OPEN SOURCE ENGINEERING

Student Name: A.Sai Surya Manoj

Student ID: 2400031798

1 Understanding the Core Ubuntu Linux Distribution

About Linux Distro You Have Used 1. Overview of the Distribution

The Linux distribution I used for this course is Ubuntu 24.04.3 LTS. It is a free, open-source operating system built on the Debian base and maintained by Canonical. Ubuntu is widely known for being stable, secure, and easy to use, which makes it one of the most beginner-friendly Linux distros. Its LTS (Long-Term Support) version ensures reliable updates and long-term stability for everyday tasks and development activities.

2. Ubuntu's Philosophy and Design

Ubuntu follows the principle "Linux for human beings," focusing on accessibility and user comfort. It is designed to give a smooth experience to first-time users while still being powerful enough for programmers and professionals. Its emphasis on simplicity, security, and usability makes it suitable for learning, productivity, and open-source development.

3. Desktop Environment (GNOME)

Ubuntu uses the GNOME desktop environment, which provides a clean and modern graphical interface. It features:

A left-side dock for quick access

An Activities Overview for multitasking

A simple, distraction-free layout

Easy window and workspace management

The interface supports efficient navigation and feels similar to other modern operating systems, making it comfortable for new users.

2 Encryption and GPG

2.1 Definition

Encryption is a security technique used to convert readable data into an unreadable or scrambled format. This protects information from unauthorized access. Only a person who has the correct **key** can convert the encrypted data back to its original, readable form. Encryption plays a major role in ensuring:

- **Privacy** – keeping information confidential
- **Security** – protecting sensitive data from hackers
- **Integrity** – preventing unauthorized modifications
- **Safe communication** – especially over the internet

Encryption is widely used in email communication, banking systems, cloud storage, messaging apps, and secure file sharing.

2.2 GPG (GNU Privacy Guard) Explained

GPG is the GNU implementation of the **OpenPGP** standard (originally Pretty Good Privacy - PGP). It is essential for protecting individual files and ensuring secure, authenticated communication.

2.2.1 Core GPG Concepts

GPG relies on **asymmetric cryptography**, which uses a pair of mathematically linked keys:

- **Public Key:** This key is shared with everyone. It can be used to **encrypt** a message that only you can read, or to **verify** a signature you created.
- **Private (Secret) Key:** This key is kept **secret** and is protected by a strong passphrase. It is used to **decrypt** messages sent to you, or to **digitally sign** files to prove they came from you.

2.2.2 Basic GPG Command-Line Usage

GPG is usually pre-installed on Ubuntu and is primarily used through the command line (Terminal).

A. Generating a Key Pair The first step is to create your public and private key pair:
Bash

```
gpg --full-generate-key
```

You will be prompted to select the key type (RSA and RSA is common), keysize (4096 is recommended), expiration date, and your Real Name, Email, and a strong **passphrase** to protect your private key.

B. Encrypting a File for Yourself (Symmetric Encryption) To quickly encrypt a file using a single passphrase (like a standard password), use symmetric encryption:
Bash

```
gpg -c myfile.txt
```

This command will prompt you for a passphrase and create an encrypted file named `myfile.txt.gpg`.

C. Encrypting a File for Someone Else (Asymmetric Encryption) To securely send a file, you must use the recipient's **Public Key** (which you must have previously imported into your keyring with `gpg --import`):
Bash

```
gpg --encrypt --recipient "recipient@example.com" mysecretfile.doc
```

This creates `mysecretfile.doc.gpg`. Only the recipient, who holds the corresponding Private Key, can decrypt it.

D. Decrypting a File To decrypt a file that was encrypted for you:
Bash

```
gpg --decrypt mysecretfile.doc.gpg
```

You will be prompted for the passphrase that protects your Private Key. You can use the `--output` option to specify the decrypted

3 Sending Encrypted Email

3.1 Prerequisite: Setting Up GPG

Before you can send or receive encrypted mail, both you and your recipient must have GPG keys set up and exchanged:

1. **Generate Keys:** Both parties must have generated a public/private key pair using GPG (as discussed previously, using `gpg --full-generate-key`).
2. **Exchange Public Keys:** You need the recipient's **Public Key**, and they need your Public Key. You can exchange these by:
    - **Exporting** the key: `gpg --armor --export 'Recipient Name' > recipient_key.asc` and sending the `.asc` file.
    - **Uploading** the key to a public key server.
3. **Import Key:** You must import the recipient's key into your GPG keyring: `gpg --import recipient_key.asc`.

