



new faces

of FreeBSD BY DRU LAVIGNE

This column aims to shine a spotlight on contributors who recently received their commit bit and to introduce them to the FreeBSD community. This past quarter was busy for new committers. In this issue's column, the spotlight is on **Fedor Uporov**, who received a src bit and **Luca Pizzamiglio**, who received a ports bit in August; **Adriaan de Groot** and **Craig Leres**, who each received a ports bit in September; **Ilya Bakulin** and **Chuck Tuffli**, who each received a src bit in September; and **Yuri Victorovich**, who received a ports bit in October.

Tell us a bit about yourself, your background, and your interests.



- **Fedor:** I received a master's degree in radio engineering nearly six years ago, and then started my career as a hardware designer. I had worked with different microcontrollers and FPGA, and then started to make my first efforts to program these, not as regular job tasks, but just as a hobby. Later, I found that my

professional interests shifted from hardware development and PCB design to programming. And then I changed my job and became a programmer.

I have really simple interests. When I have free time and when I am not tired from the monitor and keyboard, I try to find time to work with open-source code or to learn something new. Otherwise, I prefer walking or travelling.



- **Luca:** I'm 39 years old, and I was born in Casalpuusterlengo, a small city in Italy. I graduated from the Politecnico di Milano in computer science engineering in 2003. I was lucky enough to receive my first 8086 when I was 10, and since then I've been interested in computers.

I worked as consultant for some years in Italy, working with FPGA and Linux. In 2009, I moved to Berlin, Germany, where I started to work on FreeBSD and learned to speak something close to German. Now I live in Dusseldorf, still working on FreeBSD at Trivago.

When not working with computers, I play volleyball and beach-volley, and have started a fencing class, which is really a lot of fun. I also play piano, and I sang for some years in a jazz choir.



- **Adriaan:** I grew up in Calgary, Canada, and did silly things on an Apple until my high school got some PC juniors. Then I spent a short time at the U of C, where I met men with long grey beards, wearing hiking boots, seated at vt100 terminals. There was some kind of UNIX running on the other end, and like a doofus, I wrote my own build system (make used tabs, which I disagreed with) and then my own editor (vi had command-mode, which I disagreed with). Shortly after that, I moved to the Netherlands to escape the consequences of that hubris, and ended up with an email address in 1990 and "adridg" as login—which has been my username in the 25 years since.

The Netherlands continued to be my home, where I received a PhD, had some kids, and enjoyed a varied career in the software industry—currently doing free software development full-time. If you give me an hour's notice, you can stop by for dinner: there's a fair chance of dinner being vegetarian, and I'll gladly cook for groups up to 40 people. I can play electric bass (badly), Kerbal Space program (badly), and a lawyer on TV (ditto). The lawyering thing is connected to my interest in licensing issues; for a while I carried a card that said, "I talk to lawyers so you don't have to." That's the card I showed to developers; the other side said, "I talk to engineers so you don't have to," for the other end of the conversation.



- **Craig:** When I was a little kid my father taught me how to solder, and over the years I built a lot of little electronic projects. When I was about 11, my father's wealthy friend bought a PDP-8/e, had the

stock exchange wired into our house, and father wrote some code in assembly to watch and report on a couple stocks. Once his friend got bored with the project, I was able to use the PDP anytime I wanted. This is when I learned to program, and started spending more of my time playing with computers than with electronics.

My first Unix account was on the Cory EECS 11/70 at UC Berkeley, while I was a student. Later, I became an unpaid volunteer with the Computer Systems Research Group (CSRG) at Berkeley and had root and source commit privileges for Berkeley Software Distribution (BSD). I was one of the maintainers of the termcap file and my name appears in this file on Unix and Linux systems that still ship with it. I coauthored `/usr/bin/write` when BSD was working on rewriting programs that were encumbered by the System V source license.

At that time, I was also working at the Lawrence Berkeley National Laboratory in the Network Research Group (NRG). Another group had a Vax 11/780 that ran BSD. It was an ARPANET node, and when I first got my account, it used the Network Control Program (NCP) protocol. The admins would sometimes boot the experimental TCP kernel after hours and eventually transitioned to it full-time. With guidance from my boss (Van Jacobson), I was introduced to Unix network drivers, and we added a watchdog to the token ring driver used to talk to a couple of VMS systems in the building. I was also involved with another fun project, where we installed a custom-built microwave link between LBL and UCB. It provided 230.4 Kb/s site-to-site communication.

