Linux Kernel Compilation Assignment

Version: 6.11.1

Author: **Tazmeen Afroz**

Academic Institution: **FAST NUCES**

Instructor: **Saad Ahmad**

October 3, 2024

Contents

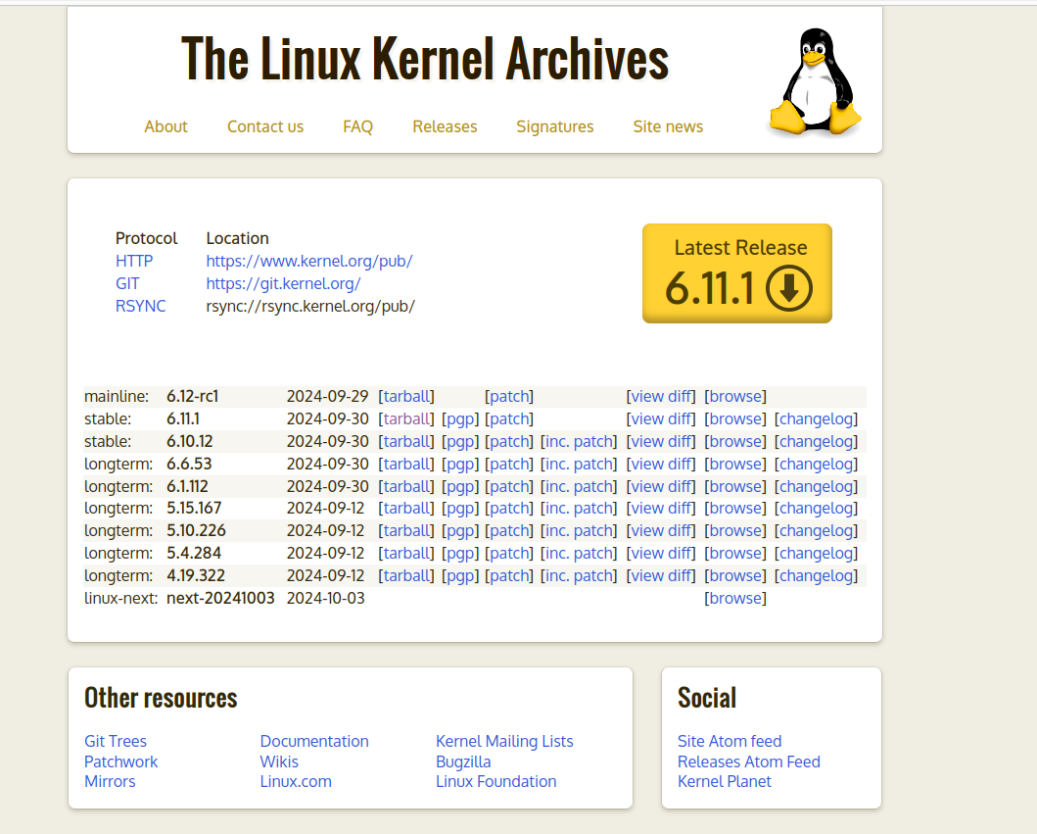
1. Introduction

This document provides a comprehensive, step-by-step account of compiling and installing Linux kernel version 6.11.1. The primary objective was to replace the default kernel with a custom-built version, incorporating specific tweaks for performance enhancements and functionality improvements. This assignment was completed as part of the Operating Systems Lab (OSLab) course.

2. Detailed Process

2.1. Downloading the Kernel Source

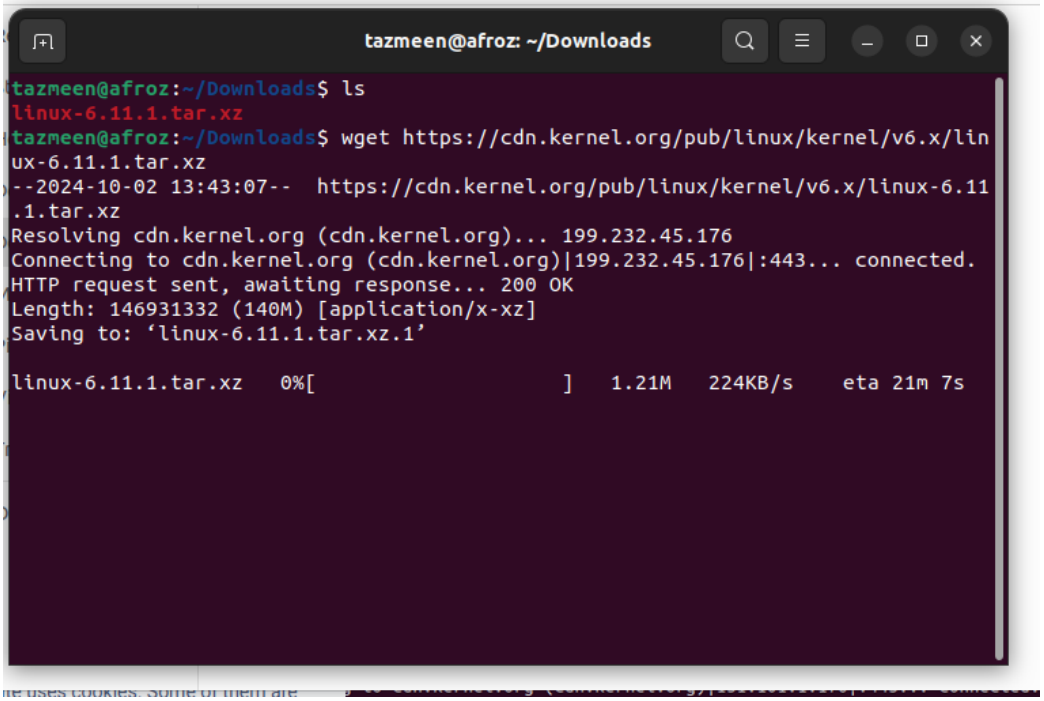
The first step involved downloading the Linux kernel source from the official repository at kernel.org. For this assignment, version 6.11.1 was used.

