Learn about the Linux Kernel and their main functions

With more than 13 million lines of code, the Linux kernel is one of the largest open source projects in the world, but what exactly are they and what do they do in the system?

With more than 13 million lines of code, the Linux kernel is one of the largest open source projects in the world, but what exactly are they and what do they do in the system?

Learn about the Linux Kernel

What is a kernel?

The kernel concept here refers to software, applications at low (*low-level*) *level* in the system, capable of changing flexibly to fit the hardware. They interact with all applications and operate in user mode, allowing other processes - or servers, to receive information from other inter-process communication (IPC) components.

Different types of kernels

In essence, there are many ways to build the structure and compile a certain set of kernels from scratch. In general, for most current kernels, we can divide into three categories: *monolithic* , *microkernel* , and *hybrid* . Linux uses a monolithic kernel while OS X (XNU) and Windows 7 use a kernel hybrid.

Microkernel

Microkernel has all the necessary features to manage microprocessor, memory and IPC. There are many other things in the computer that can be seen, exposed and managed in user mode. Microkernel has quite high flexibility, so you don't have to worry about changing a certain device, such as a video card, a hard drive, etc., or even an operating system. Microkernel with very small footprint related parameters, similar to memory and storage capacity, they also have high security because they clearly specify which processes operate in user mode, but not Authorized as supervisor mode.

Advantages :
- High flexibility
- Security
- Use less footprint installation and storage

Disadvantages :
- Hardware sometimes 'confusing' through the driver system
- Hardware operates under normal performance levels because the drivers are in user mode
- Processes must wait to receive information
- Processes are not accessible to other applications without waiting

Monolithic Kernel

With Monolithic, they have a broader function than microkernel, not only involved in managing microprocessors, memory, IRC, they also interfere with driver control, system file coordination feature. , interactions between servers . Monolithic is better when accessing hardware and multi-tasking, because if a program wants to gather information from memory and other processes, they need access rights. Direct access and no waiting for other tasks to end. But at the same time, they are also causing instability because many programs run in supervisor mode, only a small problem can cause system instability.

Advantages :
- Direct access to the hardware
- Easily handle signals and communication between multiple components together
- If fully supported, the hardware system will not need to install additional drivers and other software
- The process of processing and interacting faster because there is no need to wait

Disadvantages :
- Consumes a lot of footprint installation and storage
- Less security because they all work in supervisor mode

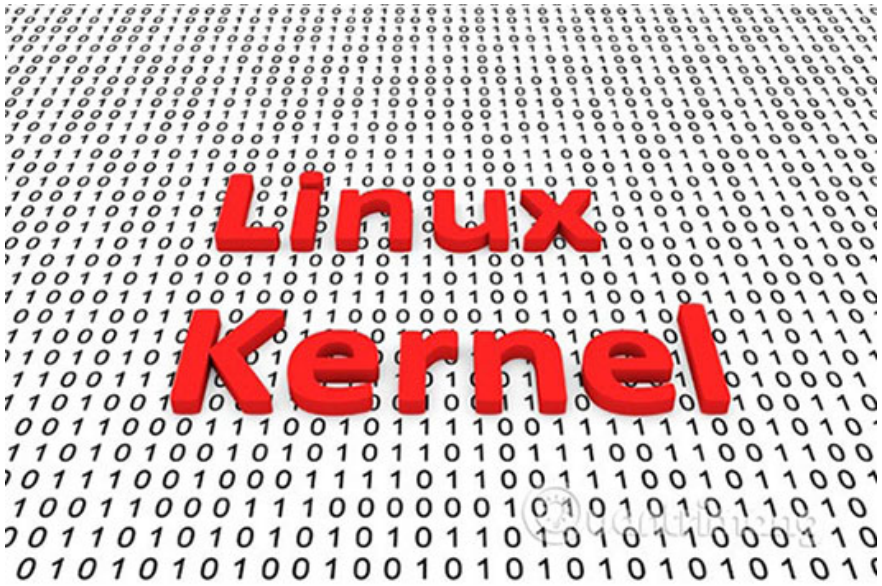
Hybrid Kernel

Unlike the above two types of kernels, Hybrid has the ability to select and decide which applications are allowed to run in user or supervisor mode. Typically, things like driver and I / O system files will operate in user mode while IPC and packets from the server are retained in supervisor mode. This feature is really useful because they ensure system efficiency, distribution and adjustment of work, easy to manage.

