

 **BECOME A PATRON**

<https://www.patreon.com/bePatron?u=10705728>

# LINUX JOURNAL (/)

[Linus Torvalds \(/tag/linus-torvalds\)](/tag/linus-torvalds/)

[Linux Journal \(/taxonomy/term/29\)](/taxonomy/term/29)

## 25 Years Later: Interview with Linus Torvalds

by [Robert Young \(/user/800409\)](/user/800409/) on April 2, 2019



*Linux Journal's* very first issue featured an interview between *LJ's* first Publisher, Robert Young (who went on to co-found Red Hat among other things), and Linus Torvalds (author of the Linux kernel). After 25 years, we thought it'd be interesting to get the two of them together again. You can read that first interview from 1994 [here](https://www.linuxjournal.com/article/2736) (<https://www.linuxjournal.com/article/2736>).

## Interview: Linus Torvalds and Robert Young

**Robert Young:** It is a great pleasure to have an excuse to reach out to you. How are you and your family? Your kids must be through college by now. Nancy and I and our three daughters are all doing well. Our eldest, Zoe, who was 11 when Marc and I started Red Hat, is expecting her second child—meaning I'm a grandparent.

**Linus Torvalds:** None of my kids are actually *done* with college yet, although Patricia (oldest) will graduate this May. And Celeste (youngest) is in her senior year of high school, so we'll be empty-nesters in about six months.

All three are doing fine, and I suspect/hope it will be a few years until the grandparent thing happens.

**Bob:** When I first interviewed you back in 1994, did you think that you'd be still maintaining this thing in 2019?

**Linus:** I think that by 1994 I had already become surprised that my latest project hadn't just been another "do something interesting until it does everything I needed, and then find something else to do" project. Sure, it was fairly early in the development, but it had already been something that I had spent a few years on by then, and had already become something with its own life.

So I guess what I'm trying to say is not that I necessarily expected to do it for another few decades, but that it had already passed the bump of becoming something fairly big in my life. I've never really had a long-term plan for Linux, and I have taken things one day at a time rather than worry about something five or ten years down the line.

**Bob:** There is a famous old quote about the danger of achieving your dreams—your running joke back in the day when asked about your future goals for Linux was "world domination". Now that you and the broader Open Source/Free Software community have achieved that, what's next?

**Linus:** Well, I stopped doing the "world domination" joke long ago, because it seemed to become less of a joke as time went on. But it always *was* a joke, and it wasn't why I (or any of the other developers) really did what we did anyway. It was always about just making better technology and having interesting challenges.

And none of that has really changed on a core level. All the details have changed—the hardware is very different, the problems we have are very different, and my role is very different. But the whole "make it better and have interesting challenges" is all the same.

For example, back in 1994, I was mostly a developer. Sure, I was the lead maintainer, but while I spent a lot of time merging patches, I was also mostly writing my own code. These days I seldom write much code, and the code I write is often pseudo-code or example patches that I send out in emails to the *real* developers. I'd hesitate to call myself a "manager", because I don't really do things like yearly reviews or budgets, etc. (thank God!), but I definitely am more of a technical lead person than an actual programmer, and that's been true for the last many years.

So the truly big-picture thing hasn't changed, but my role and all the details obviously look very very different from 1994.

**Bob:** Where will you and this code base be in another quarter century?

**Linus:** Well, I'll be 75 by then, and I doubt I'll be involved day to day. But considering that I've been doing this for almost 30 years, maybe I'd still be following the project.

And the good news is that we really do have a pretty solid developer base, and I'm not worried about "where will Linus be" kind of issues. Sure, people have been talking about how kernel developers are getting older for a long time now, but that's not really because we wouldn't be getting any new people, it's literally because we still have a lot of people around that have been around for a long time, and still enjoy doing it.

I used to think that some radical new and exciting OS would come around and supplant Linux some day (hey, back in 1994 I probably still thought that maybe Hurd would do it!), but it's not just that we've been doing this for a long time and are still doing very well, I've also come to realize that making a new operating system is just way harder than I ever thought. It really takes a lot of effort by a lot of people, and the strength of Linux—and open source in general, of course—is very much that you can build on top of the effort of all those other people.

So unless there is some absolutely enormous shift in the computing landscape, I think Linux will be doing quite well another quarter century from now. Not because of any particular detail of the code itself, but simply fundamentally, because of the development model and the problem space.

I may not be active at that point, and a lot of the code will have been updated and replaced, but I think the project will remain.