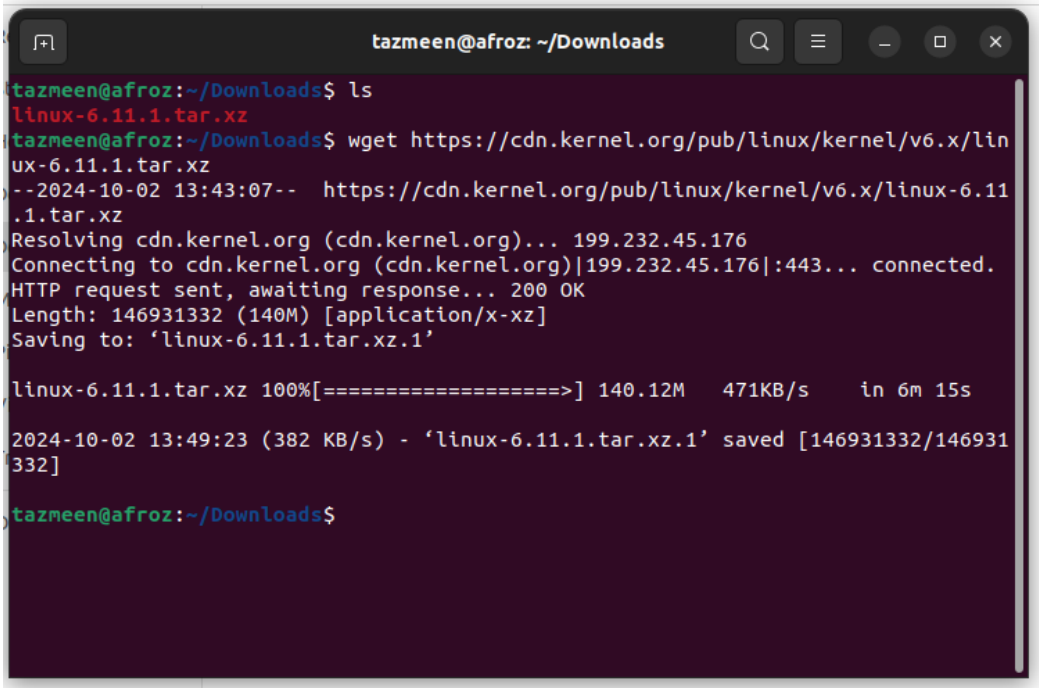


Linux Kernel Compilation Assignment

Open the terminal and use the wget command to download the Linux kernel source code:

```
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux
   -6.11.1.tar.xz
tar xvf linux-6.11.1.tar.xz
```



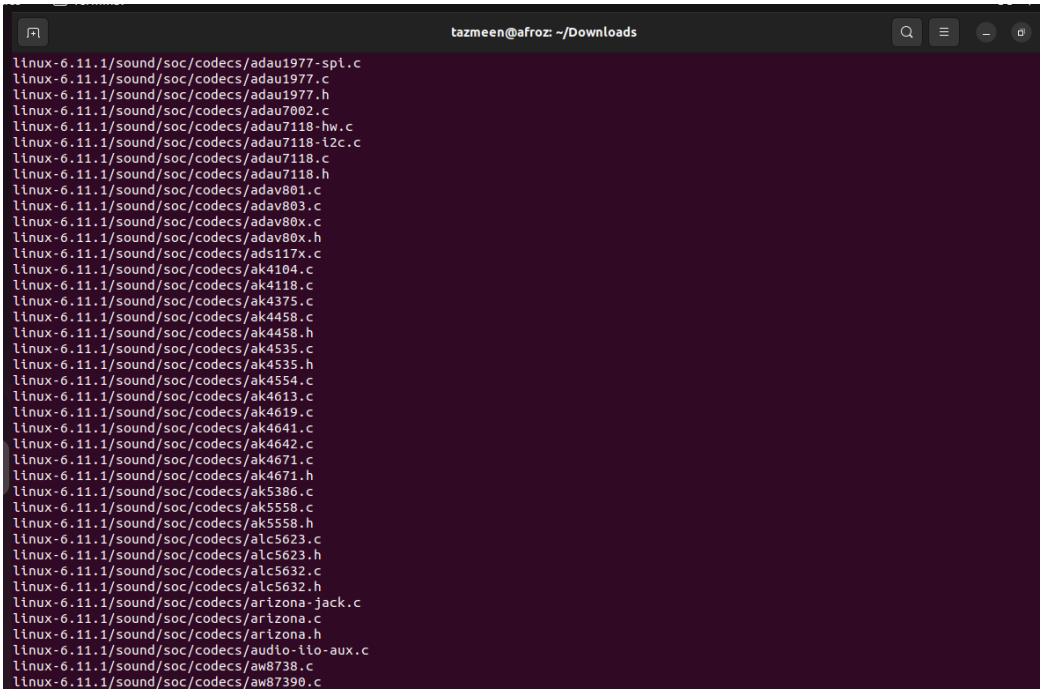


2.2. Extracting the Source Code

run the tar command to extract the source code:

Open the terminal and use the wget command to download the Linux kernel source code:

```
tar xvf linux-6.11.1.tar.xz
```

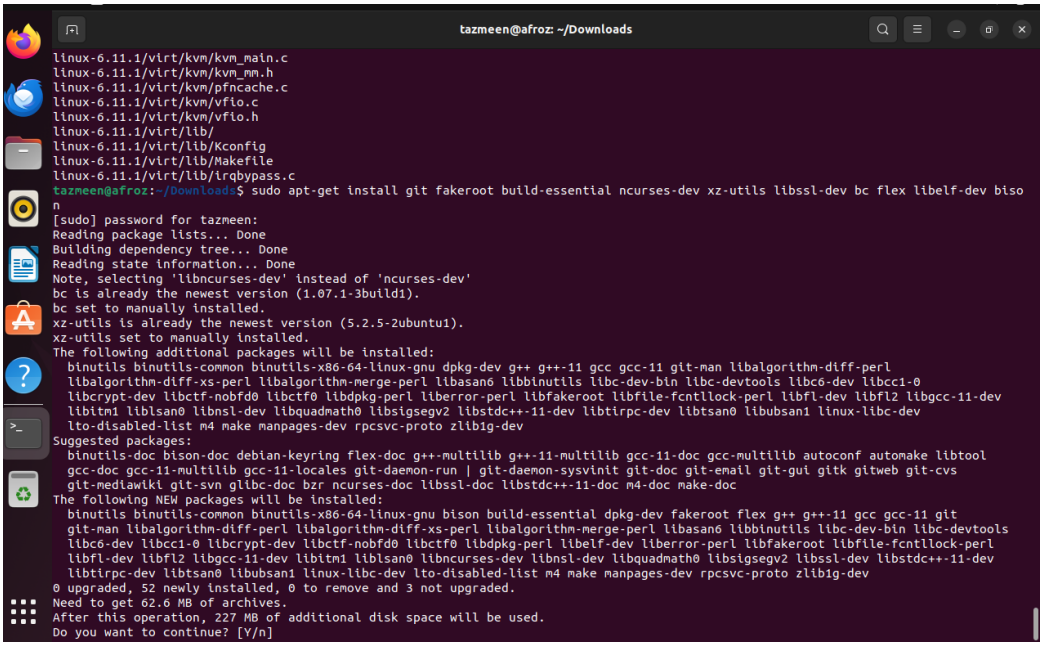


2.3. Installing Dependencies

Before proceeding with kernel compilation, the following dependencies were installed:

```
sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison
```

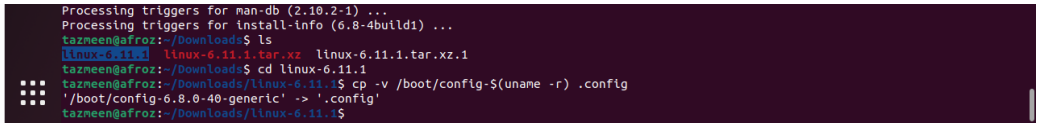
This command ensures all necessary tools and libraries are available for the compilation process.