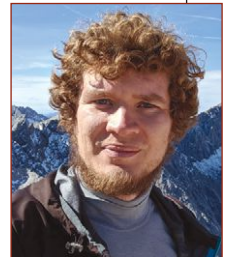
Our own group had a Vax 11/780 that ran VAX/VMS. But we also had an early version of software called Eunice that provided an environment that would run BSD network device drivers on top of VMS. We actually tested a lot of the early NRG TCP enhancements on this platform. It was common for me to port the current BSD networking stack to Eunice while Van was out of town giving a paper somewhere.

In the mid-1980s, we started getting Sun workstations, and soon I had 3/50s on my desktops at

work and home. We built our kernels from source (which was based on BSD at the time) and started using SunOS 3 to do network stack development. One memorable change was when Van noticed that mbuf clusters were 1K long, but Ethernet frames were ~1500 bytes. By doubling the size of a cluster (and making a few other tweaks), he was able to get 3/50s to transfer data at close to 10 Mb/s over ThickNet Ethernet. His testing would frequently crash our Vax/VMS system; while Ethernet uses exponential back-off, DECnet used linear back-off and VMS did not limit how much memory could be allocated to DECnet transmit retries and would panic when network utilization was unusually high.

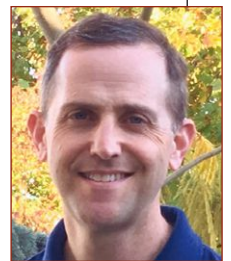
My interests include NASCAR (I'm a part-time stock car racing school instructor), electronics (digital audio and especially anything Arduino based), science fiction, and, of course, programming.

- **Ilya:** I graduated from the Moscow Engineering Physics Institute in 2009, with the intention to work in the micro- and nanoelectronics industry, but quickly changed my direction to software development. I have since been working at different companies, first in Moscow, and then in Munich, Germany, all of them employing BSD systems to build their products. Currently I work as a site reliability engineer at Google in Munich, and hack on the FreeBSD kernel in my free time.



When I'm not behind my laptop's screen, I usually go to the mountains and spend time there hiking or skiing, either alone or with my wife and kids.

- **Chuck:** My name is Chuck Tuffli, and I currently live outside of Sacramento, California. I've been using computers for "a while" (post punch cards, but before floppy drives were popular), and had the fortune to stumble into the world of "software that talks to hardware." The majority of my recent experience is with storage-related hardware (Fibre Channel, SAS, NVMe), but my past is checked with various video, graphics, and WiFi drivers. When not programming, I spend time with my family and our horses. Somewhat related, if you ever have the burning need to fix fences or irrigation, just let me know. I can hook you up.



new faces of FreeBSD



• **Yuri:** I have an MS degree in physics. At the university, I specialized in theoretical physics, computed the anomalous magnetic moment of elementary particles, and did a lot of mind-boggling math. After the university, science remained my passion, but my interests also included computers and technology.

I have worked as an engineer and later as an architect in a variety of industries: insurance, gaming, consumer electronics, optics, and electronic design automation (EDA), which develops software to design microchips. I've always gravitated toward algorithmically complex and science-related fields, since they tend to be the most challenging. I am also very interested in chemistry and biochemistry.

How did you first learn about FreeBSD and what about FreeBSD interested you?

• **Fedor:** Nearly three years ago, I started working with the Mac OS X darwin kernel, mostly VFS and filesystems. I started to learn FreeBSD, because darwin VFS was branched from BSD some time ago. And it turned out that the BSD side is more readable and understandable than the darwin kernel. So, FreeBSD really helped me to manage OSX VFS in terms of my regular job tasks.

• **Luca:** The first time I heard about FreeBSD, I was at the university and my friends were completely fascinated by the user 'toor'. At that time, I was a Linux guy, Slackware was my distro. When I moved to Berlin, I had the pleasure of working with my first mentor, vwe@, unfortunately only for a short time. What I really like in FreeBSD is how smooth it is building every piece of software that is running on your machine. It makes it really easy to hack your system and to contribute by submitting patches. I'm still building everything that runs on my laptop!

My first work on FreeBSD was GELI and different legacy ways to provide passphrases. Currently, I work mainly on ports, but I've started to play with jails and ZFS. Moreover, my interest in DTrace is growing every day, and it's incredible what you can do with it.

• **Adriaan:** There's a FreeBSD 2.2.3 boxed set on my bookshelf. I probably ran that on my home PC, which had two vt100s connected. At the same time, I was developing little networked games on X11 on SunOS, and next to the 2.2.3 boxed set is a Red Hat Linux 4 manual, so I was probably just experiment-

ing with whatever I could get my hands on.

