Lab01

Ahmed Siradj Eddine Bekkari

November 2024

Source Code Download

Download Methods:

I have used the `wget` command to download the **linux kernel**.Then i extracted it.

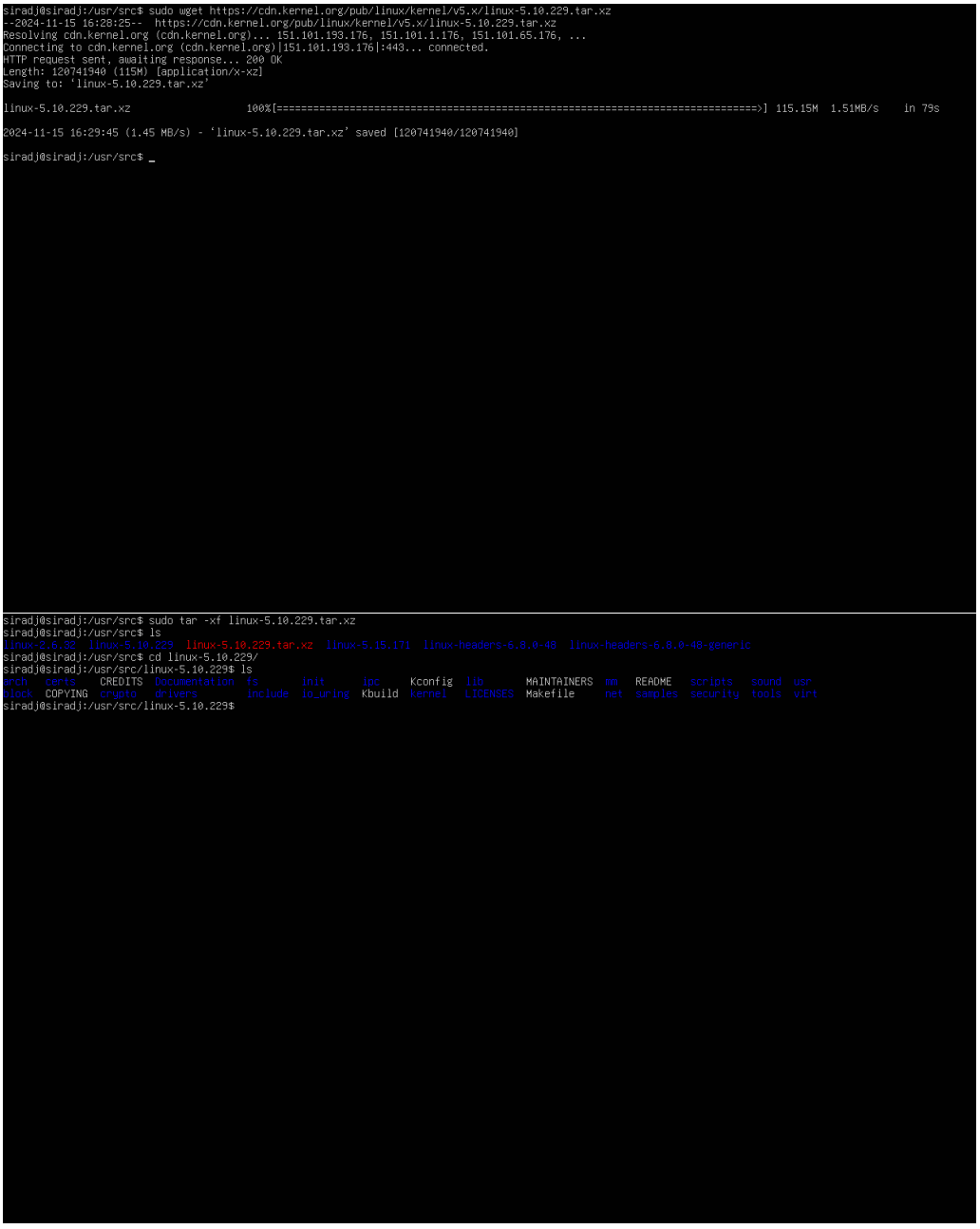


Figure 1: Download And Extract

Source Code Structure

Files and Directorie description:

- **arch/**: Contains architecture-specific code for various platforms (e.g., x86, ARM, MIPS). Each subdirectory corresponds to a specific architecture, housing code optimized for that architecture's needs.
- **drivers/**: Houses device driver code for various hardware components like network interfaces, USB, sound, and storage. This is where hardware-specific control logic is implemented.
- **include/**: Contains header files with declarations and macros used throughout the kernel. These headers are shared across different parts of the kernel, ensuring consistency and modularity.
- **kernel/**: Holds core kernel functionalities and subsystems, including process scheduling, signals, and low-level system management.
- **Documentation/**: Provides detailed documentation, guidelines, and explanations for kernel modules, APIs, and development practices. Essential for developers to understand kernel features and coding standards.
- **Makefile**: The main build script used to compile the kernel. It orchestrates how different components are compiled, assembled, and linked, handling dependencies and configuration settings.

