**[Linux Kernel Monkey Log](http://www.kroah.com/log/)**

**Random bits from Greg Kroah-Hartman**

* [RSS](http://www.kroah.com/log/index.rss)

フォームの始まり



フォームの終わり

* [Blog](http://www.kroah.com/log/)
* [Archives](http://www.kroah.com/log/blog/archives)

[**What Stable Kernel Should I Use?**](http://www.kroah.com/log/blog/2018/08/24/what-stable-kernel-should-i-use/)

**どの安定版カーネルを使用すべきか？**

Aug 24th, 2018

I get a lot of questions about people asking me about what stable kernel should they be using for their product/device/laptop/server/etc. all the time. Especially given the now-extended length of time that some kernels are being supported by me and others, this isn’t always a very obvious thing to determine. So this post is an attempt to write down my opinions on the matter. Of course, you are free to use what ever kernel version you want, but here’s what I recommend.

製品/デバイス/ラップトップ/サーバー/などにどの安定したカーネルを使用すべきか、をいつも人々は私に尋ねてきます。特にいくつかのカーネルが私や他のメンバーがサポートしているために（サポート期間が）延長されているということが常に明白に決定されているとは限りません。したがってこの投稿はこの件に関する私の意見を書き留めることを試みています。もちろんどのバージョンのカーネルを使用するかは自由ですが、これが私の推奨です。

As always, the opinions written here are my own, I speak for no one but myself.

いつものようにここに書かれた意見は私自身のもので、私以外の誰に向けたものでもありません。

**What kernel to pick（どのカーネルを選択すべきか）**

Here’s the my short list of what kernel you should use, raked from best to worst options. I’ll go into the details of all of these below, but if you just want the summary of all of this, here it is:

これが最善から最悪な選択肢までを含めた、私の使用すべきカーネルの短いリストです。私はこれらのすべての詳細についていくつもりですが、要約だけが必要な場合はここにあります

Hierarchy of what kernel to use, from best solution to worst

どのカーネルを使用すべきかの最善から最悪までの序列:は

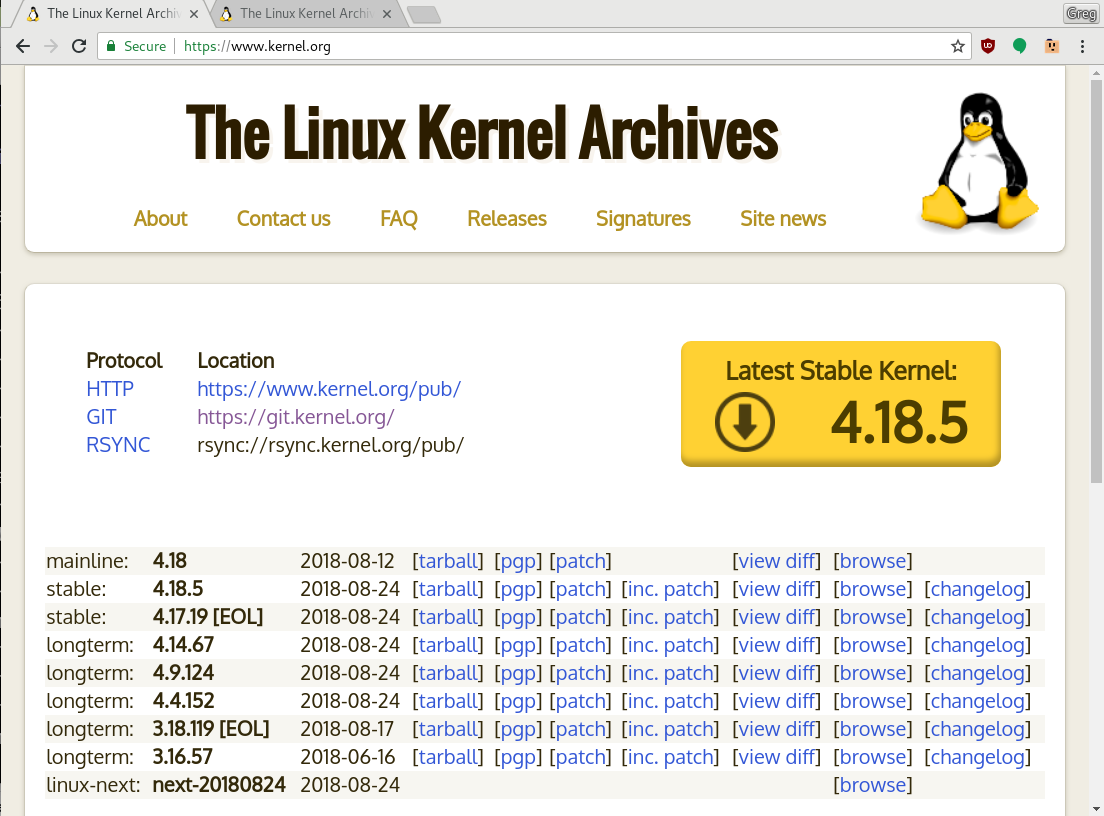
* Supported kernel from your favorite Linux distribution
* お好みのLinuxディストリビューションからのサポートされたカーネル
* Latest stable release
* 最新の安定板リリース
* Latest LTS release
* 最新のLTSリリース
* Older LTS release that is still being maintained
* メインテナンスされている古いLTSリリース

