

Bc. Václav Kubernát

vaclav@kubern.at | github.com/syyyr

EDUCATION

Faculty of Information Technology, CTU in Prague

Bachelor, Software Engineering

Prague, Czech Republic

Aug 2015 – Jun 2019

Programming classes were focused on programming concepts/algorithms rather than individual programming languages. They gave me a good foundation for improving my programming skills and learning new programming languages.

- This is where C++ became my main programming language
- I have written a multi-threaded FTP server in C++ as a final programming project

I finished my studies with a bachelor's thesis that focused on parsing input with context-free grammars in C++. The thesis was based on my work for *CESNET*.

I have also collaborated on making \LaTeX materials for the final state exam. My main contribution was the build system.

EXPERIENCE

Software engineer

CESNET

Prague, Czech Republic

February 2018 – Present

Initially, I started working at *CESNET* as a part-time programmer during my studies at *CTU*. After finishing school, I transitioned into a full-time software engineer.

- Programming mainly in C++
- Code always goes through code review using *Gerrit*
- Testing is done for nearly all changes

My biggest project is *netconf-cli* – an interactive CLI, which uses the NETCONF protocol to connect to devices over SSH and configures them according to a configuration model defined by the YANG language. This program is used to configure optical devices at *CESNET*.

- The code uses latest C++20 features and advanced C++ techniques like templates and standard algorithms
- Command parsing is implemented using the *Boost.Spirit X3* library which embeds parser grammars directly into the code
- The NETCONF communication is implemented via the *libnetconf2*, *sysrepo* and *libyang* libraries implemented by employees of *CESNET* in Brno, I have also contributed code to these libraries
- The code is available at <https://github.com/cesnet/netconf-cli>
- For the latest iteration of the software, I have also developed C++ bindings for *libyang* and *sysrepo* <https://github.com/cesnet/libyang>, <https://github.com/sysrepo/sysrepo-cpp>

Working with optical devices involved programming for embedded computers, which means that most of the software has to be written from scratch. This includes low level C++ and kernel drivers, daemons, interfaces used for configuration and even websites. Things I have worked on:

- Contributions to the Linux kernel: driver for a power supply and fixes for the hwmon subsystem
- Userspace C++ drivers for optical components
- Linux system images via Buildroot
- A website in Javascript that graphs various optical networking stats

PROJECTS

For most of my free-time projects I like to use a combination of Node.js and Typescript (and Express.js if it's a web application).

Prší | [Typescript](#), [React](#), [Websockets](#)

During the COVID-19 pandemic, I have developed an online multiplayer version of a card game popular in the Czech Republic to play with my friends.

- The entire program is written in Typescript and Node.js
- UI is written in React
- Communication between the server and the players is implemented using Websockets
- The project uses Webpack for deployment
- Code is available here: <https://github.com/syyyr/prsi>

node-shortener | [Typescript](#), [MongoDB](#), [Express](#)

A simple web application I wrote to shorten URLs. The reason I wrote this is that I wanted a simple website and other shortening services seemed too complicated.

- Very simple, uses Typescript, Node.js as the platform, Express.js for the server and MongoDB for storage
- Code is available here: <https://github.com/syyyr/node-shortener>

tradfri-mqtt | [Zigbee](#), [MQTT](#), [Typescript](#), [Node.js](#)

Interface for Raspberry PI I use to control my smart home devices like lights and remotes.

- Uses *zigbee2mqtt* which translates the standard protocol for smart devices to MQTT
- A Node.js program handles events from the devices and acts accordingly
- Code is available here: <https://github.com/syyyr/tradfri-mqtt>

obedy | [Python](#), [beautifulsoup4](#)

A web scrapper which gathers restaurant menus near my workplace. Before COVID-19 I would host this on a website for my colleagues to use.

- Written in Python
- Works by parsing website source code and transforming that into machine readable format
- Code is available here: <https://github.com/syyyr/obedy>

I also contribute to open source repositories, mainly ones that have something to do with text editor (Vim) plugins. I keep all of my configuration open source and stored in Git, so I can easily synchronize it between my devices.

TECHNICAL SKILLS

Languages

- C/C++ (7 years) - The language I am most comfortable with and the one I use at work
- Typescript/Javascript (3 years) - My favorite language for doing free-time projects
- Bash (7 years) - I've been using Linux for a long time, this is my language of choice for scripting
- Python - When Bash is not enough, I use Python
- Rust - Mostly just interest, but I wrote one small project in Rust

Developer Tools

- Experienced with Git
- Debugging via GDB
- CMake - all C++ projects I worked on use CMake
- Gerrit/Github - I am used to the workflow of Gerrit and Github and how they handle PRs
- Vim - my editor of choice

Libraries: Boost, Catch2, trompeloeil, doctest, spdlog, libssh, Express.js, Chart.js

Other: Node.js, Linux, Buildroot

SPOKEN LANGUAGES

Czech: Native speaker

English: C1 - I have no problems using English

OTHER INFORMATION

Driver's license: B (cars and small scooters)

Interests: improving my dev workflow (and text editor / shell configuration), playing the ukulele, juggling and driving