



Courtesy of Linux Foundation

Interview with Linus Torvalds

In June of this year, kernel developer Greg Kroah-Hartman sat down with Linus Torvalds in front of a live audience at LinuxCon Japan to look at the first 20 years of Linux, the state of the kernel, and the future of Linux.

Linux 3.0

Greg Kroah-Hartman: *You announced that the version number is going to change. You said 3.0.*

Linus Torvalds: It's not out yet, but I did the RC just before I left for this trip. If everything goes well – and it looks fine so far – in about seven or eight weeks, we'll have the final 3.0 release just in time for the year's festivities. I'm actually really happy about the whole thing. I'm finally getting rid of 2.6!

I'm finally getting rid of 2.6!

I don't know how many of you know this, but with our old versioning, I added a third number – we're up to 39 right now – and then to make things even more interesting, Greg does something like 2.6.39.1 to number the stable releases. It just gets really messy. Then all the distributions have their own build version, so when you actually run Linux, you will run something like 2.6.39.7-13 [audience laughs]. We've been doing this for a long time, and it's been kind of meaningless.

We used to change the version number when we hit a big milestone. So, 1.0 was networking works; 1.2 was our first multiplatform, so that was when we supported alpha, m68k; 2.0 was SMP worked ... kind of ...; and 2.2 was SMP actually scaled to two or three CPUs. We've always had this

THE VIDEO

The original video was produced by The Linux Foundation at LinuxCon Japan. To see the entire video, go to http://www.youtube.com/watch?v=_fALdvvcM0.

notion that we need to have a feature to increment the version number – and then we changed how we do development.

We don't do versioning by features anymore; we just do this 8- to 10-week release cycle – and it's working wonderfully – but it's meant that 2.6 has stuck around for a long time. Now, 20 years of Linux means that I finally had the excuse to say, "OK, enough. We've done version numbering based on time, so let's change 2.6 based on time, too." Now it will be 3.0, and you [indicates Greg] can do 3.0.1 for your stable. We'll still have a lot of numbers, but the numbers will be smaller, and I don't think we'll ever hear 40 anymore. When we get to 3.20 or something, then I'll just say, "Hey! Let's increment to 4."

When I made my 3.0 release candidate, I created a diff to the previous version. I created the diff not against 39 but against 29 – because when you're talking about big numbers like that, they all look the same. I didn't notice I was starting to upload this diff against the wrong kernel version. I'm hoping now that we've renumbered, it will be easier to remember.

GKH: *Thank you. I deal with these numbers every day, and it drives me crazy.*

LT: I hope this will clarify things, although, right now, it's also causing a lot of discussion about – do we start doing new features now because we've changed the version? No, it's all the same, we just changed the numbers so they're easier to remember.

GKH: *People have said, "Can we remove things?" People want to get rid of micro-channel or ISA or IDE.*

LT: No, we're not getting rid of features. One of the things that has worked so well for the last couple of years is trying to be stable all of the time. We used to have these big jumps when we removed features and rewrote the kernel and did "big" development changes. It was necessary back in the 1.0 days, back in the 2.0 days. You had to rewrite the world. It's been so easy in the last

five to eight years, when we decided we'll be stable all the time. I think it's helped us as developers, but I think users have been much happier, too, when they don't need to worry too much about being on a new kernel version.

Growing Up

GKH: *In the 20 years, has there been any recent features you liked or found interesting?*

LT: I actually like the really boring features. We've had many performance improvements in the last few releases. The one I like particularly is the name lookup changes. It speeded up enormously on some of my loads that I run. We literally had a 40 percent performance improvement in one release, and that's basically unheard of. But there's no new feature. There's no new interface for users. There's nothing new going on, we're just doing it 40 percent faster. That, to me, is really exciting.

GKH: *Doing things faster is a nice change. Normally, the joke is that we're getting bloated.*

LT: Well, we still are pretty big. Thinking about the machine that I ran Linux on 20 years ago, we wouldn't fit on that machine anymore. The good news is that even a cell phone has 10 times the computing power of that machine these days. Nobody really cares that we do need more resources; we use those resources very efficiently. We have been growing, but I think most of the growth has been to do things that modern hardware needs to do, things that modern usage patterns need to do.

GKH: *I know a lot of kernel developers have objected to the cgroups interface, the control groups, and some people have argued that it didn't have initial users, but now we do.*

LT: The cgroups interfaces were mainly done for certain server setups. Not very many people really used them. A lot of people were unhappy because they complicated the memory management code, they made the scheduler more complex, they

had a lot of impact on very core infrastructure. Then, it turns out, we ended up finding great use for it outside of the original target audience. It's interesting how often it does end up happening. Like all the SMP code. Ten, 15 years ago this was the "big iron," the server feature, and obviously today, every single high-end cellphone ends up using SMP code.

That's one of the strengths, I think, of Linux because we have the same kernel across so many platforms. Nobody else does this. Apple has iOS for their low end and OS X for their high end. Microsoft has Windows CE for low end and "real" Windows for high end. Linux has never had that. I think it's one reason we are doing really well in the embedded space. We never had a cut-down, kind of castrated version for the low end. We always had full features because, as it turns out, all those high-end features eventually end up percolating down.

