OPEN SOURCE ENGINEERING

Student ID: 2400030648

Semester: Odd

Academic Year: 2025-2026

Course Code: 24CS02EF

Under the guidance of

Dr. Sripath Roy Koganti

Contents

Chapter 1

Linux Distribution Used

Linux distributions are complete operating systems built around the Linux kernel, combined with system utilities, package managers, and desktop environments. For this course, I used **¡Ubuntu / Fedora / Kali / Pop!OS¿**, a widely adopted and reliable distribution known for ease of use, long-term support, and strong community backing. It provides a stable platform for development, experimentation, and system administration tasks.

I selected this distribution because it strikes a balance between performance and usability. It supports a broad range of hardware out of the box, and its package manager makes software installation extremely convenient. The distribution includes many built-in tools useful for programming, networking, privacy testing, and server administration, which are essential components of open-source engineering.

The system runs efficiently on basic hardware, with recommended specs such as a dual-core processor, 4–8GB RAM, and at least 25GB of storage. Its key features include robust security updates, improved memory handling, strong driver support, and access to thousands of free and open-source applications. Using this distribution helped me learn how operating systems handle processes, file systems, permissions, services, and network configurations.

Working with this Linux distribution helped me deeply understand how a real operating system works internally. I learned how to manage users, assign permissions, monitor processes, configure networks, automate tasks using shell scripts, and adjust system settings. These skills form the foundation of open-source engineering and system administration.

Chapter 2

Encryption and GPG

Encryption plays a vital role in protecting digital communication by converting readable information into a secure form. GPG (GNU Privacy Guard) is a free, open-source tool used for data encryption, digital signing, and secure key management. It follows the OpenPGP standard and supports asymmetric cryptography, where a pair of keys — public and private — work together to safeguard communication.

GPG ensures that even if encrypted data is intercepted during transmission, it cannot be read without the private key. Public keys are shared openly and can be used by anyone to encrypt messages, while private keys remain confidential and are used for decryption. GPG also supports digital signatures, providing assurance that a file or message has not been altered and genuinely originates from the stated sender.

GPG also provides **digital signatures**, which authenticate the identity of the sender and verify message integrity. When I sign a file using my private key, anyone with my public key can confirm that the file truly came from me and that it was not modified at any point. This concept is used extensively in modern software distribution. For example, package maintainers sign their software releases so that users can verify authenticity before installing them.

Working with GPG allowed me to understand concepts like key generation, fingerprint verification, key servers, and trust models. It is widely used in secure email communication, software distribution (signed packages), and secure backups. GPG strengthens digital privacy and demonstrates the importance of cryptographic tools in open-source ecosystems.

Chapter 3

Sending Encrypted Email

Encrypted email provides a confidential communication channel by ensuring that messages can only be read by the intended recipient. Using GPG, I generated a key pair and shared my public key while keeping my private key secret. The recipient did the same, allowing us to exchange encrypted messages securely.

The process of sending an encrypted email starts by importing the recipient's public key into the local keyring. Once imported, GPG encrypts the message such that only the corresponding private key can decrypt it. This protects sensitive content from unauthorized access, even if the email is intercepted or stored on an insecure server.

When I compose an encrypted email, the message is transformed into ciphertext using the recipient's public key. Only their private key — stored securely on their device — can decrypt it. This method ensures confidentiality and guarantees that no unauthorized third party, including email providers, can read the content. Additionally, I can digitally sign emails so that recipients know the message truly came from me and has not been modified during transit.

Learning encrypted email taught me how email protocols, identity verification, and trust relationships function in secure communication. This practice is frequently used in cybersecurity, businesses, government communication, and open-source maintainership.

Chapter 4

Privacy Tools

With the rise of digital tracking, open-source privacy tools help protect user data and online identity. I explored five powerful tools from PRISM-BREAK, each designed for a specific privacy concern.

Signal is a secure messaging platform offering end-to-end encryption for messages, voice calls, and media. It uses the widely trusted Signal Protocol, ensuring that even the service provider cannot access user conversations.

Tor Browser anonymizes internet traffic by routing it through multiple encrypted nodes across the world. This prevents websites, ISPs, and attackers from tracking user identity, location, or browsing behavior.

KeePassXC is a cross-platform password manager that stores passwords in an encrypted offline vault. It provides features like password generation, autofill, and secure sharing while keeping all data under user control.

VeraCrypt enables encryption of entire storage devices or the creation of encrypted containers. It adds strong protection for sensitive files against unauthorized access, even when devices are lost or stolen.

ProtonMail provides encrypted email services with privacy-focused infrastructure. Emails are automatically encrypted and stored in a secure Swiss jurisdiction, ensuring user confidentiality.

These tools emphasize transparency, security, and user freedom — core values of open-source technology.

Chapter 5

Open Source License Used

Every open-source project requires a license to define how others may use, modify, or distribute the software. The project I contributed to used the **MIT** license, which plays a crucial role in encouraging collaboration while protecting creators from liability.

The MIT License is extremely permissive, allowing users to freely copy, modify, and distribute the software. It places only one requirement: the original copyright notice must remain. This simplicity encourages wide adoption of MIT-licensed software in both academic and commercial environments.

