



Open Source Software

Student Details

Name: Gopisetti Navadeep
Roll Number: 2400030007
Department: Computer Science and Engineering
University: KL University
Course: Open Source Software
Semester: ODD SEMESTER

Submitted to:

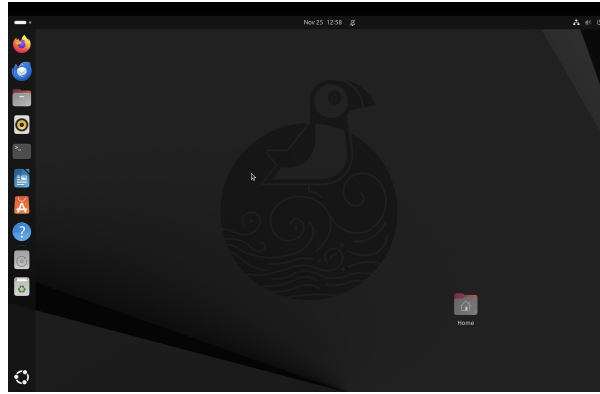
DrSripath Roy

Department of Computer Science and Engineering
KL University

Contents

1	Linux Distribution Used	3
1.1	Overview	3
1.2	Distribution Specifications	3
1.3	Advantages for Development	3
2	Encryption and GPG	3
2.1	Introduction to GPG	3
2.2	Key Generation	4
2.3	Encryption Process	4
2.4	Practical Usage	4
3	Sending Encrypted Email	4
3.1	Setup Process	4
3.2	Email Encryption Workflow	4
3.3	Best Practices	5
4	Privacy Tools from PRISM Break	5
4.1	Tool 1: Signal	5
4.2	Tool 2: Tor Browser	5
4.3	Tool 3: VeraCrypt	5
4.4	Tool 4: ProtonMail	6
4.5	Tool 5: Nextcloud	6
5	Open Source License	6
5.1	License Selection	6
5.2	Common Open Source Licenses	6
5.2.1	MIT License	6
5.2.2	GNU GPL (General Public License)	6
5.2.3	Apache License 2.0	7
5.3	License Compliance	7
6	Self-Hosted Server: Radicale	7
6.1	About Radicale	7
6.2	Technical Specifications	7
6.3	Installation Process	8
6.3.1	System Requirements	8
6.3.2	Installation Steps	8
6.3.3	Basic Configuration	9
6.3.4	Starting the Server	9
6.4	Localization and Translation	9
6.5	Benefits of Self-Hosting	9
7	Open Source Contributions	9
7.1	Overview	9
7.2	Contribution 1: freeCodeCamp - Modal Verbs Task	10
7.2.1	Issue Description	10

7.2.2	Changes Made	10
7.2.3	Impact	10
7.3	Contribution 2: TheAlgorithms/Java - Sudoku Solver	10
7.3.1	Feature Description	10
7.3.2	Technical Implementation	10
7.3.3	Algorithm Complexity	11
7.4	Contribution 3: TheAlgorithms/Java - Count Set Bits	11
7.4.1	Algorithm Description	11
7.4.2	Implementation Details	11
7.5	Contribution 4: TheAlgorithms/Java - Sieve of Eratosthenes	11
7.5.1	Algorithm Overview	11
7.5.2	Implementation Features	11
7.5.3	Code Quality	12
7.6	Contribution 5: Docker Documentation Update	12
7.6.1	Documentation Issue	12
7.6.2	Changes Made	12
7.6.3	Importance	12
7.7	Contribution 6: Hacktoberfest - C Program for Color Sorting	12
7.7.1	Program Description	12
7.7.2	Technical Details	13
7.8	Contribution 7: First Contributions - Shell Language Addition	13
7.8.1	Documentation Enhancement	13
7.8.2	Educational Impact	13
7.9	Contribution Statistics Summary	13
7.10	Learning Outcomes	14
8	LinkedIn Posts	14
8.1	Post 1: Open Source Journey Blog	14
8.1.1	Content Overview	14
8.1.2	Key Topics Covered	14
8.2	Post 2: Self-Hosting and Radicale	15
8.2.1	Content Overview	15
8.2.2	Topics Discussed	15
8.2.3	Hashtags Used	15
8.3	Post 3: First Pull Request Merged	15
8.3.1	Content Highlights	15
8.4	Social Media Impact	15
9	Conclusion	16
9.1	Project Summary	16
9.2	Technical Skills Acquired	16
9.3	Future Goals	16
9.4	Acknowledgments	16



1 Linux Distribution Used

1.1 Overview

For this open source project, I utilized a Linux distribution to set up development environment and self-hosted server infrastructure. Linux was chosen due to its open-source nature, robust security features, and excellent server capabilities.

