

## AN INTERVIEW WITH

# Kirk McKusick

by Allan Jude and Benedict Reuschling

The *FreeBSD Journal* Editorial Board suggested that some of the outstanding interviews from the BSD Now series might be of interest to readers as they reflect the state of technology when the interview took place. Here, we've transcribed and excerpted from a 2018 interview with Kirk McKusick by Allan Jude and Benedict Reuschling.



**Benedict:** In this episode of BSD Now, we're actually enjoying the holidays a little bit, so we have picked something from the shelf. We sat down at BSDCan 2018, way back when, to interview Kirk McKusick about various topics ranging about the early years of BSD, Unix, the continuing work of UFS that he is working on, the governance of FreeBSD, and more.

We're joined today by Kirk McKusick, who is one of the early committers to the FreeBSD Project and a principle of the BSD project at the University of California at Berkeley. So, welcome. What we want to know first is how you got involved with Unix and BSD.

• **Kirk:** *I was office mates with Bill Joy, and it was Bill Joy's project originally. It was impossible not to be involved because there were four desks and four people. There was one phone which was for*

everybody to use, and if the phone would ring, if I happened to be near it, I'd just pick it up and hand it to Bill because 99% of the time it was for Bill. So why even bother answering it if you were anyone else?

**Benedict:** During your early involvement with BSD, did you have any idea that it would become what it is today?

● **Kirk:** *Not a clue. I'm terrible about predicting the future anyway. But to me it just was this thing where Bill had some programs and you couldn't be around Bill without getting sort of sucked into doing things. And he'd say, "Oh, could you just do this one little thing?" For example, we had a Pascal compiler, and it compiled into byte codes and then the interpreter would interpret them, and that interpreter had been written in Assembly language.*

*And then we got a VAX and Bill said, "Could you just port this Pascal interpreter over to the VAX? There's the MOVE instruction on the PDP-11, and there's the MOVE instruction on the VAX. It will just be simple transformation." Ha-ha-ha. And three weeks later, something sort of popped out. But he just made everything sound like it was going to be so simple and so easy, and we would just do it.*

*The distribution was just Bill passing out tapes. He would go to conferences and talk up BSD, and people would get all excited and would get the latest BSD distribution. And code would be coming back, and it just sort of bubbled up and bubbled up and bubbled up, and at the time, it was like one big software project. Lots of people wrote programs and passed them around so that they would become what they became. No chance I would have guessed that.*

**Allan:** Yes.

**Benedict:** Maybe that motivated other people to contribute because they had no other chance of escaping that during his presentations?

● **Kirk:** *Yeah, the thing is that Bill would go out and he would talk with people: what are you doing, blah-blah-blah? The early EX editor started from the EM editor written by Professor George Coulouris at Queen Mary's College, London. And Bill's attitude was always, why come up with something good when you can use something that already exists and is better. So, he would find something else that someone else had done and then just embellish it to be whatever he wanted it to be.*

*And it was actually not Bill, but Keith Bostic who would say there was no end to the good that you can do in the world by giving credit to other people. And in particular, when we were trying to rewrite AT&T's programs, he would get other people to write CAT or LS or whatever it was, and contribute it to Berkeley and he put their name all over it, and 9 times out of 10 would rewrite 80% of what was in there.*

**Benedict:** [laughs] Okay.

**Allan:** But yeah, that's a big part of the BSD philosophy, that giving credit is the important thing.

● **Kirk:** *Right, and by doing so—as Keith put it at one point where AT&T was bickering over whether this was the original or not, he’d say, I always have someone else to blame. You need to go talk to so-and-so because they wrote that!*

**Allan:** There’s definitely a lot of history in the files.

● **Kirk:** *Correct.*

**Allan:** When you look at the number of people who have touched any given file, it’s just like wow!

● **Kirk:** *Yes, some of them are truly amazing. There’s a file that I thought I had written. When I went and looked, I think I found four or five lines that I had put in and which probably are part of the copyright message.*

**Benedict:** And it’s also the care and the maintenance that people have put into these programs and passed along to make it still work on your hardware or with your environments or your file formats.

