Semester Long Internship Report

On

eSim Development and Installation

Submitted by

Anshul Verma

Under the guidance of

Prof.Sumanto Kar

Engineering Department IIT
Bombay

28 May, 2025

Acknowledgment

I would like to express my heartfelt gratitude to the entire FOSSEE team for providing me with the invaluable opportunity to be a part of this prestigious internship. Working with an open-source initiative like FOSSEE has been a deeply enriching experience, allowing me to explore the technical and practical aspects of simulation software such as **eSim**, while simultaneously sharpening my problem-solving and collaborative skills. I am profoundly thankful for the chance to contribute to this impactful project and to gain exposure to the real-world challenges of open-source development. This internship has been a significant step in my academic and professional journey, and I will carry the lessons learned here into my future endeavors.

I am especially grateful to **Prof. Sumanto Kar**, my mentor and guide during this internship. His unwavering support, insightful feedback, and expert guidance have been instrumental in shaping my understanding of the project. His mentorship not only enhanced my technical capabilities but also inspired me to approach problems with a thoughtful and solution-oriented mindset.I would also like to sincerely thank the entire FOSSEE team, whose collaborative spirit and technical expertise made this internship a thoroughly rewarding experience. My sincere appreciation goes out to all team members for their encouragement, responsiveness, and for fostering an environment conducive to learning and innovation..

Overall, this internship has been a transformative experience, contributing significantly to both my personal growth and professional aspirations. As I look forward to a career in the semiconductor and electronics domain, I will always regard this experience at FOSSEE, IIT Bombay, as a foundational milestone.

Contents

Chapter.1

Introduction

This report outlines the work I carried out during my internship at IIT Bombay's FOSSEE project, specifically on the **eSim** tool, an open-source EDA (Electronic Design Automation) software for circuit design, simulation, and PCB layout.

eSim (formerly Oscad / FreeEDA) is a powerful tool developed using free/libre and open-source software such as KiCad, Ngspice, GHDL, OpenModelica, Verilator, Makerchip, and the SkyWater SKY130 PDK. It provides functionalities similar to commercial tools like OrCAD, PSpice, LTspice, Xpedition, and HSPICE — but without licensing costs, making it especially suitable for educational institutions and small-medium enterprises (SMEs).

1.1 eSim

eSim is a free/libre and open-source Electronic Design Automation (EDA) tool developed by FOSSEE at IIT Bombay. It supports circuit design, simulation, analysis, and PCB layout using open-source components like KiCad, Ngspice, and GHDL.

1.2 Flatpak

latpak is a cross-platform software packaging and distribution system that simplifies application deployment across different Linux distributions by isolating apps and managing dependencies efficiently. Flatpak is a framework for distributing desktop applications across various Linux distributions. It was created by developers with a long history of working on the Linux desktop and is run as an independent open source project.

Features of Flatpak

1. Cross-Distribution Compatibility

- Flatpak works on most major Linux distributions, allowing applications to be installed and run across systems without modification.

2. Sandboxing (Isolation and Security)

- Applications run in an isolated environment (sandbox), limiting their access to system resources and improving overall security.

3. Bundled Dependencies

- Flatpak packages include all required libraries and dependencies, ensuring that the app works consistently on any supported system.

4. Automatic Updates

- Flatpak supports automatic updates and uses delta updates to download only the parts of the app that changed, saving bandwidth.

5. Versioning and Rollbacks

- Users can easily switch between different versions of an application or roll back to a previous version if needed.

6. User-Level Installation

- Apps can be installed and run by individual users without requiring administrator privileges.

7. Central Repositories

- Flatpak uses centralized repositories like Flathub, where users can find and install a wide range of applications.

8. Portals (Permission Management)

- Portals offer a secure way for applications to request access to system resources (like files or hardware) through a standardized permission system.

1.3 Ubuntu

Ubuntu, one of the most popular Linux distributions, is widely used in educational, research, and development environments. During my internship, I worked extensively on ensuring eSim's compatibility across various Ubuntu versions and contributed to packaging eSim as a Flatpak for improved accessibility and platform independence.

Chapter 2

Task Chosen: Installation and Packaging

Currently, eSim only supports limited versions of some Linux Distributions, namely Ubuntu and Fedora due to the dependency constraints on other platforms as they might be hosted under different names, use different package managers, have different packaging formats, or have outdated packages. This means that the dependencies need to be resolved by self-hosting packages, compiling on the host machine and using alternative packaging management systems.

During the course of my internship, I was involved in several critical tasks related to the development, deployment, and user support of eSim. These tasks are outlined below:

a. Installation and Compatibility Testing

One of my key responsibilities was to ensure that eSim could be reliably installed and executed on multiple versions of **Ubuntu**, one of the most widely used Linux distributions. I carried out the following tasks:

- Installed and rigorously tested eSim on Ubuntu 22.04 (Jammy), 23.04 (Lunar), 24.04 (Noble), and the development version 25.xx.
- Diagnosed and resolved compatibility issues caused by changes in libraries, package versions, or system configurations across these Ubuntu releases.
- Addressed package dependency conflicts and environment setup errors to ensure a stable installation process.
- Provided feedback to the development team and documented the installation process for future users.

b. Windows Installation Support

In addition to Linux support, I worked on making eSim more accessible for **Windows users** by:

- Analyzing platform-specific installation challenges, including dependency resolution, file path handling, and administrative permissions.
- Creating detailed installation guides tailored to Windows, aimed at reducing the setup effort for new users.
- Troubleshooting common errors encountered during installation and contributing to a more user-friendly experience by recommending and implementing fixes.
- Testing the eSim setup process on multiple Windows systems to ensure consistency and usability.

c. Flatpak Packaging

To facilitate broader adoption of eSim across various Linux distributions, I was involved in **Flatpak-based packaging**, which provides a distribution-agnostic deployment solution. My work included:

- Building and testing Flatpak packages of eSim to ensure smooth installation and execution regardless of the underlying Linux distribution.
- Configuring **sandbox permissions** to allow eSim access to necessary system resources while maintaining security best practices.
- Debugging issues related to desktop integration, environment variables, and library linking inside the Flatpak environment.
- Updating the Flatpak manifest and associated build scripts to be compatible with the latest versions of Flatpak and the eSim codebase, ensuring long-term maintainability.

Chapter 3

Objective

The primary goal of this internship was to contribute to the ongoing development and accessibility of **eSim**, an open-source EDA tool developed by FOSSEE, IIT Bombay. The specific objectives of my internship were as follows:

- **To explore and support the installation of eSim on various operating systems**: Ensuring that eSim runs smoothly across multiple environments by identifying and resolving OS-specific issues and dependencies.
- **To work on packaging eSim for cross-platform usage**: Focusing on the creation and testing of Flatpak packages to provide an OS-agnostic method for installing eSim on Linux distributions.
- **To contribute towards improving ease of access to eSim for the user community**: Aiming to simplify the installation process for end-users, including students, educators, and professionals, by developing streamlined workflows and troubleshooting documentation.
- **Ensuring eSim Compatibility Across Multiple Operating Systems:** One of the main goals was to test and verify the compatibility of eSim on various versions of the Ubuntu Linux operating system. This involved identifying version-specific challenges, resolving dependency issues, and ensuring a stable installation process. By doing so, the objective was to make eSim more robust and accessible for users across different Linux environments.
- **Developing a Cross-Platform Installation Mechanism Using Flatpak:** To improve eSim's portability and simplify its installation across all major Linux distributions, I worked on Flatpak packaging. Flatpak is a modern software distribution system that enables developers to package applications along with their dependencies, providing consistent behavior on any supported system. The objective here was to create a Flatpak package for eSim, allowing it to be installed easily regardless of the user's distribution or environment.

Chapter 4

Implementation

4.1 Ubuntu Installation Testing and Issue Resolution

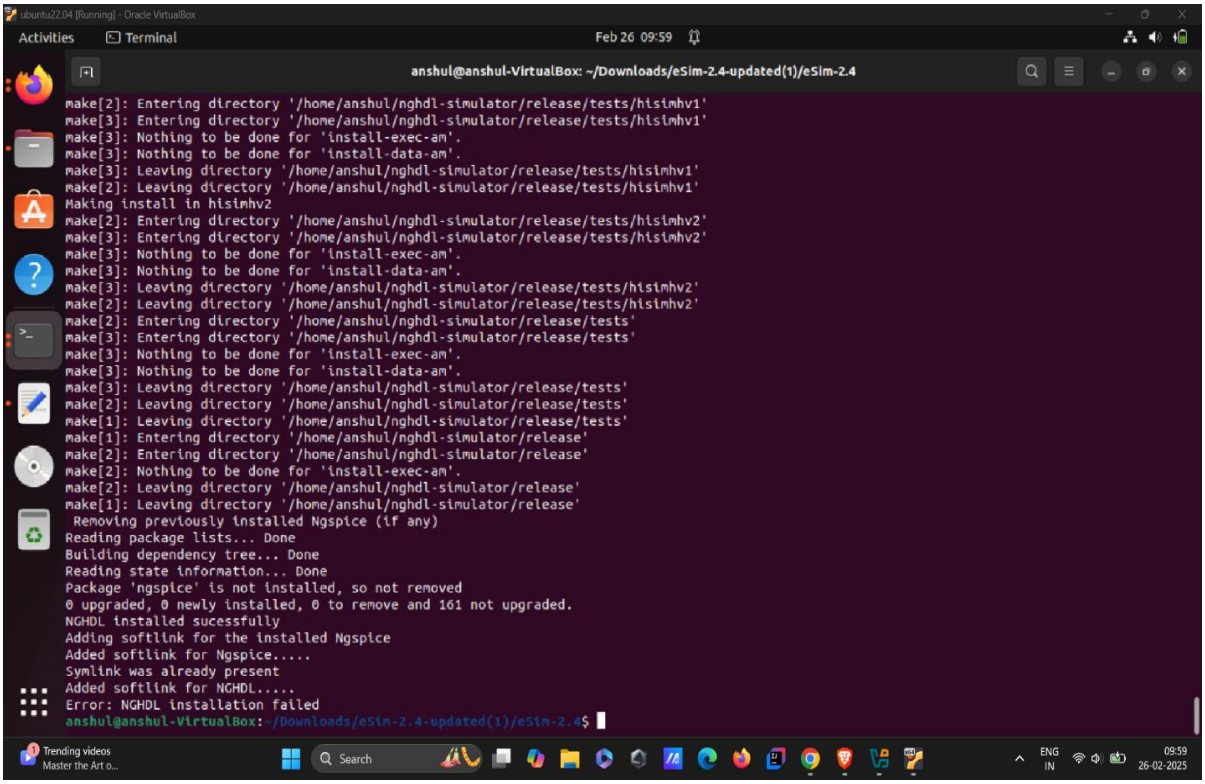
During the internship, I thoroughly tested eSim on multiple versions of Ubuntu including 22.04 (Jammy), 23.04 (Lunar), 24.04 (Noble), and 25.04(Development). The following were the observations and actions taken:

