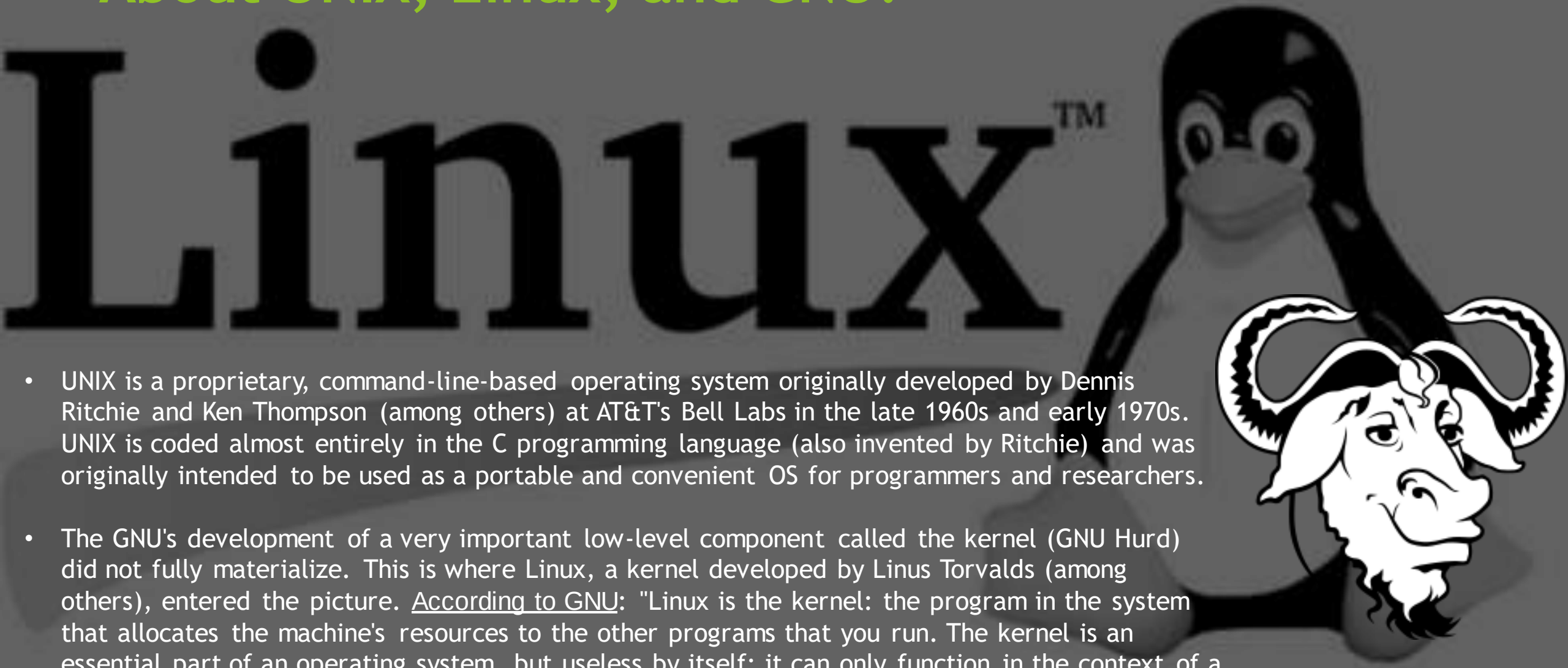
I Use Arch BTW!

About UNIX, Linux, and GNU!



- UNIX is a proprietary, command-line-based operating system originally developed by Dennis Ritchie and Ken Thompson (among others) at AT&T's Bell Labs in the late 1960s and early 1970s. UNIX is coded almost entirely in the C programming language (also invented by Ritchie) and was originally intended to be used as a portable and convenient OS for programmers and researchers.

