Dmitry Safonov *Kernel Engineer*

0x7f454c46@gmail.com • github.com/0x7f454c46 • linkedin.com/in/0x7f454c46
Currently in Dublin, Ireland

Summary

Low-level Engineer, opensource contributor and maintainer. Focused on long-term projects, that have priority on correctness and "doing the right thing". Takes self-testing seriously. Author of long email-threads and many code reviews that significantly decreased coding issues per LOC.

Born in USSR/Russia. Got a university degree of an engineer and self-taught Linux. Eventually fell from regression analysis, numerical methods, differential equations and statistical C++ projects into assembly, kernel C and multi-threaded Linux memory ordering developments.

Started as home-projects enthusiast that recompiled OpenBSD kernel on a home-server and learned Linux virtualization/replication basics. Moved to bash-scripting projects, including a home web-scrapper that gathered some data and statistics (a scripting skill that was very handy since then).

Got a chance to participate in many wonderful projects. From being an author of PoC on non-documented hardware to creator of GCC compiler plugin. One of CRIU (Checkpoint Restore in Userspace) contributors and maintainers. Author of some thousands lines in upstream Linux kernel. Contributed to many parts, including x86, memory management, networking, selftests, tty drivers and core code.

Emigrated to Ireland in 2017. Since then, worked remotely from other team members.

Experience

Arista Networks   DUBLIN, IRELAND
    Linux kernel engineer   *Sept '17 – present*

Currently working on TCP-AO Linux kernel support. I've spent time designing uAPI; writing kernel selftests to verify the functionality and to discover uAPI corner-cases; writing documentation on the project. Together with my colleagues, we wrote code that implements RFC5925 in Linux kernel; we communicated with the BGP userspace team to get feedback and worked with the network interoperability testing team for verification and scale testing. Besides writing the code, I was reviewing all the changes and did some Gitlab automation. Later I polished the result, implemented missing features, debugged the code and split it apart into upstreamable patches.

At the moment of writing the patch set is version 15, it received approvals from some of the network maintainers and reviewers and all pre-required changes that I've made (crypto, jump_label and TCP-MD5) were merged. I'm addressing the feedback from reviews, fixing items from my todo-list and writing more selftests in the background. I'm quite pleased with the result: it has about 5.5 thousand good-quality code changes, that are properly split into patches, well documented and tested. The project went smoothly, the customers are satisfied, no fires lit.

Previously I participated in Linux kernel upgrade projects. The goal was to move from older kernels to newer LTS (Long Term Support) kernels, porting company-specific patches and addressing any issues that arise on testing. During these projects, I've focused on reviewing non-upstream company kernel code and attempting to either eliminate it or upstream it. That decreased the future work (technical debt) for maintaining and upgrading the Linux kernel in the company as well as improved the quality. Participating in that project involved fixing proprietary drivers from different allocation issues (leaks/use-after-free) to scaling issues and introducing proper locking. A little cherry on top was debugging GNU/Make to fix a race in jobserver that was rarely hitting the build system.

One of the exciting things during my time here was debugging 2 CPU issues in one of the major x86 vendors. Another issue I'm quite proud to have debugged and fixed was missed locking in the generic tty driver. The reason is that besides missing semaphore in tty_reopen(), I had to fix the tty semaphore realization itself: from missed wake-up on reader/writer contention, a theoretical issue on non-default timeout, providing more priority to line discipline changes rather than write/read and optimizing the fast-path by not taking the writer lock. It was quite challenging to fix this tty mess as for most of those fixes I had only one report which was dmesg log from syzbot or a person. From that little log I've managed to find and fix the issues in semaphore realization!

Another big and interesting project that I did in the background was fixing IPSec/XFRM tunnels on ia32. It ended up also having a nice selftest.

At this job, I participated in investigating customers' kernel escalations, analyzing and doing triage for kernel issues in the testing lab. I also mentored two people: one who joined on full-time and an intern.

CRIU
Maintainers team *Dec '16 – present*
...a project by various mad Russians...

Likely, the most interesting project I've worked on. It involves everything: understanding CPU architecture, adding and modifying Linux kernel code, network stack and etc. I was working on various things in this project. From maintaining build/Makefiles and adding cross-compilation for different architectures, to making arm/arm64 work. I made Linux kernel patches to support 32-bit x86 (ia32) Checkpoint/Restore. Addressed issues with x86 mmap allocator and added support to CRIU itself. Together with my good friend Andrei, we introduced a new time namespace in Linux kernel, that helps migrating containers.

Today CRIU is used by hosting providers, Google task manager (Borg) and even by Microsoft's ChatGPT. Unfortunately, these days I'm not as active contributor to CRIU project, as I wished to be, but I'm trying.

Virtuozzo
Kernel engineer Moscow, Russia
Jan '16 – Jul '17

I was working on VZ kernel (based on RHEL7, Linux v3.10). I was fixing kernel crashes, adding OpenVZ-specific features and addressing bugs from testing or customers. That's where I started working and contributing to CRIU, being responsible for VZ CRIU (product version of CRIU). I was reviewing patches, creating releases, backporting needed commits and being liable for the resulting product.

Parallels
Kernel engineer Moscow, Russia
Dec '15 – Jan '16

Started working in a team that later was separated into Virtuozzo company.

Samsung Research & Development
Kernel engineer Moscow, Russia
Jul '14 – Nov '15

Developing Samsung Tizen TV kernel (Linux v3.10 based). I've worked on feature requests on different parts of the kernel: IRQ handling, process scheduler, memory management, etc. Also, that job involved analyzing crash logs and providing bug fixes.

One of the interesting projects I did by myself was adding FIQ support to Linux and to closed-source Samsung Secure OS (running in ARM TrustZone extension). That provided a reliable way to interrupt another ARM CPU core, even if it was running in an IRQs-disabled context.

Another PoC/research project that was made by me solo was a GPU swap. That added functionality to an open-source Mali GPU driver that we didn't have specifications for. As a result, it created a project that preserved 150 - 200 MB on a 1.5 - 2 GB Tizen TV set and similar Android Tablets/Phones by moving inactive GPU pages into ZRAM (compressed graphical memory swapping).

I was in a team that worked on ARM big.LITTLE Linux high-performance scheduler. One of the most interesting bugs I was proud of fixing was the incorrect voltage set for one of the CPU frequencies. That made Linux kernel crash on overnight stress-testing in various random places. Ouch!

Education

Moscow Aviation Institute
Engineer Specialist degree in Informational Systems and Technologies Moscow, Russia
2008 – 2014

Diploma thesis was "Developing an algorithm for searching ground objects in UAV live video stream". Quite glad I didn't continue this project in PhD as that would likely have been used in illegal invasion in Ukraine.

Skills

Technical expertise: Linux, multi-threaded kernel-version of C, bash, x86 and ARM assembly, GNU/Make, Git, Vim.

Languages: Russian (*native*), English (*fluent*), Spanish (*elementary*), Ukrainian (*elementary*), Greek (*beginner*) and Latin (*beginner*).

Interests/hobbies

In alphabetical order: Book-worm (esp. non-fiction), culture-learning, functional programming, hiking, learning languages, math/statistics, motorcycling and vegetarian cooking.