

Pratham K

prathamIN@proton.me | [blog](#) | [@git-bruh](#)

TECHNICAL SKILLS

Tools: Git, Docker, GDB, Strace, GNU Make, CMake, Meson, Autotools

Languages: Bash / POSIX sh, C, C++, Python, Rust, TypeScript, Zig

Miscellaneous: CI/CD, Cross Compiling, Linux Kernel Configuration, Software Packaging, Systems Administration

OPEN SOURCE EXPERIENCE

Package Maintainer and Core Team Member

2022 – Present

KISS Linux Community

- Participated in the packaging and upkeep of software packages including Compiler Toolchains, Containerization Tools and Browsers, working with build systems like **CMake** and **Meson**, and writing **runit**-based service scripts
- Developed automation projects to streamline the maintenance workflow, including the implementation of a sandboxed multi-stage rootfs bootstrap script utilizing **unshare** and **bubblewrap**: [maintainer-utils](#)
- Wrote patches to resolve build failures and runtime crashes, ensuring software portability across different compiler toolchains (**GCC** and **LLVM/Clang**) and libc implementations (**glibc** and **musl**)
- Improved stability and performance by fixing various issues in the shell-based **KISS** package manager: [kiss](#)

Open Source Contributor

2022 – Present

Contributions To Projects Used In Personal FOSS Endeavours

- Resolved a bug in **Chromium** that caused page crashes on **GCC** builds due to undefined behavior: [#4546610](#)
- Contributed new interfaces, portability fixes, and support for mouse events to **termbox2**, a TUI library: [termbox2](#)
- Wrote a detailed report about Wayland application crashes inside the guest machine when using **Qemu**'s **virglrenderer** backend, helping uncover a bug in **Mesa**: [virglrenderer/#291](#)
- Discovered a libcurl usage bug in **Flatpak** and **OSTree** causing crashes on certain systems: [#5074](#), [#2706](#)

CONTRACTUAL PROJECTS

libcurl Cross Build | *C, Rust, Docker, Autotools, CMake, Cargo, Android NDK*

Feb 2023

- Wrote a **Dockerfile** using **cross-rs** to cross-compile the **cURL** library with its Rust bindings and dependencies like **BoringSSL** for multiple Android targets, involving debugging build failures, and backporting upstream patches

Kaldi ASR Client | *C++, Python, CMake, gRPC, NVIDIA Triton*

Dec 2022

- Developed a C++ client library to perform audio inference on WAV files with **Kaldi** and **NVIDIA Triton Inference Server**, utilizing **gRPC** for communication
- Created Python bindings with **ctypes**, addressing signal and exception handling concerns in the library interface
- Created a build pipeline to build C++ libraries like **Kaldi** from source and integrate them with Python bindings into a Python wheel, addressing build system quirks with tools like **patchelf**
- Created a daemon script for Triton server provisioning, mitigating a memory leak bug reported upstream: [#4814](#)

PERSONAL PROJECTS

Landbox | *C, Make, Linux Syscalls*

Aug 2021 - Present

- Explored the Landlock API as an alternative to User Namespaces for filesystem sandboxing by creating a CLI program inspired by **bubblewrap**, facilitating restriction of read, write and execute permissions for specified paths
- Created a reusable helper library for the Landlock API, including runtime feature checks, syscall wrappers, and constants mapping UNIX file permissions to Landlock rules

Matrix TUI | *C, Meson, cJSON, libcurl, lmbd, termbox2*

Aug 2021 - Present

- Developed a minimal TUI for the Matrix communications protocol to interact with the REST APIs laid down in the specification
- Effectively leveraged queues, pthreads, pipes, signals, and the poll() syscall to implement an asynchronous TUI
- Wrote custom TUI widgets such as scrollable input fields, tree views and message views, and organized reusable code into a sub-library, [termbox-widgets](#)
- Created [libmatrix](#), a sub-library that utilizes **libcurl** and **cJSON** to wrap the Matrix REST APIs, offering streamlined interfaces with API helpers and patterns such as tagged unions and iterators
- Implemented an efficient key-value event store with the **LMDB** database and designed clean abstractions over it to facilitate serialization of events received from the server
- Wrote unit tests, and integrated sanitizers, static analyzers and coverage reports to improve code quality & safety