What kernel to never use:使用してはいけないカーネルは

* Unmaintained kernel release
* メインテナンスされていないカーネルリリース

To give numbers to the above, today, as of August 24, 2018, the front page of kernel.org looks like this:

上記の数についてですが2018年8月24日時点でkernel.orgのフロントページはこのようになっています。



So, based on the above list that would mean that:

したがって上記のリストは以下を意味します。

* 4.18.5 is the latest stable release
* 4.18.5 が最新の安定版リリース
* 4.14.67 is the latest LTS release
* 4.14.67 が最新LTSリリース
* 4.9.124, 4.4.152, and 3.16.57 are the older LTS releases that are still being maintained
* 4.9.124, 4.4.152, と 3.16.57 がまだメインテナンスされているより古いLTSリリース
* 4.17.19 and 3.18.119 are “End of Life” kernels that have had a release in the past 60 days, and as such stick around on the kernel.org site for those who still might want to use them.
* 4.17.19 と 3.18.119 が それらをまだ使用したい人たちのためにkernel.orgに居残っている、過去60日内にリリースがあった “寿命の終わり”のカーネル。

Quite easy, right?

非常に簡単ですよね？

Ok, now for some justification for all of this:

では、今からこれすべての根拠を示します。

**Distribution kernels**

**ディストリビューションのカーネル**

The best solution for almost all Linux users is to just use the kernel from your favorite Linux distribution. Personally, I prefer the community based Linux distributions that constantly roll along with the latest updated kernel and it is supported by that developer community. Distributions in this category are Fedora, openSUSE, Arch, Gentoo, CoreOS, and others.

殆どすべてのLinuxユーザーへの一番いいソリューションはお好きなLinuxディストリビューションのカーネルを使用することです。個人的には私はコミュニティーをベースにし、常に最新のアップデートされたカーネルと一緒に進み続けていて、コミュニティーにサポートされたディストリビューションを好みます。このカテゴリのディストリビューションはFedora, openSUSE, Arch, Gentoo, CoreOS, やその他です。

All of these distributions use the latest stable upstream kernel release and make sure that any needed bugfixes are applied on a regular basis. That is the one of the most solid and best kernel that you can use when it comes to having the latest fixes ([remember all fixes are security fixes](http://kroah.com/log/blog/2018/02/05/linux-kernel-release-model/)) in it.

これら全てのディストリビューションは最新の安定した上流のカーネルリリースを用い定期的に必要なバグ修正が適用されているかを確認しています。これは最新の修正([remember all fixes are security fixes](http://kroah.com/log/blog/2018/02/05/linux-kernel-release-model/))が適用されればあなたの使用可能な最も堅牢で最善のカーネルです。

There are some community distributions that take a bit longer to move to a new kernel release, but eventually get there and support the kernel they currently have quite well. Those are also great to use, and examples of these are Debian and Ubuntu.

新しいカーネルリリースに移行するのが少し長くかかりますが、最終的には移行し、現在のカーネルを非常によくサポートするいくつかのコミュニティーディストリビューションがあります。

Just because I did not list your favorite distro here does not mean its kernel is not good. Look on the web site for the distro and make sure that the kernel package is constantly updated with the latest security patches, and all should be well.

あなたが好きなディストリビューションをここに挙げていないということが、そのカーネルがよくないわけではありません。ディストリビューションのWebサイトを見て、カーネルパッケージが常に最新のセキュリティパッチで更新されていることを確認してください。

Lots of people seem to like the old, “traditional” model of a distribution and use RHEL, SLES, CentOS or the “LTS” Ubuntu release. Those distros pick a specific kernel version and then camp out on it for years, if not decades. They do loads of work backporting the latest bugfixes and sometimes new features to these kernels, all in a Quixote quest to keep the version number from never being changed, despite having many thousands of changes on top of that older kernel version. This work is a truly thankless job, and the developers assigned to these tasks do some wonderful work in order to achieve these goals. If you like never seeing your kernel version number change, then use these distributions. They usually cost some money to use, but the support you get from these companies is worth it when something goes wrong.