- The GNU's development of a very important low-level component called the kernel (GNU Hurd) did not fully materialize. This is where Linux, a kernel developed by Linus Torvalds (among others), entered the picture. According to GNU: "Linux is the kernel: the program in the system that allocates the machine's resources to the other programs that you run. The kernel is an essential part of an operating system, but useless by itself; it can only function in the context of a complete operating system."

More...

- GNU is **a Unix-like operating system developed by the Free Software Foundation in 1984**. It is a collection of free software, including applications, libraries, developer tools, and games. GNU can be used as an operating system or in parts with other operating systems. The Linux kernel is the core component of GNU/Linux, which provides basic services and allocates OS resources.
- Linux is a powerful and flexible open-source operating system that has become increasingly popular in the technology industry. Developed by a global community of developers, Linux offers a wide range of features and customization options for users of all skill levels.
- Arch Linux is a general-purpose rolling release Linux distribution that is very popular among DIY enthusiasts and hardcore Linux users.
- It starts with a very basic setup and allows users to build it up to fit their needs.
- Many people who enjoy learning about computers and software like to use Arch Linux because it offers a lot of flexibility and teaches them how Linux systems work.

What Is a Distro?

- The modern operating systems we use every day, such as Windows and macOS, are made of many, many different (and very technical) components, including kernels that help software communicate with hardware and the graphical user interface (GUI) elements you see on screen.
- A distro (short for distribution) is best thought of as a neatly wrapped package of the core software components that make up a GNU/Linux operating system. Consider distros like Fedora, Elemntary OS, Linux Mint, Manjaro, Ubuntu as roughly the functional equivalent of Windows and macOS.
- A typical GNU/Linux distribution includes the Linux kernel; GNU tools and libraries; a windowing system for displaying windows on screen and interacting with input devices; a desktop environment for performing actions with OS programs; and additional parts. Even the preceding description is a vast simplification. Some of the most common desktop environments are GNU's GNOME, KDE's Plasma, MATE, and XFCE.

How to install Arch Linux

As of now, there are two ways using which you can install Arch Linux:

- Without the guided installer (mannual installation)
- With the help of Archinstaller (guided installer)

  Using Cmd : archinstall

Did you know about Boot Procedure?