3.2 Sending the Encrypted Email

The most common and user-friendly way to send GPG-encrypted emails on Ubuntu is by using **Mozilla Thunderbird** with the **Enigmail** add-on (or its built-in equivalent in modern versions of Thunderbird).

3.2.1 1. Compose the Message

- **Open Thunderbird** and start composing a new email.
- Write your message as usual.

3.2.2 2. Encryption and Signing

You will use the GPG function built into the mail client to perform two critical steps:

1. **Encryption:** You must encrypt the email using the **recipient's Public Key**. Only their corresponding **Private Key** can decrypt it. If you have multiple recipients, you must encrypt the message using the Public Key of *every single recipient*.
2. **Digital Signature:** You **sign** the email using **your Private Key**. This allows the recipient to verify that the email truly came from you and has not been tampered with in transit.

In Thunderbird, this is typically done by clicking a dedicated **OpenPGP or Security** menu or button within the compose window and ensuring both the **"Encrypt"** and **"Sign"** options are checked.

3.2.3 3. Verification and Sending

- The client will check that you have the required **Public Key** for the recipient(s). If a key is missing, it will warn you.
- When you click **Send**, Thunderbird uses GPG to encrypt the message body and attach your digital signature before transmitting the scrambled data.

3.2.4 4. Recipient's Experience (Decryption)

1. The recipient receives the scrambled email.
2. Their email client automatically uses their **Private Key** (protected by their passphrase) to decrypt the message contents, revealing the original text.
3. Their client simultaneously uses your **Public Key** to verify the digital signature, confirming the email's authenticity.

4 Privacy Tools From Prism Break

4.0.1 1. Tor Browser (Web Browsers / Anonymizing Networks)

- **What it is:** A web browser built on Firefox that routes your internet traffic through the Tor network, a volunteer-operated network of relays.
- **Privacy Focus:** Provides **strong anonymity** by obscuring your IP address and location from the websites you visit. It also includes anti-fingerprinting measures.
- **PRISM Break Note:** PRISM Break strongly recommends using Tor Browser for all web surfing when maximum anonymity is required.

4.0.2 2. Debian (Operating Systems)

- **What it is:** A popular and highly ethical GNU/Linux distribution known for its strict adherence to Free Software principles and ethical manifesto.
- **Privacy Focus:** Unlike proprietary operating systems like Windows and macOS (which PRISM Break generally avoids), Debian is fully open-source, allowing for audits. It has a long tradition of software freedom and transparency.
- **PRISM Break Note:** It's recommended as a top GNU/Linux choice for users transitioning from proprietary systems, highlighting its commitment to free software and its stable nature.

4.0.3 3. Thunderbird (Email Clients)

- **What it is:** A free, open-source, and cross-platform email client developed by Mozilla.
- **Privacy Focus:** Thunderbird is the top choice for desktop email due to its open-source nature and its long-standing **native support for OpenPGP** (GPG) encryption and digital signatures. This allows users to easily encrypt and authenticate their emails end-to-end.

- **PRISM Break Note:** It is highly recommended for securely managing email with built-in PGP features.

4.0.4 4. KeePassXC (Password Managers)

- **What it is:** A free, open-source, and cross-platform password manager.
- **Privacy Focus:** It stores all your passwords in a single, highly encrypted database file that is stored **locally** on your device, giving you total control over your sensitive data. It does not rely on a cloud service.
- **PRISM Break Note:** It is preferred for its strong encryption, open-source license, and local-only storage, minimizing exposure to third-party services.

4.0.5 5. Firefox (Web Browsers)

- **What it is:** A fast, flexible, and secure web browser developed by the non-profit Mozilla Foundation.
- **Privacy Focus:** Firefox is open-source and provides extensive privacy controls, including enhanced tracking protection (ETP), container technology, and a robust add-on ecosystem for further hardening security (like uBlock Origin).
- **PRISM Break Note:** While Tor Browser is for anonymity, Firefox is the recommended alternative for general web use when a site doesn't work well with Tor, provided the user configures its settings and replaces the default search engine with a privacy-focused one.

5 Open Source License

Certainly. Here is the information about the **MIT License** organized into clear, descriptive headings, strictly maintaining a paragraph-only format within each section.