Ubuntu 22.04 (Jammy):

I have tried to Download **eSIM** on **Ubuntu 22.04 OS** the Issues that i have encountered and their Solutions are listed below:

Issue 1:

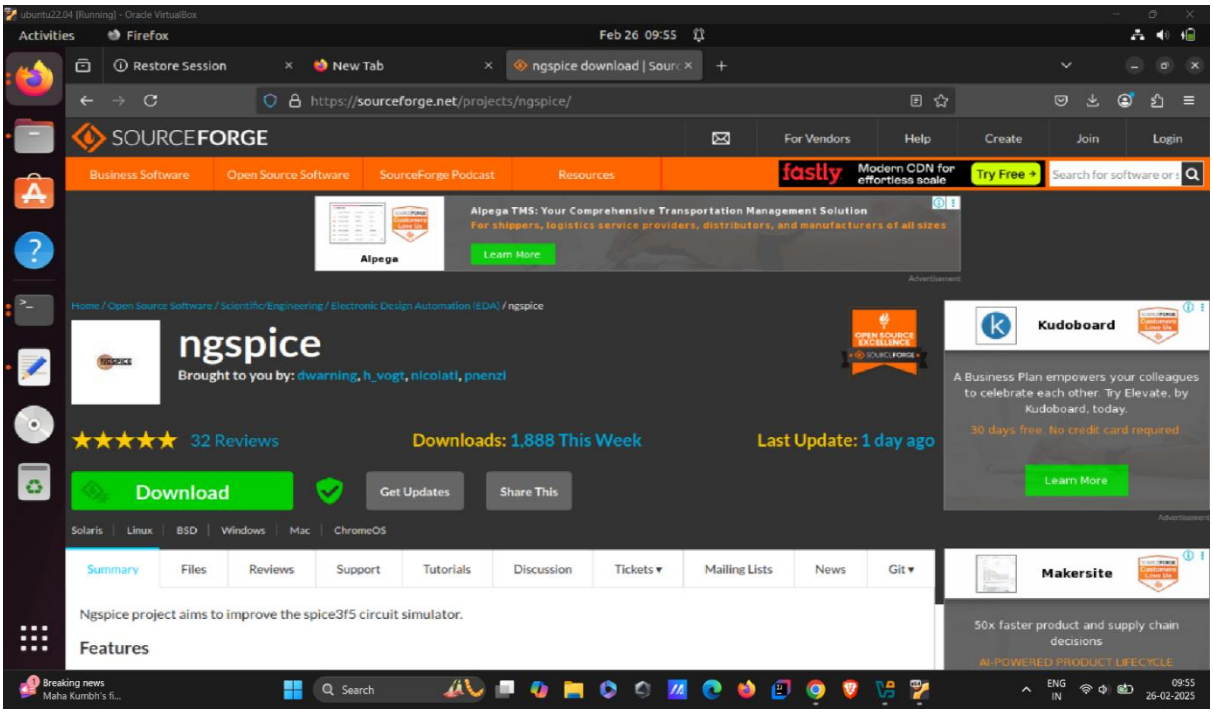
This Is the first Issues I have encountered. **NGHDL installation** is failed. And the installation is aborted. This is an dependency issue of eSim.



After searching a lot across web, files and documentation of Ubuntu OS. I have found a solution.

Solution to Issue 1:

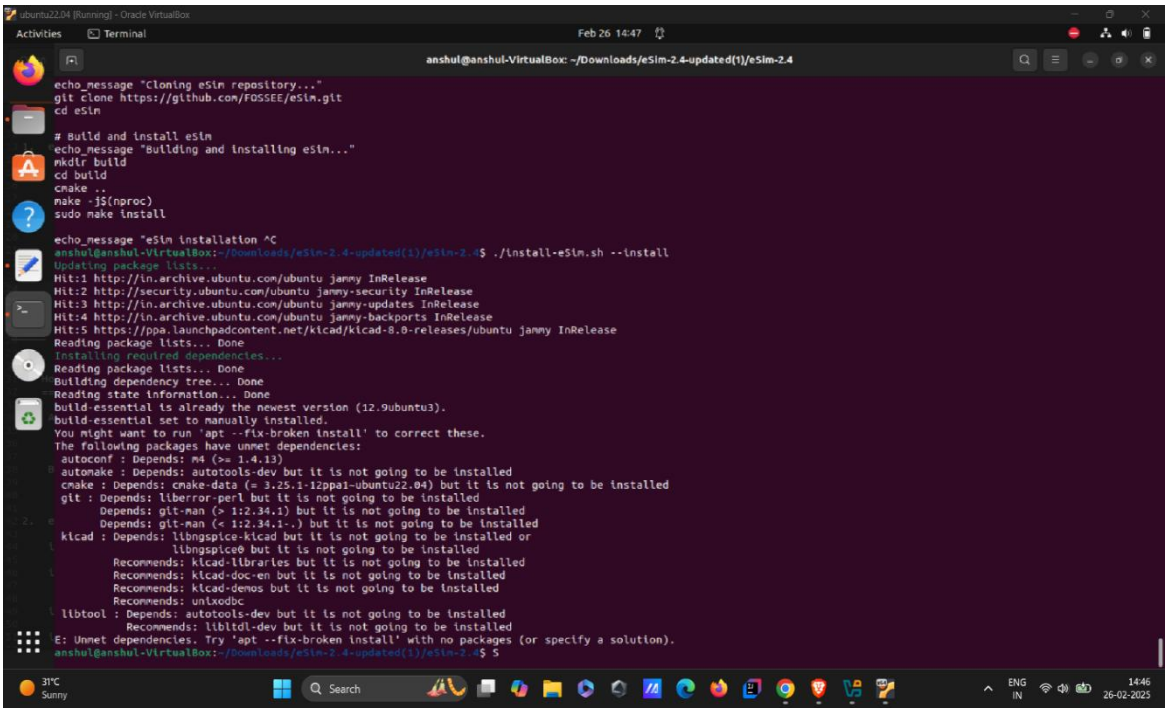
Step1) Go the website link and download **ngspice** from there.



Download it and it will resolve error.

Issue 2: Fix broken Installs **(E : Unmet dependencies)**

This error occurs because of broken or not found dependencies.



Solution to Issue 2:

To resolve this error we need to run the following command.

Run the following command:

```
sudo apt –fix-broken install
```

This will resolve the error if not then we need try to install eSim on fresh VM of Ubuntu 22.04

Issue 3: You are not able to run that command you are not sudo user.

Step1) To resolve this error. Run the following command.

```
su –
```

Step2) Then run.

```
sudo visudo
```

This opens the file in a safe text editor (usually vi or nano, depending on your system).

Step3) Give a user **sudo** access without a password

Add this line at the end of the file:

username ALL=(ALL) NOPASSWD:ALL

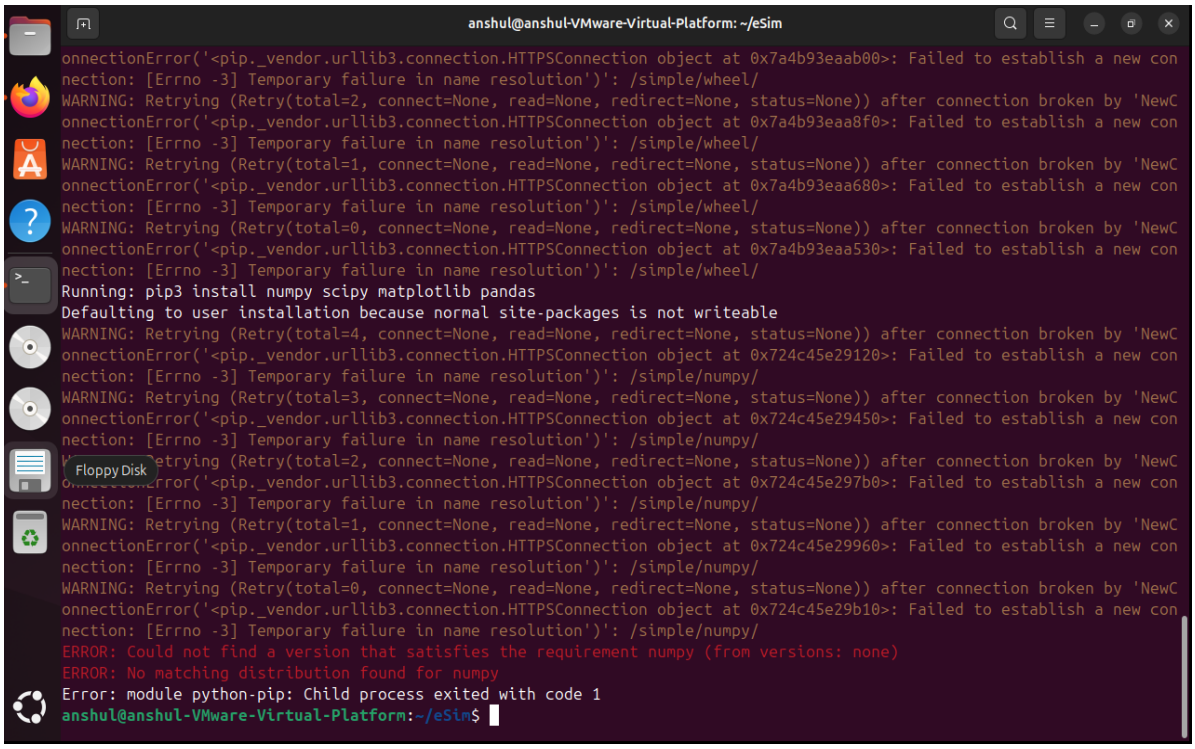
Step4)

If using **nano,** press **Ctrl + X, then Y**, then Enter.

If using **vi,** press **Esc, type :wq,** and press Enter.

Ubuntu 23.04 (Lunar):

Issue 1: Encountered issues with Python3 bindings in the packaging script. No matching distribution found for **numpy.** This is another dependency issue and it automatically resolves sometimes after a fresh OS start.



Solution to Issue 1:

To resolve this error we need to start installing **eSim** with a **fresh OS.** This will resolve most of the erros.

