

Pratham K

prathamIN@proton.me | [@git-bruh](https://github.com/git-bruh)

TECHNICAL SKILLS

Build Systems: Cargo, CMake, Meson, Autotools, Scons, Make, Waf

Developer Tools: Git, Git Bisect, Docker, Docker Compose, Podman, Github Actions, Shellcheck

Debugging Tools: GDB, Strace

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

Linux Kernel APIs: Landlock, Seccomp, User Namespaces

Systems Programming Concepts: Filesystem Permissions, Memory Management, Threads, Processes, Signals, IPC

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

OPEN SOURCE EXPERIENCE

Package Maintainer and Core Team Member

2022 – Present

KISS Linux Community

- Forked the official inactive KISS Linux main and community repositories with the help of various active community members, ensuring regular updates to the package repositories while upholding quality standards
- Helped in the packaging and regular maintenance of large software packages such as Compiler Toolchains, Browsers like **Firefox** and **Chromium**, tools like **Docker** etc. involving working with various build systems, syncing with other distributions, and patching software to be more portable, allowing usage on esoteric system configurations such as **musl libc**, pure **Clang/LLVM** toolchains, no **D-Bus** etc.
- Debugged and Communicated portability and build system related issues with upstream developers in an effort to ensure widespread availability of fixes
- Contributed various bug fixed to the shell-based **KISS** package manager, involving the **ldd**-based dynamic dependency detector and the package alternatives system: [kiss-community/kiss](https://kiss-community.github.io/kiss/)
- Provided regular support for issues relating to custom kernel configuration, package build failures and packaging/contributing new software in the [#kisslinux](https://libera.chat) IRC channel on libera.chat
- Maintained an extra soft-fork of the main repositories - **GKISS**, that uses the GNU C library, **glibc** rather than **musl** for users desiring wider software compatibility: [kiss-community/grepo](https://kiss-community.github.io/grepo/)
- Used **Github Actions** to automate **shellcheck** runs on package build scripts, participated in regular review of Pull Requests adding new packages, and performed regular follow up on Issues
- Lead the development of various infrastructure automation related projects, reducing the scope of human error and helping reproducibility: [kiss-community/maintainer-utils](https://kiss-community.github.io/maintainer-utils/)
 - * Wrote a multi-stage bootstrap script to build the rootfs tarball from scratch in a rootless fashion using **unshare** and **bubblewrap**
 - * Wrote various helper scripts for maintainers to check package versions against the latest upstream versions on **repology** and semi-automate the package maintenance process
- Developed familiarity with various projects powering the Linux ecosystem such as **Clang**, **LLVM**, **GCC**, **glibc**, **musl**, **Mesa** etc.

Open Source Contributor

2022 – Present

Various Contributions To Projects Used In Personal FOSS Endeavours

- Made various contributions to a terminal UI rendering library, **termbox**, including portability-related bug fixes, and features like ANSI escape sequence parsing for Mouse events: [termbox/termbox2](https://termbox.github.io/termbox2/)
- Wrote a detailed report including findings from git bisect regarding a bug in the **virglrenderer** backend used by **Qemu** that caused Wayland applications inside the guest machine to crash, helping uncover a bug in **Mesa**: [virglrenderer/#291](https://virglrenderer.org/#291)
- Contributed a script to the DevelopersIndia community for creating weekly job posts on the subreddit by consuming an RSS feed using Python, **PRAW**, and **FeedParser**, using a simple JSON store for persistence and integration with **Github Actions**: [deviras/#9](https://deviras.github.io/#9)
- Uncovered a small bug related to improper **libCURL** usage in **Flatpak** and **OSTree**: [flatpak/#5074](https://flatpak.org/#5074), [ostree/#2706](https://ostree.org/#2706)
- Various similar instances...

PERSONAL PROJECTS

S6 Scripts | *POSIX sh, execline, s6-rc, mdev*

Feb 2023 - Present

- Ported KISS Linux's init scripts from **busybox** **runit** to the **s6-rc** service manager for better reliability, logging and dependency management
- Wrote system initialization scripts performing various tasks such as mounting pseudo-file systems, loading kernel modules, coldplugging devices along with service definitions for daemons like **dhcpcd** with a clean and explicit dependency graph, allowing maximum parallelization in the boot process