**Bob:** Have you and the kernel team been updating the kernel code to your satisfaction through the years? Is there any need or pressure to re-write any of the 25-year-old ever-expanding Linux code base? Perhaps in a more "modern" language than C?

**Linus:** We've gone through many many big rewrites of most of the subsystems over the years—not all at once, of course—and many pieces of code end up being things that nobody really wants to modify any more (most often because they are drivers for ancient hardware that very few people really use, but that we still support). But one of the advantages of a big unified source base for the whole kernel has been that when we need to make some big change, we *can* do so. There may be a few out-of-tree drivers, etc., around (both source and binary), but we've always had a policy that if they are out of tree, they don't matter for development. So we can make radical changes when necessary.

As to C, nothing better has come around. We've updated the kernel sources for new and improved features (the C language itself has changed during the years we've been doing this), and we've added various extensions on top of C for extra type-checking and runtime verification and hardening, etc., but on the whole, the language is recognizably the same except for small details.

And honestly, it doesn't look likely to change. The kind of languages people see under active development aren't for low-level system programming. They are to make it easier to create user applications with fancy UIs, etc. They explicitly don't want to do things a kernel needs, like low-level manual memory management.

I could imagine that we'd have some "framework" language for generating drivers or similar, and we internally actually have our own simplified "language" just for doing configuration, and we do use a few other languages for the build process, so it's not like C is the only language we use. But it's the bulk of it by far, and it's what the "kernel proper" is written in.

**Bob:** What's your hardware instrument of choice? Is there a Stradivarius of Linux (or any) laptops out there? Or tablet or phone?

**Linus:** My main development machine is a very generic PC workstation. It's a franken-machine with different parts cobbled together over the years. It's nothing particularly special, and it's actually been two years since I made any big changes to it, so it's not even anything bleeding-edge. My main requirement at home is actually that it be basically entirely silent. Outside a couple fans, there are no moving parts (so no spinning disks anywhere), and the fans are not even running most of the time.

On the road (which is happily not *that* often), my main requirement is a good screen and being lightweight. My target weight is 1kg (with charger), and honestly, I've not been able to hit that ideal target, but right now, the best compromise for me is the XPS13.

**Bob:** It seems Linux on the desktop's success was not on the PC desktop but on the device desktop via Android. What are your thoughts on this?

**Linus:** Well, the traditional PC is obviously no longer quite the dominant thing it used to be. Even when you have one (and even when it's still running Windows or OS X), lots of people mainly interact with it through a web browser and a couple random apps. Of course, then there are the "workstation" users, which is kind of the desktop I was personally always envisioning. And while still important, it doesn't seem to drive the market the way the PC did back when. Powerful desktop machines seem to be mostly about development or gaming, or media editing. The "casual" desktop seems to have become more of a browser thing, and quite often it's just a tablet or a phone.

Chrome seems to be doing fine in some of that area too, of course. But yes, in just numbers of people interacting daily with Linux, Android is obviously the huge bulk of it.

*[Note from Bob: In the strict sense of "dominant", this is probably fair. But despite the recent fall in total numbers of PCs shipped in the last couple years, the cumulative growth in the PC market between 1994 and, say, 2014 is such that even in a slow PC market today, the world is still installing four or five times as many PCs every year compared to 1994 (<https://www.marketwatch.com/story/one-chart-shows-how-mobile-has-crushed-pcs-2016-04-20>).]*

**Bob:** If you had to fix one thing about the networked world, what would it be?



*Linus Torvalds (Image Courtesy of Peter Adams, [The Faces of Open Source Project](http://www.facesofopensource.com/) (<http://www.facesofopensource.com/>))*

**Linus:** Nothing technical. But, I absolutely detest modern "social media"—Twitter, Facebook, Instagram. It's a disease. It seems to encourage bad behavior.

I think part of it is something that email shares too, and that I've said before: "On the internet, nobody can hear you being subtle". When you're not talking to somebody face to face, and you miss all the normal social cues, it's easy to miss humor and sarcasm, but it's also very easy to overlook the reaction of the recipient, so you get things like flame wars, etc., that might not happen as easily with face-to-face interaction.

But email still *works*. You still have to put in the effort to write it, and there's generally some actual content (technical or otherwise). The whole "liking" and "sharing" model is just garbage. There is no effort and no quality control. In fact, it's all geared to the *reverse* of quality control, with lowest common denominator targets, and click-bait, and things designed to generate an emotional response, often one of moral outrage.

Add in anonymity, and it's just disgusting. When you don't even put your real name on your garbage (or the garbage you share or like), it really doesn't help.