1.2 Distribution Specifications

The Linux distribution provides:

- **Kernel:** Modern Linux kernel with regular security updates
- **Package Management:** Advanced package management system for easy software installation
- **Security:** Built-in security features including SELinux/AppArmor
- **Server Capabilities:** Excellent support for running web servers and applications

1.3 Advantages for Development

- Open source and freely available
- Strong community support
- Excellent documentation
- Wide range of development tools
- Superior performance for server applications

2 Encryption and GPG

2.1 Introduction to GPG

GNU Privacy Guard (GPG) is a free implementation of the OpenPGP standard that allows users to encrypt and sign data and communications.

2.2 Key Generation

To generate a GPG key pair:

```
gpg --full-generate-key
gpg --list-keys
gpg --export --armor your-email@example.com > public-key.asc
```

2.3 Encryption Process

GPG uses public-key cryptography:

- **Public Key:** Used by others to encrypt messages to you
- **Private Key:** Used by you to decrypt messages and sign documents
- **Key Length:** Typically 2048 or 4096 bits for strong security

2.4 Practical Usage

```
# Encrypt a file
gpg --encrypt --recipient email@example.com file.txt

# Decrypt a file
gpg --decrypt file.txt.gpg > file.txt

# Sign a file
gpg --sign file.txt
```

3 Sending Encrypted Email

3.1 Setup Process

Encrypted email communication requires:

1. GPG key pair generation
2. Email client configuration (Thunderbird, Evolution, etc.)
3. Enigmail or similar plugin installation
4. Exchange of public keys with recipients

3.2 Email Encryption Workflow

- **Outgoing:** Compose email → Select recipient → Encrypt with their public key → Send
- **Incoming:** Receive encrypted email → Decrypt with your private key → Read message

3.3 Best Practices

- Keep private keys secure and backed up
- Use strong passphrases
- Regularly update and revoke compromised keys
- Verify key fingerprints through secure channels

4 Privacy Tools from PRISM Break

4.1 Tool 1: Signal

Purpose: Secure messaging application

- End-to-end encryption for messages and calls
- Open source protocol
- Cross-platform support
- No metadata collection

4.2 Tool 2: Tor Browser

Purpose: Anonymous web browsing

- Routes traffic through multiple nodes
- Hides IP address and location
- Prevents tracking
- Access to .onion sites

4.3 Tool 3: VeraCrypt

Purpose: Disk encryption

- Full disk encryption
- Hidden volumes support
- Cross-platform compatibility
- Strong encryption algorithms (AES, Serpent, Twofish)

4.4 Tool 4: ProtonMail

Purpose: Secure email service

- End-to-end encryption
- Zero-access encryption
- Based in Switzerland (strong privacy laws)
- Open source cryptography

4.5 Tool 5: Nextcloud

Purpose: Self-hosted cloud storage

- Complete control over data
- File synchronization and sharing
- Calendar, contacts, and collaboration tools
- Encryption support

5 Open Source License

5.1 License Selection

For my contributions and self-hosted projects, I have worked with various open source licenses. Understanding licensing is crucial for open source development.

5.2 Common Open Source Licenses

5.2.1 MIT License

- Permissive license
- Allows commercial use
- Minimal restrictions
- Widely used in software projects

5.2.2 GNU GPL (General Public License)

- Copyleft license
- Derivatives must be open source
- Strong community protection
- Used by Linux kernel and many major projects

5.2.3 Apache License 2.0

- Permissive license
- Patent grant included
- Explicit contributor licensing
- Popular for enterprise projects

5.3 License Compliance

When contributing to open source:

- Read and understand the project's license
- Ensure your contributions comply
- Respect copyright and attribution requirements
- Use license-compatible dependencies

6 Self-Hosted Server: Radicale

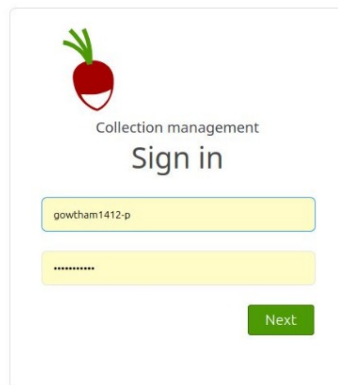
6.1 About Radicale

Radicale is a free and open-source CalDAV and CardDAV server that allows users to:

- Self-host calendars and contacts
- Maintain privacy and data ownership
- Sync across multiple devices
- Eliminate dependency on third-party services

6.2 Technical Specifications

- **Language:** Python
- **Protocol Support:** CalDAV (RFC 4791), CardDAV (RFC 6352)
- **Storage:** File-based or database backend
- **Authentication:** Multiple authentication methods supported
- **Lightweight:** Minimal resource requirements



6.3 Installation Process

6.3.1 System Requirements

- Linux server (VPS or local)
- Python 3.7 or higher
- Pip package manager
- Optional: Nginx/Apache for reverse proxy