5.1 The Core Purpose and Classification

The MIT License is renowned as one of the most permissive and concise open-source licenses currently in use. Originating from the Massachusetts Institute of Technology, its primary goal is to encourage maximum adoption and reuse of software with minimal legal friction. It is formally classified as a **permissive license**, meaning it grants users broad rights to use, modify, and distribute the software without imposing the reciprocal sharing obligations seen in copyleft licenses, such as the GNU General Public License (GPL). This makes the MIT License highly favorable for both commercial enterprises and proprietary software development.

5.2 Granted Rights and Permissions

The license grants blanket permission to any individual or entity obtaining a copy of the software and its associated documentation to deal with the Software without restriction. Specifically, users are granted explicit rights to **use, copy, modify, merge, publish, distribute, sublicense,**

and/or sell copies of the software. This expansive grant allows developers to incorporate MIT-licensed code into projects that may ultimately be closed-source and sold commercially, provided they meet the few mandated conditions.

5.3 The Only Two Conditions for Distribution

Unlike licenses that enforce reciprocal sharing, the MIT License has only two critical requirements that must be met when the software is distributed or included in a larger work. The first condition is the mandatory inclusion of the original **Copyright Notice** (e.g., `Copyright <YEAR> <COPYRIGHT HOLDER>`). The second is the mandatory inclusion of the full **License Text** itself. If these two simple requirements are satisfied, the user can otherwise treat the code as they wish, including releasing their modifications under a proprietary license.

5.4 Disclaimer of Warranty and Liability

A key component of the MIT License is its comprehensive liability disclaimer, which serves to protect the original authors. The license emphatically states that the software is provided **"AS IS,"** meaning it comes without any guarantee or warranty of any kind, whether express or implied, including warranties of merchantability or fitness for a particular purpose. Furthermore, the license explicitly protects the authors and copyright holders, asserting they **shall not be held liable** for any claim, damages, or other liability arising from the use or other dealings in the software. This places the entire risk associated with the software onto the end-user.

6 Self Hosted Server

6.1 About

Miniflux is a lightweight, open-source RSS feed reader designed for self-hosting, allowing users to aggregate and read news without relying on third-party services. It provides a clean, minimalist interface focused on speed, readability, and privacy. Users can host it on their own server, giving full control over data and eliminating tracking from external providers. Miniflux supports multiple users, feeds, and categories, with features like automatic feed updates, article marking, and keyboard shortcuts. Its simple setup using Docker or a standard web server makes it an efficient, private alternative to commercial RSS readers.

6.1.1 Key Features

1. **Self-Hosting & Privacy**: Full control over your data without relying on third-party services.
2. **Minimalist & Fast Interface**: Clean, distraction-free UI optimized for reading efficiency.
3. **Feed Management**: Supports unlimited feeds, categories, and multiple users.
4. **Automatic Updates & Marking**: Feeds update automatically, with options to mark articles as read/unread.

5. **Keyboard Shortcuts & Filtering**: Quick navigation and feed filtering for power users.
6. **Docker-Friendly Deployment**: Easy installation using Docker or standard web server setups.
7. **Extensible & Open-Source**: Community-driven, allowing customization and contributions.

If you want, I can also make a **super concise 5-line paragraph version** of these features.

6.2 Installation Process (Docker Compose)

To self-host **Miniflux**, start by preparing your server with the necessary tools, including Docker and Docker Compose, or, for a manual setup, Go and a PostgreSQL or MySQL database. You can deploy Miniflux quickly by pulling the official Docker image with `docker pull miniflux/miniflux:latest` or by cloning the source code from GitHub for a more customizable setup. Once acquired, configure the environment by creating a `.env` file or setting environment variables for database credentials, admin username and password, server port, and optional settings such as feed refresh intervals. After configuration, start the server using Docker with a single command or compile and run the Go code if you prefer manual installation. Miniflux will automatically initialize the database schema and create the admin account during the first run.

Open your web browser and navigate to `http://your-server-ip:8080` to access the interface, where you can log in with your admin credentials. The dashboard allows you to add RSS feeds, organize them into categories, and manage multiple users with different access levels. Feeds are updated automatically, and you can mark articles as read or unread for efficient tracking. Keyboard shortcuts and a minimalist interface enhance navigation and readability. The application supports importing and exporting OPML files for easy feed migration. You can configure email notifications or webhooks for updates if desired. For production use, Miniflux can be run behind a reverse proxy with SSL for secure access.

