

Pavel Dolgov

Software Developer. Email: pdolgov99@gmail.com

GitHub: <https://github.com/zer0main>

LinkedIn: <https://www.linkedin.com/in/zer0main>

Homepage: <https://zer0main.github.io>

THEORETICAL KNOWLEDGE:

Data structures and algorithms in informatics, cryptography, strong background in software design patterns and site reliability engineering.

Blockchain & decentralized finance.

TECHNICAL SKILLS:

Programming languages (general-purpose):

Go, C, C++ (7 years), Lua, Python, Haskell.

Programming languages (domain-specific) and markup languages:

Bash, SQL, Makefile, cmake, LaTeX, Markdown, Wiki, XML, JSON, YAML.

Operating systems:

Linux (Debian) - user for 7 years.

System administration experience:

Docker, Docker Compose, Nginx, ssh, Vim, Tmux, VirtualBox.

Libraries/frameworks:

Protobuf and gRPC, Qt, boost, Wt, Lua, LuaBind, MXE, NSIS, Love2d, pygccxml, yaml-cpp, LevelDB, PostgreSQL.

Version control systems: git, mercurial, svn.

Unit testing frameworks:

Busted, Boost.Test.

Continuous integration:

Travis CI, Coveralls.io, Gitlab-CI, Github-CI.

Profiling:

valgrind, gdb, delve.

EDUCATION:

The Moscow School of Programming (Yandex branch). 2014 – 2017 (finished).

Specialized in C++, algorithms and data structures for programming contests.

<https://informatics.ru/>

Trinity College Dublin – Computer Science. 2018-2018 (not finished).

WORK EXPERIENCE

Lead Developer

SCP, Corp, Full-time

Dates Employed: Aug 2020 – Present

Working on Decentralized Data Storage.

Senior backend Developer

Waves Platform, Full-time

Dates Employed: Jan 2019 – Aug 2020

Employment Duration 1 yr 8 mos

Location Moscow, Russian Federation

Blockchain developer.

I was working in a team of 3 developers and we built Go implementation of full node for Waves blockchain. My main focus was storage structure of the node and low level optimizations. I came up with a number of solutions which allowed to optimize blockchain import speed and rollback of blocks a lot compared to other existing implementations.

The key ideas were saving transactions to plain file instead of putting them into the database while importing blocks (with storing offsets in this file in the database) and storing block IDs

with each state record to enable fast rollback by only removing IDs of discarded blocks. Then when the records are read, their corresponding IDs are checked against the list of valid blocks, this way we filter out discarded blocks.

The code is open source and can be found here:

<https://github.com/wavesplatform/gowaves>

CERTIFICATIONS:

IELTS (International English Language Testing System): CEFR Level C1.

PUBLIC SPEAKING AND CONFERENCES:

February 2017

FOSDEM conference, Brussels.

Presenting LuaWt, Lua bindings for a C++ Web Toolkit Library:

<https://archive.fosdem.org/2017/schedule/event/luawt/>

LANGUAGE SKILLS:

Russian, Native or bilingual proficiency.

English, Full professional proficiency. CEFR Level C1 (Advanced English), experience of participation in international conferences and giving talks, experience of living and studying in English-speaking country for one year.

PROFESSIONAL INTERESTS:

Algorithms, information security, functional programming.

OPEN SOURCE SOFTWARE WRITTEN BY ME AND OTHER PROJECTS:

LuaWt, Lua bindings for a C++ Web Toolkit library.

Written in: C++, Lua, Python. Libraries: Wt, Lua.

<https://github.com/LuaAndC/luawt>

(2015 - Present)

Battleship, classic battleship game. Written in: C++. Libraries: Qt.

Example project aimed to demonstrate the usage of MXE (M cross environment) with Continuous Integration Systems for flexible and configurable build and deploy. It also shows practical application of MVC design pattern.

<https://github.com/zer0main/battleship>

(2014 - 2017)

Problems, a collection of implementations of multitude of algorithms and data structures, solutions of programming contest tasks, notes on mathematical problems.

Written in: C++, C, Python.

<http://github.com/zer0main/problems>

(2015 - Present)

Bacteria-core, cellular-automata game (academic project at The Moscow School of Programming).

Written in: C++, Python. Libraries: Qt, boost.

Program written for studying concepts like bytecode (it required to implement simple language as part of the project), unit testing, integration testing and working with Continuous Integration and Coverage-measuring systems.

<https://github.com/zer0main/bacteria-core>

(2016)

I also contributed to scientific project NPG-explorer (Nucleotide PanGenome explorer) by configuring its build and fixing some bugs.

<https://github.com/npge/npge>