**Allan:** Yes. Entire architectures like RISC-V are brand-new but are going to run all this old code with only a little bit of help.

● **Kirk:** *Yes, exactly. Care and feeding is such a huge amount of what keeps a system viable. Of course, it is what’s new, what’s exciting, that attracts interest.*

*Documentation is an area that is important but not exciting for most programmers. I remember now many years ago someone came up and said, “Oh, I’m interested in FreeBSD. And I’m just a person who writes documentation. I can’t write any code. Is there anything I can do?” And I said, “Well, as chance would have it, we have these Manual pages.”*

**Benedict:** We’ve been waiting for you.

● **Kirk:** *Right, I think documentation is a big part of what has made FreeBSD successful, and the thing that made the documentation successful is that we don’t have this whole social hierarchy.*

**Benedict:** Yeah.

● **Kirk:** *If you are a committer, you get just as much right to vote for core whether you’re committing to documentation, source, ports, system administration scripts, or whatever.*

**Allan:** Or even to be on the core team.

**Benedict:** Yeah.

**Allan:** It’s not only source committers who are eligible to stand in the election.

● **Kirk:** *Correct. People realize that it’s more than just code that’s important in the FreeBSD Project.*

**Benedict:** Yeah, and it’s also the tendency that people who got appreciation for the work they are doing are doing even more work. So, documenters become ports committers or source committers, and do both bits.

**Allan:** That’s how they sucked me in. I wrote some notes on the train ride here in 2013, and then in 2014, I was a documenter. In 2015, I was a source committer. And then in 2016, I was on the core team.

● **Kirk:** *Yeah, exactly!*

**Benedict:** And especially at this conference, we identified new people with like, hmmm, that guy could be a perfect new source committer or let's just give him a bit.

● **Kirk:** *One of the things that gives FreeBSD or BSD a leg up over the Linux world is that we do understand you've got to keep bringing in new blood. Linux is aging out on the people at the top of the tree, and it's going to be very interesting to see how that carries over.*

*When Linus Torvalds retires, who's going to step up to that? And how is that going to happen?*

**Benedict:** Yeah, and as the system grows, there's more code to maintain, and that's just too much for a single person or just a small group to do. And there are also people leaving of course. Life happens to every one of us, but as long as we bring in more people than are leaving, then the project is still healthy.

● **Kirk:** *There was a period in BSD where if I went on vacation for two weeks, things couldn't get fixed. And my goal was if I get hit by a bus, BSD needs to be able to keep going.*

**Benedict:** Yeah, the Project is still live.

● **Kirk:** *And I'm way past the point where I'm a critical cog.*

**Allan:** I guess that leads into our next question, which is, what do you think is the biggest accomplishment of the FreeBSD Project over the last 25 years?

● **Kirk:** *Well, that's my governance talk, right? I think that we have evolved a system that allows it to continue moving forward, bringing in new people, and letting those people rise. There was a study that was done quite a few years ago now, and it was by people in economics at MIT and at the University of Toulouse in southern France. And they just studied source code commit logs to see who's in charge.*

*And if you look at Linux, for example, it's been Linus Torvalds and his lieutenants, who are a fairly static set—throughout the lifetime of Linux. So, they've had one leadership. Most open-source projects go dark within about five years, where dark is defined as no commits for more than a year. Usually the person who started the project goes away. Occasionally they manage one leadership rollover but then it usually keels over. And only a couple of projects—Apache and FreeBSD and just one other—have had three or more successful changes in leadership. I think we're on our fifth, if you count BSD at Berkeley as one of them. So, four leadership changes in FreeBSD.*

**Allan:** Or can you count core teams in FreeBSD?

● **Kirk:** *Look at the people that make up the core team. The Jordan Hubbard era. There was a Robert Watson era. There's the current set of people. There have been people who have moved in and many of them are still sort of kicking around but aren't really leading at this point.*

**Benedict:** But the ability—the change of leadership is there—

● **Kirk:** *Right.*

**Benedict:** —and the ability to continue on after that successfully.

● **Kirk:** *The committer population has continued to evolve, and our average and median committer age is still under 40 after 25 years, though it has crept up. It started in the low 30s, and it's now high 30s. I think it's up to 40 now. If I actually chop off every committer who's over 60, I think we drop to around 38 as the average committer age. We just need to get rid of us old geezers. There's something like 12 of us.*

**Benedict:** I guess it's also a test of the durability of the Project. And I guess they lost a couple of people or the trust of some people that this is going to be a viable thing in the future? But, then once there, new people come and see that it's working and continue from that point on.