Option1)

If not we need to download them using following commands:

```
sudo apt update
sudo apt install python3-pip
```

and then

```
pip3 install numpy scipy matplotlib
```

Option2) Using **apt (System packages)**

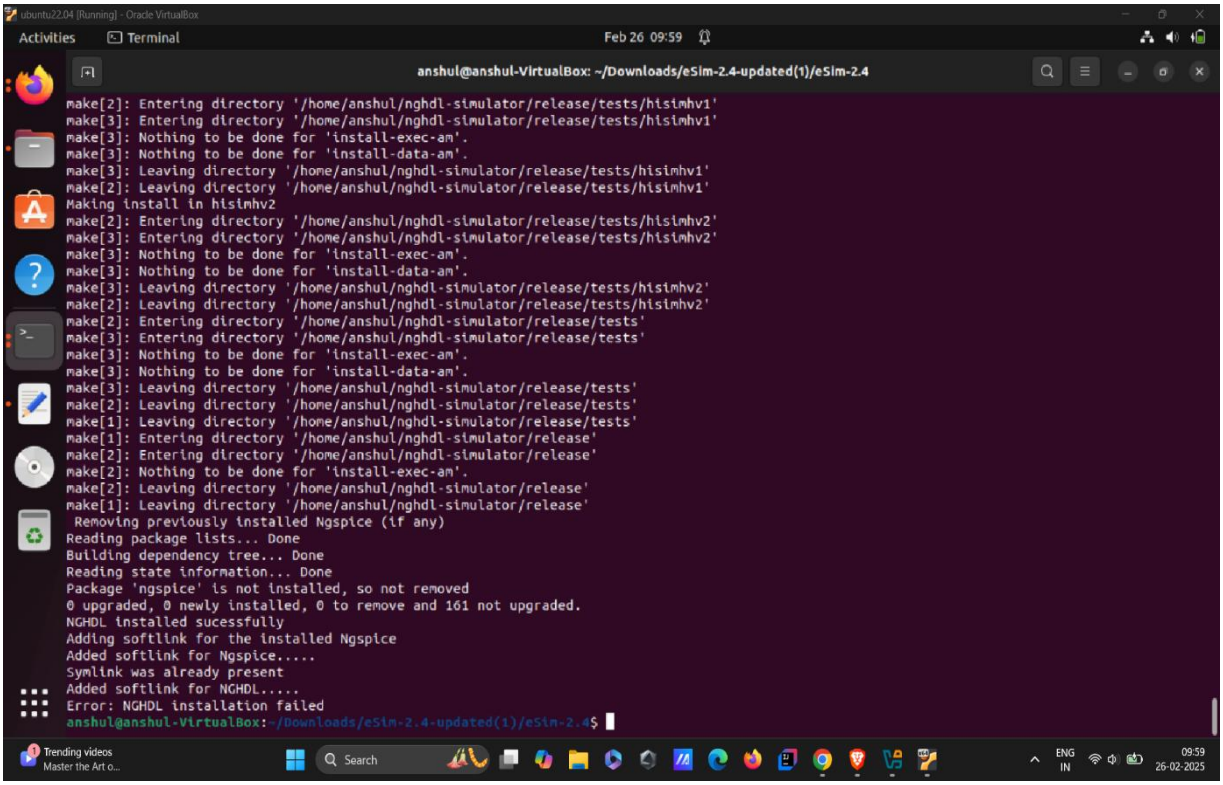
```
sudo apt update
sudo apt install python3-numpy python3-scipy python3-matplotlib
```

This will resolve the error.

Issue 2:

I have encountered. **NGHDL installation** is failed and **Ngspice** dependency not found. And the installation is aborted. This is an dependency issue of eSim.

This error occurs multiple times working with **eSim installation** and packaging this error also depend on several **Linux distributions.**



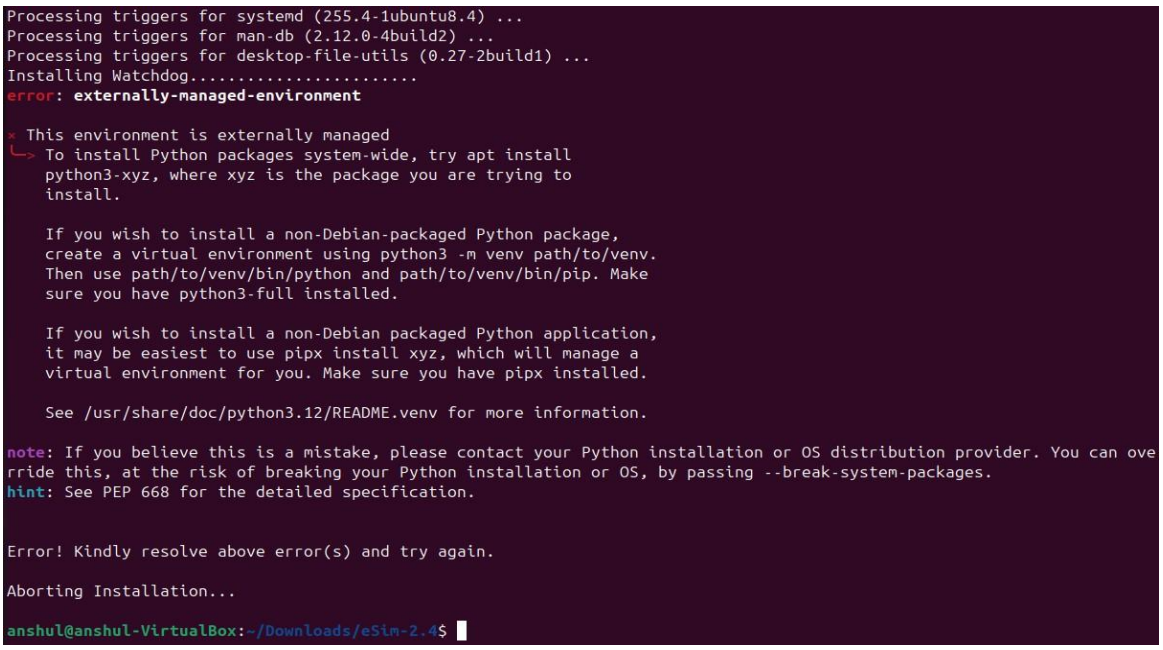
Ubuntu 24.04 (Noble)

This is the stable version of the Ubuntu OS, and our main objective is to download eSim on it. Below are the issues I encountered while testing and packaging, along with their respective solutions.

Issue 1:

Error: externally-managed enviroment(While installing Watchdog).

The **"externally-managed environment" error** in Ubuntu typically appears when you try to use pip to install Python packages **system-wide** on certain Linux distributions, including newer versions of Ubuntu (like 22.04 and later). This is a deliberate restriction introduced by the maintainers of Python and Linux distros to prevent breaking system-managed Python packages. The error is caused by an externally managed environment. To resolve this error we need to create a virtual environment using python3.



Solution to fix this Issue 1)

Step1): Open your Ubuntu CMD and type the following command:

```
sudo apt update
sudo apt install python3-venv
```

Step2): Create a Virtual Environment: Navigate to the directory where you want the virtual environment and run:

```
python3 -m venv myenv
```

Step3): To activate the Virtual Environment run:

```
source myenv/bin/activate
```

Step4): Now run:

./install-eSim.sh –install

Issue 2:

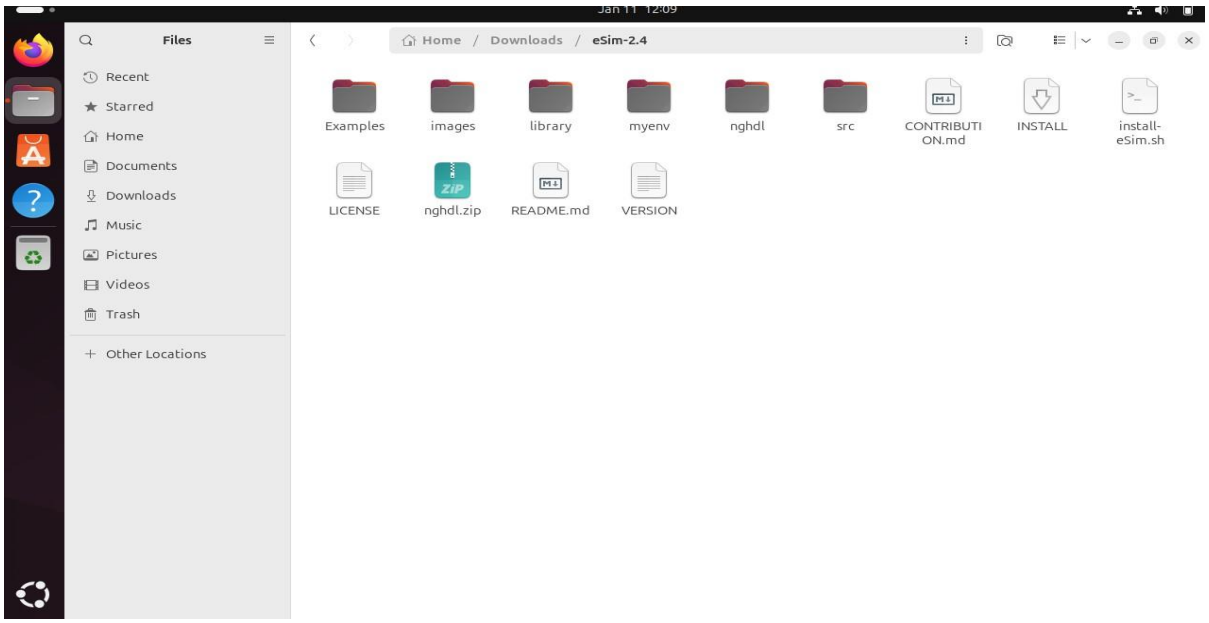
Package **‘python3-distutils’** has no installation candidate.

This issue states that there is no **'python3-distutils' module** found or is not available.

This issue takes hours to solve but I have found a way to solve it.