2.4. Configuring the Kernel

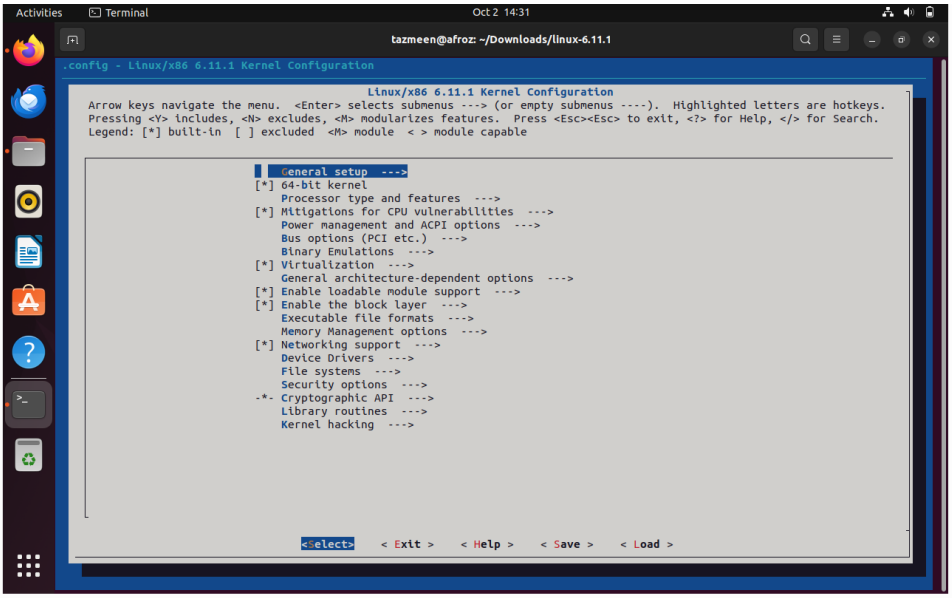
After installing dependencies, I navigated to the extracted kernel source directory and copied the existing system configuration:

```
cd linux-6.11.1
cp -v /boot/config-$(uname -r) .config
```



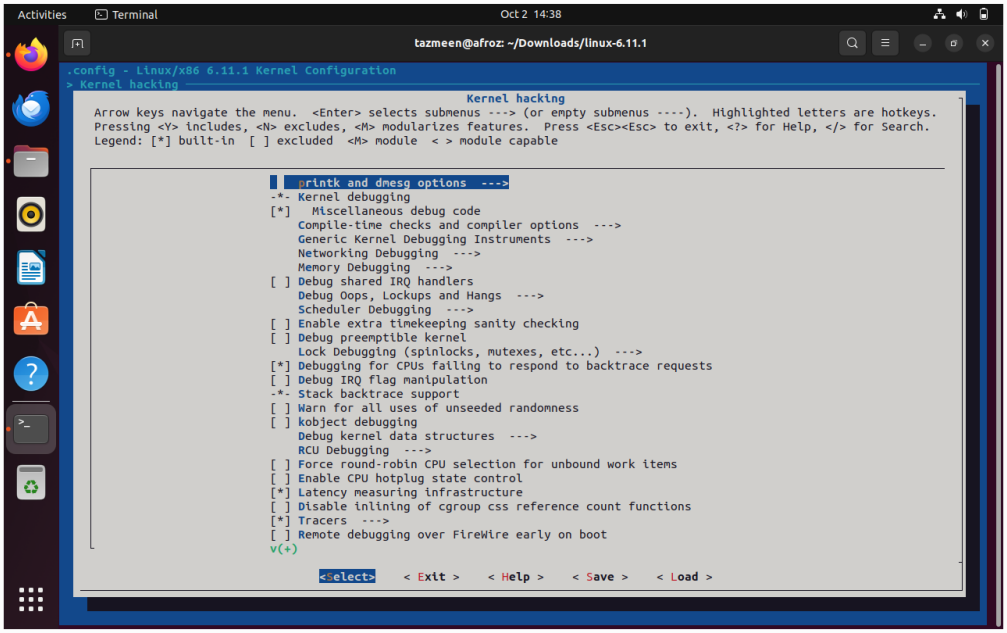
To customize the kernel settings, I used the menuconfig interface:

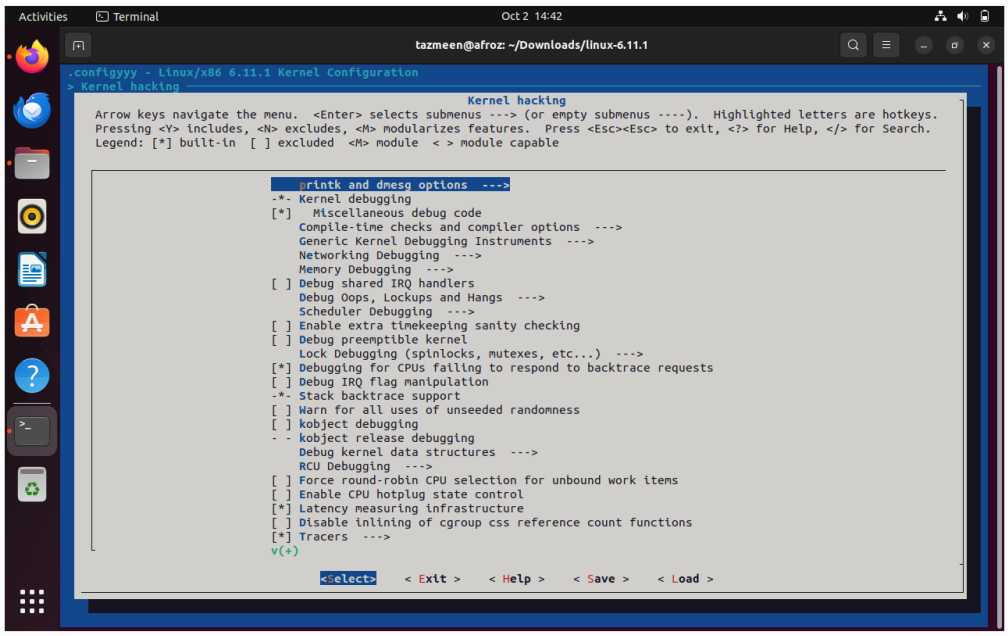
```
make menuconfig
```

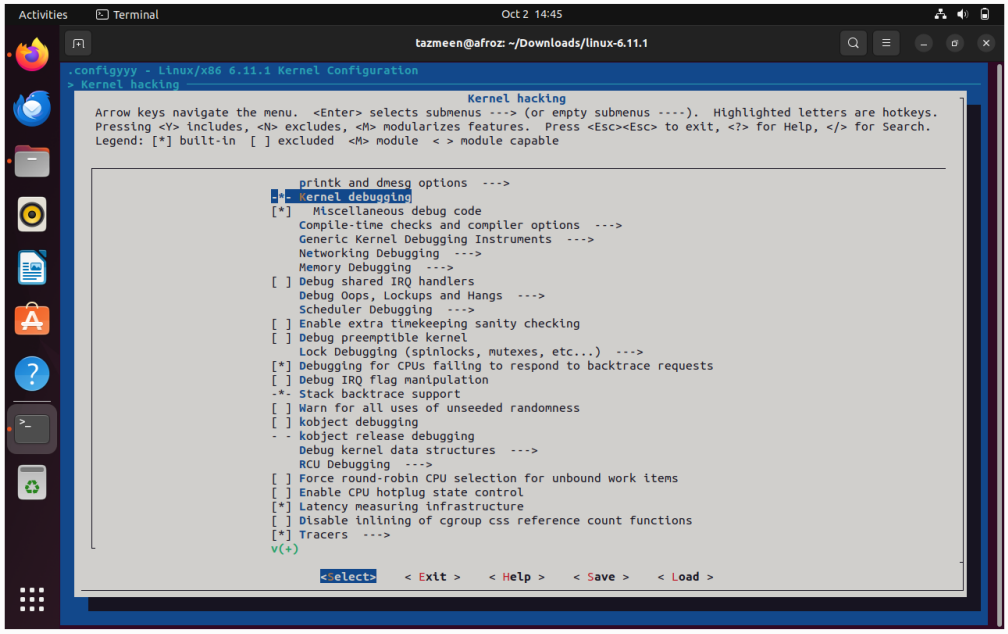


2.4.1. Bonus Tasks

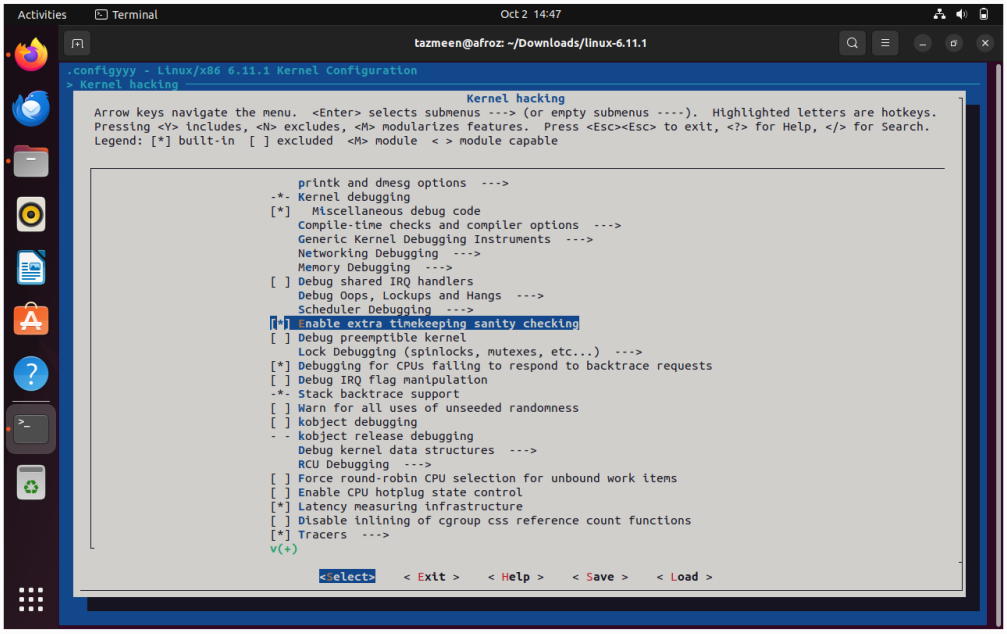
In the configuration menu, I made the following custom adjustments:

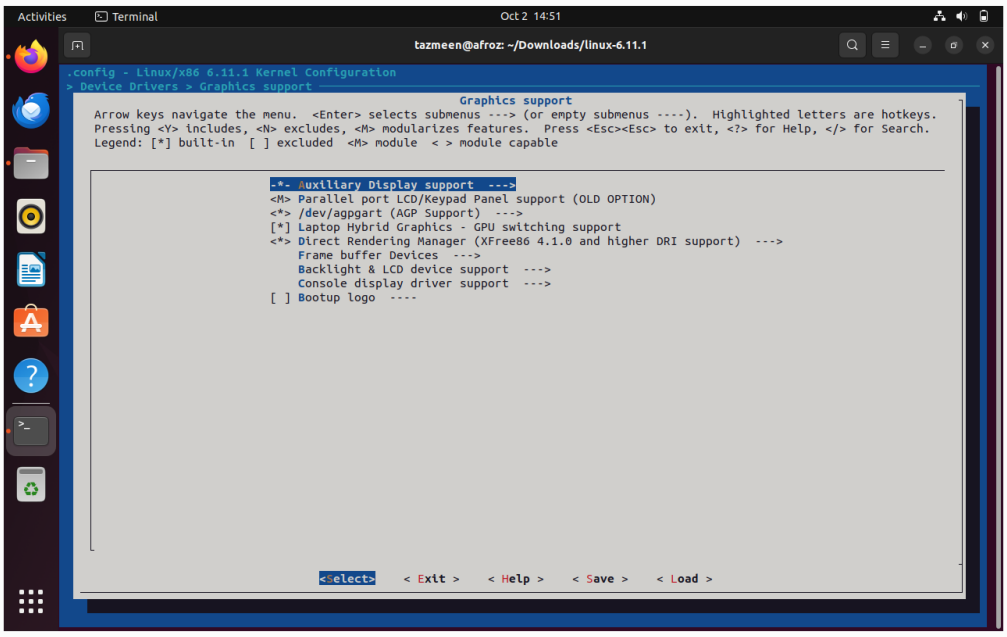




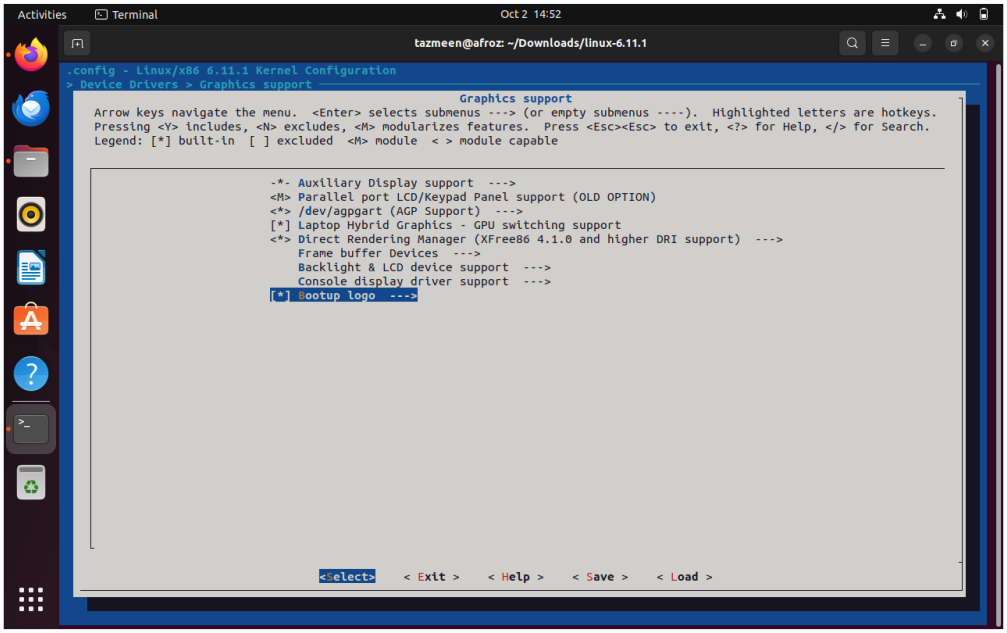


1. Enabled extra time keeping sanity check in kernel hacking

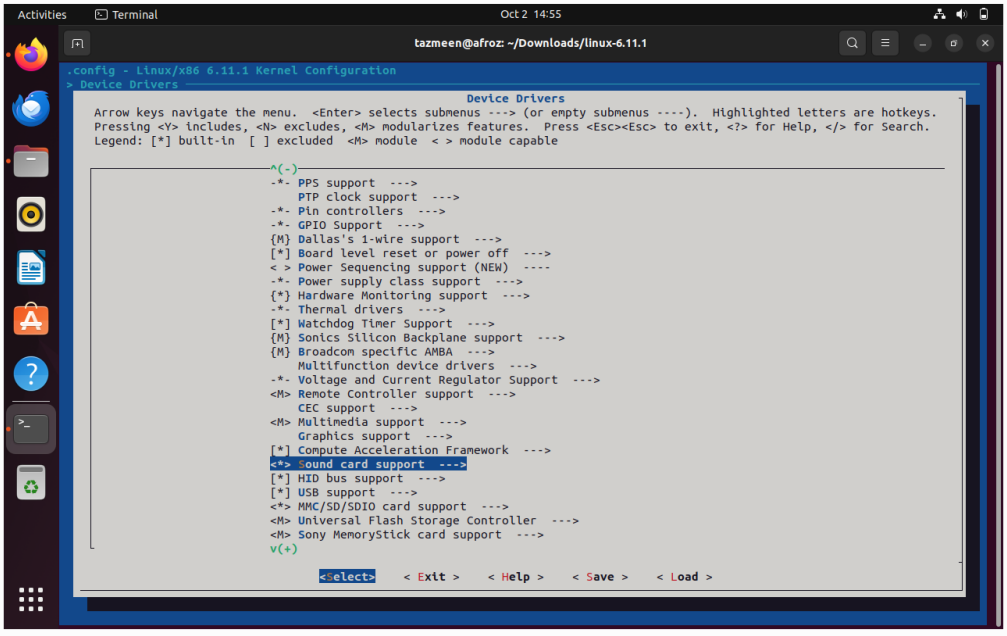




2. Enable bootup logo in graphics support



3. Adjusted sound card settings for improved audio performance



The modified configuration was saved and applied.

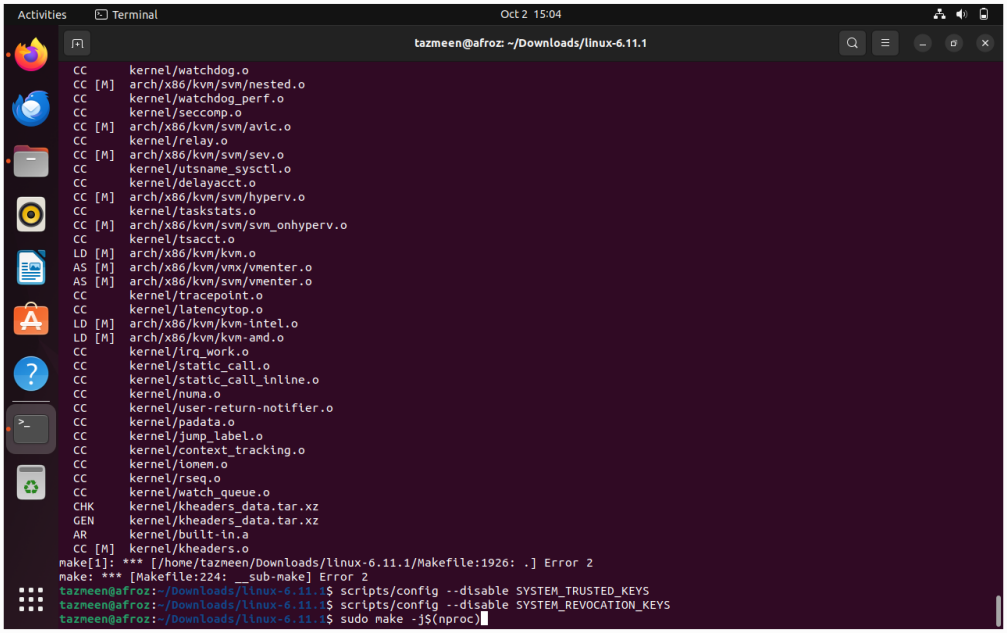
2.5. Compiling the Kernel

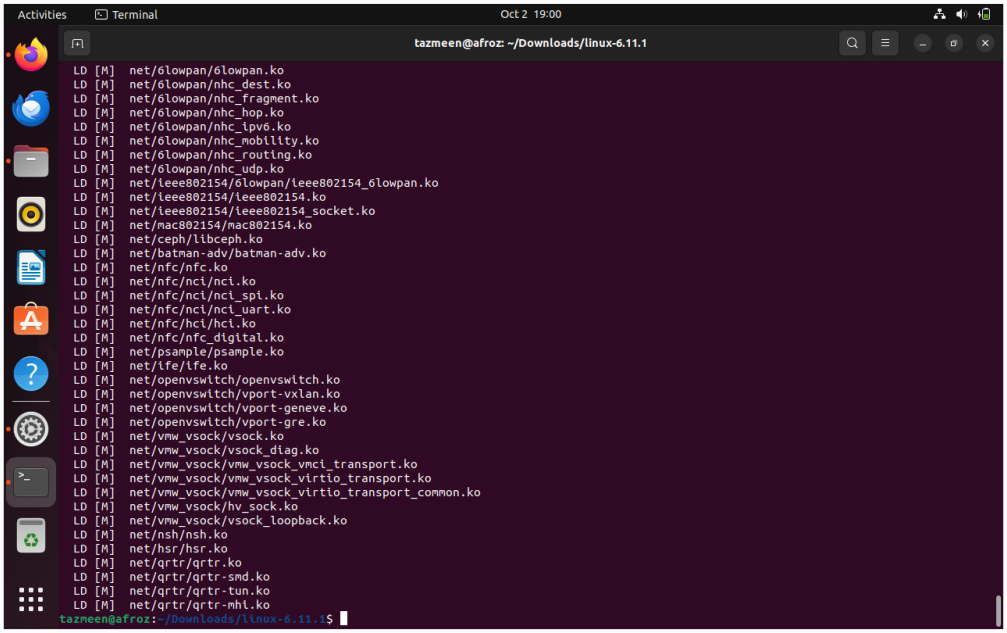
The kernel compilation process was initiated using the following command:

```
make -j$(nproc)
```

This command utilizes all available CPU cores for faster processing. The compilation process took approximately 3 hours and 30 minutes to complete.