● **Kirk:** *Right. I mean one of the other things that I'm not sure I would call an accomplishment per se but the FreeBSD code base has a more studied advancement. The Linux Foundation is very proud of the fact that they've added a million lines of code to the Linux kernel in the space of a year or some short period like that.*

*And to me that's an unmitigated disaster. More is not better. Windows learned this when they grew to where they had 50 million lines of code, and they couldn't stabilize it. Even given their enormous resources, they just couldn't.*

**Allan:** Right, too many moving parts. There's too much bulk.

● **Kirk:** *Right, and fixing bugs—with that many lines of code, you just can't—you never get the bugs fixed to the point where it becomes stable. Linux, just the kernel, is 20 million lines. The FreeBSD kernel is under 2 million lines and the entire FreeBSD distribution today is 16 million lines. But that includes the kernel and commands and libraries in the base system.*

**Benedict:** Yes, so what is your biggest surprise about what the project has become 25 years later.

● **Kirk:** *That it's still here and wildly successful. I mean given that the half-life of open-source software projects is five years, we've made it through five lifetimes. And if you count the original BSD in the total, from 1978 to now, it is 40 years. I gave a talk at the Usenix FAST conference. They asked me to give the keynote, and the keynote was on the evolution of the Fast File System (UFS) over 30 years. I started by saying if you had told me 30 years ago that I was going to be giving a keynote about this software 30 years later, I would have wondered what version of acid you would be on.*

*Software has an even shorter half-life than projects have. So, three to five years is the lifetime of most software. To still have UFS running is quite unusual. Just as a lark, I took a disk image of a UFS filesystem from 1982—which I had just saved as an image of the very first one I ever built—and I could actually read and write that on the current UFS implementation. Though it did get a little bit cranky about some of the cylinder table mapping stuff. But it worked.*

**Benedict:** Yeah, there were some changes there.

● **Kirk:** *Another interesting question is whether the much more modern Zettabyte File System (ZFS) is going to replace UFS. And the answer is no, because they each solve different problems. When you've got a lot of resources and huge filesystems, it's a no-brainer that ZFS is what you're going to use. But with an embedded system, you're just not going to use ZFS. You don't have the resources to run it. You need UFS, which is lean and mean.*

**Allan:** Yeah, and we're heading into an era where we're going to have billions more of these tiny devices, and they're all going to need a filesystem.

● **Kirk:** *Yeah. And it's not going to be ZFS.*

**Allan:** Yeah, and I'm sure you didn't expect to ever anticipate billions of installs of UFS. Or a billion volumes.

● **Kirk:** *Right, no.*

**Allan:** I don't think anybody thought there'd be a billion computers.

**Benedict:** And you want to use something that's reliable and has been tested for so long and not just use a fresh development. Of course, they always happen but if you have something available, like UFS that's been driven and tested, it's a no-brainer to use that rather than just getting something off the ground.

**Allan:** Well, it's not like UFS is stagnant either, right? It's got support and works well on modern SSDs.

● **Kirk:** *UFS needs care and feeding. When I first started at Berkeley, I didn't understand that concept.*

**Benedict:** You have to go back, yeah.

● **Kirk:** *So, thankfully, Rick Macklin did a lot of the initial work on NFS, and he more or less has been caring for it and feeding it ever since. And David Greenman originally, and then later Alan Cox has taken the VM system to places where you never would have guessed in a million years. And even Jeff Roberson does a lot of the UFS stuff these days.*

*So, I can just sort of sit back and say, Oh! You ought to try doing this.*

**Allan:** Yes, yes.

**Benedict:** So here history repeats itself.

● **Kirk:** *Yes. Being the "grand old man" can be kind of fun. You can have all the ideas and not have to do any of the work.*

