

BENJAMIN JUREWICZ

 +48 504 028 732

 github.com/BenJurewicz

 benjamin.jurewicz204@gmail.com

 linkedin.com/in/benjaminjurewicz

ABOUT ME

I am a 3rd year Computer Science student at Gdańsk University of Technology. I have been interested in computers since my youth, developing my skills more broadly in high school, where in my free time I learned about topics that interested me. I became most interested in C++ programming and the Linux system. I also learned the basics of web development, network configuration, and server administration. Currently I am interested in creating web applications and learning about developing programs for embedded systems.

EDUCATION

Gdańsk University of Technology 2023 — Current

Bachelor's degree | Computer Engineering

1st High School named after Jan Kochanowski in Olecko

2019 — 2023

Advanced level subjects: Mathematics, Computer Science, English

PROJECTS

Linux Kernel HID Driver for controlling mouse LEDs

Source: <https://github.com/BenJurewicz/razer-naga-trinity-linux-driver>

A Linux kernel driver for controlling the RGB lighting of the Razer Naga Trinity mouse. The driver was created by reverse-engineering the USB protocol and integrates with the standard HID subsystem. Functionality is exposed to any userspace application through a sysfs interface. A GUI application is provided as an example of how to interact with the driver.

- Language: **C**

Symmetric Encryption Demonstration App

Try at: benjurewicz.github.io/szyfrowanie-symetryczne

Source: github.com/BenJurewicz/PG-CB-Projekt-Szyfrowanie-Symetryczne-GUI

A cross-platform application for interactive demonstration of symmetric encryption using ECB, CBC, CTR modes. Supports encryption of raw text as well as files and allows for the demonstration of how the corruption in the encrypted data affects the different decryption methods.

- Language: **Rust**
- UI Library: Dioxus

Key Value ISAM Database

Source: <https://github.com/BenJurewicz/PG-SBD-Proj2-ISAM>

The database allows for key-value storage and fast retrieval of data. It is written to generalize easily to any value that implements the **Serializable** concept. The UI assumes a value of string with length up to 30 characters for ease of demonstration. Additionally all file IO is buffered using a custom class for easy index based file access. Written using modern C++ functionalities.

- Language: **C++23**

SKILLS

Software: Linux (Mainly Fedora and Ubuntu) | Windows | MacOS | Docker Basics

Languages: C/C++ | Python | Java | HTML, CSS, JavaScript/TypeScript | Rust | x86 Asm