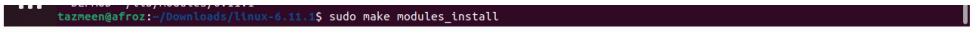
In case of make error run following commands

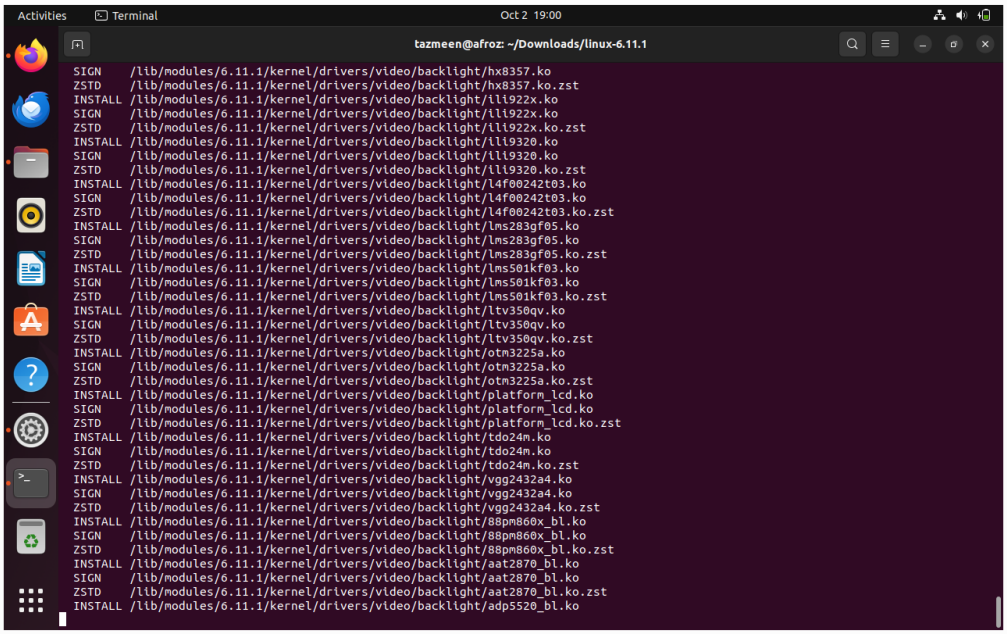


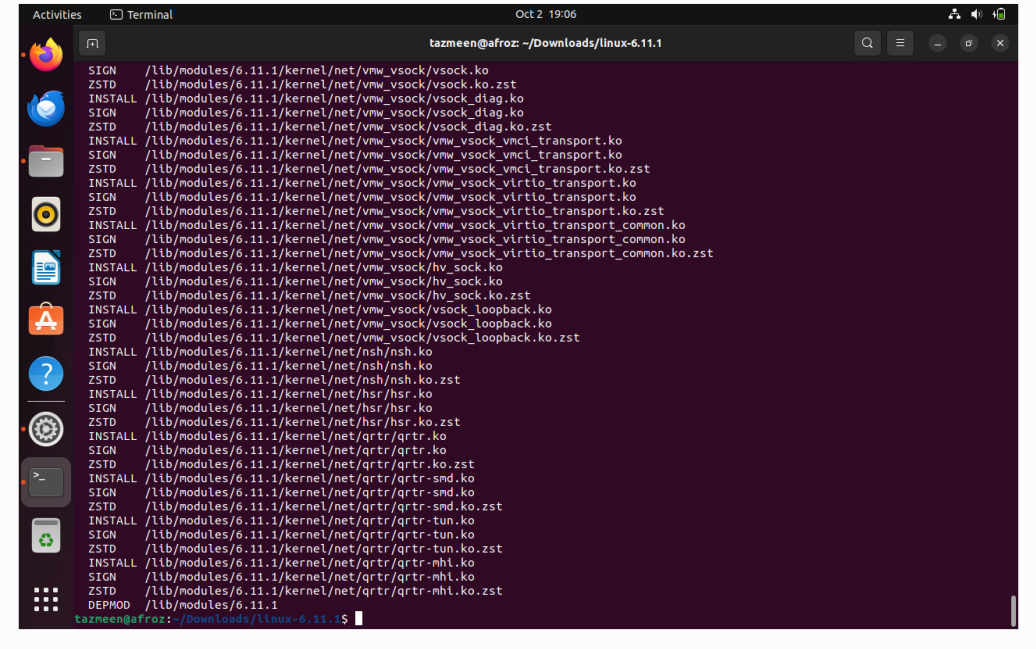


After the main compilation, I installed the required modules:

```
sudo make modules_install
```