The location of drivers for your hardware is in `/dev`, it's stand for **devices**

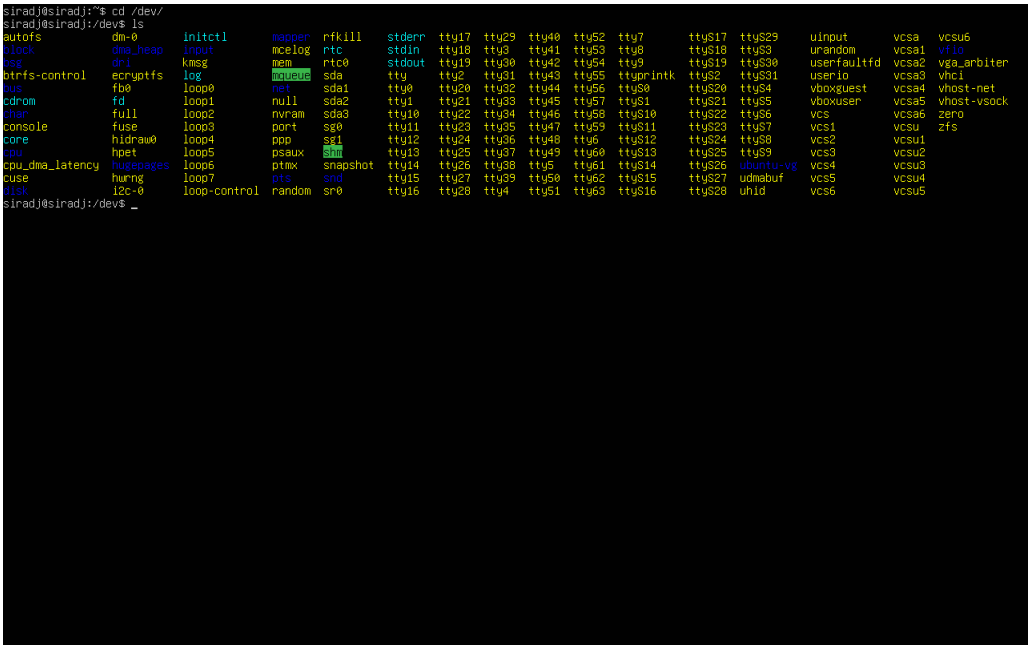


Figure 2: Drivers path

Kernel Configuration

Understanding Configuration Options

- **CONFIG_*:** In the Linux kernel, configuration options are prefixed with CONFIG_. These are settings used to enable or disable specific kernel features and modules. Each CONFIG_* option corresponds to a feature, hardware driver, or subsystem within the kernel.
  For example : **CONFIG_NET** enables networking capabilities.

- **Option types:**
  - **y:** Yes. Compiles the feature directly into the kernel.
  - **m:** Compiles the feature as a loadable module, meaning it can be loaded and unloaded from memory as needed.
  - **n:** No. Disables the feature entirely.

- **Configuration locations:** On `.config` file, for example in my machine i have found it on this path :
  `/usr/src/linux-5.10.229/.config`

Figure 3: config file path

Configuration Methods:

I will use the default configuration using this command `sudo make defconfig`

Figure 4: default config

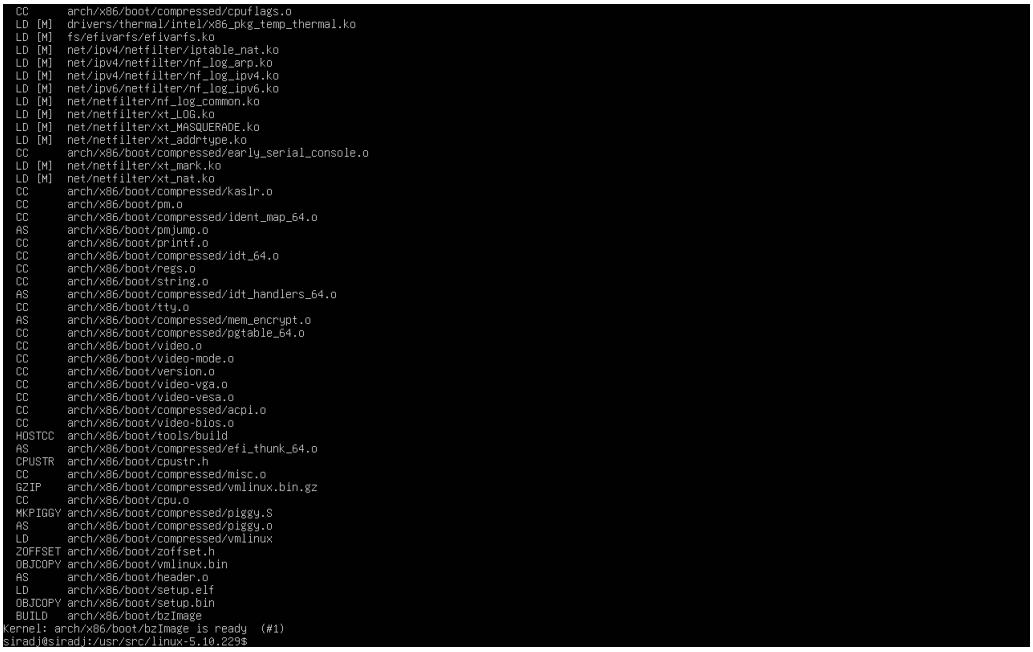
Essential Configuration Options:

- **Processor type and features:** This option determines the type of CPU your system has and enables/disables various CPU features. It's necessary to ensure compatibility and optimal performance.
- **Clock and power management:** This option includes settings for power saving and frequency scaling. It's necessary to manage power consumption and heat generation, especially in laptops and mobile devices.
- **Memory management:** This option includes settings for managing system memory. It's necessary to ensure efficient memory usage and to prevent out-of-memory errors.
- **File systems:** This option includes settings for the file systems your kernel will support. It's necessary to include the file systems that your operating system and applications require.
- **Device drivers:** This option includes settings for the drivers of your system's devices. It's necessary to include the correct drivers for your hardware to function properly.
- **Networking:** This option includes settings for network protocols and drivers. It's necessary for network communication and internet connectivity.
- **Security:** This option includes settings for kernel security features. It's necessary to protect the system from unauthorized access and attacks.
- **Executable file formats:** This option includes settings for the types of executable files your kernel can run. It's necessary to include the formats that your applications use.
- **Kernel modules:** This option determines whether kernel modules are enabled or not. Kernel modules allow for dynamic loading and unloading of kernel code, which can be necessary for supporting certain hardware or features.
- **Debugging:** This option includes settings for kernel debugging features. It's necessary for troubleshooting and fixing issues in the kernel.

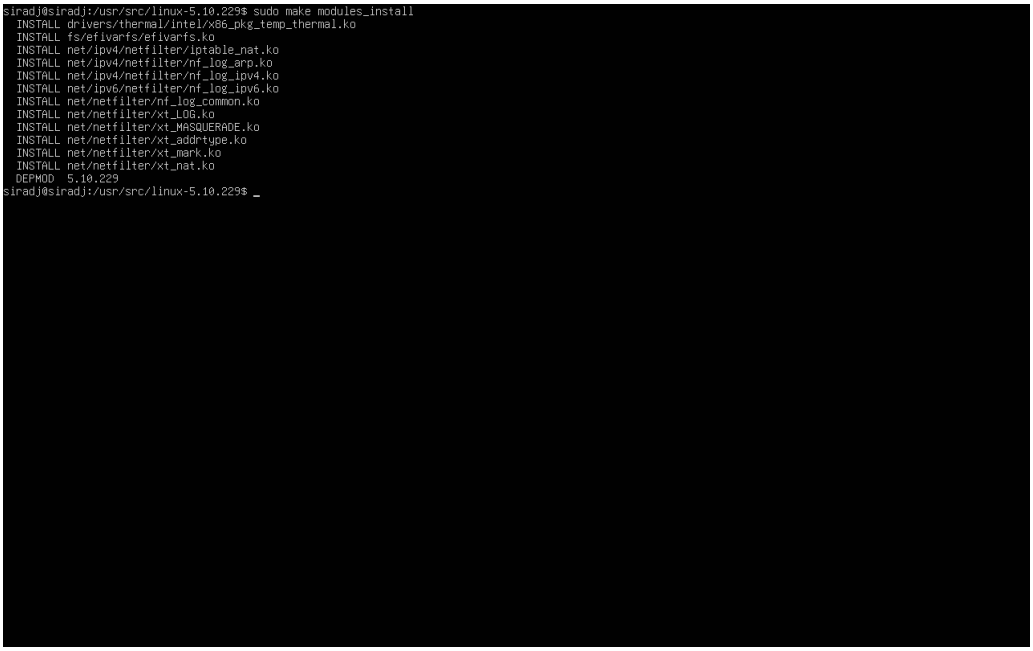
Kernel Compilation:

To clean previous build run this two commands `sudo make clean` and `sudo make mrproper`

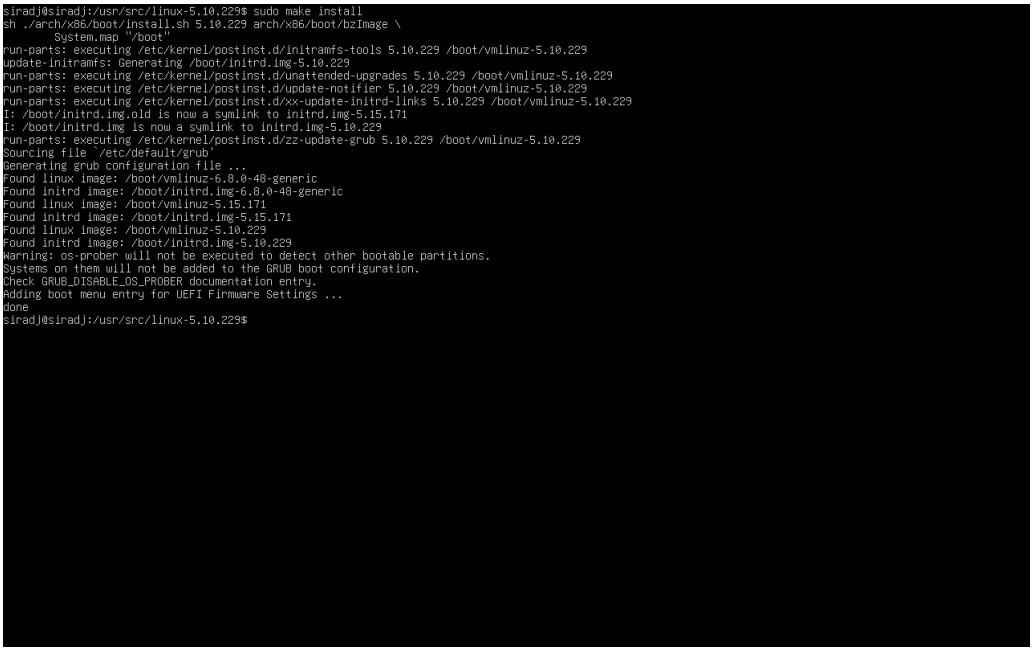
1. After i run the default configuration command `sudo make defconfig`, irun this command `sudo make -j(nproc)` to compile the kernel using all cpu cores when compiling. After a few minutes i got this :