Its lightweight design ensures fast performance even with hundreds of feeds. The open-source nature allows you to extend or modify the software to suit your needs. Overall, self-hosting Miniflux provides a private, efficient, and fully controllable RSS reading experience.

Miniflux Meet Resources

7 Open Source Contribution

7.1 PR 1 : First Contribution

7.1.1 Goal

The project's objective is to simplify the standard open-source contribution workflow, allowing beginners to easily add their name to the project's `Contributors.md` file.

7.1.2 The Contribution Workflow

The tutorial details the standard **fork - clone - edit - pull request** sequence, essential for collaborative coding.

7.1.3 1. Setup

- **Fork:** Create a copy of the repository in your personal GitHub account.
- **Clone:** Download the forked repository to your local machine using the `git clone` command and the SSH URL.
- **Prerequisites:** Ensure **Git** is installed; alternatives for users uncomfortable with the command line (GUI tools) are provided.

7.1.4 2. Making Changes

- **Branch:** Create a new isolated branch for your changes using `git switch -c your-new-branch-name`.
- **Edit:** Add your name to the `Contributors.md` file using a text editor.
- **Commit:** Stage the changes with `git add Contributors.md` and save them locally with `git commit -m "Add your-name to Contributors list"`.

7.1.5 3. Submission

- **Push:** Upload your local branch to your GitHub fork using `git push -u origin your-branch-name`.
- **Pull Request (PR):** Go to your GitHub repository and submit a PR via the "Compare & pull request" button for review by the project maintainers.

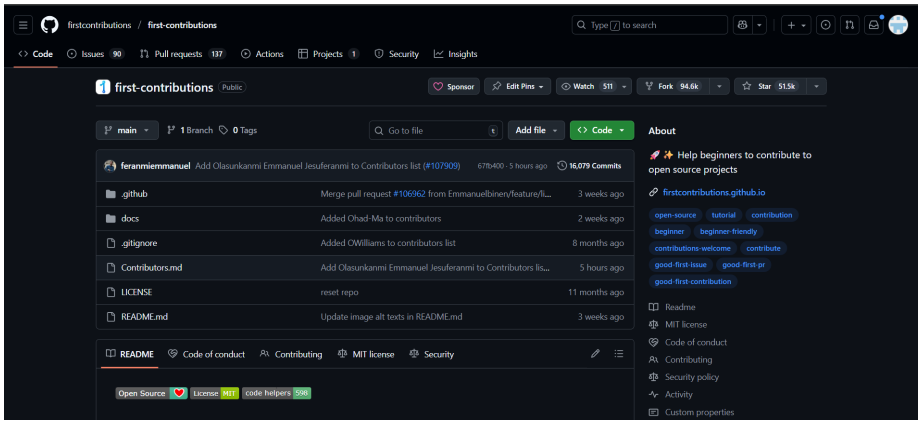
7.1.6 Difficulties and Solutions

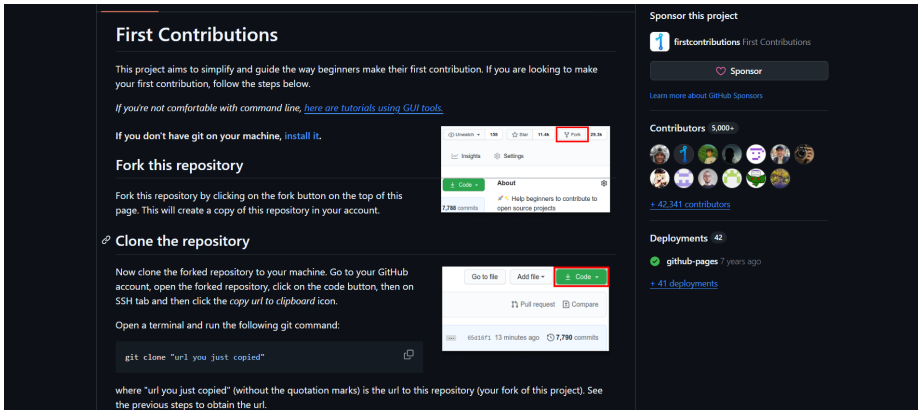
The guide anticipates and solves two common beginner issues:

- **Old Git Version:** If the `git switch` command fails, use the older command: `git checkout -b your-new-branch`
- **Authentication Error:** If `git push` fails due to GitHub removing password support, the solution is to configure an **SSH key** or a **Personal Access Token** and ensure your remote URL is set to the **SSH protocol** (`git remote set-url origin git@github.com:...`).