Solution to Issue 2:

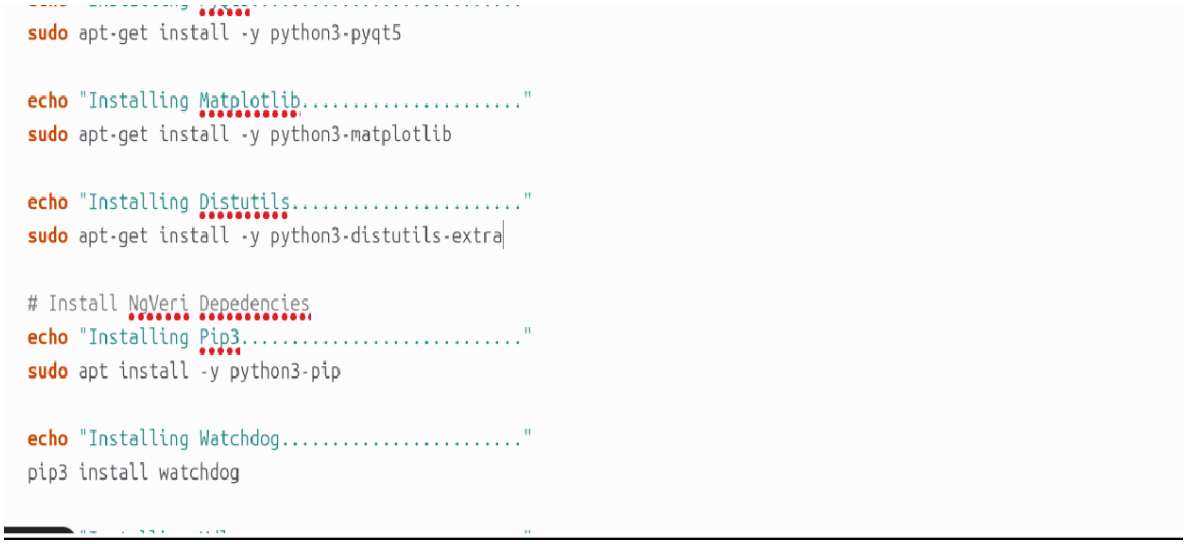
Step1) Go to your eSim-2.4 unzip folder. Go to your downloads and open the **eSim** folder



Step2) Open the install-eSim.sh and look for the function installDependency and in that function look for **"Installing Distutils".**

```
echo "Installing PyQt5........................."
sudo apt-get install -y python3-pyqt5

echo "Installing Matplotlib....................."
sudo apt-get install -y python3-matplotlib

echo "Installing Distutils....................."
sudo apt-get install -y python3-distutils

# Install NgVeri Depedencies
echo "Installing Pip3.........................."
sudo apt install -y python3-pip

echo "Installing Watchdog......................."
pip3 install watchdog

echo "Installing Hdlparse......................."
```

Step3) Now change **sudo apt-get install -y python3-distutils** to **sudo apt-get install -y python3-distutils-extra.**



Issue 3:

The **kicad/kicad-6.0-releases does'nt** have a Release file. This error shows that kicad library does'nt have a release file.

The **KiCad library** refers to the collection of predefined electronic components and symbols used within the **KiCad EDA (Electronic Design Automation)** software suite. KiCad is an open-source tool used for designing **schematics** and **printed circuit boards (PCBs)**.

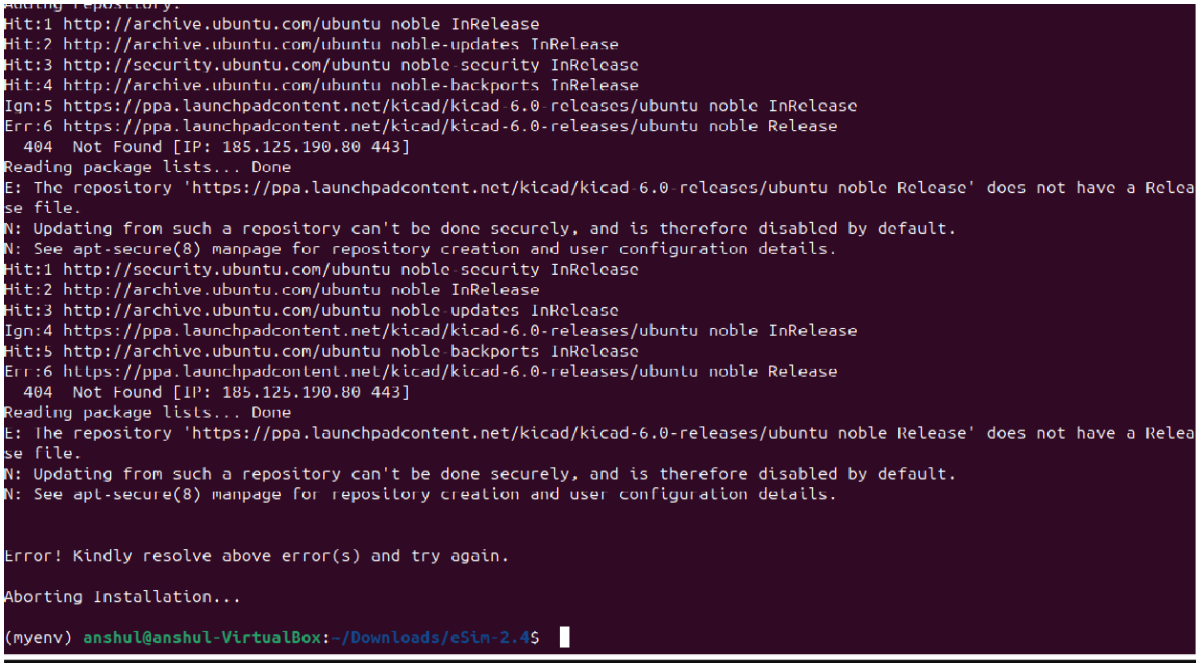
Types of KiCad Libraries

KiCad libraries are broadly categorized into three main types:

1. **Symbol Libraries**
   - Contain schematic symbols (used in the schematic editor).
   - Examples: resistors, capacitors, ICs, connectors.
2. **Footprint Libraries**
   - Contain physical layouts for components (used in the PCB editor).
   - Define pad layout, dimensions, and shape.

3. **3D Model Libraries**

   - Provide 3D models of components for 3D visualization of PCBs.
   - Often associated with footprints.



Solution to Issue 3)

Step1) To resolve this problem we have to download Kicad Library manually.Open your Ubuntu CMD and type the following command:

```
sudo apt update
sudo apt install git
```

Step2) Clone the KiCad Libraries Repository. Type the following command:

```
git clone https://gitlab.com/kicad/libraries.git
```

Step3) Navigate to the Downloaded Libraries Folder.

```
cd libraries
```

Download Using apt Package Manager:

If you do not want to download it from git you can directly download it by using apt package manager. Skip this if you have downloaded from Github.

Step1) Install KiCad Using apt: Run the following command:

```
sudo apt update
```

Step2) Install KiCad: For the latest stable release available in the Ubuntu repositories, run:

```
sudo apt install kicad
sudo apt install kicad-libraries
```

This measure fixes the problem.

Issue 4:

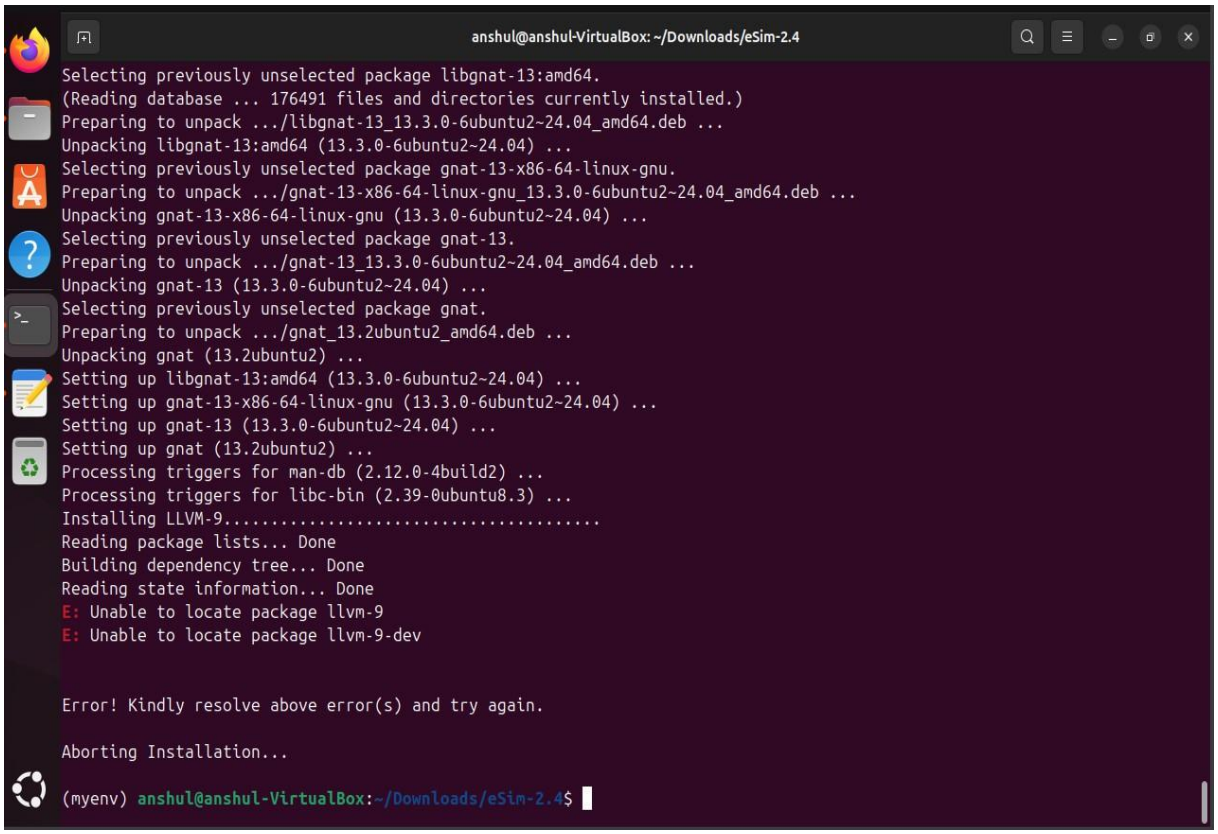
Error while Installing **llvm-9** and **llvm-9 dev**

This problem states that **'Unable to locate package llvm-9 and llvm-9-dev'**. To resolve this problem we need to download llvm packages manually in our machine. This error indicates that the script is unable to find the **llvm-9 and llvm 9-dev packages.**

Both llvm-9 and llvm-9-dev are packages related to **LLVM version 9**, a powerful open-source **compiler infrastructure** used for building languages like **C, C++, Rust, Swift**, and others.

llvm-9

- This is the **runtime** package.
- It includes tools like:
    - llvm-as: LLVM assembler
    - llvm-dis: LLVM disassembler
    - opt: LLVM optimizer
    - llc: LLVM static compiler
- Used primarily to **run** or **interact with** LLVM-based tools.
- Installs command-line tools and basic runtime libraries for LLVM 9.



Solution to Issue 4)

There are two ways to solve the problem depending on different systems:

The 1st Way: To resolve the problem we need to download llvm and llvm dev using apt package manager.