たくさんの人が古い「伝統的な」流通モデルを気に入っており、RHEL、SLES、CentOS、または「LTS」Ubuntuのリリースを使用しているようです。これらのディストリビューターは、特定のカーネルバージョンを選び、数十年ではないにしても、それを数年間使用します

So again, the best kernel you can use is one that someone else supports, and you can turn to for help. Use that support, usually you are already paying for it (for the enterprise distributions), and those companies know what they are doing.

But, if you do not want to trust someone else to manage your kernel for you, or you have hardware that a distribution does not support, then you want to run the Latest stable release:

**Latest stable release**

This kernel is the latest one from the Linux kernel developer community that they declare as “stable”. About every three months, the community releases a new stable kernel that contains all of the newest hardware support, the latest performance improvements, as well as the latest bugfixes for all parts of the kernel. Over the next 3 months, bugfixes that go into the next kernel release to be made are backported into this stable release, so that any users of this kernel are sure to get them as soon as possible.

This is usually the kernel that most community distributions use as well, so you can be sure it is tested and has a large audience of users. Also, the kernel community (all 4000+ developers) are willing to help support users of this release, as it is the latest one that they made.

After 3 months, a new kernel is released and you should move to it to ensure that you stay up to date, as support for this kernel is usually dropped a few weeks after the newer release happens.

If you have new hardware that is purchased after the last LTS release came out, you almost are guaranteed to have to run this kernel in order to have it supported. So for desktops or new servers, this is usually the recommended kernel to be running.

**Latest LTS release**

If your hardware relies on a vendors out-of-tree patch in order to make it work properly (like almost all embedded devices these days), then the next best kernel to be using is the latest LTS release. That release gets all of the latest kernel fixes that goes into the stable releases where applicable, and lots of users test and use it.

Note, no new features and almost no new hardware support is ever added to these kernels, so if you need to use a new device, it is better to use the latest stable release, not this release.

Also this release is common for users that do not like to worry about “major” upgrades happening on them every 3 months. So they stick to this release and upgrade every year instead, which is a fine practice to follow.

The downsides of using this release is that you do not get the performance improvements that happen in newer kernels, except when you update to the next LTS kernel, potentially a year in the future. That could be significant for some workloads, so be very aware of this.

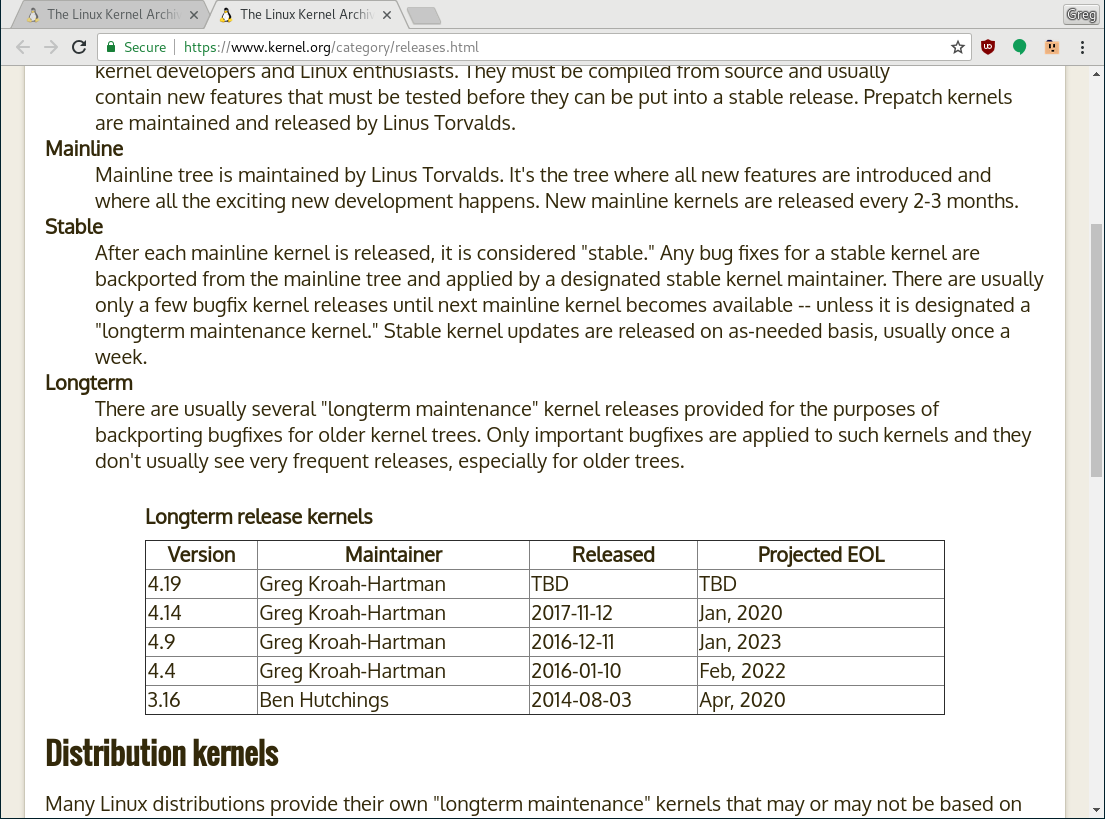
Also, if you have problems with this kernel release, the first thing that any developer whom you report the issue to is going to ask you to do is, “does the latest stable release have this problem?” So you will need to be aware that support might not be as easy to get as with the latest stable releases.

