Google outlines plans for mainline Linux kernel support in Android

reader comments

117 with 81 posters participating

Share this story

- Google’s eventual plan for Android, one kernel for every device, instead of the device-specific kernels that exist now.
- If we use a Galaxy S10 as an example here, the mainline Linux LTS kernel (blue) gets forked by Google for the Android Common Kernel, then Android Common gets forked by Qualcomm for a Snapdragon SoC, then Samsung forks the SoC Kernel for the specific S10 Device Kernel.
- Linaro got the mainline Kernel up and running on a Poco F1 and (with a few patches) Pixel 3.
- A closeup of the Linux/Android phone.

It seems like Google is working hard to update and upstream the Linux kernel that sits at the heart of every Android phone. The company was a big participant in this year’s Linux Plumbers Conference, a yearly meeting of the top Linux developers, and Google spent a lot of time talking about getting Android to work with a generic Linux kernel instead of the highly customized version it uses now. It even showed an Android phone running a mainline Linux kernel.

But first, some background on Android’s current kernel mess.Currently, three major forks happen in between the “mainline” Linux kernel and a shipping Android device (note

that “mainline” here has no relation to Google’s own “Project Mainline“). First, Google takes the LTS (Long Term Support) Linux kernel and turns it into the “Android Common kernel”—the Linux kernel with all the Android OS-specific patches applied. Android Common is shipped to the SoC vendor (usually Qualcomm) where it gets its first round of hardware-specific additions, first focusing on a particular model of SoC. This “SoC Kernel” then gets sent to a device manufacturer for even more hardware-specific code that supports every other piece of hardware, like the display, camera, speakers, usb ports, and any extra hardware. This is the “Device Kernel,” and it’s what actually ships on a device.

This is an extremely long journey that results in every device shipping millions of lines of out-of-tree kernel code. Every shipping device kernel is different and device specific—basically no device kernel from one phone will work on another phone. The mainline kernel version for a device is locked in at the beginning of an SoC’s initial development, so it’s typical for a brand-new device to ship with a Linux kernel that is two years old. Even Google’s latest and, uh, greatest device, the Pixel 4, shipped in October 2019 with Linux kernel 4.14, an LTS release from November 2017. It will be stuck on kernel 4.14 forever, too. Android devices do not get kernel updates, probably thanks to the incredible amount of work needed to produce just a single device kernel and the chain of companies that would need to cooperate to do it. Thanks to kernel updates never happening, this means every new release of Android usually has to support the last three years of LTS kernel releases (the minimum for Android 10 is 4.9, a 2016 release). Google’s commitments to support older versions of Android with security patches means the company is still supporting kernel 3.18, which is five years old now. Google’s band-aid solution for this so far has been to team up with the Linux community and support mainline Linux LTS releases for longer, and they’re now up to six years of support.

Last year, at Linux Plumbers Conference 2018, Google announced its initial investigation into bringing the Android kernel closer to mainline Linux. This year it shared a bit more detail on its progress so far, but it's definitely still a work in progress. "Today, we don't know what it takes to be added to the kernel to run on a [specific] Android device," Android Kernel Team lead Sandeep Patil told the group at LPC 2019. "We know what it takes to run *Android* but not necessarily on any given hardware. So our goal is to basically find all of that out, then upstream it and try to be as close to mainline as possible."

Google says that "compared to [Linux] LTS (4.14.0), the Android common kernel has 355 changes, 32,266 insertions, and 1,546 deletions (as of February 2018)." Progress is being made here every year, but keep in mind there are still two more forks after this.

While there is still work to be done for getting the wider Android ecosystem and myriad device configurations on board, for specific models of device, things seem to be progressing well. Tom Gall, the director of the Linaro Consumer Group, showed off a Xiaomi Poco F1 *actually running* the mainline Linux kernel under the usual Android OS. Gall told the audience, "There are major, major props to be given to the Google Kernel Team in particular for getting their code upstream so that we can boot devices with a mainline kernel."

Along with the plan to upstream as much kernel code as

possible, Google is also aiming to replicate its earlier work at modularizing Android, Project Treble. Just as Treble created a stable interface between the Android OS and the device HALs (Hardware Abstraction Layers), Google's proposal for bringing Android closer to mainline Linux (How is there not a silly "project" name for this yet?) involves stabilizing Linux's in-kernel ABI and having a stable interface for the Linux kernel and hardware vendors to write to. Google wants to decouple the Linux kernel from its hardware support.

The Linux community has been against the idea of a stable interface for some time, with the suggestion that if you want the ability to quickly update a kernel, open source your drivers and get them in the main kernel tree, where any changes will be taken care of for you. Open sourcing drivers is an absolute deal breaker for many hardware companies, though, and no amount of advocacy or product degradation is going to change that. The fact that Google is stuck supporting five-year-old versions of Linux shows this model isn't working.

Enlarge / Instead of a bunch of forks, Google imagines the new kernel as a series of modules.
Matthias Männich, a senior software engineer at Google, displayed a diagram labeled “Android Next Generation” that showed what this proposed kernel architecture would look like. Google would ship a “Generic Kernel Image (GKI)” along with several “GKI Modules” that would expose this stable API/ABI in the kernel. The hardware-specific drivers (probably closed source drivers) would be loaded as kernel modules. So instead of forks on top of forks, everything gets modularized into additive packages.

For now, Google is only proposing that the in-kernel ABI be stable for a single LTS version. So this wouldn’t allow devices to upgrade from one version of the Linux kernel to another—it would just allow for a single generalized kernel

image to work across multiple devices, instead of the device-specific kernel forks we have today. It would definitely allow for easier security updates, and hopefully it would get new LTS releases to market faster.

We would be premature if we said this is a change that is *definitely* coming to Android and the Linux kernel. There are still tons of problems to solve (both technically and politically, I would imagine), and some of these talks were more like public brainstorming sessions between the Google presenters and the Linux engineers (many of which were also Googlers!) in the audience. This is the second year in a row Google has showed up to LPC and pitched the idea of using the mainline Linux kernel in Android, though, and with Android being one of the biggest Linux sort-of-distros on Earth, it makes sense for everyone to work together, right?

“We have miles to go, and we know that, but we’ve come along from where we started,” Patil told the crowd. “Last year, I talked it into existence. This time, we actually have problems to discuss.”

Listing image by Eric Adeleye / Flickr

https://arstechnica.com/?p=1603607