Step1) Open your CMD. And paste the following command:

sudo apt update

Step2) Install the **llvm and llvm-dev** using cmd. Paste the following command:

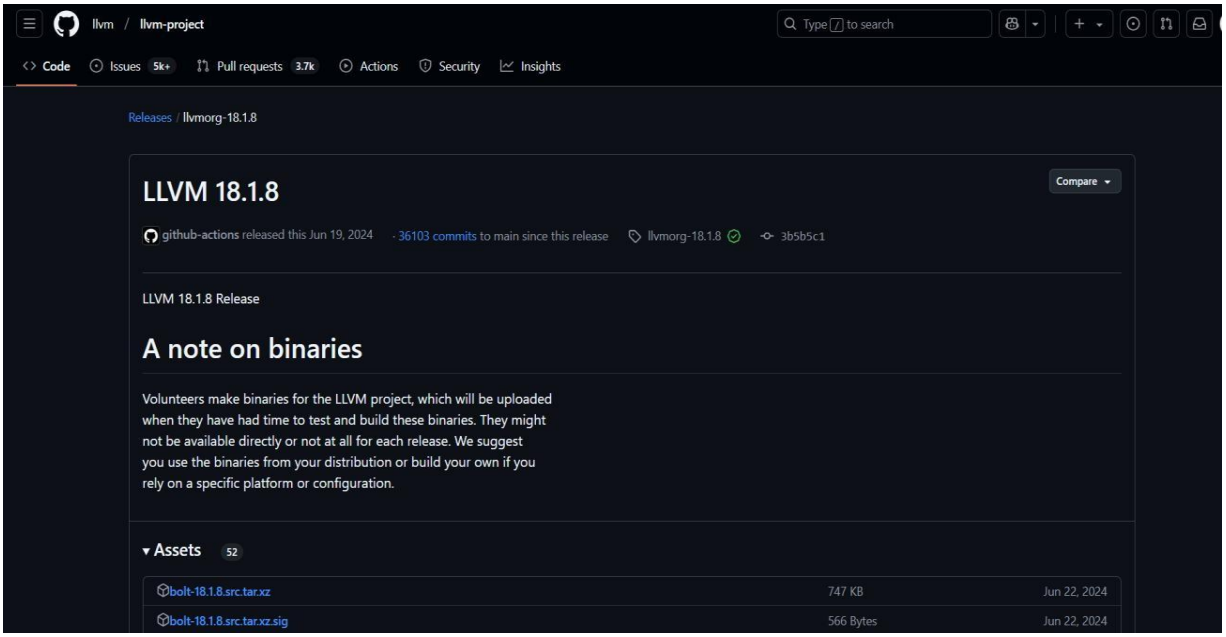
sudo apt install llvm

sudo apt install llvm-dev

The 2nd Way: In this way we will download **llvm and llvm-dev** from **github**.

Step1) Open your browser and paste the following link: https://github.com/llvm/llvm-project/releases/tag/llvmorg-18.1.8

Step2) Scroll down and you will find **llvm and llvm-dev** and download them.



I followed the official instructions for installing eSim on Ubuntu 24.04. During the process, I encountered several errors related to missing dependencies, including the python3-distutils module, externally-managed environments, issues with the KiCad library, and problems with installing llvm-9 and llvm-9 dev. I conducted extensive research through official documentation and community forums to resolve these issues.

4.2 Windows Installation Work

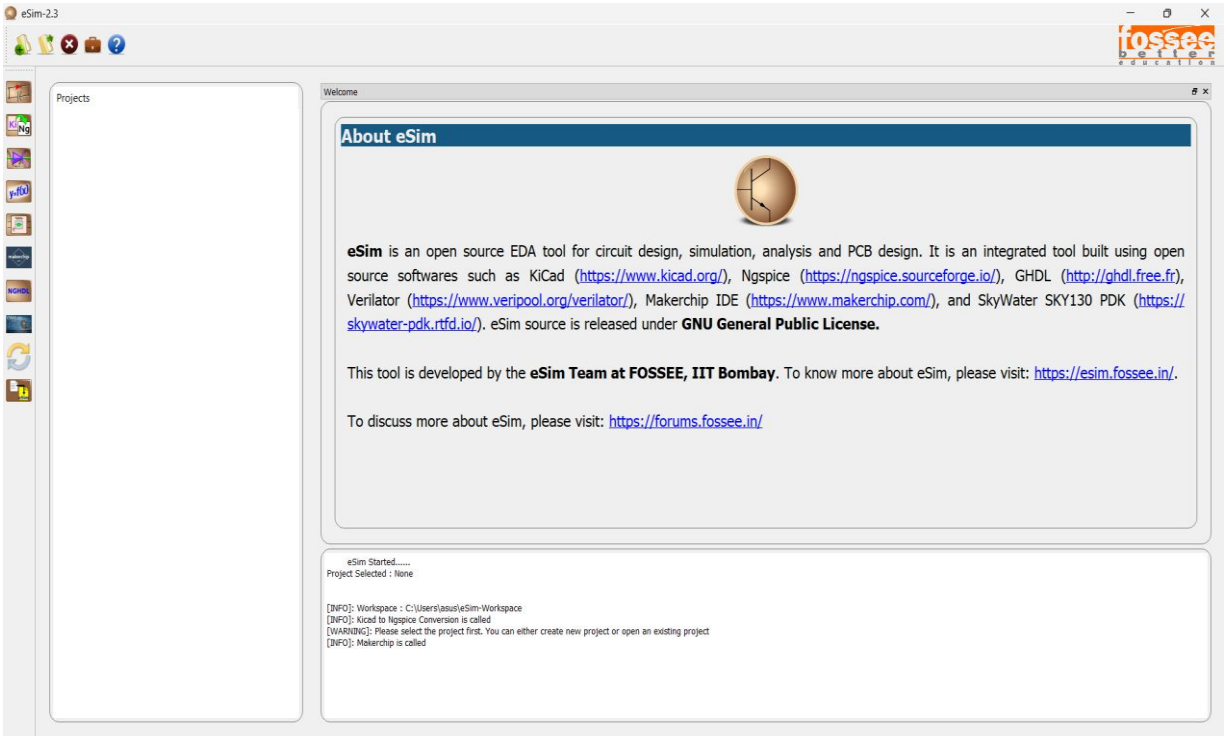
While eSim is primarily designed for Linux environments, extending its usability to **Windows** is crucial for increasing its adoption—especially in educational institutions and among users unfamiliar with Linux. During my internship, I contributed to supporting and improving the **Windows installation process** of eSim through the following activities:

- **Initial Environment Setup and Testing**
  I began by setting up clean Windows 10 and Windows 11 environments (both virtual and physical) to test the existing installation process of eSim. This allowed me to reproduce common user issues in a controlled setup and document any inconsistencies.

- **DependencyResolution and Installer Debugging**
  One of the major challenges was handling external dependencies such as Ngspice, Python, and KiCad components on Windows, where automatic package management (as seen in Linux) is not available. I reviewed the installer script and modified it to ensure all necessary dependencies were correctly installed and configured. Special attention was given to:

    - **PATH Environment Variables**: Ensured all required binaries and executables were correctly added to the system path.
    - **Python Environment**: Verified compatibility with specific Python versions and libraries, and ensured scripts ran without requiring manual intervention.
    - **Admin Permissions**: Resolved installation issues that required elevated privileges by modifying script prompts and improving error messages for clarity.

- **Error Handling and User-Friendly Workflows**
  I identified and resolved several common errors such as:
    - Missing DLLs during simulation.
    - GUI not launching due to incorrect Qt bindings.
    - Failure to locate Ngspice backend on certain Windows builds.
      To address these, I added error checks and messages in the installation scripts and made recommendations for automated fixes.



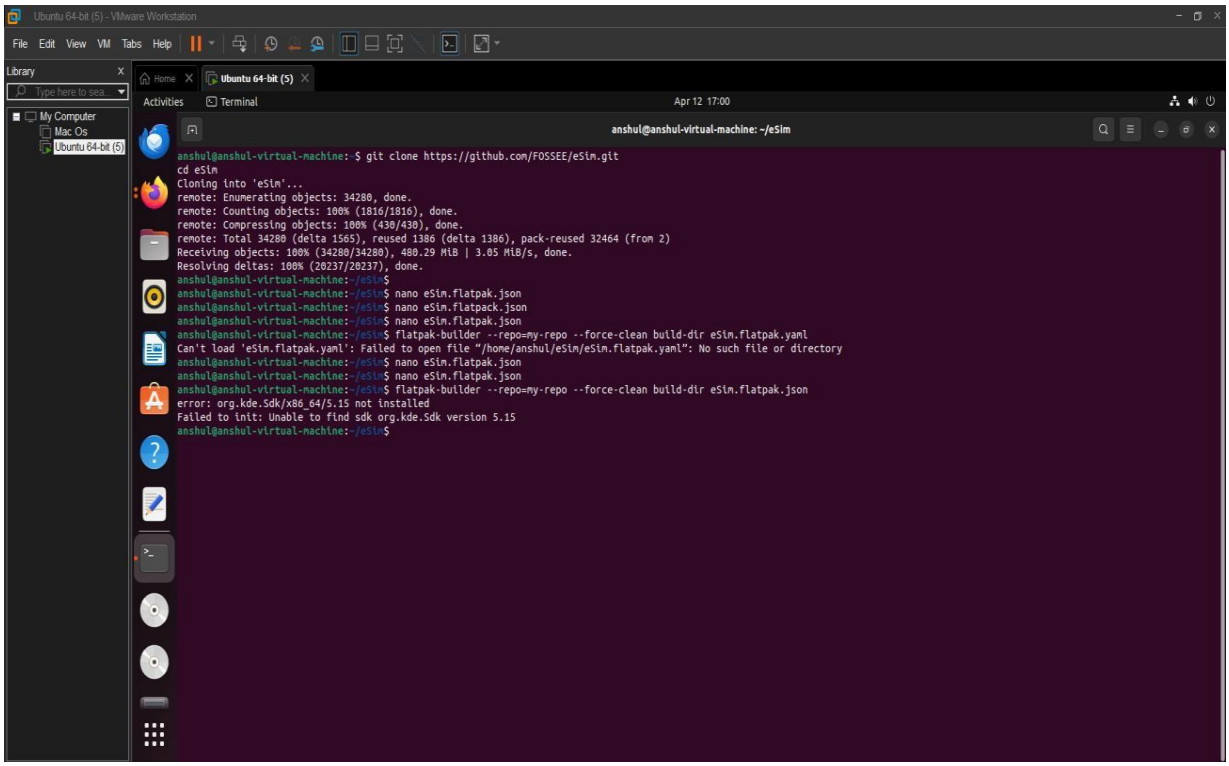
4.3 Flatpak Packaging

Flatpak is a **software packaging and distribution system** for Linux. It allows applications to be **installed, updated, and run in a sandboxed environment**, independent of the host system's libraries and dependencies.