Around 1998, I got into international free software development, contributing patches to KDE and becoming maintainer of KPilot, the Palm Pilot synchronization tool. Somehow, I ended up chatting with Lauri Watts, the documentation coordinator for KDE, and she ran FreeBSD and invited me over to the dark side. That must have been around 2001, and so I ran FreeBSD at home for a long time, and Linux at work, where I needed a (binary only) CMU LISP system to run a proof checker. After a few years, the KDE-FreeBSD community weakened and fell asleep—IRC nicks like tap, wca, lioux, and lofi fell silent on the KDE-FreeBSD channel.

I spent a few years working with OpenSolaris; by then I had a pretty clear idea of one thing I could do well: apply interesting tools to source code and find code quality issues—and then fix them. OpenSolaris had the Sun Studio compiler, which was free for a while. That had the primary incentive that it wasn't gcc, and, therefore, had a different view of what good C++ is. Applying that compiler to a huge codebase turned up tons of subtle gcc-isms, and plenty of Linux-and-BSD-isms that didn't apply to Solaris.

Somewhere in this history, SATA appeared on the horizon, and I spent a couple of months toying with Silicon Motion controllers and port replicators. I sent patches to the FreeBSD ATA subsystem maintainer—for a while I had possibly the only FreeBSD system that supported Sil3124 RAID with a port multiplier, although I didn't have six disks to attach to it. Shortly after that, the CAM subsystem took over, and I don't think anything ever happened with my code. The parts are still kicking around in my box-of-cruft. I should try them again someday.

Once OpenSolaris had been shuttered, I was looking for a platform to develop on again; with Clang on the horizon and FreeBSD's approach of nuts-and-bolts-included, I could return to applying interesting tools to KDE's codebase quickly. Most Linux distros have wandered off into developer-unfriendly territory, in my opinion, and, certainly for casual contributors, it's hard to start hacking on something in a clean-and-shiny distro. FreeBSD might have an ASCII art bootloader, but it gets you a system you can work with as a developer from the get-go.

Cue another few years of working with the renewed, reborn KDE-FreeBSD group, with tcberner@ and rakuco@ in the lead. I was the KDE guy doing FreeBSD, and they the FreeBSD guys

doing KDE, and it worked pretty well together. On the KDE side, we got FreeBSD added as a first-class citizen of the continuous integration system (snarkily, I'll add that one Linux distro was just canned for being notoriously difficult to keep running).

Linuxisms are now routinely fixed, and upstream is generally more aware of "things could also be different." That may be a side effect of (KDE) also producing mobile and Windows packages, but I'll take it.

I think I was part of the organizing committee of a EuroBSDCon once, too, but I can't find any reliable references (neither is my memory).

- **Craig:** Having been a user, contributor, and fan of BSD, FreeBSD was the obvious next step when BSD was mothballed. Using FreeBSD always felt like using BSD to me. I also liked FreeBSD's reputation as a high-performance server platform.

By the mid-1990s we started running FreeBSD 2.X at LBL, first on our laptops and eventually on commodity PC hardware. I've run on FreeBSD desktops ever since then.

- **Ilya:** I set up my first FreeBSD server (based on FreeBSD 6.2-RELEASE i386) to serve as a router for the corporate network of the company I worked for. It replaced a big, noisy Windows-based server and was able to provide fast file storage and network access while running on an old PC. Back then, I was impressed with how effective and easy to maintain the system can be, and how it can squeeze everything out of aging hardware.

One of the companies I worked at built a FreeBSD/ARM-based telemetry solution, which is how I got in touch with running FreeBSD on embedded hardware. I have also been running FreeBSD on my laptop for a while.

- **Chuck:** Around FreeBSD 4.7, my employer had a driver development kit with a working driver for Red Hat Linux. At this time, there was a great deal of uncertainty around GPL enforcement and a worry that the Linux driver might force them to reveal proprietary information about their hardware. In order to hedge their bets, they decided it would be wise to also include a driver for a non-GPL, open-source operating system. After experimenting a little with the BSDs, FreeBSD became the reference design platform.

If the open-source license was all FreeBSD offered, I wouldn't still be here today. To paraphrase an old TV commercial, "come for the license, stay for the design." At the time, developing device drivers on FreeBSD was an order of mag-

nitude better than Linux. Between the documentation, advanced storage features like SCSI target mode, and other well-thought-out and implemented subsystems, it was, and still is, an amazing development environment.

- **Yuri:** A friend mentioned FreeBSD in 1998, and I decided to try it. Since then, I've never gone back to Windows. What attracted me was the ability to dive into it, to learn how its various parts worked and interacted, and to hack it on all levels. FreeBSD is very well structured and cleanly designed. FreeBSD made it possible for me to learn how an operating system functions inside, an opportunity that no commercial OS can ever allow.