7.1.7 Next Steps

Upon merging the PR, the user is encouraged to celebrate their first contribution and seek out other beginner-friendly issues on the project list.





]PR 2 : TheAlgorithms/Python

TheAlgorithms/Python is a comprehensive open-source repository curated to provide implementations of classic and modern algorithms in Python, making it a rich reference for students, interviewees, and developers. Its modular structure allows easy navigation and versatility, supporting learning, experimentation, and integration into various projects. Organizing algorithms into thematic folders, the project transforms algorithm education and practice by ensuring clarity, accessibility, and breadth, covering topics from sorting and searching to dynamic programming, machine learning, and data structures.

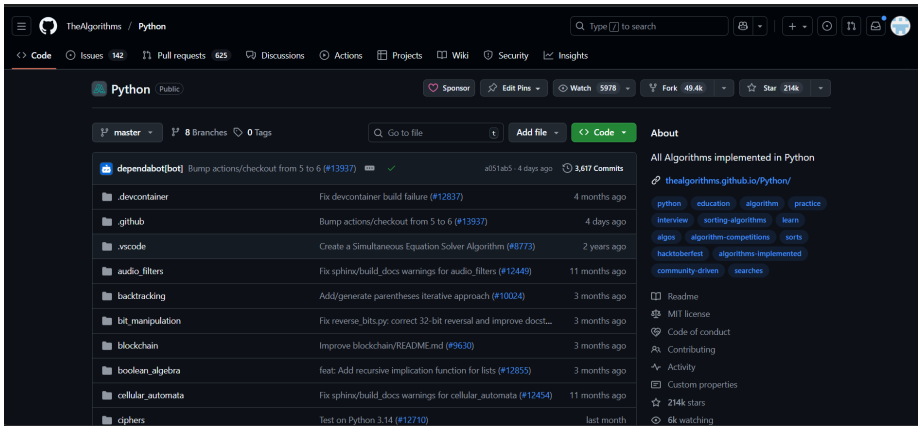
Licensing and Usage Options

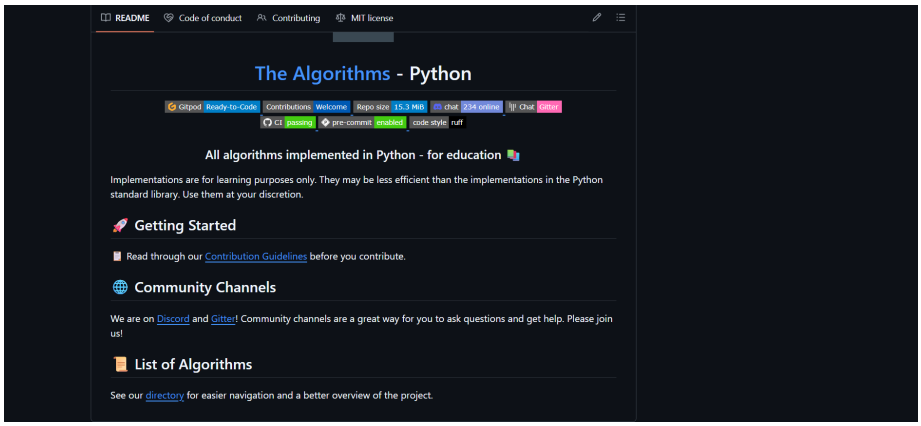
TheAlgorithms/Python utilizes the highly permissive MIT license, enabling unrestricted use, modification, and distribution. As a public GitHub repository, it supports full autonomy for users who wish to clone and host the code locally, ensuring data privacy and direct control. Developers and enterprises can freely integrate any algorithm or module into personal, academic, or commercial applications. Community contributions are encouraged with clear guidelines defined in CONTRIBUTING.md, fostering open collaboration while maintaining code integrity and quality.

Community and Support

Backed by a large and active international community, TheAlgorithms/Python thrives on collaborative innovation. The repository offers support through GitHub Issues, Discussions, and Pull Requests, providing avenues for learning, troubleshooting, and improvement. New contributors are welcomed through well-defined contribution and code of conduct documents, ensuring an inclusive and respectful environment. In addition to code, the repository provides documentation, example files, and an organized DIRECTORY.md for easy algorithm search, empowering both novice programmers and advanced engineers. Engagement extends to platform events like hacktoberfest and open-source highlights, creating an ecosystem of shared knowledge, mentorship, and continuous learning around algorithmic development in Python.