**Allan:** You get to watch the project happen and get all the joy.

**Benedict:** Yeah, enjoy the fruits of the labors.

**Allan:** Yes, less stress.

● **Kirk:** *Someone asked me when am I going to run for core? It's like ha-ha-ha. No, never! I did that for 10 years at Berkeley. I've served my time.*

**Allan:** Yes. In looking back, what is your favorite FreeBSD-related memory?

● **Kirk:** *Oooh! Favorite FreeBSD-related memory? I mean there are so many different things that I can pick. Seeing things like the SMP finally ship and—well, not just ship in FreeBSD 5, but finally getting made to work in FreeBSD 6. Most of my memories are conferences like BSDCan, which kind*

*of started out as an ad hoc thing, and I go to it and it's like, wow, I get to see all my friends.*

**Benedict:** Yes, breaks records.

● **Kirk:** *Also things like EuroBSD. It started as EUUG, and then later it bifurcated into several groups but ultimately became the BSD conferences. And the fact that they came up with a model where they could really move around was great. Asia started out moving around, but the problem was everyone had to start from scratch, and so you'd get conferences that didn't work because they didn't really know what they were doing.*

*A group of folks from around Europe created the NUUG Foundation that manages conference venue commitments, registration logistics, and money handling. That is what makes the conference moves between cities possible because the organizers need only focus on the logistics of their country and getting together the program.*

*When it's always in the same place, then you don't get the local draw. So for example, in Tokyo, I can really only teach my class about once every four or five years because you have to build up to having enough new attendees. Whereas in Europe, I can pretty reliably just teach it year after year because we're in a different country and there's a whole bunch of new local people. They want to take it. And that works.*

**Benedict:** I like to see the faces behind all the commit emails, the community-building, the act of meeting people.

**Allan:** Every time you get an email from them, you read it in their voice.

● **Kirk:** Yes.

**Allan:** And it makes a very big difference in the communication streaming because suddenly any ambiguity is slanted in a good way.

**Benedict:** Yes, you touched on it a little bit before with the SMP work, but do you have a favorite milestone for the Project?

● **Kirk:** *Milestones that were interesting to me were things like getting soft updates into the system, and it's some of the technical things that I've been involved with directly or indirectly that are most memorable to me. But that is mostly because I've been directly involved in working on them.*

**Allan:** I know 64-bit inode numbers—that went on for some time.

● **Kirk:** *Yes, but that one finally came about because I found the person who was the most obstreperous and basically collared them into being involved in it. Once they were involved in it, they stopped being obstreperous and they got it done. I guess part of the thing that's sort of so interesting is it gets back to the interpersonal situations, where you get to know someone and figure out how they can be most effective.*

**Allan:** Yes, or how you can make the message make sense from their point of view.

● **Kirk:** *Or sometimes it's convincing other people that their point of view is correct. They may not be the best at getting that view out there. And one of the drawbacks about being the grand old man is that whatever I*

*say must be correct, even if it's complete bullshit. So, I try not to do stuff that's complete bullshit, obviously, but I will often find people who haven't been in the community long enough and are having trouble getting their ideas out, and just give it a push.*

**Allan:** Lend a bit of weight to the argument?

● **Kirk:** *Right. And when I say you should really listen to what Allan is saying, there are some good ideas here, doing it like this is the right way to go, people go, Oh, yeah, I guess I hadn't really thought about that. And then they come and engage you, or whoever, and once engaged, you're very good at expressing why your idea is good and then it gets launched.*

*So, to me a lot of the time when I'm just walking around these conferences and I'm talking to people, I hear good ideas, and I make mental notes that I need to engage with them and perhaps get them a little more connected into the community. Especially people who are not committers.*

**Allan:** Yes.

● **Kirk:** *There's one person who sent me something about a bug report that was six years old, and no one had really paid any attention to it, and I just went into the source code control and saw who was involved in it. The reporter of the bug had included a patch that worked, and he kept updating it for the new systems. I said, just let it go in. And it did, it went in.*

*So, it's little things like that where one email from me makes something good happen. I like to take advantage of that. Indeed, that's mostly what I'm contributing these days, more than code or anything else.*

**Allan:** Right.

**Benedict:** We touched on the Project itself and the community and also the core team that's been elected by the community. But there's also the FreeBSD Foundation that you're involved with. Is that also a contributing factor to the health of the Project in certain ways?