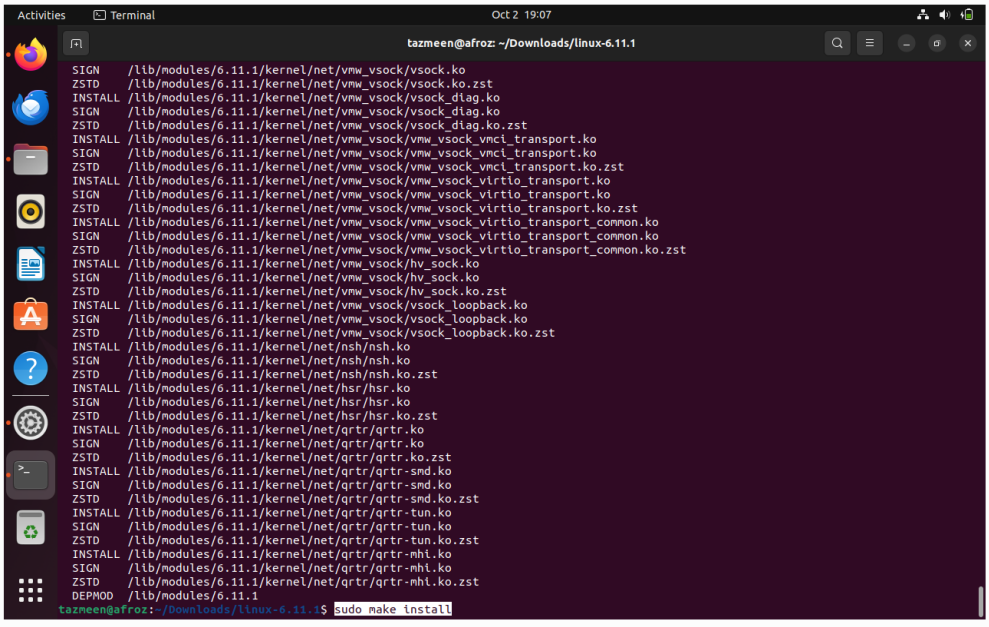


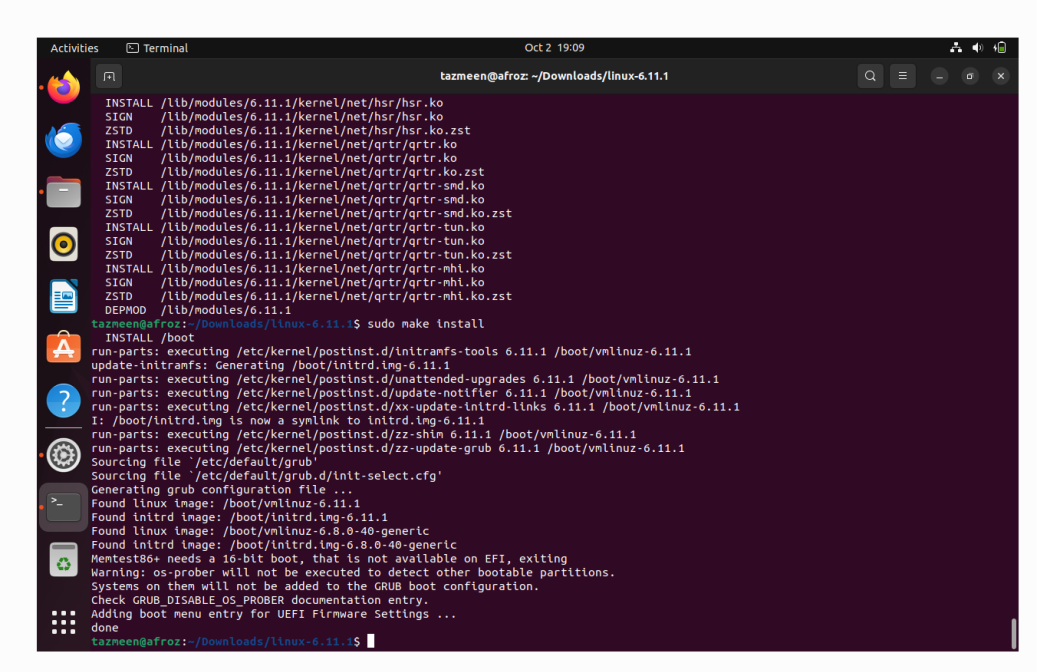




Finally, the kernel itself was installed:

```
sudo make install
```





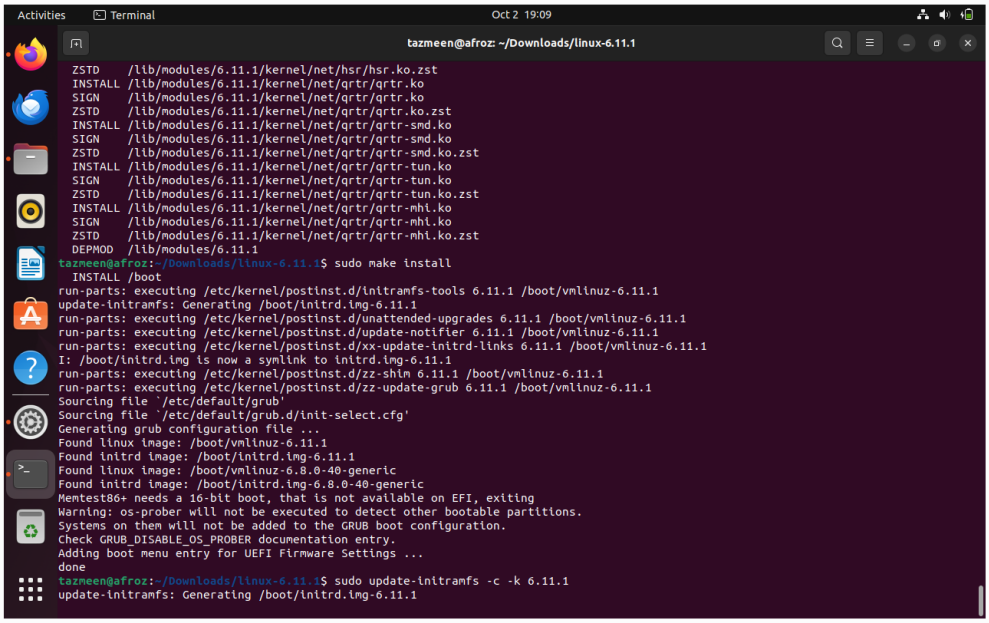
2.6. Updating the Bootloader

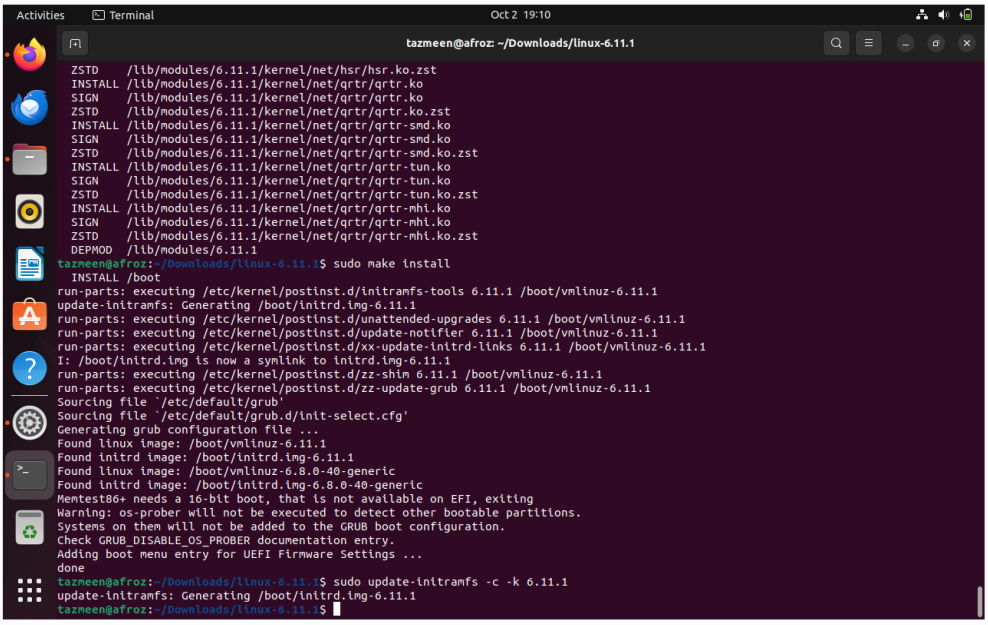
Once the kernel was installed, I updated the bootloader configuration to include the newly installed kernel:

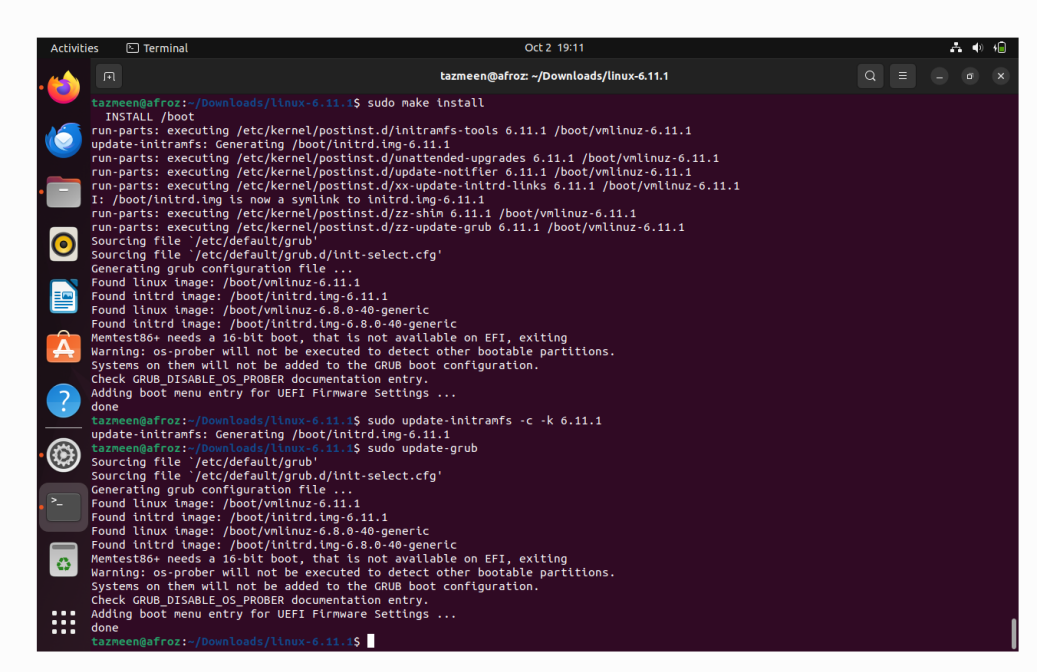
```
sudo update-initramfs -c -k 6.11.1
sudo update-grub
```

These commands create a new initial RAM disk for the new kernel and update the GRUB bootloader configuration.