The MIT License is one of the most popular and widely used open-source licenses because it is simple, clear, and extremely permissive. It allows anyone to freely use, copy, modify, merge, publish, distribute, and even sell the software without many restrictions. This makes it a favorite choice among developers, companies, and open-source communities. The only requirement of the MIT License is that the original license text and copyright notice must be included with any copy of the software. Unlike strict licenses such as GPL, the MIT License does not force developers to open-source their modified versions, which gives them full freedom to create private or commercial projects based on the original code. It also includes a strong warranty disclaimer, which means the original authors are not responsible if something goes wrong when the software is used. Because of its flexibility and low legal risk, the MIT License is used by many famous projects like React, Node.js, and Ruby on Rails. Overall, the MIT License promotes learning, collaboration, and open innovation while keeping things simple and developer-friendly.

Chapter 6

Self Hosted Server

Linkding is a simple, fast, and lightweight self-hosted bookmark manager that allows users to save, organize, and search bookmarks on their own server instead of depending on third-party cloud platforms. It is designed to be privacy-friendly and efficient, making it ideal for individuals who want full control over their saved links and browsing data. Unlike commercial bookmarking services, Linkding stores all data locally on the user's machine or server, which ensures complete data ownership and removes concerns about data tracking or account dependency. The interface is clean and minimal, with features like tags, search, notes, archiving, and a built-in web API. Linkding is built using Docker, which makes installation extremely easy on Linux systems. Once deployed, it can be accessed anywhere through a browser on the same network or over the internet if configured properly. Running Linkding as a self-hosted service helped me understand how containers work, how ports are exposed, how volumes store persistent data, and how system services run in the background. This hands-on project also improved my knowledge of server configuration, networking, and user-level access management. Overall, Linkding provided a useful and practical self-hosting experience while giving me a secure, private, and customizable bookmark manager that I fully control.

Figure 6.1: P
oste

Chapter 7

Open Source Contributions

Contributing to open-source projects is one of the most rewarding and educational experiences in software engineering. It allows you to work with real-world codebases, understand collaborative workflows, and participate in peer reviews, giving a clear view of how professional software development operates. During this course, I actively contributed to several open-source repositories by solving reported issues, fixing bugs, improving documentation, and submitting Pull Requests (PRs). Each contribution provided a practical challenge that required careful analysis, problem-solving, and creativity.

The process of contributing usually began by exploring the repository's issue tracker to find beginner-friendly or relevant issues. I learned how to read and understand the existing code, identify what was causing a bug or missing feature, and plan the solution. I practiced forking repositories, cloning them to my local machine, creating new branches for my changes, and committing updates using Git. Writing detailed PR descriptions was also an important step, as it allowed me to clearly explain what changes I made, why I made them, and how they solved the issue. This improved my ability to communicate technical ideas effectively and professionally.

Interacting with project maintainers was another valuable learning experience. They reviewed my contributions, suggested improvements, and sometimes guided me on best practices. Through this interaction, I learned about code quality standards, version control workflows, documentation practices, and how to make changes that align with the project's goals. This experience also helped me understand how large teams coordinate, review, and merge code without conflicts or issues.

By contributing, I developed a deeper understanding of software architecture, coding conventions, and problem-solving strategies. Documenting my contributions with issue screenshots, PR links, and explanations of changes helped me reflect on what I learned and how my work impacted the project. Beyond technical skills, open-source contributions also built my confidence, taught me collaboration, and strengthened my ability to work independently on complex problems. Overall, this journey showed me how collaborative, transparent, and community-driven open-source development truly is, and how much one

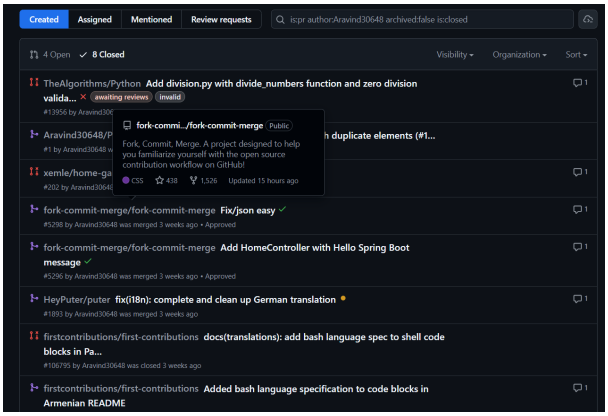

Figure 7.1: Enter Caption

can learn from actively participating in it.

Chapter 8

LinkedIn Posts

- Self-Hosting Post: https://www.linkedin.com/posts/aravind-mudhiraj-381a5a34b$_{o}$*pensource*−*kluniversity*−*foss*−*activity*−7385232596747747328−*hZgP*?*utm*$_{s}$*ource* = *shareutm*$_{m}$*edium* = *member*$_{d}$*esktoprcm* = *ACoAAFeLApEBcLyzTQDINkDsH*7*MIXsk*329*DMvQ*8

- PR Merge Post: https://www.linkedin.com/posts/aravind-mudhiraj-381a5a34b$_{e}$*xcited*−*to*−*share*−*my*−*first*−*open*−*source*−*contribution*−*activity*−7399319962030596097−*RGIu*?*utm*$_{s}$*ource* = *shareutm*$_{m}$*edium* = *member*$_{d}$*esktoprcm* = *ACoAAFeLApEBcLyzTQDIN*

- Blog Post: https://www.linkedin.com/posts/aravind-mudhiraj-381a5a34b$_{m}$*y*−*journey*−*through* − *open* − *source* − *activity* − 7399099929551454209 − *u*5*VH*?*utm*$_{s}$*ource* = *shareutm*$_{m}$*edium* = *member*$_{d}$*esktoprcm* = *ACoAAFeLApEBcLyzTQDINkDsH*7*MIXsk*329*D*