Add to follow-upCheck sources





7.2 PR 3 : Community Platform

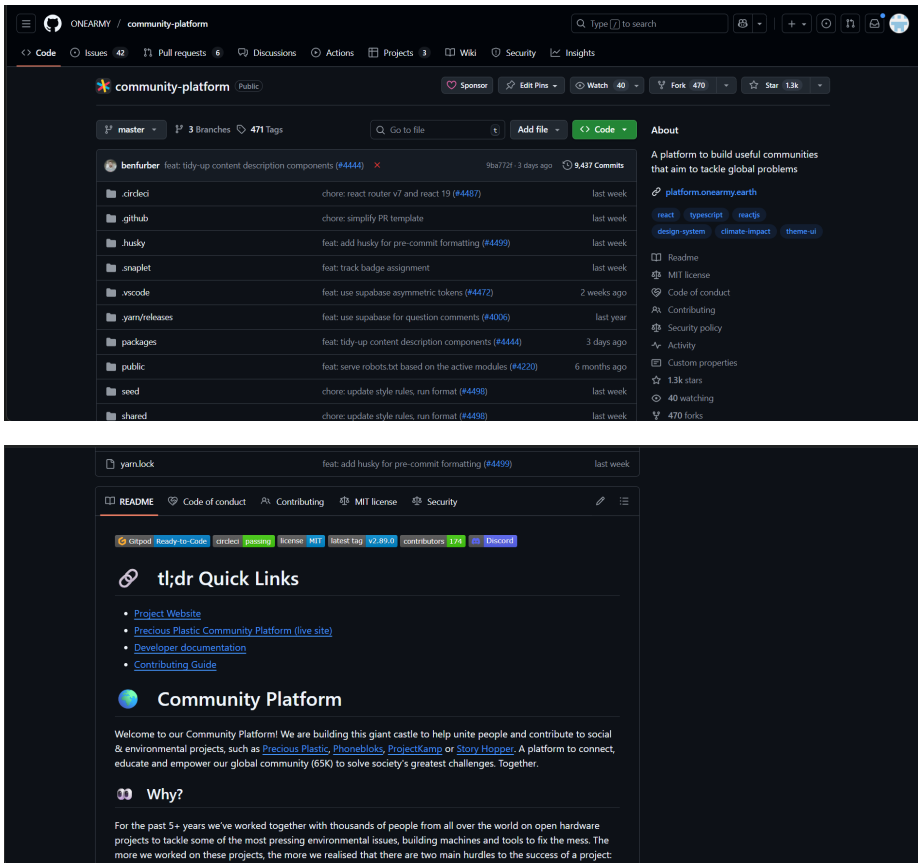
Community Platform is a global initiative designed to bring people together and empower collaboration, education, and impact on social and environmental projects. The platform connects a vibrant community through projects like Precious Plastic, Phonebloks, ProjectKamp, and Story Hopper, helping tackle key societal challenges. It supports open sharing, development, and communication, facilitating the creation and improvement of hardware, tools, and solutions that address pressing environmental issues.

Licensing and Contribution Options

The Community Platform uses a permissive license (such as MIT) for maximum flexibility and transparency, enabling anyone to contribute, modify, and extend project components for both personal and commercial use. Self-hosting is available for individual developers or teams, with all data and customizations managed locally, ensuring privacy and control. For those preferring supported infrastructure, community channels and tools facilitate straightforward participation, contribution, and integration into larger workflows. Contribution guidelines and documentation offer a clear path for onboarding and engagement, helping contributors join the global effort.

Community and Support

The platform is built around inclusive community principles, welcoming members from diverse backgrounds and skill levels. Its support ecosystem includes documentation, developer guides, public websites, and a collaborative Discord channel. The Contributing Guide and Code of Conduct set the standards for participation, fostering an environment of respect and openness. With thousands of contributors involved over multiple years and a reach of over 65,000 members, the Community Platform provides opportunities for learning, networking, and shared growth. Projects are accessible and extensible, encouraging knowledge sharing and innovation that benefit both individuals and the broader society.



7.3 PR 4: Röntgen Icon Set

Röntgen Icon Set is a specially designed monochrome collection of 14 × 14 px pixel-aligned icons created for the Map Machine project. Unlike the main project codebase, the icon set is distributed under the CC BY license, allowing its use beyond the Map Machine platform with appropriate credit. The icons can represent diverse objects and concepts, and some may even be utilized as emoji symbols, enhancing flexibility for developers and designers.