GKH: *In the low end, the embedded guys argue "nobody's going to care about power management but us, so let's do it in our own little tree." But [look at] Google's servers – people have real power needs and big iron.*

LT: It goes that way, too. Power management grew up instead of growing down.

Playing Nice

GKH: *That leads to [the idea that] people sometimes feel their features are just for them. The ARM community has been very, um – how do I put it nicely – insular, thinking that they're doing things only in their own sandbox, so it's not going to affect everybody else. For years, we've been trying to push back, and recently, you pushed back really hard. Do you want to talk about how that's working out?*

LT: I'm actually happy to talk about how it's working out. In 3.0rc1, one of the things I'm really happy about was the ARM people were starting to react. A couple of months ago, we had a big flame war, and I basically called them bad names, and people were not very happy, and some people thought it was just me being difficult – which sometimes happens.

The ARM people are realizing that they shouldn't think of only their small platform but try to make the code generic so they share code among themselves, which is a big step. Now, they're also trying to share it outside of ARM. We'll see how well that actually works. It at least means that now the people outside of ARM that are working on similar features see the ARM code, because it's not hidden somewhere in the deep confines of the ARM tree. There's a history of

people thinking of one particular platform and not thinking of themselves as being a part of a bigger ecosystem. ARM is growing up; along with that, they're starting to hit all the same issues that the other platforms hit a long time ago.

Linux Everywhere

GKH: *Speaking of ARM. Linux now is everywhere. It's in our small cellphones to ... everything. A long time ago, when you did the 1.0 release, you said "total world domination" was a goal, and we're doing pretty well. But one of the things you said then was for us to achieve that, we'd have to have the applications. The kernel was boring, and all the interesting stuff was going to happen in the applications. Do you think that's still true?*

LT: It's less true than it used to be. We have the applications to a large degree. I don't say "world domination" anymore. It was funny 15 years ago because it was so obviously a joke. For the last 10 years, it's not been so much a joke anymore, so it's no longer funny, so I stopped saying it.

I don't say 'world domination' anymore.

We're doing really well on the low end; we're doing really well on the high end, servers – pretty much every little niche. The desktop is where we actually have the applications now. It's a hard market to get into, and it's still the market that I started Linux for. I wanted it on my desktop. It's what I use every day. I have a Linux phone and I'm really happy about that, but it's still the laptop and the desktop that I really work with.

It just takes a long time to convince people to change what they're using, so they're still stuck on Windows and some are still stuck on OS X, and we'll get there some day.

GKH: *Can we do anything to help that in the kernel, or is it just working with people?*

LT: I've been trying to think what we can do in the kernel, and I don't know. We have worked very hard at making the kernel do as well as it can. We have worked hard on the whole interactive performance and user interface issues. There are still application issues, but I think it's really up to distributions to very aggressively target the desktop. I'm happy that some of them clearly are.

Past and Future

GKH: *So, 20 years has been a long time to work on one project.*

LT: Some people flitter from project to project, and I've always been somebody who likes concentrating on one thing. I'm not a multitasker. I've had a few small projects in between, but I'm really happy doing one thing and feeling like I'm doing a good job at that one thing. I didn't think I'd do it for 20 years, but now, if I didn't do it, I'd be bored.

GKH: *Do you see yourself doing it for another 20?*

LT: I'll be old by then [laughs]. I mean, I was really young when I started, and another 20 years ... at some point there will be somebody young and hungry and energetic who comes along and shows that he's really good at it. That's the point where I think I'd stop. I'm perfectly happy to say, "Hey, you're clearly doing a better job than I. Go. Take it."

GKH: *How do we keep Linux relevant to be able to hand it off in 20 years? How do we make sure we succeed?*

LT: I really don't think that's a problem. If you look at all the work we do today, a lot of it is hardware maintenance, and that doesn't seem to be going away. A lot of what the kernel does is day-to-day stuff people did in the '60s.

The whole Unix architecture is 40 years old, and I don't think it's any less relevant today than it was back then. I don't think 20 years will make a big difference, but we will have to update for new hardware and new usage cases. With new hardware comes new software and new places where that new hardware gets used. So, I think we'll be relevant in 20 years.

The one thing I don't want to be is in maintenance mode, where we don't live, where we don't make changes.

GKH: *Yesterday at the open forum discussion here, there was lots of talk about how open source software can help with disaster relief efforts. Is there any role do you think that Linux can play in that?*

People are using Linux for things I never envisioned.

LT: One of the things that personally is very gratifying to me is how people are using Linux for things I never envisioned. Not just the markets, but people using Linux and open source for reasons that were not my reasons – going into developing worlds and spreading knowledge of technology and making it a teaching tool. I find that to be really exciting and gratifying.

Disaster relief – I’m sure it works really well. I must admit, it’s not something I’ve been personally thinking about.

Questions from the Audience

Q: *What is your single most memorable moment?*

LT: That really is hard to answer. It’s been small ideas with lots of different people being involved. It’s plodding day-to-day work, and when you look at it a year or two or five years afterward, you say, “Wow, we really made a huge change,” but at no point was there the “Aha!” moment.