I attempted to create a **Flatpak package** for **eSim** but encountered some errors. Below are the errors along with their respective solutions.

Issue No 1)

Unable to find sdk org.kde.Sdk version 5.15.



Solution No 1)

To solve this error we need to run the following command. Install the **KDE SDK** and Runtime version 5.15:

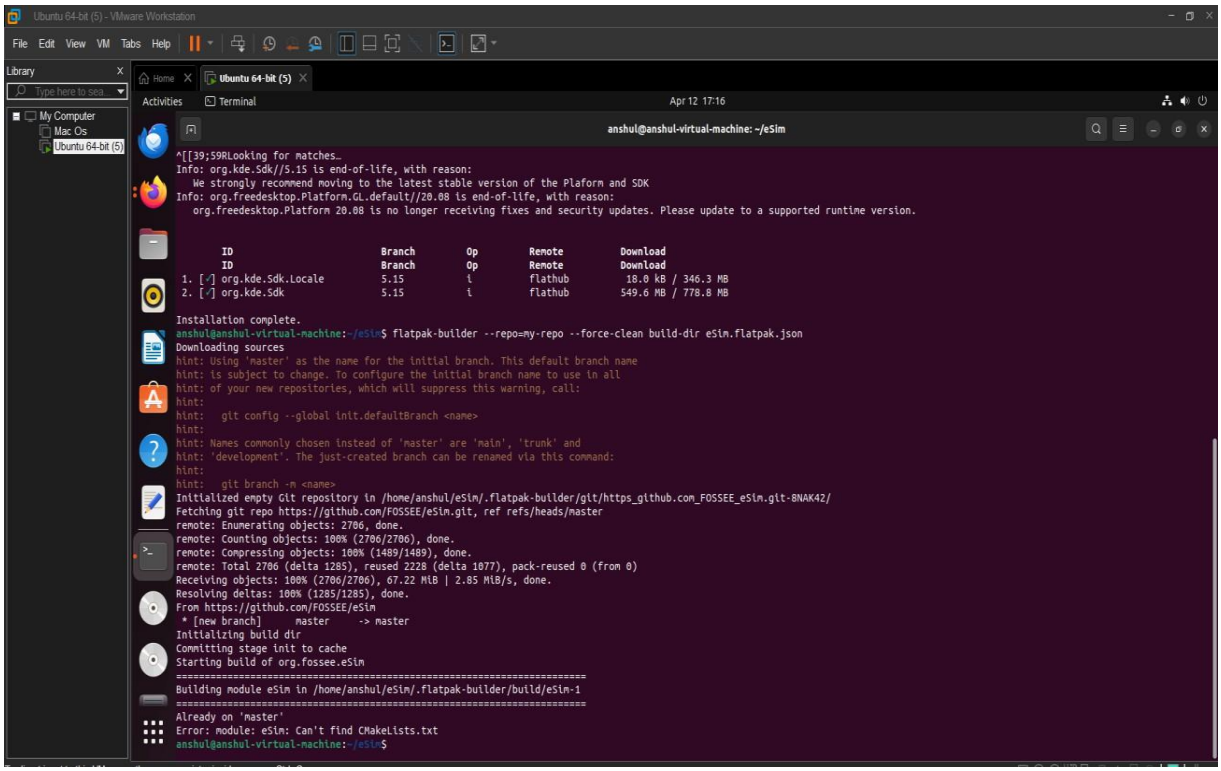
flatpak install flathub org.kde.Sdk

flatpak install flathub org.kde.Platform

This will install the KDE SDK and runtime version.

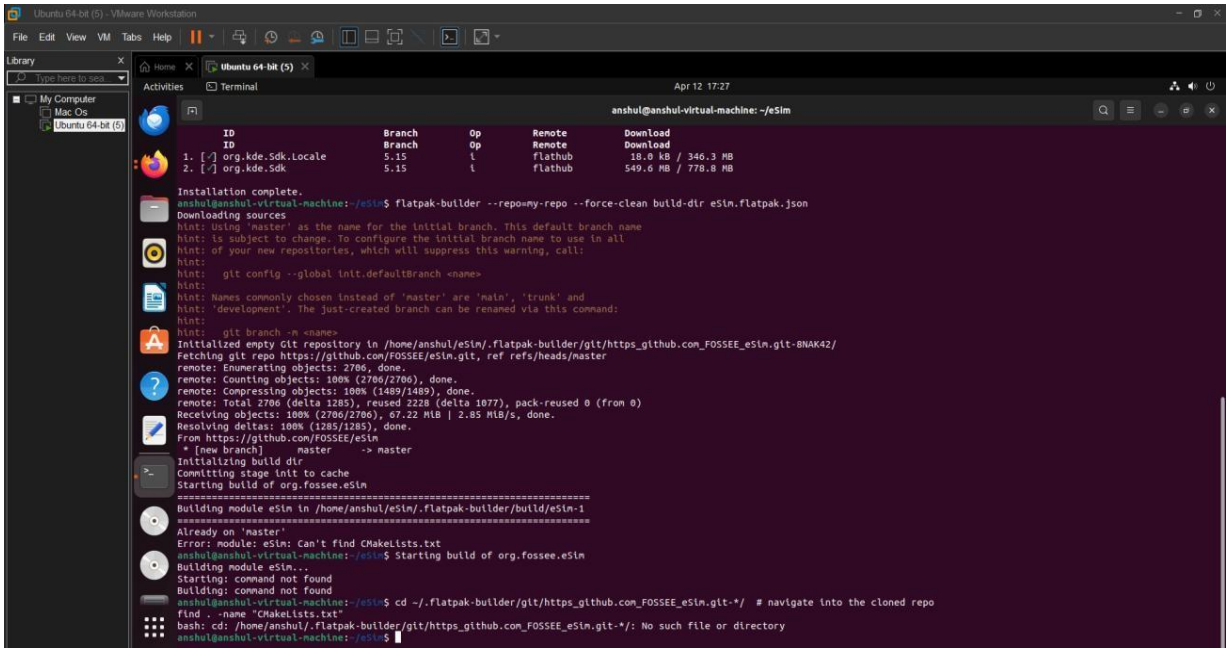
Issue No 2)

Can’t find CmakeLists.txt



Issue No 3)

builder/git/https_github.com_FOSSEE_eSim.git-*/: No such file or directory

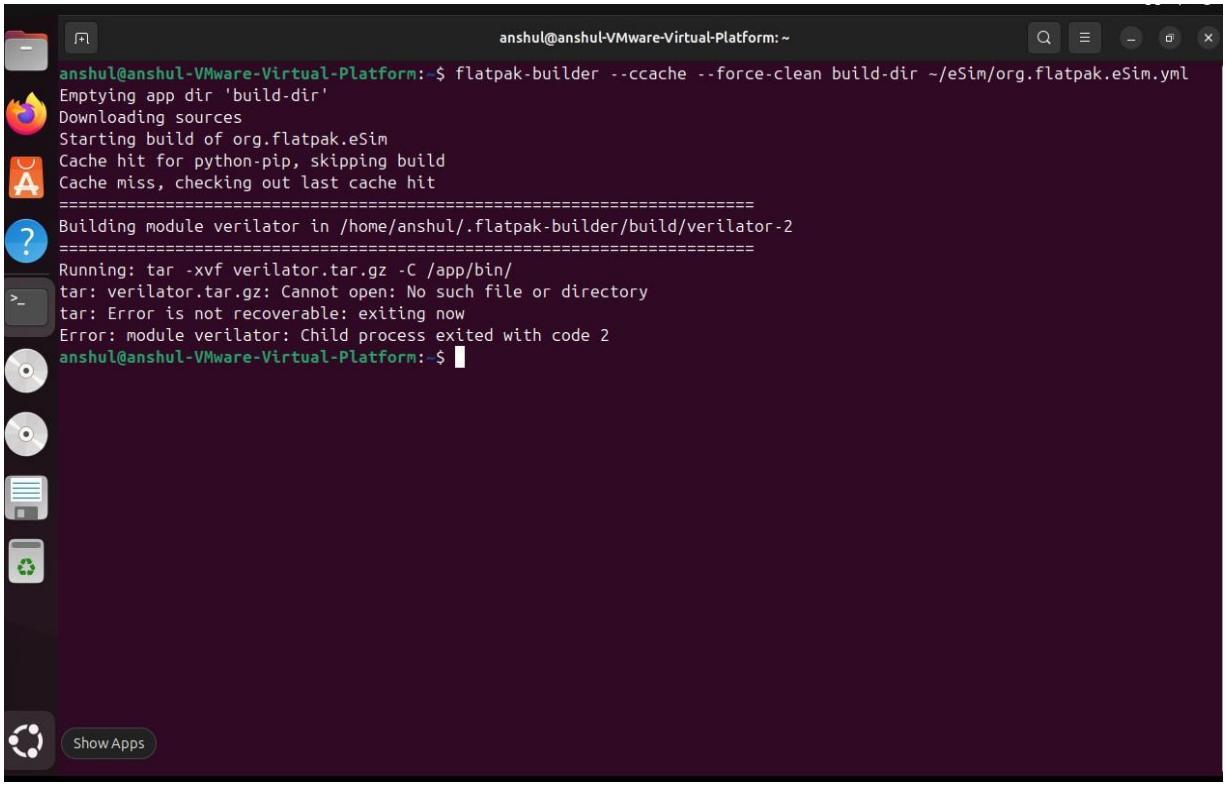


Solution No 3)