Linux Kernel Compilation Assignment







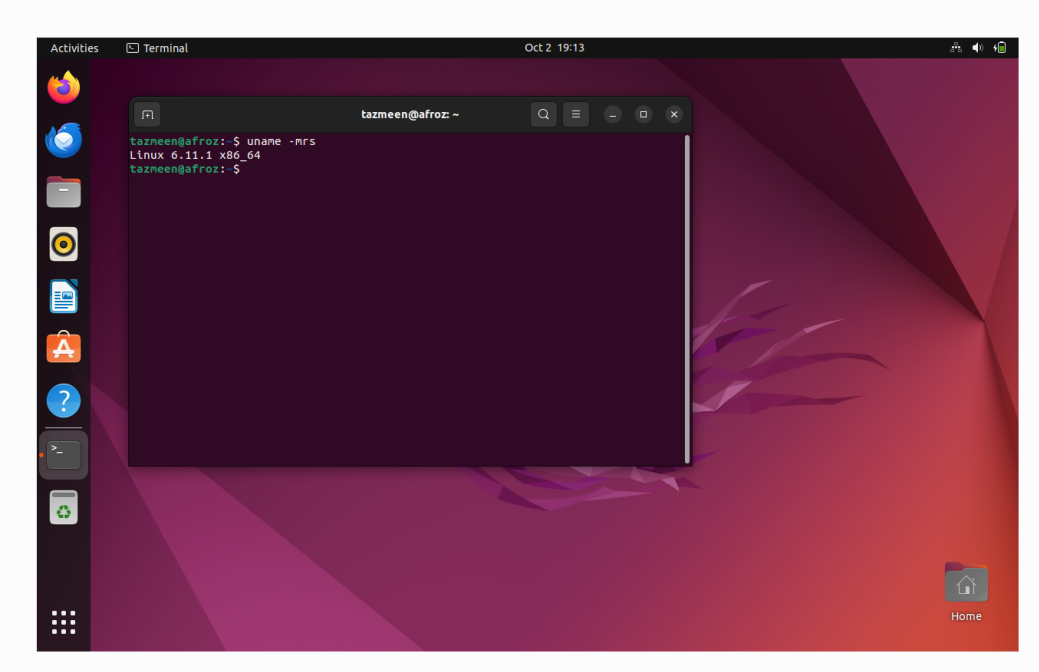
2.7. Rebooting and Verification

After completing the installation and bootloader update, I rebooted the system:

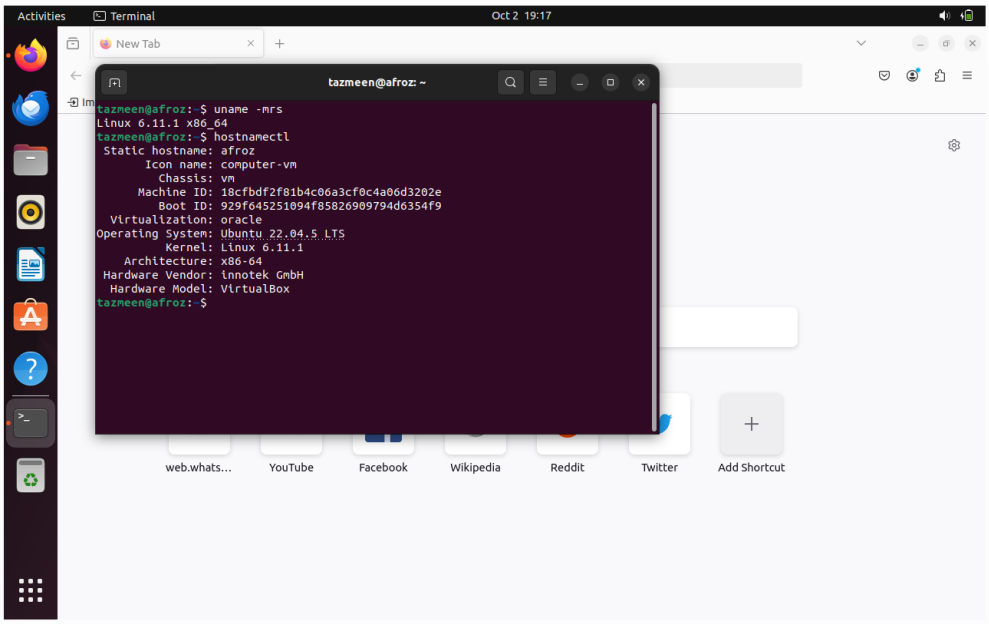
```
sudo reboot
```

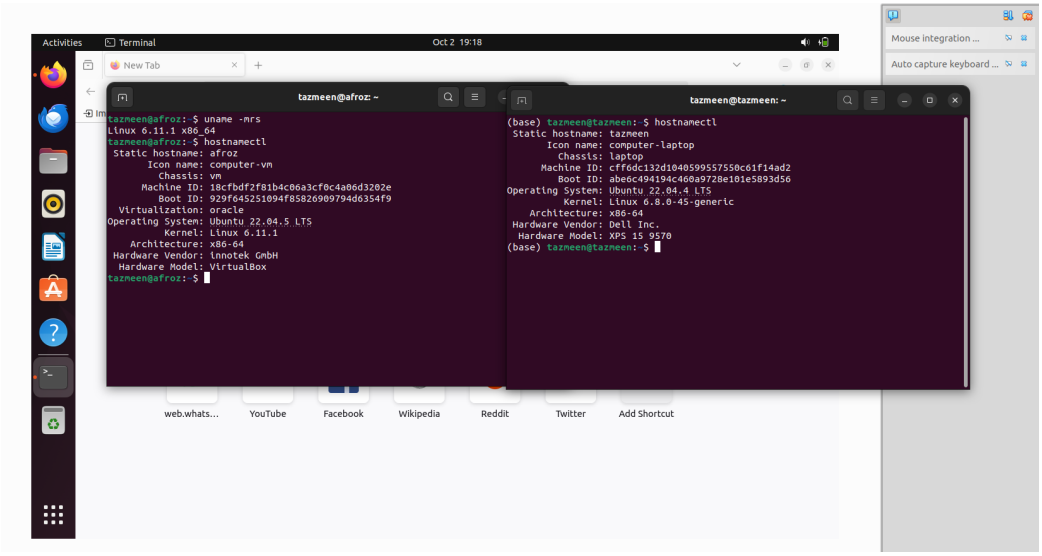
Upon rebooting, I confirmed that the new kernel was active by running:

```
uname -r
```



The output confirmed that kernel version 6.11.1 was successfully running.



comparison 

3. Compilation Time and Effort

The entire process, from initial setup to final verification, took 4 hours. This included:

- 15 minutes for setup and dependency installation
- 10 minutes for kernel configuration
- 3 hours for compilation and module installation
- 5 minutes for bootloader updates and final checks

The process required significant attention to detail and patience, especially during the lengthy compilation phase.