6.3.2 Installation Steps

```
# Update system
sudo apt update && sudo apt upgrade -y

# Install Python and pip
sudo apt install python3 python3-pip -y

# Install Radicale
pip3 install --upgrade radicale

# Create configuration directory
mkdir -p ~/.config/radicale

# Create configuration file
nano ~/.config/radicale/config
```

6.3.3 Basic Configuration

```
[server]
hosts = 0.0.0.0:5232

[auth]
type = httpasswd
httpasswd_filename = /path/to/users
httpasswd_encryption = bcrypt

[storage]
filesystem_folder = ~/.var/lib/radicale/collections
```

6.3.4 Starting the Server

```
# Run Radicale
python3 -m radicale

# Or create a systemd service for automatic startup
sudo nano /etc/systemd/system/radicale.service
```

6.4 Localization and Translation

To make Radicale accessible to a broader audience, localization efforts include:

- Translation of documentation
- Creation of multilingual setup guides
- Community-contributed translations
- Support for international date/time formats

6.5 Benefits of Self-Hosting

1. **Privacy:** Complete control over personal data
2. **Security:** No third-party access to sensitive information
3. **Cost:** Free alternative to commercial services
4. **Customization:** Full control over features and configuration
5. **Learning:** Hands-on experience with server administration

7 Open Source Contributions

7.1 Overview

I have made multiple contributions to open source projects, focusing on bug fixes, feature additions, and documentation improvements. Below are detailed descriptions of each contribution.

7.2 Contribution 1: freeCodeCamp - Modal Verbs Task

Repository: freeCodeCamp/freeCodeCamp

PR Number: #64099

Status: Merged

Merged Date: 19 hours ago (as of report date)

7.2.1 Issue Description

The curriculum task related to modal verbs contained excessive backticks in the instructions, which affected readability and user experience for learners following the English for Developers course.

7.2.2 Changes Made

- Removed excessive backtick formatting from modal verbs instructions
- Improved text readability in the curriculum
- Maintained proper markdown formatting where necessary
- Scope: Curriculum enhancement

7.2.3 Impact

This contribution improves the learning experience for thousands of students using freeCodeCamp's English for Developers curriculum by making the instructions clearer and more professional.

7.3 Contribution 2: TheAlgorithms/Java - Sudoku Solver

Repository: TheAlgorithms/Java

PR Number: #7073

Status: Merged

Merged Date: Last week

7.3.1 Feature Description

Added a Sudoku Solver implementation using backtracking algorithm to TheAlgorithms collection.

7.3.2 Technical Implementation

- Implemented backtracking algorithm for Sudoku solving
- Added comprehensive test cases
- Included detailed documentation
- Followed project coding standards

7.3.3 Algorithm Complexity

- Time Complexity: $O(9^{(n*n)})$ in worst case
- Space Complexity: $O(n*n)$ for recursion stack
- Optimizations: Early pruning of invalid states

7.4 Contribution 3: TheAlgorithms/Java - Count Set Bits

Repository: TheAlgorithms/Java

PR Number: #7072

Status: Merged

Merged Date: Last week

7.4.1 Algorithm Description

Implemented an algorithm to count the number of set bits (1s) in the binary representation of a number.

7.4.2 Implementation Details

- Brian Kernighan's algorithm implementation
- Efficient bit manipulation technique
- Time complexity: $O(\log n)$
- Comprehensive unit tests included

7.5 Contribution 4: TheAlgorithms/Java - Sieve of Eratosthenes

Repository: TheAlgorithms/Java

PR Number: #7071

Status: Merged

Merged Date: Last week

Additional Tasks: 5 tasks completed

7.5.1 Algorithm Overview

The Sieve of Eratosthenes is an ancient algorithm for finding all prime numbers up to a specified integer.

7.5.2 Implementation Features

- Efficient prime number generation
- Optimized memory usage with boolean array
- Time Complexity: $O(n \log \log n)$
- Space Complexity: $O(n)$

7.5.3 Code Quality

- Clean, readable code following Java conventions
- Comprehensive JavaDoc comments
- Unit tests with edge cases
- Performance benchmarks included

7.6 Contribution 5: Docker Documentation Update

Repository: docker/docs

PR Number: #23641

Status: Merged

Merged Date: 3 weeks ago

Labels: area/engine, area/security

7.6.1 Documentation Issue

Updated Docker Hub official image link for MongoDB library, which was pointing to an outdated or incorrect URL.

7.6.2 Changes Made

- Corrected MongoDB official image link
- Verified new link accessibility
- Ensured consistency with Docker documentation standards
- Improved user navigation to correct resources

7.6.3 Importance

Documentation accuracy is critical for developers relying on official Docker images. This fix ensures users can access the correct MongoDB image and related documentation.