Landbox | *C, Make, Linux Syscalls*

Oct 2022 - Present

- Developed a small sandboxing program with a CLI interface inspired by **bubblewrap** using the Linux Landlock APIs rather than User Namespaces, with the ability to restrict read, write and execute permissions for specified paths
- Implemented a reusable helper library for checking available kernel features at runtime and wrapping raw syscalls, along with helper constants correlating traditional file permissions (read, write, execute) to Landlock rules

Matrix TUI | *C, Meson, cJSON, libCURL, lmbd, termbox*

Aug 2021 - Present

- Developed a minimal TUI for the Open Source Matrix communications protocol, interacting with the REST APIs laid down in the specification
- Wrote various TUI widgets from scratch, including scrollable and auto resizing input fields, treeviews, message views etc., putting general-purpose, reusable parts of the code into a sub-library, termbox-widgets
- Wrote another sub-library, libmatrix, wrapping the REST APIs with **libCURL** and **cJSON**, providing a clean library interface with structs and enums corresponding to event types, iterators over the received events, and helpers for event creation and HTTP requests
- Designed the architecture of the application while keeping in mind the overall requirements for integrating various components, resulting in an asynchronous and multi threaded architecture
- Implemented an asynchronous UI with constraints such as handling Input while simultaneously receiving events from producer threads by effectively leveraging queues, threads, pipes, signals, and the poll() syscall
- 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
- Prevented memory safety bugs by writing extensive unit tests with the **Unity** test framework and integrating them with **ASAN** (Address Sanitizer) and **TSAN** (Thread Sanitizer)
- Used **Meson** as the build system, leveraging features like subprojects, along with **clang-format** for code formatting

Matrix Discord Bridge | *Python, Bottle, Sqlite3, Urllib3, Websockets*

Nov 2020 - Present

- Developed a simple, self-hosting friendly bridge to share communications across Discord channels and Matrix rooms with a 1:1 mapping of features like replies, emotes, and mentions
- Used the Python **Websockets** library to interact with the Discord API and handle various quirks such as heartbeats and disconnects
- Implemented a webhook endpoint with the **Bottle** framework to receive events from the Matrix home server along with **Sqlite3** for caching/persistence of relevant data
- Reviewed various code-related contributions and provided regular assistance to users with their filed Issues

CONTRACTUAL PROJECTS

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

Feb 2023

- Developed an end-to-end build pipeline for cross compiling the **CURL** library with its rust bindings to various architectures targeting the Android platform
- Used **cross-rs** to setup the **Android NDK** toolchain images serving as a base for the **Dockerfile**
- Wrote a **Dockerfile** to build the **CURL** library, along with transitive dependencies such as **Brotli**, **NGhttp2** and **BoringSSL**, involving debugging of build failures, backporting upstream patches, and figuring out appropriate build options for various targets
- Effectively leveraged **Docker layers** to minimize image size

PlayStation India Discord Bot | *Python, Flask, BeautifulSoup, Sqlite3*

Feb 2023

- Developed a bot for the PlayStation India Discord server for scraping user trophies from an external website and exposing a paginatable leaderboard on Discord for the same

- Designed a method to get authwalled information about linked external accounts from Discord with the help of Discord's **implicit OAuth2 grant flow**, implementing a basic frontend with JavaScript to POST relevant data to the backend
- Implemented a webhook endpoint with **Flask** for receiving events from Discord along with Ed25519 signature verification in accordance with Discord's API documentation
- Wrote a background worker to scrape the external service at regular intervals with **BeautifulSoup**, implementing heuristics to prevent rate limiting, along with **Sqlite3** for storing fetched stats

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

Dec 2022

- Developed a client library in C++ for performing audio inferences on WAV files with the **NVIDIA Triton Inference Server**, using **GRPC** for communication
- Wrote Python bindings for the same using the **ctypes** module and handled potential issues such as exception handling and propagation of signals such as SIGINT back to Python on the C++ side
- Developed an end-to-end build pipeline for installing build dependencies on the host, building unpackaged C++ libraries such as **Kaldi** from source using **CMake** and integrating the build artifacts along with Python bindings into a Python wheel
- Resolved various issues arising during the build process due to build system quirks such as non-relative RPATHs being set in shared libraries, preventing their relocation to other paths, resolved with tools like **patchelf**
- Wrote a simple daemon script using Netcat to provision Triton server instances on-demand in an effort to work around a memory leak bug which was extensively documented and reported upstream; [kaldi/#4814](#)
- Documented the entire end-to-end deployment process in detail including building libraries, configuring the Triton server and demonstrating the usage of Python bindings