Ulya Trofimovich

London, UK
E-mail: skvadrik@gmail.com
Mobile: +44 7899 602 542

Summary

I am a C++/Linux developer with background in parsing theory and compiler construction.

Skills

- **Programming languages**: C++ (proficient, C++98/C++11, over 8 years of experience), Haskell (basic), Bash (basic), Python (beginner), Assembly (reading skills, mostly x86).
- **Operating systems**: Gentoo Linux (basic usage, over 7 years of experience).
- **Profiling and debugging tools**: Valgrind, Perf, Strace, GDB, GCC/Clang sanitizers, Objdump, etc.
- **Parsing tools**: lexer/parser generators, parser combinators, regular expressions.
- **Compiler theory**: dataflow, SSA, pointer modelling, compiler optimizations.
- **VCS**: Git.

Education

Belarusian State University (BSU) — Minsk, Belarus
Bachelor's Degree in Mathematics and Systems Programming — *2008 – 2013*

Experience

VirusBlokAda LTD — Minsk, Belarus
Software Engineer — *Aug. 2011 – Aug. 2014*

- **Anti-Virus Kernel**: co-developed a processor emulator for x86 ISA family. Added emulation of specific instructions and wrote a tool that builds statistics on instruction usage in malware code.
- **JavaScript emulator**: developed a lightweight ECMAScript-5.1 conforming interpreter for malware emulation in a sandboxed environment (written from scratch in C++ and optimized for performance).
- **Malware classification system**: developed a fast signature-based pre-filter to the anti-virus kernel that classifies incoming malware samples into categories.
- **Regular expression library**: developed an internal library for Perl5-compatible regular expressions.

Perforce (former PRQA) — London, UK
Software Engineer — *Nov. 2017 – Feb. 2019*

- **Dataflow engine of a C/C++ static analyzer**: found and fixed numerous bugs in program modelling and transformation; implemented checks for coding standard compliance (CERT C, RePhrase); extended logging and debugging capabilities; helped training new members of the dataflow team.
- **Analysis of multithreaded code**: co-developed a new component for detecting bugs in multithreaded C/C++ code, in particular, I worked on deadlock detection.
- **Scripting**: wrote multiple Bash/Python/C-Reduce scripts to help automate routine tasks, organized a centralized repository for developer scripts and helped document them.
- **Administration**: maintained a Gentoo server used for dataflow testing.

Open Source
Software Engineer — *2011 – present day*

- **RE2C lexer generator**: (`http://re2c.org`) RE2C is a tool for generating fast lexical analyzers for C/C++ (used by such projects as PHP and Ninja). Contributions: added Unicode support for the generated lexers, researched and developed efficient algorithm for submatch extraction, developed self-testing and self-validation tools, implemented numerous optimizations for speed and size of the generated code, updated project documentation, extended portability, fixed many bugs, provided help and support for RE2C users.
- **LALR2C parser generator**: (`https://github.com/skvadrik/lalr2c`) Developed an experimental tool in Haskell for generating very fast LALR(1) parsers for C/C++.
- **Other**: Contributed occasional patches to open source projects such as GHC Haskell compiler, Midnight Commander, xxdiff, yasm.

Research

Tagged Deterministic Finite Automata with Lookahead *Feb. 2015 - Aug. 2017*
Studied efficient submatch extraction in regular expressions, POSIX disambiguation semantics, and their practical application to performance-oriented lexer generators like RE2C. Wrote a paper:
`http://re2c.org/2017_trofimovich_tagged_deterministic_finite_automata_with_lookahead.pdf`

Fields of interest

Compiler construction, program analysis and transformation, code generation, formal languages and automata.