The one single moment for me, personally, was back 19 years ago, when it went literally from a personal project to being something where I no longer knew the people involved. Now, it was not my toy anymore. Now there were hundreds – at that time it was hundreds – of people using this project of mine that I never met. That was a big thing for me.

Other than that, there’s been lots of exciting developments, like when Oracle announced they’d port their database to Linux. Now we’re in the big league because if you are a Unix and you have Oracle running on top of you, you’re a real Unix. But for me, personally, it was when it went from this pet project that I taught to a few people that I knew to going to hundreds of people that I didn’t know. But I really want to stress that it really isn’t about the innovation. The reason I’m here talking about Linux is, I think, the persistence and hard work of the kind of person who sticks to a project and does it every day, 10 hours a day – there’s thousands of them now – and that was what’s brought Linux to where it is. It’s sweat and hard work.

Q: *How will applications, in moving from running on the OS on the desktop to running on the web in a browser affect the development of Linux?*

LT: This has already helped Linux enormously. I don’t know how it was in Japan, but both in Finland and then in the US, you had specialized applications, like for banking, that you had to run on Windows to talk to your bank. That’s all gone, obviously. Almost everything is a web application. That helps Linux because, all of the sudden, the differences between operating systems aren’t as important. When the differences aren’t as glaring, now the technology matters a lot more. Now the licensing and the price and just being available matters a lot more, and that really helps Linux in the long run.

That said, I don’t think the web applications will take over the world entirely. I think it’s

replacing a lot of local applications, but I think we’ll still have a lot of native applications, and I don’t think, for example, that the Unix background of Linux is ever going to be a problem or something we want to forget about. The Unix architecture has been very successful. One of our strengths is that we were able to build on top of good design made by smart people long ago.

Q: *How happy are you with Ubuntu? And if you’re not happy, what would you change?*

LT: [Sighs] This question should actually go to Greg, because I suspect the answer from Greg would be way more amusing because he feels much stronger about this than I do. I think Ubuntu is interesting because they’re taking a different approach from a lot of other Linux distributions. I don’t mind that. I think it’s been very helpful to have a distribution that takes a very different and maybe less technical approach and a more user-interface and user-centric approach. Ubuntu, I think, has been very successful, thanks to that. That’s good, and it’s showed the other distributions to some degree a piece that they were missing. At the same time, we’ve had some issues at the kernel front, where some kernel developers who I won’t name [pointing at Greg] felt that Ubuntu wasn’t pulling their weight and helping as much as they should.

GKH: I go around giving a lot of talks about who helps develop the kernel. At those talks, people ask how Canonical is helping out, and I say they’re very low on the list. A lot of people found that interesting. They didn’t realize that. My only response was that they weren’t contributing to the kernel. They’re still not a very large contributor to the kernel or most upstream source projects at all. Other people have verified that as well. My only objection was, I want to see the kernel community grow. In order to grow, we need to have more developers contributing. Canonical has some very, very good developers. They are contributing more now. Over the past couple of years they’re doing much better – a lot more patches.

Q: *What is the toughest technical problem that you have faced during the development of Linux so far?*

LT: I’ll give two answers to that. They have never been technical. Even when we make a wrong decision and take a wrong turn somewhere, eventually we figure out that was a bad decision and we can fix it. The two areas where we’ve had serious problems was documentation and help from hardware manufacturers. Some hardware manufacturers have not been supportive,

which has always irritated me immensely and has sometimes been a problem for users, too. It’s kind of been going away since Linux has been growing so much. A lot of the manufacturers are finding out that it really hurts them not to help us.

The toughest problems have never been technical.

The other big problem is that it’s hard to develop a big project with thousands of people and tens or hundreds of companies that are major contributors involved, and they have completely different ideas of where they want to go. So, there’s been many times during the 20 years where we’ve had big disagreements between developers. We’ve had people who were really unhappy with how development was done. We’ve had people who were really unhappy about their feature not getting used when somebody else’s feature was picked. If I lose sleep over something, it’s always about politics and people. I’m happy to say that I think we usually solve our problems, but we’ve had times when we’ve had really bad blood in the community.

Q: *Are you still happy with the license, or do you think it needs an upgrade, or do you regret choosing the GPL?*

LT: I’m very happy with the GPL. I started out with my own personal license that was one paragraph, and it basically said “you can charge no money for this, you have to give source code back,” and that was it. It was probably not a license that would ever stand up in court. The “no money can change hands” became a problem very early on. I looked around, and I thought the

I’m very happy with the GPL.

GPL version 2 was exactly what I was looking for. I think it’s a very fair license, a license that’s very successful, and I think it’s something that speaks to people at a very deep level – the whole fairness notion that I give you something, you give me something back. It’s worked very well. I personally don’t care for the GPL version 3. I think it extended it to, “I give you something, you give me the code back, and you promise not to do certain things with it.” That was never what I wanted to do. I’m very happy with the license. It’s clearly worked very well. Why change it? ■■■