Advantages :
- Developers can select and classify which applications will run in the appropriate mode
- Use less footprint than the monolithic kernel
- Has the highest flexibility and mobility

Disadvantages :
- Can be left in the process of causing the system to hang similar to the microkernel
- The device drivers must be managed by the user

What is Linux kernel?



Technically, it is not accurate to consider Linux as a complete operating system. Linux actually refers specifically to the kernel, named after the founder of Linus Torvalds. Everything you see on the screen comes from other projects and developers.

Torvalds created the Linux kernel in 1991. Initially, he named the project Freax (a combination of 'free', 'freak' and 'UNIX'). But another Torvalds colleague liked the Linux name (then Linux became the official name). Torvalds released the first Linux version in 1992 under the GNU copyleft license (the license is free to copy, modify, distribute and not exclusively), becoming an important part of the success of the project. project.

Much of the Linux desktop experience comes from the GNU project, an old initiative that created a nearly complete desktop operating system. All it needs is a kernel and Linux has met that need. This is why some people call this operating system GNU / Linux.

Other free and open source desktops, like FreeBSD, look similar to Linux because they run most of the same GNU software.

Because the Linux kernel is available under the GNU license, little attention is paid to continuing to develop a separate kernel as part of the GNU project. And instead of creating other competitive kernels, like in Windows and macOS, many companies have chosen to use and contribute to the Linux kernel.

The Linux kernel has grown into a large project containing millions of lines of code. Thousands of people and more than a thousand companies contributed to the development of the kernel. This is one of the most outstanding examples of free and open source software in the world.

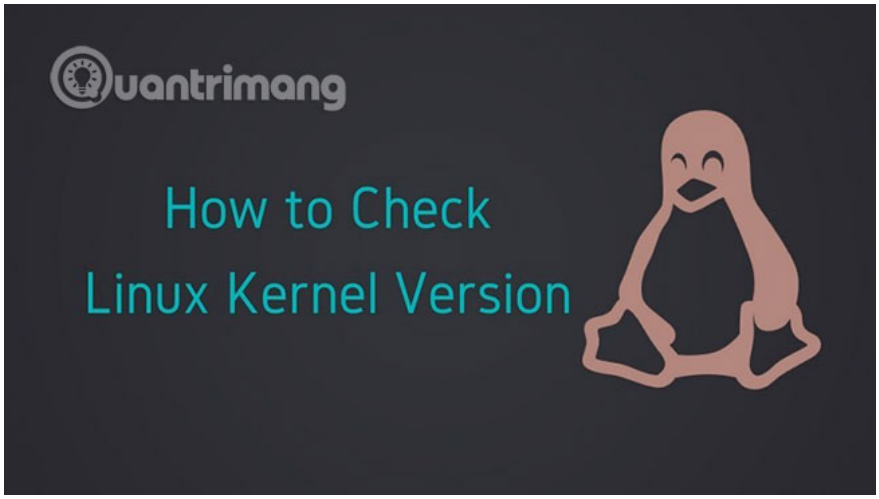
What is Linux kernel used for?

While Linux is still a relatively desktop operating system 'cocoon', the kernel is widely used in many places. Thanks to Android, Linux kernel has appeared on most smartphones, all kinds of mobile devices, including devices worn on people and cameras.

Linux is the "soul" of the 500 most powerful supercomputers and most of the Internet infrastructure. Talking about the cloud means that we are primarily referring to the connected Linux support servers.

The tiny Raspberry Pi is a Linux-powered computer, sized just by credit card, completely open for people to modify and use on-demand projects.

How to check the Linux kernel version



The kernel is still actively developing, so new Linux versions always appear.

The easiest way to see which release is running on your computer is to use the **uname** command **.** This is a command line tool that provides system information. You can find the Linux kernel version you are using by opening a terminal window and entering the following command:

```
uname -r
```

Examples are running Linux kernel version **4.20.16-200.fc29.x86_64.** These parameters have the following meanings:

1. **4** refers to the kernel version.
2. **20** refers to the current major revision.
3. **16** refers to the current sub-revision.
4. **200** mentioned bug fixes and patches applied to this version.
5. The last bit will tell us more about the running distribution. This string shows an example running 64-bit versions of Fedora 29.

Should you update the Linux kernel?