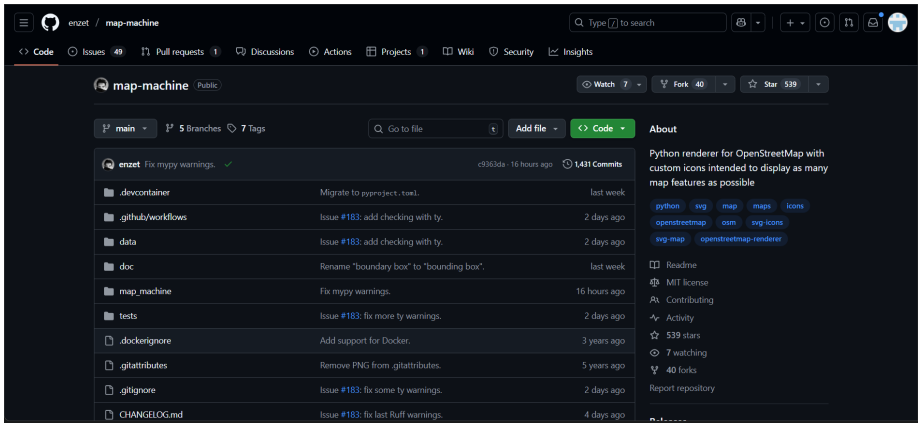
Licensing and Usage Options

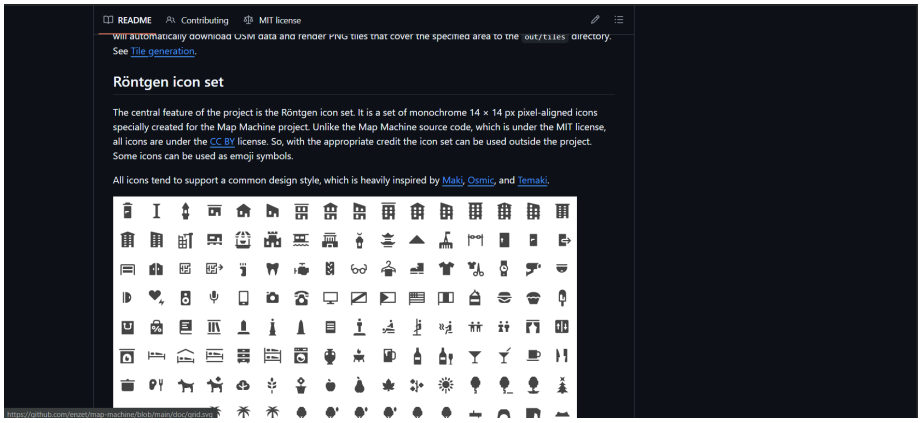
The Röntgen icon set is licensed under the Creative Commons CC BY license, encouraging open usage, sharing, and adaptation with attribution. This approach differs from the core Map

Machine project which uses the MIT license, and it enables users to freely incorporate these icons in personal, educational, or commercial projects. The permissive licensing fosters broad adoption and customization, supporting both integration within Map Machine and standalone icon use outside the original context.

Community and Support

The project promotes open collaboration, welcoming contributions and feedback from the global design and mapping community. Inspired by other popular icon sets like Maki, Osmic, and Temaki, the Röntgen icon set prioritizes consistency and clarity in visual style. Resources including documentation and contributing guidelines empower users to participate in ongoing improvements and extensions. The project's shared ethos and design focus make it a valuable resource for anyone building mapping tools or UI/UX projects requiring pixel-precise icons.