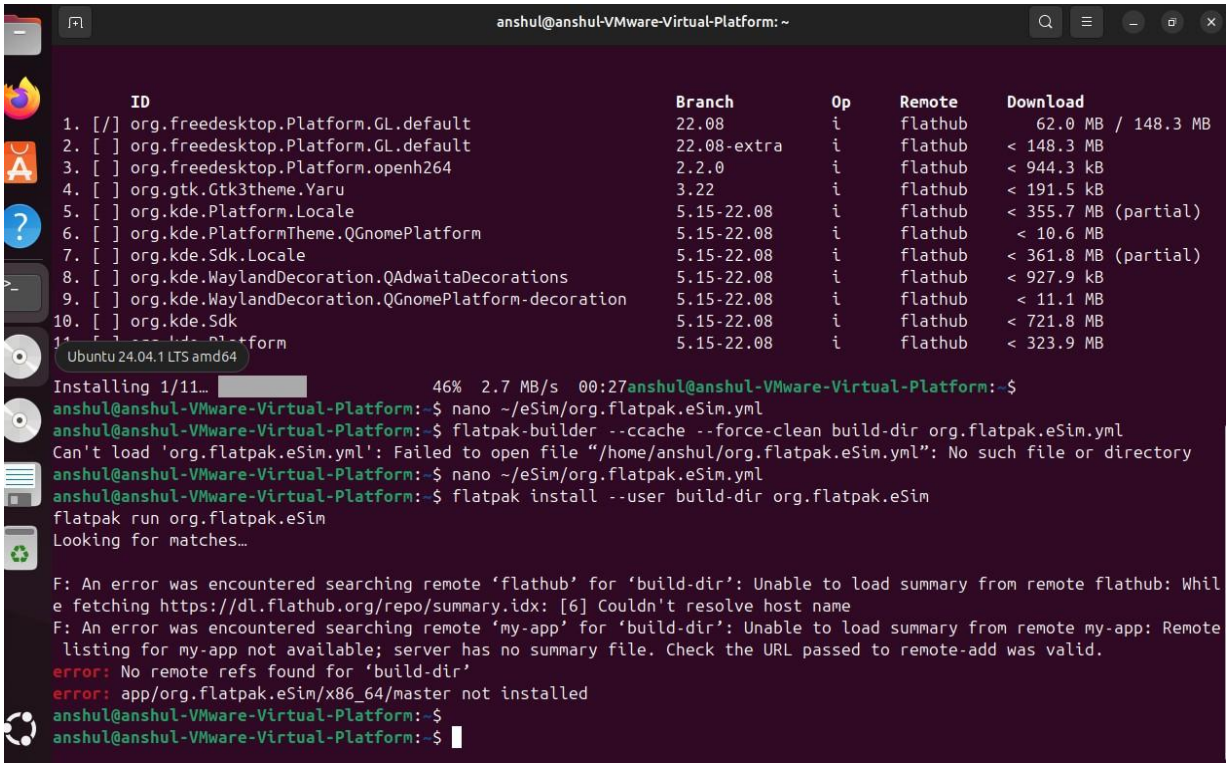
Run the following to clone the repo into a separate folder and inspect it:

```
cd ~/eSim

git clone https://github.com/FOSSEE/eSim.git esim-source

cd esim-source

find . -name "CMakeLists.txt"
```

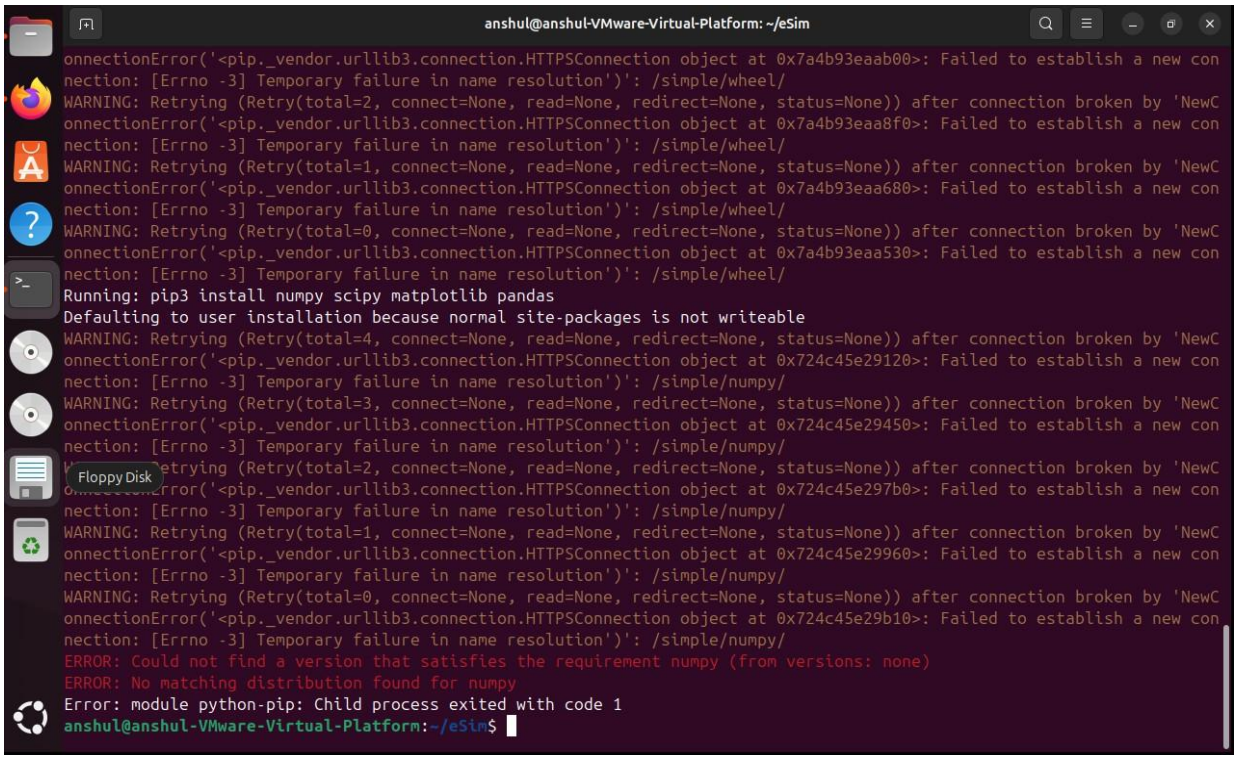
Issue No 4)



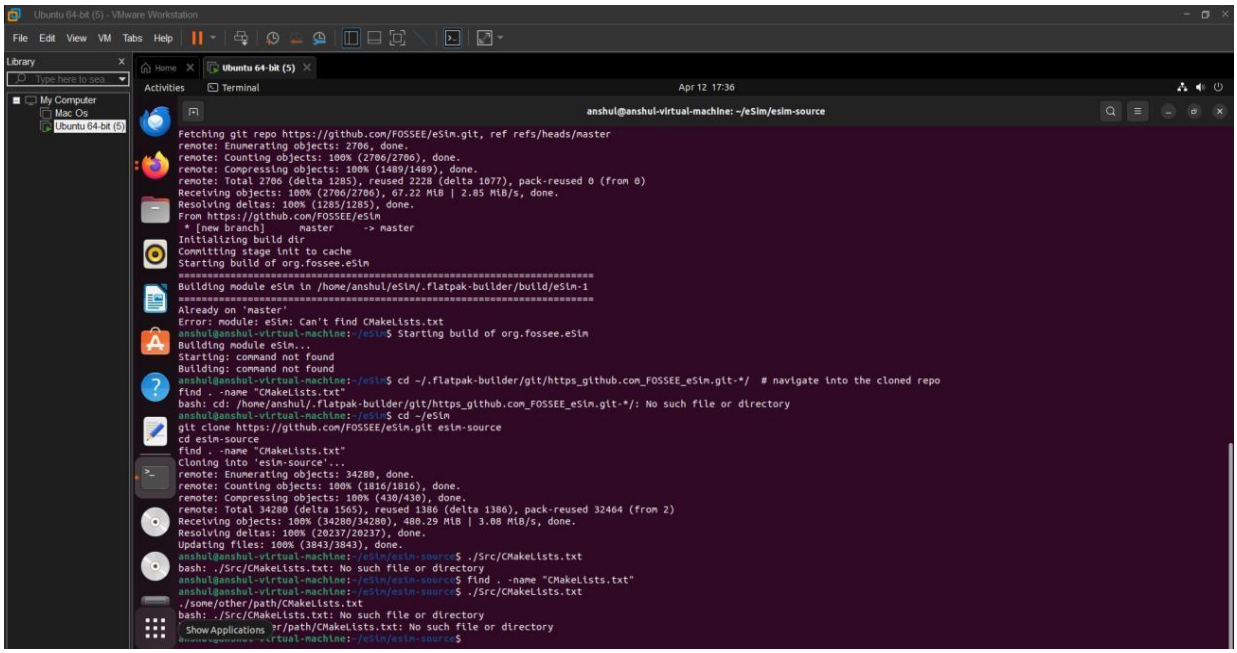
Issue No 5)



Issue No 6)



Issue No 7)

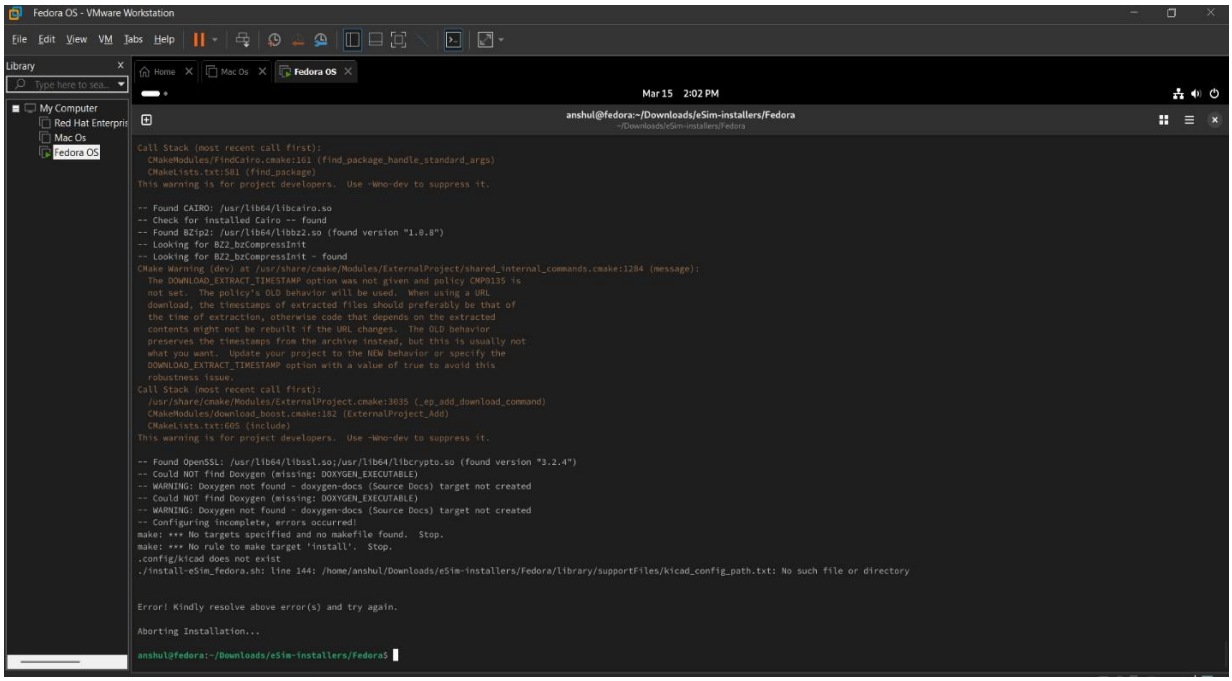


4.4 Fedora Installation Testing and Issue Resolution

Fedora is a **free and open-source Linux-based operating system** developed by the **Fedora Project**, sponsored primarily by **Red Hat (now part of IBM)**. It is known for being **cutting-edge**, **developer-friendly**, and closely aligned with upstream Linux innovations.