I'm actually one of those people who thinks that anonymity is overrated. Some people confuse privacy and anonymity and think they go hand in hand, and that protecting privacy means that you need to protect anonymity. I think that's wrong. Anonymity is important if you're a whistle-blower, but if you cannot prove your identity, your crazy rant on some social-media platform shouldn't be visible, and you shouldn't be able to share it or like it.

Oh well. Rant over. I'm not on any social media (I tried G+ for a while, because the people on it weren't the mindless usual stuff, but it obviously never went anywhere), but it still annoys me.

**Bob:** This issue of *Linux Journal* focuses on Kids and Linux. Is there any advice you'd like to give to young programmers/computer science students?

**Linus:** I'm actually the *worst* person to ask. I knew I was interested in math and computers since an early age, and I was largely self-taught until university. And everything I did was fairly self-driven. So I don't understand the problems people face when they say "what should I do?" It's not where I came from at all.

**Bob:** The very first time you and I met was at a Digital Equipment Company (DEC) tradeshow. It was on your very first trip to the US that Jon "maddog" Hall and DEC financed.

**Linus:** I think actually that was my second trip to the US. The first was, I believe, a trip for me to Provo, Utah, to talk with Novell about Linux (for a project inside Novell that was then to become Caldera).

But yes, the DECUS tradeshow (in New Orleans? Maybe I misremember) was certainly among my earliest trips to the US.

**Bob:** I asked how you were going to catch up with all the emails you missed by the time you returned to Helsinki. Your answer surprised me, and I've been quoting you ever since. You simply said you would send the backlog of emails to /dev/null. I expressed shock and asked you, "but what if there were important emails in your inbox?" You shrugged and replied, "If it was important, the writer would just send it again." Possibly the most liberating piece of advice anyone had ever given me. Do you still follow that philosophy of email handling?

**Linus:** It's still somewhat true, but at the same time, I've also changed my workflow a lot so that travel wouldn't be as disruptive to my work as it used to be. So these days I often strive to have people not even notice when I'm on the road all that much. I will give people a heads-up if I expect to be without much internet connectivity for more than a day or two (which still happens in some places of the world—particularly if you're a scuba diver), but most of the time, I can do my work from anywhere in the world. And I *try* (and sometimes fail) to time my trips so that they're not in the merge window for me, which is when I get the most pull requests.

So these days I keep all my email in the cloud, which makes it much easier to switch between machines, and it means that when I travel and use my laptop, it's not nearly as much of a pain as it used to be back in the days when I downloaded all my email to my local machine.

And it's not just about my email—the fact that almost all the kernel development ends up being distributed through git also means that it's much less of an issue what machine I am at, and synchronization is so much easier than it used to be back when I was working with patches coming in individually through email.

Still, my "if it's really important, people will re-send" belief stands. People know that I'm around pretty much 7/365, and if I don't react to a pull request in a couple days, it still means that it might have gotten lost in the chaos that is my email, and people send me a follow-up email to ping me about it.

But it's actually much less common than it used to be. Back in 1994, I wasn't all that overworked, and being gone a week wasn't a big deal, but it got progressively worse during the next few years, to the point where our old email-and-patches-based workflow really meant that I would sometimes have to skip patches because I didn't have the time for them, knowing that people would re-send.

Those times are all happily long gone. BitKeeper made a big difference for me, even if not all maintainers liked it (or used it). And now git means that I don't get thousands of patches by email any more, and my inbox doesn't look as bad as it used to be. So it's easier to stay on top of it.

By the way, perhaps even more important than the "If it was important the writer would just send it again" rule is another rule I've had for the longest time: if I don't *have* to reply, I don't. If I get a piece of email and my reaction is that somebody else could have handled it, I will just ignore it. Some busy email people have an automatic reply saying "sorry, I'll try to get to your email eventually". Me, I just ignore anything where I feel it doesn't absolutely concern me. I do that simply because I feel like I can't afford to encourage people to email me more.

So I get a lot of email, but I don't actually *answer* most of it at all. In a very real sense, much of my job is to be on top of things and know what's going on. So I see a lot of emails, but I don't usually write a lot.

**Bob:** At a talk at the Washington DC Linux user group meeting back in May 1995, that Don Becker organized, you stopped halfway through and asked the audience if anyone knew the score of the Finland-Sweden men's world championship hockey game. As the token Canadian in the room, I was able to assure you that Finland won that game. On that topic: Finland's recent win of the World Junior Championship must have been fun for you. Or were you cheering for the US?

