lawsuit against IBM claiming that IBM had violated copyrights that SCO claimed to hold over the Unix source code, by contributing portions of that code to Linux. Additionally, SCO sent letters to a number of companies warning that their use of Linux without a license from SCO may be a violation of copyright law, and claimed in the press that they would be suing individual Linux users. IBM then promised to defend its Linux customers on their behalf. This controversy has generated lawsuits by SCO against Novell, DaimlerChrysler (partially dismissed in July, 2004), and AutoZone, and retaliatory lawsuits by Red Hat and others against SCO.

In early 2007 SCO filed the specific details of the purported copyright infringement. Despite previous claims that SCO was the rightful owner of 1 million lines of code, they specified 326 lines of code, most of which were uncopyrightable. In August 2007, the court in the Novell case ruled that SCO did not actually own the Unix copyrights to begin with, though the Tenth Circuit Court of Appeals ruled in August 2009 that the question of who owned the copyright properly remained for a jury to answer. The jury case was decided on 30 March 2010 in Novell's favour.

8.3 Architecture

The **Linux kernel** is ubiquitously found on various hardware and is supported by an abundance of both free and open-source and also proprietary software.

See also: [vmlinux](#)

Linux is a monolithic kernel. Device drivers and kernel extensions run in kernel space (ring 0 in many CPU architectures), with full access to the hardware, although some exceptions run in user space, for example filesystems based on FUSE. The graphics system most people use with Linux does not run within the kernel, in contrast to that found in Microsoft Windows. Unlike standard monolithic kernels, device drivers are easily configured as modules, and loaded or unloaded while running the system. Also unlike standard monolithic kernels, device drivers can be pre-empted under certain conditions. This latter feature was added to handle hardware interrupts correctly, and to improve support for symmetric multiprocessing.¹ By choice, the Linux kernel has no Binary Kernel Interface.

The hardware is also incorporated into the file hierarchy. Device drivers interface to user applications via an entry in the /dev and/or /sys directories. Process information as well is mapped to the file system through the /proc directory.

Linux supports true preemptive multitasking (both in user mode and kernel mode), virtual memory, shared libraries, demand loading, shared copy-on-write executables (via KSM), memory management, the Internet protocol suite, and threading.