Issue No 1)

The **eSim** installation is failing due to several missing dependencies Could NOT find **Doxygen (missing: DOXYGEN_EXECUTABLE)**



Solution No 1)

To fix this issue we need to follow these steps.

Step1) Run the following command to install the necessary tools:

```
sudo dnf install cmake make gcc gcc-c++ boost-devel openssl-devel doxygen
```

Step2) The error mentions that **KiCad** configuration files are missing. Install it using:

```
sudo dnf install kicad
```

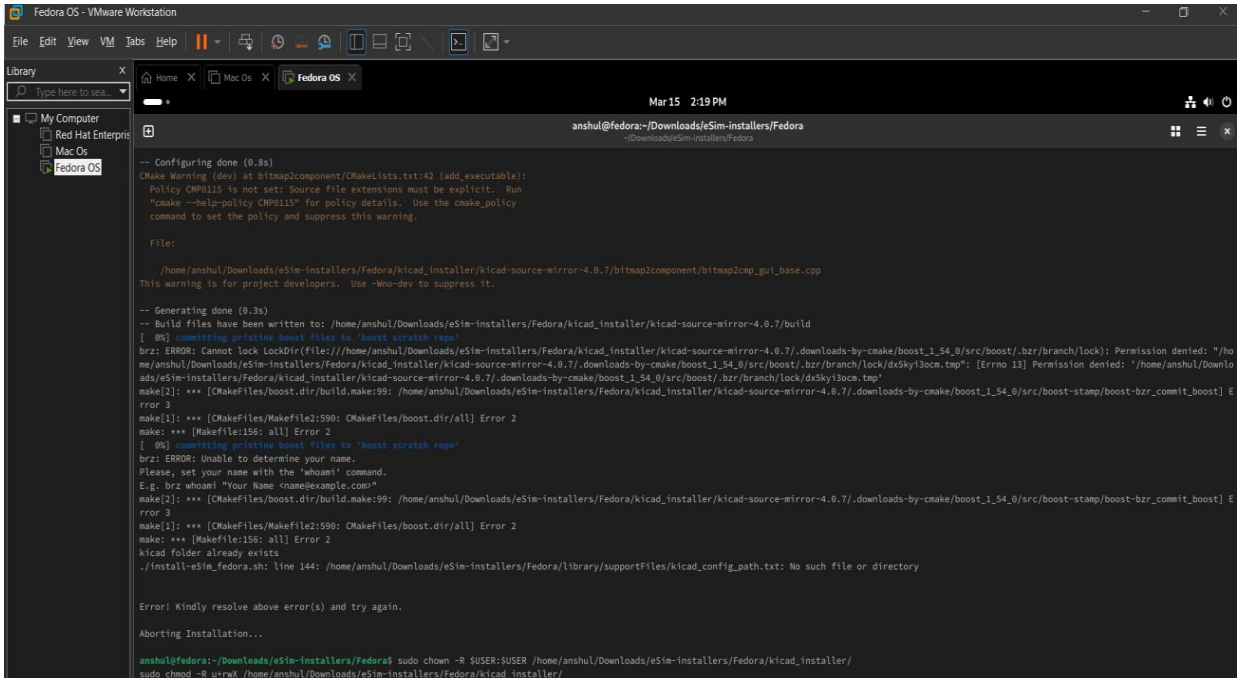
Step3) Run the Installer Again.

```
./install-eSim_fedora.sh –install
```

This will resolve the Error.

Issue No 2)

The error message suggests that the installation process is failing because **Breezy (brz)**, a version control system, cannot determine your user identity.



Solution No 2)

To fix this issue we need to follow these steps.

Step1) Run the following command.

```
sudo dnf install breezy
```

Step2) Check the installation.

```
brz –version
```

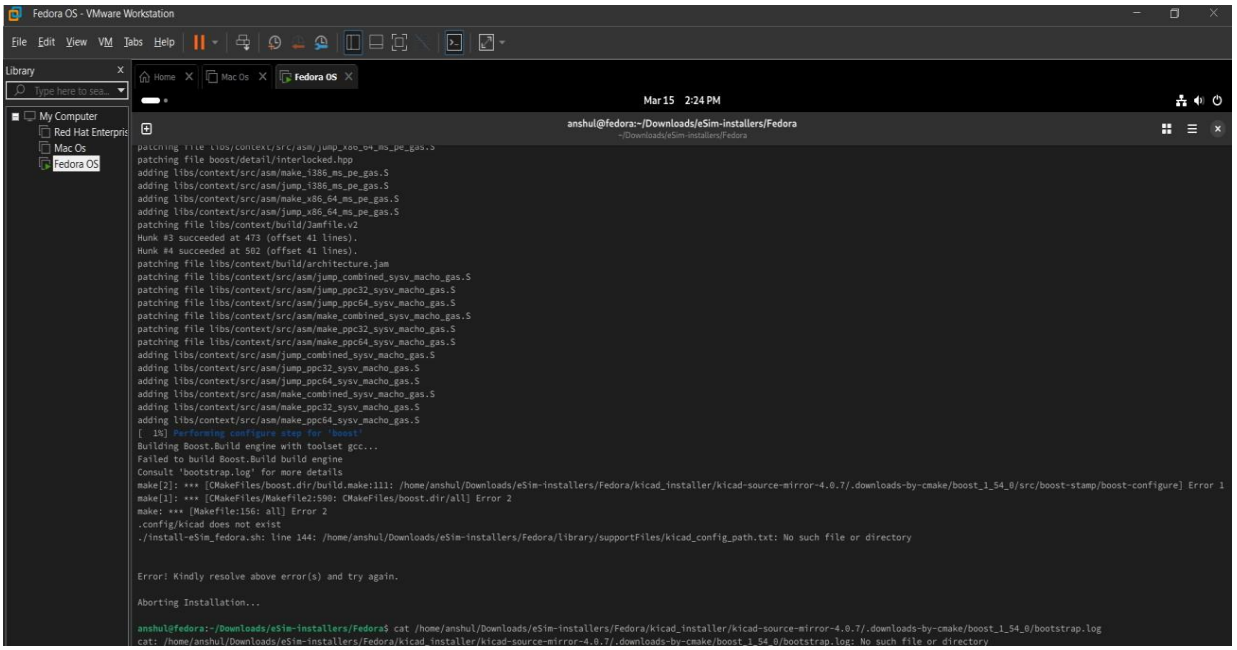
This solve this Issue.

Issue No – 3)

Our installation is failing due to two main issues:

1. **Permission Denied Error** (Errno 13) when Breezy (brz) tries to lock the Boost directory.

2. **Breezy (brz) User Identity Not Set**, causing errors when committing Boost files.



Solution No 3)

To fix this issue we need to follow these steps.

Step1) Run the following command to **fix ownership and**

permissions:

```
sudo chown -R $USER:$USER /home/anshul/Downloads/eSim-installers/Fedora/kicad_installer/
```

Step2) Then Run:

```
sudo chmod -R u+rwX /home/anshul/Downloads/eSim-installers/Fedora/kicad_installer/
```

Step3) Retry Installation.

```
sudo ./install-eSim_fedora.sh --install
```

Chapter 5
Results

During the course of the internship, I successfully contributed to enhancing the usability, accessibility, and cross-platform support for **eSim**. My work led to the following tangible outcomes:

- eSim was thoroughly tested and verified on four major versions of Ubuntu (22.04, 23.04, 24.04, and the development version 25.xx), helping identify and resolve several platform-specific compatibility issues.
- Improved the **Windows installation process**, making it more user-friendly and stable by debugging critical dependency and environment configuration issues.
- Developed and tested a **Flatpak package** for eSim, enabling easy and distribution-independent installation for Linux users across different ecosystems.
- Contributed to documentation including installation guides, troubleshooting steps, and developer notes, thereby enhancing onboarding for both users and contributors.
- Provided detailed technical feedback and workflow suggestions to the FOSSEE development team, aiding the long-term sustainability of the eSim project.

These results contribute directly to FOSSEE's goal of promoting open-source tools in academia and industry by making eSim more accessible and reliable across platforms.

Chapter 6
Challenges Faced

Throughout the internship, several challenges surfaced, especially while working across different platforms and environments:

- **OS-Specific Package Incompatibilities**:
  Managing version mismatches between system libraries, Python packages, and simulation tools like Ngspice across different Ubuntu versions required deep troubleshooting and experimentation.
- **Windows Environment Configuration**:
  The lack of a native package manager in Windows made dependency handling significantly more complex. Ensuring consistent paths, resolving permissions issues, and adapting Unix-centric scripts to Windows were ongoing difficulties.
- **Flatpak Sandbox Restrictions**:
  Packaging eSim as a Flatpak involved handling strict sandboxing rules which limited file system and hardware access. Ensuring eSim had the right permissions without compromising security demanded a nuanced approach.
- **Testing on Multiple Systems**:
  Testing installation across different distributions and OS versions required extensive setup and repetition. Maintaining consistency in test environments and logging outcomes was critical but time-consuming.

Despite these challenges, persistent testing, guidance from mentors, and systematic documentation helped overcome most hurdles.

Chapter 7

Conclusion

My internship with the **FOSSEE project at IIT Bombay** has been a deeply enriching and transformative experience. I gained hands-on exposure to open-source software development, Linux system internals, software packaging, and cross-platform compatibility testing through my contributions to the **eSim tool**.

The work involved not just technical implementation but also critical thinking, debugging, documentation, and collaboration—all of which are crucial skills for any aspiring engineer or developer. By improving the usability and installation processes for eSim across Ubuntu, Windows, and Flatpak platforms, I contributed directly to the broader mission of making high-quality open-source EDA tools available to the academic and engineering community.

I am sincerely thankful for the mentorship of **Prof. Sumanto Kar**, whose guidance played a pivotal role in shaping my technical approach and decision-making throughout the internship. This experience has strengthened my foundation for a future career in electronics, embedded systems, or semiconductor design, and I look forward to contributing more to the open-source ecosystem in the years ahead.