Pratham K

prathamIN@proton.me | blog | @git-bruh

Technical Skills

Tools: Git, Docker, GDB, Strace, GNU Make, CMake, Meson
Languages: Bash / POSIX sh, C, C++, Python, Rust, TypeScript
Miscellaneous: CI/CD, Cross Compiling, Linux Kernel Configuration, Systems Administration

Open Source Experience

Package Maintainer and Core Team Member — 2022 – Present

KISS Linux Community

- Participated in the packaging and regular maintenance of large software packages such as Compiler Toolchains, Browsers like **Firefox** and **Chromium**, and tools like **Docker**, involving working with various build systems, and patching software to work on systems using esoteric components such as **musl libc**
- Improved stability and performance by fixing various issues in the **KISS** package manager: kiss-community/kiss
- Developed automation projects to streamline processes and enhance reproducibility: maintainer-utils

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 to **termbox2**, a terminal UI rendering library, by fixing portability-related bugs and adding parsing for mouse events: termbox/termbox2
- Wrote a detailed report regarding a bug in **Qemu**'s **virglrenderer** backend that caused Wayland applications inside the guest machine to crash, helping uncover a bug in **Mesa**: virglrenderer/#291
- Discovered a **libCURL** usage bug in **Flatpak** and **OSTree** causing crashes on specific systems: #5074, #2706

Contractual Projects

libCURL Cross Build | *C, Rust, Docker, CMake, Cargo, Android NDK* — Feb 2023

- Created a **Dockerfile** using **cross-rs** to cross-compile the **CURL** library with its Rust bindings for multiple architectures on the Android platform
- Wrote build steps for the **CURL** library and dependencies like **BoringSSL**, involving debugging build failures, backporting upstream patches, and figuring out appropriate build flags for various target architectures

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 such as hardcoded absolute RPATHs with tools like **patchelf**
- Created a daemon script for Triton server provisioning, mitigating a memory leak bug reported upstream: #4814
- Created comprehensive documentation for the complete deployment process, covering library building, Triton server configuration, and Python bindings usage

Personal Projects

Landbox | *C, Make, Linux Syscalls* — Oct 2022 - 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, lmdb, termbox2* — Aug 2021 - Present

- Developed a minimal TUI for the Matrix communications protocol to interact with the REST APIs laid down in the specification
- Implemented input handling and bidirectional inter-thread communication in an asynchronous manner by effectively leveraging queues, pthreads, pipes, signals, and the poll() syscall

- 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
- Enhanced memory safety by extensively testing with the **Unity** test framework, integrating **ASAN** and **TSAN**, and utilizing **Clang Tidy** for static analysis

Matrix Discord Bridge | *Python, Bottle, Sqlite3, Urllib3, Websockets* Nov 2020 - Present

- Developed a bridge to seamlessly share communications between Discord channels and Matrix rooms, preserving key features such as replies, emotes, and mentions
- Utilized the **Websockets** library to interact with the Discord API, managing heartbeats and disconnects within event loop constraints
- Created a webhook endpoint with **Bottle** to receive Matrix homeserver events and used **Sqlite3** for persistence

Education

Manipal University Jaipur Part Time, Remote

Bachelor of Computer Applications *2023 – Present*