FreeBSD has the best software packaging system (ports) among all OSes. It allows any user to easily rebuild any package with a clear and easy procedure, anytime. As a matter of policy, absolutely all source downloads for ports are fingerprinted. This eliminates the possibility of MITM attacks, which no other system does. Other systems just never address some security problems in their packaging systems, and those are long solved in FreeBSD.

How did you end up becoming a committer?

- **Fedor:** I worked with the Linux ext file system in my job, so I tried to make some patches to improve the FreeBSD implementation. When my patches became relatively complex, I got the src commit bit.

- **Luca:** I've been attracted by open source as a concept since I was a student. I didn't feel very confident at that time, and felt that I wasn't skilled enough to be a valuable contributor. Then I started to work on FreeBSD, and thread support on devel/gdb had serious issues. I submitted some patches, and suddenly I became the port maintainer of gdb.

Joining Trivago, I had more opportunities to submit patches in the ports tree, so Lars (lme@) and Olivier (olivier@) proposed me as a ports committer.

- **Adriaan:** Early in 2017, tcberner@ and rakuco@ approached me saying that I should go for a ports committer bit; partly so that kde@ ports could be spread across a few more people, so that we would be more resilient—as we all have jobs and busy periods and real-life commitments too. After all, I've been using (and contributing to) FreeBSD off-and-on for at least 15 years, probably 20, and this feels like putting a ring on it.

- **Craig:** I've been a longtime contributor to Unix

new faces of FreeBSD

in general, and FreeBSD in particular, submitting my first FreeBSD PR in 1999. I became a ports maintainer in 2008 (sysutils/lbl-hf), and created or picked up maintainership for additional ports over the following years. This is probably when I made becoming a committer a personal goal.

I ran into Kirk McKusick at a party last summer and mentioned my interest in being a committer. We know each other from when I was a CSRG volunteer, and he immediately offered to look into the process for me. Larry Rosenman and Matthew Seaman kindly volunteered to be my mentors. A short while later, I was invited to join FreeBSD as a ports committer.

- **Ilya:** Once I bought a somewhat expensive ARM-based home server, GlobalScale DreamPlug. It was possible to run FreeBSD on it; however, after setting it up, I found out that the onboard WiFi was connected to the system via SDIO bus. Back then, I didn't even know what SDIO was, and after reading some docs, I decided it would be a fun project to try to add support for SDIO bus. It turned out to be, well, a rabbit hole. After implementing the initial bits and discussing the problems I encountered with Warner Losh (imp@) and Adrian Chadd (adrian@) during EuroBSDCon 2013 in Malta, I started implementing a totally new SD/MMC stack based on the CAM framework. It ended up being an excellent project for learning a lot about FreeBSD kernel internals and for working with different people in the project to solve problems. Then Warner volunteered to be my mentor and let me improve FreeBSD MMC and SDIO support!

- **Chuck:** Michael Dexter's talk about diskctl got me curious about how to represent SMART buffers across the ATA, SCSI, and NVMe protocols. In the process, I wrote a little application that uses CAM(3) to send SMART commands to drives. But the CAM NVMe driver didn't have a couple of calls needed to make this work. Ten or so patches later to the NVMe driver, CAM, sgdd, and others, Ken Merry and Warner Losh got tired of committing code for me and asked if I'd like to be a committer.

- **Yuri:** I was discovering software that I found useful, and most times when this software wasn't available on FreeBSD, I was porting it. This way, I created about 200 ports, and in the process of doing this, I also fixed bugs in hundreds of other ports and submitted patches. Eventually, one committer offered for me to join the club. I agreed, but something unexpected came up and he didn't

have time to be a mentor. Later, in late October, another committer, Tobias Berner, asked me again. I agreed, and that is how this happened!

.....
How has your experience been since joining the FreeBSD Project? Do you have any advice for readers who may be interested in also becoming a FreeBSD committer?

- **Fedor:** I should say that the FreeBSD developers' community is really a friendly and fine group of people. I got this impression when I sent the self-introduction mail to the developers list. People may reply to your self-introduction mail in three to five days—it is really awesome. You get the feeling that your ideas and efforts are really needed for this project.

Also, the FreeBSD mentorship is a really great way to help people pass their introduction phase and become committers. Here I should say thanks to Pedro Guffini (@pfg), who took responsibility for being my mentor. It has been a real pleasure for me to work with him.