**Linus:** Heh. Hockey may be the Finnish national sport (and playing against Sweden makes it more personal—I *speak* Swedish as my mother language, but I'm *Finnish* when it comes to nationality), but I'm not a huge sports fan. And moving to the US didn't mean that I picked up baseball and football, it just meant that ice hockey lost that "people around me cared" part too.

**Bob:** Many of us admire your willingness to call a spade a spade in public debates on Linux technology decisions. Others, um, dislike your forthright style of arguing. Do you think you are becoming more or less diplomatic as time has goes on?

**Linus:** If anything, I think I have become quieter. I wouldn't say "more diplomatic", but perhaps more self-aware, and I'm trying to be less forceful.

Part of it is that people read me a different way from how they used to. It used to be a more free-wheeling environment, and we were a group of geeks having fun and playing around. It's not quite the same environment any more. It's not as personal, for one thing—we have *thousands* of people involved with development now, and that's just counting actual people sending patches, not all the people working around it.

And part of the whole "read me in a different way" is that people take me seriously in a way they didn't do back in 1994. And that's absolutely *not* some kind of complaint about how I wasn't taken seriously back then—quite the reverse. It's more me grumbling that people take me much too seriously now, and I can't say silly stupid cr\*p any more.

So I'll still call out people (and particularly companies) for doing dumb things, but now I have to do it knowing that it's news, and me giving some company the finger will be remembered for a decade afterwards. Whether deserved or not, it might not be worth it.

**Bob:** Anything else you want to comment on, either publicly or otherwise?

**Linus:** I've never had some "message" that I wanted to spread, so ...

## About Robert Young and What He's Been Up to in the Past 25 Years

Graduating from the University of Toronto in 1976 after studying history, Young took a job selling typewriters. In 1978, he founded his first company and then spent 15 years in Canada at the helm of two computer-leasing companies. He sold the second of these to a larger firm who moved him to Connecticut in 1992 to grow their small US subsidiary. Shortly after, the new parent company ran into financial difficulties, otherwise known as bankruptcy, and Young found himself working out of his wife's sewing closet.



*Robert Young, LJ's First Publisher*

Although that event led directly to, in 1993, co-founding Red Hat (NYSE: RHT) with Marc Ewing, a young North Carolina-based software engineer. Both of them had fallen in love with free software, now known as open source—Ewing because he could innovate with software that came with source code and a license that allowed him to innovate, and Young because he could see how technology customers could be better served with open technology than the closed proprietary alternatives the industry offered at the time. Serving as CEO from

founding through Red Hat's IPO in 1999, he then moved to the role of Chairman, and the brilliant Matthew Szulik took over as CEO, building the early Red Hat into a great business. Red Hat is now a member of the S&P 500 Index of the largest US public companies.

In 2000, Young and Ewing co-founded the Center for Public Domain, a non-profit foundation created to bolster healthy conversation of intellectual property, patent and copyright law, and the management of the public domain for the common good. Grant recipients included the Electronic Frontier Foundation and the Creative Commons.

In 2003, Young purchased the Hamilton Tiger-Cats of the Canadian Football League, and he currently serves as the league's Vice-Chairman.

Working with a talented team led by Gart Davis, he helped launch Lulu.com in 2004 as the first online self-publishing services to use print-on-demand technology to enable a new generation of authors to bring their works directly to market, avoiding the delays, expense and limited profitability of publishing through traditional channels. Under the direction of Kathy Hensgen, Lulu continues to be a leading innovator helping authors bring their works to market.

In 2012 Young invested in PrecisionHawk, a small drone company led by Ernie Earon and Christopher Dean. PrecisionHawk, based in Raleigh, has become one of the leading drone technology companies in the US. He continues to serve as Chairman, with CEO Michael Chasen.

Since 2016, Young has been involved with Scott Mitchell and a team based in Toronto, helping organize the Canadian Premier League, a professional soccer league in Canada. He owns the Hamilton Forge franchise. The league will begin play this month (April 2019).

His favorite current project is helping his wife Nancy run Raleigh-based Elizabeth Bradley Design Ltd and its [Needlepoint.com](https://needlepoint.com) (<https://needlepoint.com>) store, a leading needlepoint supplier. Their mission is nothing less than to make the world a more beautiful place, by growing the community of enthusiastic needlepointers around the world.

His most beloved pastime is spending time with his growing family. He and his wife Nancy welcomed their first grandchild a year ago. Young also enjoys pursuing a bunch of hobbies, always badly. These include fly fishing, kite boarding, golf, and he collects the occasional antique typewriter—a nod to his beginnings as a typewriter salesman.