● **Kirk:** *Undoubtedly it is. Look at the history of how the FreeBSD Project starts. The first distributions were done by Walnut Creek CD-ROM. And they paid the initial developers so they could do it as a full-time job, and that was a huge step up, but that wasn't a sustaining model because we needed a lot more infrastructure and network connectivity.*

*And again, sort of fortuitously, we fell into Yahoo. They were using it for everything, and they had all the machines and so on. And, they provided our infrastructure, and that went on for—I don't know—more than a decade.*

**Allan:** I think it's still going on.

● **Kirk:** *It's still going on to some extent.*

**Benedict:** Sure.

● **Kirk:** *But the problem is, what happens if Yahoo one day decides to switch to Linux—which they're doing.*

**Benedict:** Or the box, or whatever.



● **Kirk:** *Yes, ... whatever. And suddenly our infrastructure disappears from underneath us. And so, Justin Gibbs saw this as an issue. He realized that the FreeBSD Project needed to have some way of funding at least its basic infrastructure. So he started the FreeBSD Foundation, and it took at least five or eight years before there was enough income to contemplate even basic stuff like infrastructure, and in fact we have only converted over to that in the last four or five years. I mean it's been a slow process. But, if Yahoo went away today, we'd be fine.*

*There are many ways that projects can get infrastructure through organizations like GitHub or the Apache Foundation. The FreeBSD Project set up our own foundation. By having our own foundation and not being under Apache, for example, we have a lot more control over how things work, though that can be both good and bad. But overall, the Foundation is an important keystone.*

*I think the biggest fear of the Project is that somehow the Foundation is going to try and take over. To date, the Foundation has done its best to position itself so that it is driven by the Project goals rather than trying to drive the Project. But the Foundation is important. It's very important.*

**Allan:** Yes, and at the same time, the Foundation and core are usually the people who are able to contribute the most. Both of them are drawing from the same feedstock so it's very hard not for them to have some overlap.

**Benedict:** Right. Or to have enough trust from the community that they say, Okay, that person should be my voice, my representative.

● **Kirk:** *And so, I like to think that Linux is the dinosaur and FreeBSD is the mammal. The meteor is going to hit and then the mammals will take over. But I'm not even sure I really want that to happen because if we get to be the dominant one, then we're going to fall off.*

**Benedict:** Yeah, so always a little bit under the radar but swimming in the big pond with the other folks.

**Allan:** And you know, the rumors of FreeBSD's demise have been exaggerated. And part of that is just how FreeBSD gets used and that people don't know about it or notice it's there and even to the point where the FreeBSD Project isn't always aware of just how much FreeBSD is being used. Or how it's been used.

● **Kirk:** *Yeah, we get some unexpected contributions to the Foundation from companies we had no idea were using FreeBSD. It's like, why did they contribute? So of course, the first thing we do is say to the contributor that we want to put you on our list of users of FreeBSD. We've had a couple of them say, Oh, no, no, no. Absolutely not. We consider the fact that we use FreeBSD as a proprietary secret. They don't want to promote that.*

**Allan:** Yes.

● **Kirk:** *Or there are others that are using it even though they promote themselves as being a Linux shop. And they don't want to NOT be perceived as a Linux shop. But nevertheless, that's what happens.*

**Allan:** So, where do you see FreeBSD going in the future?

● **Kirk:** *Well, that kind of gets back to this other thing. I think it has a very*

*strong base. I think we can't just take that for granted. Companies fail or go into obscurity. Yahoo was our absolute poster child for a long time, and now not so much. And that's in part because they are not a complete FreeBSD shop anymore, and also because their star is waning in the business world.*

*So, we need to keep going out there and evangelizing and bringing in new companies and new people. But I think we're pretty successful at that. I think that we really need to get into more universities, which we're trying to do with some success. I've actually been very interested in Philip Paeps's efforts. He travels to all these conferences in Africa and Asia and other places where you might not normally think of going. There's a lot of FreeBSD that's being used out there. Philip is evangelizing FreeBSD far and wide.*

**Benedict:** That's interesting.

**Allan:** It reminds me of the mid to late 1990s, almost every ISP was based on FreeBSD because it was the easiest and cheapest way to have a reliable dial-up server or whatever. And we're seeing some of these developing countries where FreeBSD is the answer again. And we just have to do the right things to capitalize on that.

