

Starting FreeBSD Development

AN INTERVIEW WITH KAJETAN STASZKIEWICZ
BY TOM JONES

TJ: Can you tell me a bit about yourself and your background with FreeBSD?

KS: My name is Kajetan Staszkiwicz. I was born in 1984 in the magical city of Kraków in Poland. I was introduced to computers when I was around 8 years old, starting with 8-bit machines like ZX Spectrum and Atari 65XE in a computer club. Later my parents got me a Commodore 64. I have a lot of fond memories of this machine and still, I have a working one on my desk.

In 1996 we got our first PC clone at home and very soon the Internet became available for home users in Poland through a dial-up connection. This was good enough only for a short time and around 1999 my neighbors and I decided to build a neighborhood network. We were around 15 people sharing a 115kb/s connection to an Internet cafe over 10Mb/s Ethernet cables hanging haphazardly from the roof to the balconies of our flats.

Internet sharing was done by a Linux box running IP chains provided by the sysadmin from the Cafe, which was soon replaced by another system managed by some older friends from our network. Later I took over managing it myself. I wrote some CGI scripts in Perl to manage the DHCP server and traffic shaping through a web interface.

With this knowledge, I managed to get a part-time job at a company running a network not unlike ours, albeit a commercial one. At that time, I was also studying computer science at the Jagiellonian University. One of my classmates had introduced me to FreeBSD, possibly version 5.0. At my new job, routing and traffic shaping had been performed by Linux systems, but FreeBSD's jails seemed like the perfect solution for our hosting platform. We needed to serve web pages, email, webmail, and so on. Virtualization was not yet a thing back then and with jails I was able to accomplish the separation of those services.

My next job was at a company using FreeBSD for most of its services, some employees were brave enough to even use it on their workstations.

Currently, I live in Hamburg, Germany. At my current job, FreeBSD is our choice for routers and load balancers.

“

My next job was at a company using FreeBSD for most of its services, some employees were brave enough to even use it on their workstations.

”

TJ: What were your steps to making your first FreeBSD contribution? How did you pick what to work on to begin with?

KS: The choice was simple: I had to fix what was broken. At \$WORK we depend on source tracking of pf(4), and we came across a rather nasty bug where deleting source nodes caused each source node to scan all states. With hundreds of source nodes and hundreds of thousands of states, this caused a lot of looping in kernel code, effectively locking the system.

With the help of glebius@, I've managed to fix it and land my first patch to FreeBSD.

TJ: How did you track down this problem? Were there existing tools in FreeBSD that helped, or did you have to develop your own?

KS: I already had some general idea when the issue happened, because it was caused by our custom tool we use for load balancing.

At the time I didn't have much experience with FreeBSD, especially with kernel development, so I went with the dumb brute force approach: debugging by adding printf's until I found where the code froze.

It was very helpful, though, that the system comes with all the tools needed to build a custom kernel.

The ability to produce memory dumps on demand which then can be analyzed with a debugger comes in very handy too.

TJ: You must have learned a lot in that process. Are there any shortcuts you would like to share for new developers that would make the debugging and development you had to do easier?

KS: It's not only knowledge but also additional tools and workflows that make my job easier.

The first one is having a proper development and testing environment. At \$WORK we only use FreeBSD on bare metal and my initial bug fixes had to be deployed on this hardware. Those are "real" servers, which means that they POST for a few minutes before they boot, and they all serve production traffic affecting our customers.

Testing bug fixes in such conditions was very stressful and slow. Nowadays I have 3 tiers of systems involved: virtual machines which can boot new kernel code in seconds, testing hardware which gets some traffic, but if they break, customers won't be affected, and staging hardware, which handles a bit more traffic, but still no paying customers.

Another thing is the testing suite of FreeBSD. That is a real lifesaver!

I can test my changes within a few minutes (yes, it can run in parallel!) and be sure that I'm not breaking any existing functionalities. Sadly, I occasionally find that there are things for which no tests have been written yet. But we're slowly getting there.

The last thing that comes to my mind is the option to enable various debugging options during kernel compilation. One can track lock order, various variable assertions, and even in-

“

My next job was at a company using FreeBSD for most of its services, some employees were brave enough to even use it on their workstations.

”

valid memory access in the kernel.

The *FreeBSD Developers' Handbook* documents those options quite well.

TJ: Is there any advice to new contributors or anything you would like to add?

KS: Open source projects, including FreeBSD, depend on contributors. FreeBSD is a mature project, and even though there is a lot of complex code, there's always room for contributions, including small and simple ones. The community is friendly, supportive, and always happy to help with code reviews and constructive feedback. Don't be intimidated!

I hope that contributions can bring in new, useful features, and give everyone, including the newcomers, a chance to not only learn new skills and grow professionally but also do something fun with computers.

TOM JONES is a FreeBSD committer interested in keeping the network stack fast.

KAJETAN STASZKIEWICZ is a system administrator specializing in networks and virtualization. He occasionally contributes to FreeBSD, primarily working on the Packet Filter.

Books that will help you. Or not.

“While we appreciate Mr Lucas' unique contributions to the Journal, we do feel his specific talents are not being fully utilized. Please buy his books, his hours, autographed photos, whatever, so that he is otherwise engaged.”

— John Baldwin
FreeBSD Journal Editorial Board Chair

<https://mwl.io>