For most cases, Linux kernel works quietly in the background. You don't know its presence there and there is little reason to care about the kernel. The best way to update your Linux kernel version is to upgrade to the latest version of your preferred operating system based on Linux.

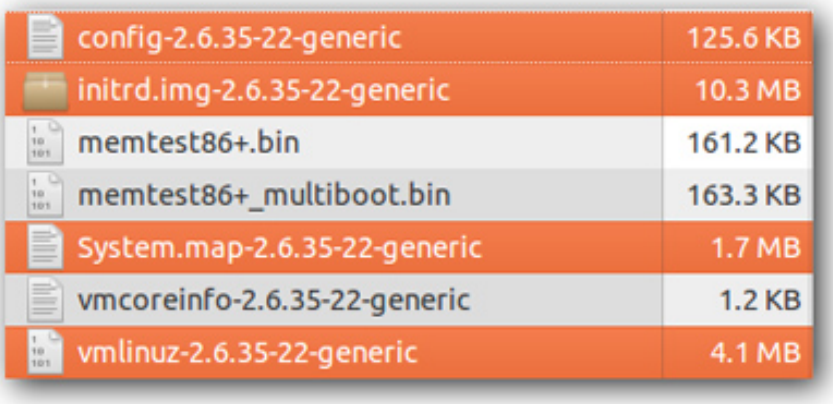
For example, new versions of Ubuntu and Fedora appear about every 6 months and come with a newer version of Linux kernel.

Unlike on Windows, Linux hardware drivers come with Linux kernel. So, if you have a relatively new laptop with speakers, WiFi or touchpad that the Linux version is not yet detected, you may have to wait for a newer version. The releases also come with improved stability and speed, so the computer can run more smoothly on this version than the other versions.

So where are these Linux Kernel files?

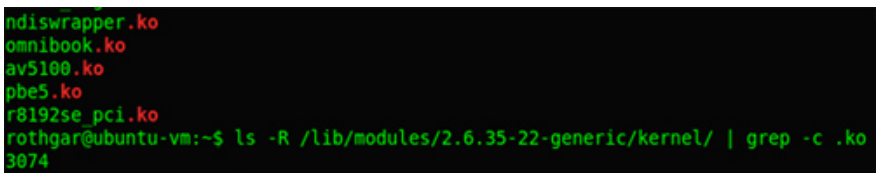
These kernel files are stored in / boot and vmlinuz-version in Ubuntu. When virtual memory begins to be developed to perform multi-threaded tasks, the vm prefix will be placed at the top of the kernel files to distinguish the ability to support virtualization technology. Since then, the Linux kernel has been called vmlinux, but this kernel system has been growing at a much faster rate than the standard boot memory capacity of the operating system, so these kernel files have been compressed. standard zlib - and the added z character is thus. There are also some other common compression formats: LZMA or BZIP2, but they are still called zImage.

The versions are ordered in ABCD format, where AB is usually 2.6, C represents the version, and D is the patch or patch symbol:

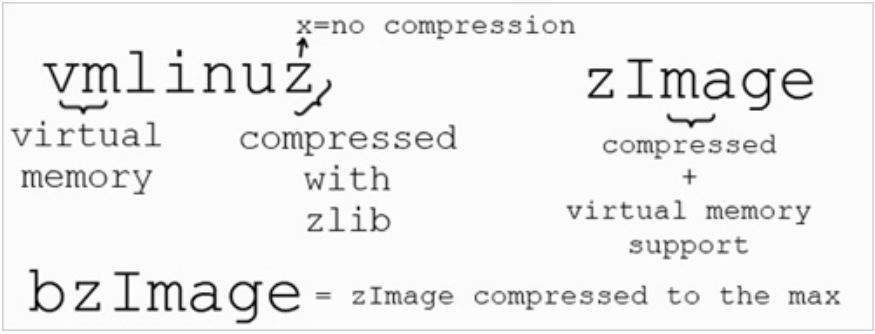


In the / boot directory, there are many other important files, such as initrd.img-version, system.map-version, and config-version. The initrd file is used as a RAM drive to extract and activate the actual kernel files, and the system.map file is used to manage the memory before the kernel is fully loaded, and the config file serves as a notification task for The kernel knows which options or modules will be loaded into the system boot process.