- **Luca:** The knowledge that you can find in the FreeBSD community is smashing. Moreover, I was really welcomed by all developers. I've already made some small mistakes, and other port committers were so kind to let me learn from my errors, instead of just correcting them. I've been involved in constructive discussions from the beginning, and not just passively accepting the decision of senior contributors.

To become a FreeBSD committer, the community has to know that you are a good contributor. So, contribute with patches, learn how to improve the quality of your contribution, and be visible on IRC and mailing lists. It's a process that needs some time, but it's also very rewarding.

- **Adriaan:** To some extent, things haven't changed much: I still run the same system, still spend most of my time on ports and pushing stuff into the KDE-FreeBSD ports github tree. I've picked up a couple of ports myself, for the heck of it (e.g., sayonara, because I want something slightly more capable than mpg123, but not much).

For newcomers, I'd say "keep doing the things you love." And, since I've made my share of mistakes in my first months, "mind your p's and q's." Things can be very exacting in ports land, and it's natural to miss some little details at some point. Your mentors are there to help, but they're human too. The first "RE: svn commit" message you get

from mat@ pointing out an omission is a rite of passage, and just a reminder that we're all keeping an eye out to create the best systems we can.

• **Craig:** It's been great fun getting to see behind the curtain, gaining access to internal resources, learning the procedures, and seeing how the overall project fits together in greater detail. Everyone I've interacted with has been helpful and polite. It was very satisfying when I made my first "production" commit (as opposed to the commits related to becoming a committer). And being a committer is clearly the most efficient way to maintain a port, both for the maintainer/committer and other members of the FreeBSD Project.

I don't know if my path to becoming a committer is typical, but I suspect it is a good model to follow. File PRs, become maintainer of several ports, and learn how the ports system works; read the documentation (obviously), but also /usr/ports/Mk makefiles. Look at existing ports to pick up tricks and optimal methods for making a port function well.

• **Ilya:** I had been contributing to the FreeBSD Project long before I became a committer—I maintained some ports, attended conferences, talked with people, and discussed ways of solving all sorts of problems. What you see pretty often is that when the system lacks certain features that many users need, people just go ahead and stuff gets implemented!

What is important to understand is that this is an open-source project—so don't expect people to solve

your problems and write code for you, and don't rely on any timelines until the feature really lands in the tree. Instead, try to research, debug and implement yourself, and come back with patches and ideas, and build a better system for everyone.

A future FreeBSD developer is active in the mailing lists, goes to the conferences and actively participates in the discussions, and eventually finds a project that is fun to work on and is useful for other people in the community. Eventually their commit bit will find them!

• **Chuck:** FreeBSD has been a very welcoming community to me as both a contributor and committer. As for advice, my life is more often cited as the "cautionary tale" than in the "how to" category. That said, just go build stuff and be open to feedback from the community. The depth and expertise available here is a true gift and will make you better if you let it.

• **Yuri:** I have always had good experiences with FreeBSD, and can't imagine using any other system. I like the technology and I like the people. FreeBSD people are very friendly, skilled, and intelligent. FreeBSD committers are among the world's best computer experts. I was really touched by many warm welcome replies to my introduction email after I joined the committers community.

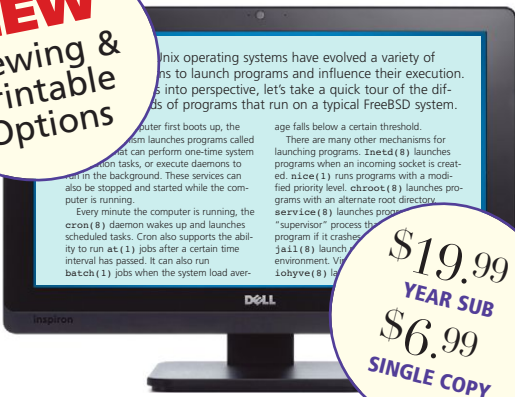
I have only had a commit bit for two weeks now; it is probably too early to offer any advice. Just don't hesitate. It's well worth it! •

.....
DRU LAVIGNE is a doc committer for the FreeBSD Project and Chair of the BSD Certification Group.



FreeBSD JOURNAL

NEW
 Viewing &
 Printable
 Options



\$19.99
 YEAR SUB
\$6.99
 SINGLE COPY

The Browser-based Edition (DE) is now available for viewing as a PDF with printable option.

The Browser-based DE format offers subscribers the same features as the App, but permits viewing the *Journal* through your favorite browser. Unlike the Apps, you can view the DE as PDF pages and print them.

To order a subscription or get back issues, and for other Foundation interests, go to

www.freebsd.foundation.org

The DE, like the App, is an individual product. You will get an email notification each time an issue is released.