Now if you are stuck with a large patchset and can not update to a new LTS kernel once a year, perhaps you want the older LTS releases:

**Older LTS release**

These releases have traditionally been supported by the community for 2 years, sometimes longer for when a major distribution relies on this (like Debian or SLES). However in the past year, thanks to a lot of suport and investment in testing and infrastructure from Google, Linaro, Linaro member companies, [kernelci.org](https://kernelci.org/), and others, these kernels are starting to be supported for much longer.

Here’s the latest LTS releases and how long they will be supported for, as shown at [kernel.org/category/releases.html](https://www.kernel.org/category/releases.html) on August 24, 2018:



The reason that Google and other companies want to have these kernels live longer is due to the crazy (some will say broken) development model of almost all SoC chips these days. Those devices start their development lifecycle a few years before the chip is released, however that code is never merged upstream, resulting in a brand new chip being released based on a 2 year old kernel. These SoC trees usually have over 2 million lines added to them, making them something that I have started calling “Linux-like” kernels.

If the LTS releases stop happening after 2 years, then support from the community instantly stops, and no one ends up doing bugfixes for them. This results in millions of very insecure devices floating around in the world, not something that is good for any ecosystem.

Because of this dependency, these companies now require new devices to constantly update to the latest LTS releases as they happen for their specific release version (i.e. every 4.9.y release that happens). An example of this is the Android kernel requirements for new devices shipping for the “O” and now “P” releases specified the minimum kernel version allowed, and Android security releases might start to require those “.y” releases to happen more frequently on devices.

I will note that some manufacturers are already doing this today. Sony is one great example of this, updating to the latest 4.4.y release on many of their new phones for their quarterly security release. Another good example is the small company Essential which has been tracking the 4.4.y releases faster than anyone that I know of.

There is one huge caveat when using a kernel like this. The number of security fixes that get backported are not as great as with the latest LTS release, because the traditional model of the devices that use these older LTS kernels is a much more reduced user model. These kernels are not to be used in any type of “general computing” model where you have untrusted users or virtual machines, as the ability to do some of the recent Spectre-type fixes for older releases is greatly reduced, if present at all in some branches.

So again, only use older LTS releases in a device that you fully control, or lock down with a very strong security model (like Android enforces using SELinux and application isolation). Never use these releases on a server with untrusted users, programs, or virtual machines.

Also, support from the community for these older LTS releases is greatly reduced even from the normal LTS releases, if available at all. If you use these kernels, you really are on your own, and need to be able to support the kernel yourself, or rely on you SoC vendor to provide that support for you (note that almost none of them do provide that support, so beware…)

**Unmaintained kernel release**

Surprisingly, many companies do just grab a random kernel release, slap it into their product and proceed to ship it in hundreds of thousands of units without a second thought. One crazy example of this would be the Lego Mindstorm systems that shipped a random -rc release of a kernel in their device for some unknown reason. A -rc release is a development release that not even the Linux kernel developers feel is ready for everyone to use just yet, let alone millions of users.

You are of course free to do this if you want, but note that you really are on your own here. The community can not support you as no one is watching all kernel versions for specific issues, so you will have to rely on in-house support for everything that could go wrong. Which for some companies and systems, could be just fine, but be aware of the “hidden” cost this might cause if you do not plan for this up front.

**Summary**

So, here’s a short list of different types of devices, and what I would recommend for their kernels:

* Laptop / Desktop: Latest stable release
* Server: Latest stable release or latest LTS release
* Embedded device: Latest LTS release or older LTS release if the security model used is very strong and tight.

And as for me, what do I run on my machines? My laptops run the latest development kernel (i.e. Linus’s development tree) plus whatever kernel changes I am currently working on and my servers run the latest stable release. So despite being in charge of the LTS releases, I don’t run them myself, except in testing systems. I rely on the development and latest stable releases to ensure that my machines are running the fastest and most secure releases that we know how to create at this point in time.