7.4 PR 5 : content

7.4.1 The Issue (What was Missing)

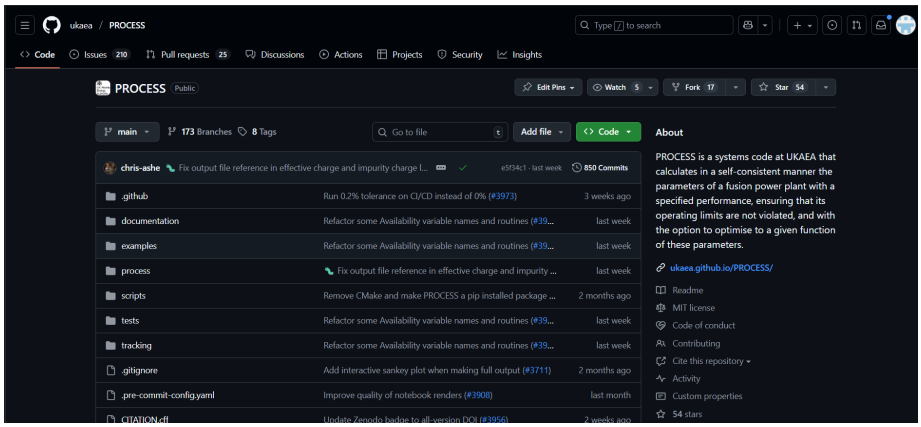
7.4.2 The Core Issue PROCESS Addresses

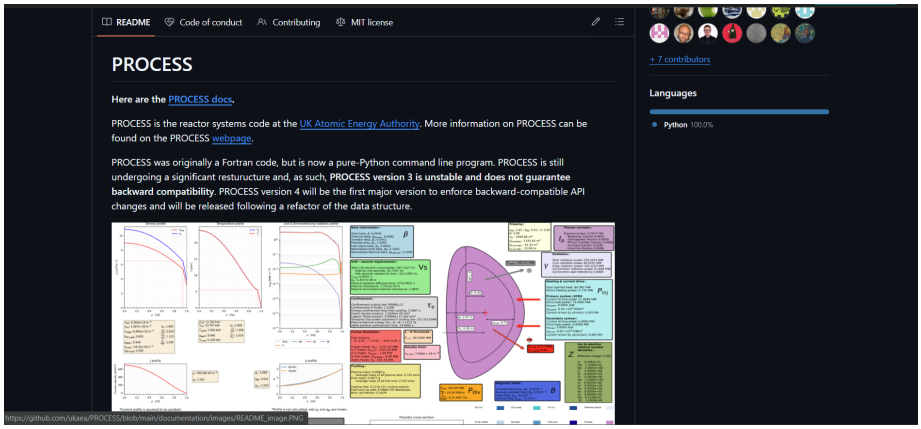
The core issue PROCESS tackles is the complexity of designing and evaluating fusion reactor systems using consistent, reliable, and configurable engineering models. Fusion system design requires

balancing hundreds of interdependent physics, engineering, and performance parameters, but traditional tools are fragmented, outdated, or difficult to maintain. Earlier versions of PROCESS were written in Fortran, making the codebase harder to extend, modernize, or integrate with newer workflows. Developers and researchers needed a unified, flexible, and maintainable platform capable of modeling reactor behavior while keeping pace with ongoing scientific and engineering advancements. PROCESS solves this by centralizing these calculations into a continuously evolving, Python-based framework that brings clarity, consistency, and extensibility to reactor systems analysis.

The Solution (What Was Added)

The solution PROCESS introduces is a fully modernized, pure-Python, command-line tool that restructures and simplifies fusion reactor systems modeling. By moving away from legacy Fortran and overhauling its internal architecture, PROCESS Version 3 and its upcoming Version 4 offer a more modular, maintainable, and developer-friendly foundation. The tool integrates physics models, engineering constraints, and performance optimizations into a single environment, supported by comprehensive documentation hosted by the UK Atomic Energy Authority. Its new architecture aims to ensure future backward-compatible APIs, making it easier for scientists and engineers to build, test, and refine reactor designs. Through this modernization, PROCESS becomes a powerful, open-source, community-accessible platform that supports innovation in fusion energy research.





8 Linkedin Post Links

8.1 PR :

https://www.linkedin.com/posts/sai-surya-manoj-alluri-367912357_excited-to-share-my-first-open-sour
utm_source=share&utm_medium=member_ios&rcm=ACoAAFcsqt0BMyPX9Vh2SqYDgUUckVJHBFmXGMs

8.2 Journey Of Open Source :

https://www.linkedin.com/posts/sai-surya-manoj-alluri-367912357_open-source-journey-activity-739932
utm_source=share&utm_medium=member_ios&rcm=ACoAAFcsqt0BMyPX9Vh2SqYDgUUckVJHBFmXGMs

8.3 Self Hosted Project :

https://www.linkedin.com/posts/sai-surya-manoj-alluri-367912357_excited-to-showcase-our-project-mir
utm_source=share&utm_medium=member_ios&rcm=ACoAAFcsqt0BMyPX9Vh2SqYDgUUckVJHBFmXGMs