Ilnur Iskhakov

e-mail: job.ilnur.iskhakov@outlook.com
GitHub: github.com/iDawer

Software Engineer

I'm looking for **Junior Rust Developer** position.

- Language: prefer *Russian*, can speak *English* and planning to take an exam for it.
- Location: Ufa, Russia. Open for relocation. Prefer full-time remote. Any timezone.
- Good knowledge in *Rust, Bash, Git, Linux, Python*.
- Basic knowledge in *C/C++, cryptography primitives, Java, C#, JavaScript, HTML/CSS, Lua, SQL, Assembler*.
- Hobbies: I'm pretty into bicycling.

Background

Contributing into Rust infrastructure

I have committed into *rust-analyzer* worth of top 12 contributors (by GitHub [1]). My contributions include fixing bugs and adding features into core of *r-a* [2]. In particular I worked on exhaustiveness check subsystem to synchronize it with Rust compiler, improved type system implementation and various IDE assists.

I learned *type systems*, *theorem proving*, *abstract syntax tree* and *git*.

Making Rust target for ANTLR

ANTLR is a parser generator and I'm using *Java* to extend ANTLR to generate parsers in Rust language. It also has Rust runtime I'm writing for generated parsers. For memory management I got inspirations from *C++* target and it showed me how Rust's ownership model restricts one to one translation from other languages.

This is the project where I learned Rust language: *syntax*, *trait system*, *lifetimes*.

Fixing Linux driver

I revived an old EciAdsl usermode driver for USB ADSL modems [3]. The project is written in *C* and the last commit is dated 2007. I had to adopt it to recent Linux kernel changes and adjusted it to make it work with my modem. I fixed includes and transitioned from deprecated *usbdevfs* Linux subsystem to new *Linux USB API*.

The instrumentaries I used was *usbcap* to log USB bus, *gdb* for debugging and *Wireshark* to view *PPP* packets within USB bus traces.

Authoring jbeam-flow: Python add-on for Blender [4]

I used *Python* to extend 3D computer graphic tool *Blender* to make it understand vehicle files (in *JSON*-like format) of BeamNG.drive game. The idea is to map vehicle parts to *Blender*'s 3D primitives to make them editable with visual tools.

I wrote a *parser* in *EBNF* with ANTLR and implemented user interface with help of *Python*'s *metaclasses*.

Reference

[1]: github.com/rust-lang/rust-analyzer/graphs/contributors
[2]: github.com/rust-lang/rust-analyzer/issues?q=author:iDawer
[3]: github.com/iDawer/eciadsl-usermode
[4]: github.com/iDawer/jbeam-flow