**Allan:** Because places where Linux hasn't gotten into the market yet, if we can just get there first, it means that there's not enough of an advantage to change.

**Benedict:** And it's also looking at it from an operating system's perspective, being UNIX and that it evolved over these decades and it's also adaptable to different systems, different environments to different architectures.

**Allan:** Yes, or the fact that the concept of containers was invented on FreeBSD, even though it gained its success somewhere else. And some of that is—sometimes we have this bad habit of getting 90% of the way to the right thing and then miss that last little bit, to productize it.

● **Kirk:** *Yes, well, you know what they say, it's the pioneers that get the arrows in the back and the settlers that get the land.*

**Allan:** Yes.

● **Kirk:** *Unfortunately, we've been pioneers way too much.*

**Allan:** Yes.

● **Kirk:** *But I think some things are important if we're going to be successful in the future. We've got to get things like a container solution and running in the cloud. The fact that we're on EC2—first-class on EC2 and Azure and other things like that—is definitely important.*

*And mostly, tooting our own horn. Android, which everyone talks about—oh, you know Android's all Linux. Well, yes, it is a Linux operating system but nearly the entire user level on top of that is BSD because the GNU license doesn't work for user-level programs. The GPL by design forces applications that include its code to release their source code. We're in a lot of places we are not well-known for being in. I think we need to make a little more of a push to let the people know that.*

**Allan:** Yes, but I mean you're right that one of the areas we need to push harder

on is the universities and so on, so that we get to that next generation of people.

● **Kirk:** *Right, well, one of the beauties of young people is they like to do something different than their parents did. At this point, their parents are Linux.*

**Benedict:** Yes! And it's also giving them the right engineering skills and teaching them how to do proper debugging or any kind of skills that are helpful and not just in this one area but that they can apply to many others.

**Allan:** Well, and that's why I think the Teach BSD Project for university-level things, where instead of learning about the concept of computer science in a toy operating system environment, using a real operating system you can run on your real computer to do real work is important.

● **Kirk:** *Yes, the toy operating system has kind of faded out. But mostly what they're learning is Linux. But Linux is now so huge and complex that it's really hard to wrap your brain around it, whereas BSD is still at a size where you can still understand it.*

**Allan:** And at the same time, it's that much more observable. We have the right tools to actually be able to watch the parts inside while they're moving.

● **Kirk:** Yes.

**Allan:** Some other machines are big black boxes over in the corner that make a lot of noise, but you can't tell what's happening.

● **Kirk:** *Yes, things like DTrace are much more advanced in FreeBSD than they are in, for example, Linux. I saw the DWatch talk today, which is a very impressive wrapper around DTrace that brings it to the masses. You don't have to even understand most of what DTrace is doing.*

**Allan:** The hard question has always been coming up with the question to ask, not just finding the answer. We have lots of tools to find the answer. But it is you that has to decide what the question is.

● **Kirk:** *Right, but the problem is that DTrace is kind of the assembly language for answering questions. All right, I've got this question. But now I have to write this 100-line program to answer it. Whereas something like DWatch is like, well, I've got this question and 10 lines of stuff and I can get a graphic answer.*

**Benedict:** And that also appeals to more younger folks who want something more interesting, like colors and things moving around.

● **Kirk:** *So, I'm very excited to see stuff like that because I think that's—again, it's going to draw in a broader group of people.*

**Benedict:** Good.

**Allan:** Yes.

**Benedict:** Well, then, thank you for having this interview with us and hopefully, see you again in the next 25 years. Or hopefully sooner than that!

● **Kirk:** *All right, yeah, well, five minutes, huh, no problem.*