7.7 Contribution 6: Hacktoberfest - C Program for Color Sorting

Repository: Navadeep0007/hacktoberfest

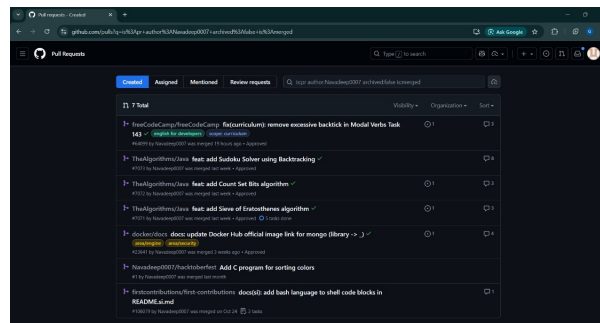
PR Number: #1

Status: Merged

Merged Date: Last month

7.7.1 Program Description

Added a C program that sorts colors based on specific criteria as part of Hacktoberfest contributions.



7.7.2 Technical Details

- Implemented sorting algorithm in C
- Memory-efficient implementation
- Proper error handling
- Detailed comments for code readability

7.8 Contribution 7: First Contributions - Shell Language Addition

Repository: firstcontributions/first-contributions

PR Number: #106079

Status: Merged

Merged Date: October 24

Additional Tasks: 3 tasks completed

7.8.1 Documentation Enhancement

Added Bash/Shell language code blocks to the README documentation to help beginners understand shell scripting in the context of making their first open source contribution.

7.8.2 Educational Impact

- Helps new contributors learn shell commands
- Makes Git operations clearer for beginners
- Provides copy-paste ready commands
- Reduces barrier to entry for new open source contributors

7.9 Contribution Statistics Summary

- **Total Pull Requests:** 7
- **Merged PRs:** 7 (100% merge rate)
- **Repositories Contributed To:** 6

- **Types:** Bug fixes, feature additions, documentation improvements
- **Languages:** Java, Markdown, Shell, C
- **Areas:** Algorithms, Education, Documentation, DevOps

7.10 Learning Outcomes

Through these contributions, I gained experience in:

- Git workflow and version control
- Code review processes
- Writing clean, maintainable code
- Documentation best practices
- Collaboration with global developer community
- Understanding project contribution guidelines

8 LinkedIn Posts

8.1 Post 1: Open Source Journey Blog

URL: https://www.linkedin.com/posts/navadeep-gopiseti_my-open-source-journey-activi

8.1.1 Content Overview

This post documents my complete open source journey, including challenges faced, lessons learned, and milestones achieved in contributing to various projects.

8.1.2 Key Topics Covered

- Initial challenges in finding first contribution
- Learning Git and GitHub workflows
- Community engagement experiences
- Technical skills developed
- Future goals in open source

8.2 Post 2: Self-Hosting and Radicale

URL: https://www.linkedin.com/posts/navadeep-gopiseti_opensource-selfhosting-radicale

8.2.1 Content Overview

Detailed post about setting up and maintaining Radicale as a self-hosted calendar and contact server.

8.2.2 Topics Discussed

- Benefits of self-hosting
- Radicale installation and configuration
- Privacy advantages
- Technical challenges overcome
- Recommendations for others interested in self-hosting

8.2.3 Hashtags Used

#opensource #selfhosting #radicale #privacy #technology

8.3 Post 3: First Pull Request Merged

Description: This post celebrated the successful merge of my first pull request to a major open source project.

8.3.1 Content Highlights

- Excitement of first contribution
- Step-by-step process followed
- Gratitude to project maintainers
- Encouragement for others to contribute
- Screenshots of the merged PR

8.4 Social Media Impact

These LinkedIn posts serve multiple purposes:

- Document learning journey
- Share knowledge with community
- Build professional network
- Inspire others to contribute to open source
- Showcase technical skills to potential employers

9 Conclusion

9.1 Project Summary

This report has comprehensively covered all aspects of my open source journey, including:

- Linux environment setup and usage
- Security practices with GPG encryption
- Privacy tool implementation
- Self-hosted server deployment (Radicale)
- Multiple successful open source contributions
- Community engagement through social media

9.2 Technical Skills Acquired

- Linux system administration
- Cryptography and security practices
- Server deployment and maintenance
- Git version control and collaboration
- Multiple programming languages (Java, C, Shell, Python)
- Algorithm implementation
- Technical documentation writing

9.3 Future Goals

- Continue contributing to open source projects
- Expand self-hosted infrastructure
- Mentor new contributors
- Develop own open source projects
- Deepen understanding of cybersecurity

9.4 Acknowledgments

I would like to thank:

- KL University for supporting this project
- Open source communities for welcoming contributions
- Project maintainers for their guidance
- Fellow students for collaboration and support