## Sidenote: the Faces of Open Source Project

*The photo of Linus (<http://www.peteradamsphoto.com/people/linus-torvalds>) in this article is by Peter Adams (<http://www.peteradamsphoto.com>), a photographer I met a few months ago when he introduced me to a series he started in 2014 called [Faces of Open Source](http://facesofopensource.com) (<http://facesofopensource.com>). On that site, Peter writes, "Despite its wide ranging impact, the open source revolution remains all but unknown to most people who now, more than ever before, depend on its survival. This project is an attempt to change that." His purpose applies not only to the muggles who rely on open source, but to the wizards who write their own code and put it to use. Knowing who created the Open Source world we have now will surely help as we code up a future that embodies the same good values.—Doc Searls*

Load 9 comments ([https://www.linuxjournal.com/content/25-years-later-interview-linus-torvalds#disqus\\_thread](https://www.linuxjournal.com/content/25-years-later-interview-linus-torvalds#disqus_thread))





(<https://www.embeddedarm.com/products/TS-4900>)

## Recent Articles



Linux Journal Ceases Publication: An Awkward Goodbye  
(/content/linux-journal-ceases-publication-awkward-goodbye)

*Kyle Rankin (/users/kyle-rankin)*

(/content/linux-journal-ceases-publication-awkward-goodbye)



Oops! Debugging Kernel Panics (/content/oops-debugging-kernel-panics-0)

*Petros Koutoupis (/users/petros-koutoupis)*

(/content/oops-debugging-kernel-panics-0)



Loadsharers: Funding the Load-Bearing Internet Person  
(/content/loadsharers-funding-load-bearing-internet-person)

*Eric S. Raymond (/users/eric-s-raymond)*

(/content/loadsharers-funding-load-bearing-internet-person)



Documenting Proper Git Usage (/content/documenting-proper-git-usage)

*Zack Brown (/users/zack-brown)*

(/content/documenting-proper-git-usage)



Understanding Python's asyncio (/content/understanding-pythons-asyncio)

*Reuven M. Lerner (/users/reuven-m-lerner)*

(/content/understanding-pythons-asyncio)



RV Offsite Backup Update (/content/rv-offsite-backup-update)

*Kyle Rankin (/users/kyle-rankin)*

(/content/rv-offsite-backup-update)

## Corporate Patron



Making IT Heroes

Pulseway

[. \(https://www.pulseway.com?rfid=linuxjournal\)](https://www.pulseway.com?rfid=linuxjournal)

## Linux Journal Week in Review

Sign up to get all the good stuff delivered to your inbox every week.

 Enter your email. Get the newsletter.

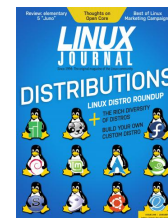
SIGN UP

☐ I give my consent to be emailed

## The Value of Open Source Journalism

Subscribe and support our coverage for technology's biggest thinkers – with up to 52% savings.

**Subscribe »** [. \(https://www.linuxjournal.com/subscribe\)](https://www.linuxjournal.com/subscribe)



## Connect With Us


[. \(https://youtube.com/linuxjournalonline\)](https://youtube.com/linuxjournalonline)

[. \(https://www.facebook.com/linuxjournal/\)](https://www.facebook.com/linuxjournal/)

[. \(https://twitter.com/linuxjournal\)](https://twitter.com/linuxjournal)

Linux Journal, currently celebrating its 25th year of publication, is the original magazine of the global Open Source community.

© 2019 Linux Journal, LLC. All rights reserved.

Powered by [PRIVACY POLICY \(/CONTENT/PRIVACY-STATEMENT\)](/CONTENT/PRIVACY-STATEMENT) | [TERMS OF SERVICE \(/TERMS-SERVICE\)](/TERMS-SERVICE) | [ADVERTISE \(/SPONSORS\)](/SPONSORS)

| [SUBSCRIBE \(/SUBSCRIBE\)](/SUBSCRIBE)

[MASTHEAD \(/CONTENT/MASTHEAD\)](/CONTENT/MASTHEAD)

[RSS FEEDS \(/RSS\\_FEEDS\)](/RSS_FEEDS)



privateinternetaccess®

[\(/SUBS/CUSTOMER\\_SERVICE\)](/SUBS/CUSTOMER_SERVICE)

[LETTERS TO EDITOR \(/CONTACT\)](/CONTACT)

[CONTACT US \(/ABOUTUS\)](/ABOUTUS)

[. \(https://www.privateinternetaccess.com/pages/buy-vpn/linuxjournal\)](https://www.privateinternetaccess.com/pages/buy-vpn/linuxjournal)