2. install modules : `sudo make modules_install`



3. Install kernel : `sudo make install`



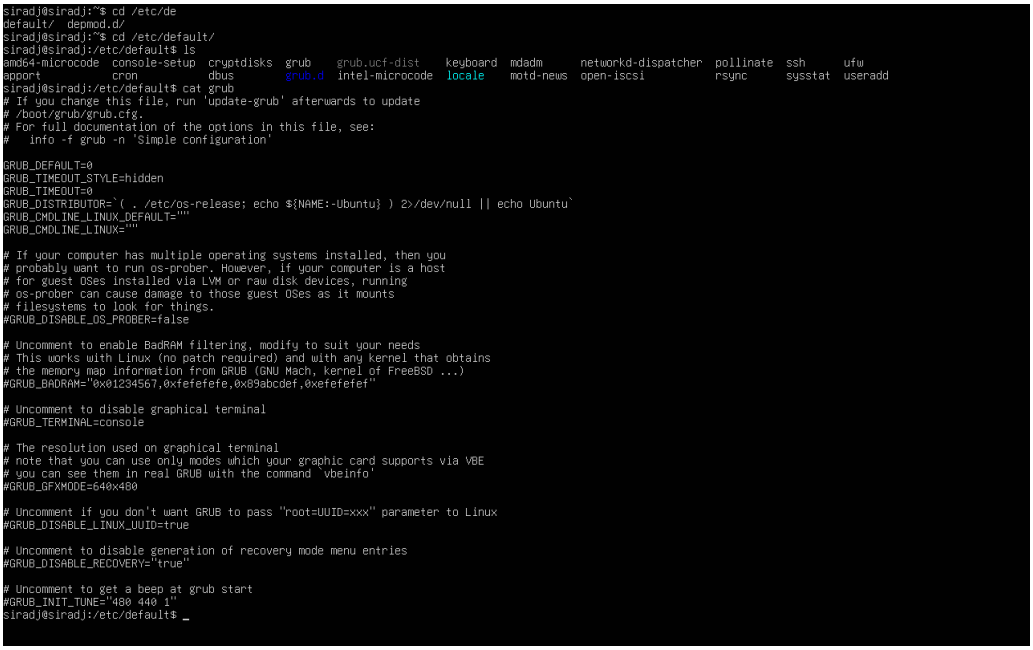


Figure 5: Grub file Before Modification

4. configure the `GRUB_DEFAULT` parameter in `/etc/default/grub` file. change it to `GRUB_DEFAULT="Advanced options for Ubuntu>Ubuntu, with Linux 5.10.229"`

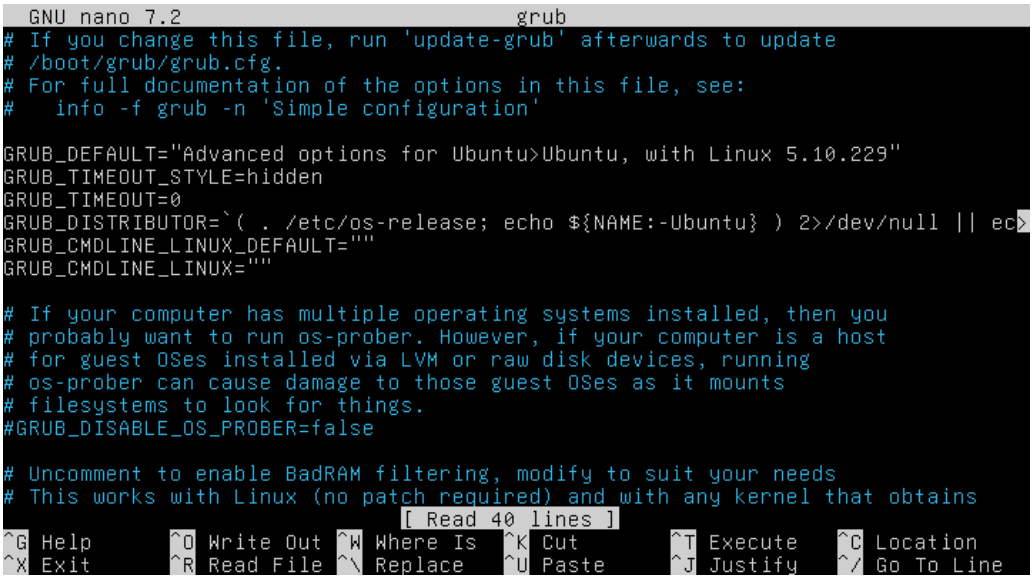


Figure 6: After Modification

5. then update the grub command : `sudo update-grub`
6. Finally : reboot the system, command : `sudo reboot`


siradj@siradj:/usr/src/linux-5.10.229$ sudo update-grub
Sourcing file `/etc/default/grub'
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-6.8.0-48-generic
Found initrd image: /boot/initrd.img-6.8.0-48-generic
Found linux image: /boot/vmlinuz-5.15.171
Found initrd image: /boot/initrd.img-5.15.171
Found linux image: /boot/vmlinuz-5.10.229
Found initrd image: /boot/initrd.img-5.10.229
Warning: os-prober will not be executed to detect other bootable partitions.
Systems on them will not be added to the GRUB boot configuration.
Check GRUB_DISABLE_OS_PROBER documentation entry.
Adding boot menu entry for UEFI Firmware Settings ...
done
siradj@siradj:/usr/src/linux-5.10.229$

7. Result:

Figure 7: Original Kernel

Figure 8: Custome Kernel