Linux Kernel file structure



In fact, Windows already has all available drivers and users just need to activate the corresponding drivers to use. And that is also the task that Linux kernel modules are responsible for, also called loadable kernel modules (LKM), which are necessary to keep the functions that come with the entire hardware system working without affecting Memory. A module normally assigns basic functions to kernels such as driver control, file system . LKM has an extension of .ko and is stored in / lib / modules directory, the user can set the autostart attribute, allow the load or not while the operating system starts, using the menuconfig command, intervene in / boot / config file, or using the modprobe command.



Third-party modules are usually available on some distributors, such as Ubuntu, or not built into the default mode. Software developers (eg nVidia, ATI or others) do not provide the source code but they have built and compiled the necessary modules themselves, then provided the user with the .ko file. And of course, there are many modules that are completely free, while others do not.

Two serious vulnerabilities in the Linux Kernel

Security researchers have revealed two serious vulnerabilities in the Linux Kernel, allowing attackers to gain root privileges on Linux systems.

The first vulnerability was discovered by researchers from Qualys security and was tracked under the name CVE-2018-14634. This vulnerability lies in the **create_elf_tables** () function of the Linux Kernel and can be exploited on 64 bit systems of local users, with access to SUID binary files.

According to Qualys researchers, who named this vulnerability is Mutagen Astronomy. This error has existed for the Linux Kernel for about a decade. The fix **b6a2fea39318** was introduced on July 19, 2007. Another fix, called **da029c11e6b1** , was released on July 7, 2017.

However, the fact is that not all Linux distributions use the da029c11e6b1 fix for their kernel and are still vulnerable. Distributions affected by this vulnerability include Red Hat Enterprise Linux (RHEL) 6, 7 and Red Hat Enterprise MRG 2, as well as CentOS, based on RHEL and Debian 8 Jessie (oldstable).

'This issue does not affect 32-bit systems because they do not have enough address space to exploit this vulnerability,' Red Hat said in a security consultation. 'Systems with less than 32GB of memory are very difficult to be affected by this problem due to memory requirements during operation'.

Red Hat plans to fix this problem in the future kernel update, but until then, there are manual solutions to protect systems against being exploited. Alternatives need to be applied again after restarting the system.

The second flaw, called **CVE-2018-17182,** was found by Jann Horn, a security researcher in Google's Project Zero project. This vulnerability can also be exploited to execute arbitrary code as root and affect all kernel versions since 3.16.

Horn showed how the vulnerability could be exploited on unconfigured kernels to enhance security, but warned that with additional effort, the code could trigger the bug right in the sandbox: Seccomp. Seccomp is the sandbox of Docker's main gVisor storage component and policy seccomp.

Horn said in a blog post: 'To make things easier, my exploits use different kernel interfaces, and therefore not only work from within sandboxes like that. In particular, it uses **/ dev / kmsg** to read dmesg logs and use an eBPF array to spam the kernel's page allocation tool (the page allocation is changeable) by the user '. "However, an attacker who is willing to invest more time in an exploit will avoid using such interfaces."

This error was reported to the Linux kernel maintainers on September 12 and a fix was created two days later, extremely fast compared to the error correction time of other software vendors.

However, he also pointed out a famous issue in the Linux ecosystem, highlighted by the Mutagen Astronomy vulnerability: When a vulnerability is patched in the Linux kernel, that does not mean that the user's system has been protect.

In fact, it may take a long time until the end user's system receives the patch. That's because most users use a specific Linux distribution and rely on receiving security fixes through it. These Linux distributions often use stable kernels, including older kernels, so they need to wait until the maintainers issue fixes.

Any delay can open a door for attackers, and sometimes, like in Mutagen Astronomy's case, some patches may not matter to a distribution at the time. At present, it can bring security meaning many years later.

Horn's patch for CVE-2018-17182 was returned to the 4.18, 4.14, 4.9 and 4.4 kernels on September 19, 4 days after the flaw was discovered, and became the official patch. However, when Horn published his detailed post on the blog, on September 26, Debian still uses the 4.9 kernel that has not been updated since August 21, while Ubuntu 16.04 migrates to an unused kernel. Sunday since August 27.

So, you can already imagine the importance of the kernel. Linux kernel is different from Mac OS X and Windows by driver driver as well as other management and support methods. Here are some basic and necessary information to help people understand what a kernel is, how it works and why they are so necessary.

Good luck!

See more:

1. How to create a Custom Kernel on Ubuntu
2. Compile the kernel
3. Basic steps to mastering a Linux system