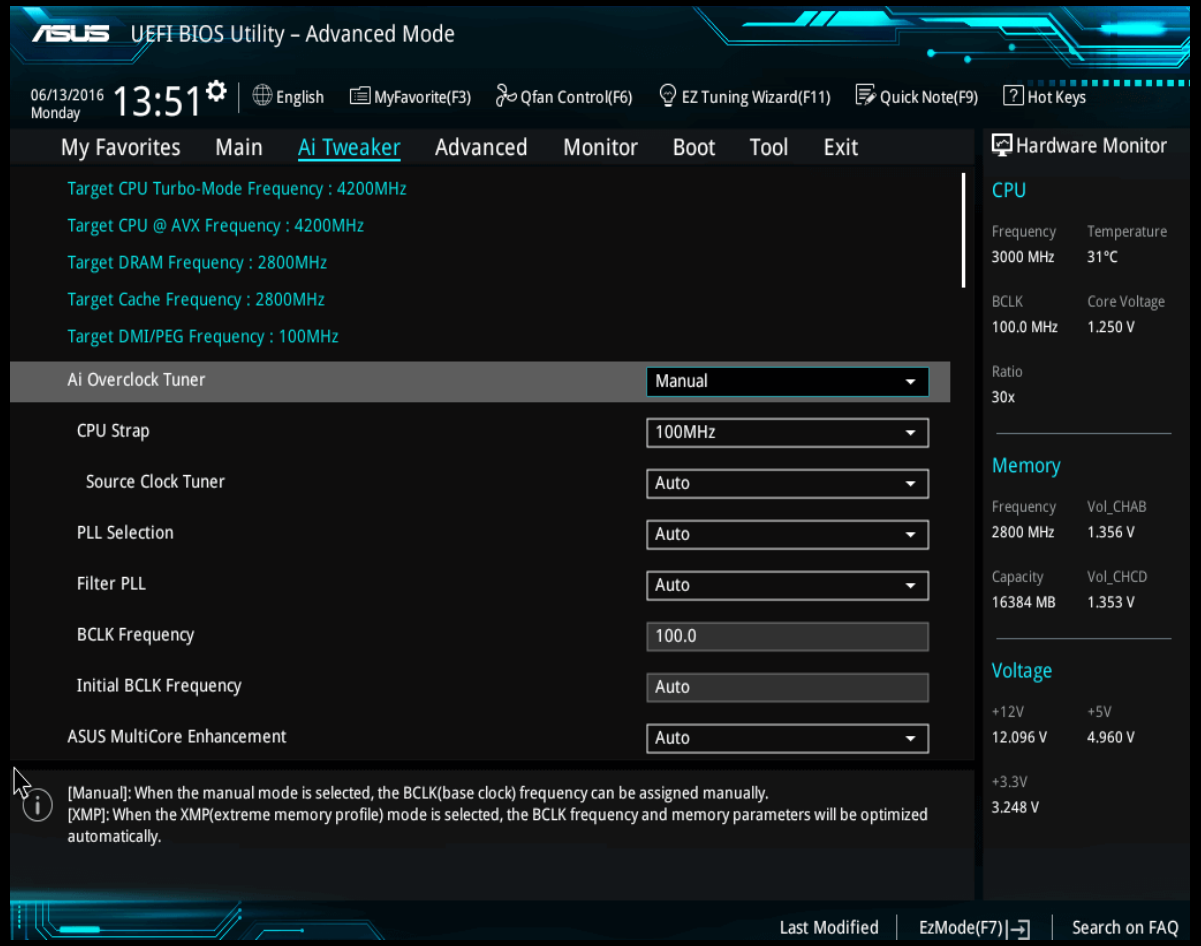
- So, how does a computer boot? Let's go step by step:
- You press the power button on your laptop/desktop.
- The CPU starts up, but needs some instructions to work on (remember, the CPU always needs to do something). Since the main memory is empty at this stage, CPU defers to load instructions from the firmware chip on the motherboard and begins executing instructions.
- The firmware code does a Power On Self Test (POST), initializes the remaining hardware, detects the connected peripherals (mouse, keyboard, pendrive etc.) and checks if all connected devices are healthy. You might remember it as a 'beep' that desktops used to make after POST is successful.
- Finally, the firmware code cycles through all storage devices and looks for a boot-loader (usually located in first sector of a disk). If the boot-loader is found, then the firmware hands over control of the computer to it.

- So now that the boot-loader is loaded, its job is to load the rest of the operating system. GRUB is one such boot-loader that is capable of loading unix-like operating systems and is also able to chain-load Windows OS. Boot-loader is only available in the first sector of a disk, which is 512 bytes. Given the complexity of modern operating systems, some of these boot-loaders tend to do multi-stage loading, where the main boot-loader loads the second-stage-boot-loader in an environment which is not restricted to 512 bytes.
- The boot-loader then loads the kernel into memory. Unix-like operating systems then run the init process (the master process, from which other processes are forked/executed) and finally initialize the run-levels.
- In Windows, wininit.exe is loaded along with some other processes like services.exe for service control, lsass.exe for local security and authority (similar to run-levels) and lsm.exe for local session management.
- After all this, and after some other drivers are initialized, the Graphical User Inferface (GUI) is loaded and you are presented with the login screen.

BIOS

- BIOS stands for Basic Input/Output System, the firmware we talked about in the above boot procedure.
- It is stored on an EPROM (Erasable Programmable Read-Only Memory), allowing the manufacturer to push out updates easily.
- It provides many helper functions that allow reading boot sectors of attached storage and printing things on screen. You can access BIOS during the initial phases of the boot procedure by pressing del, F2 or F10.

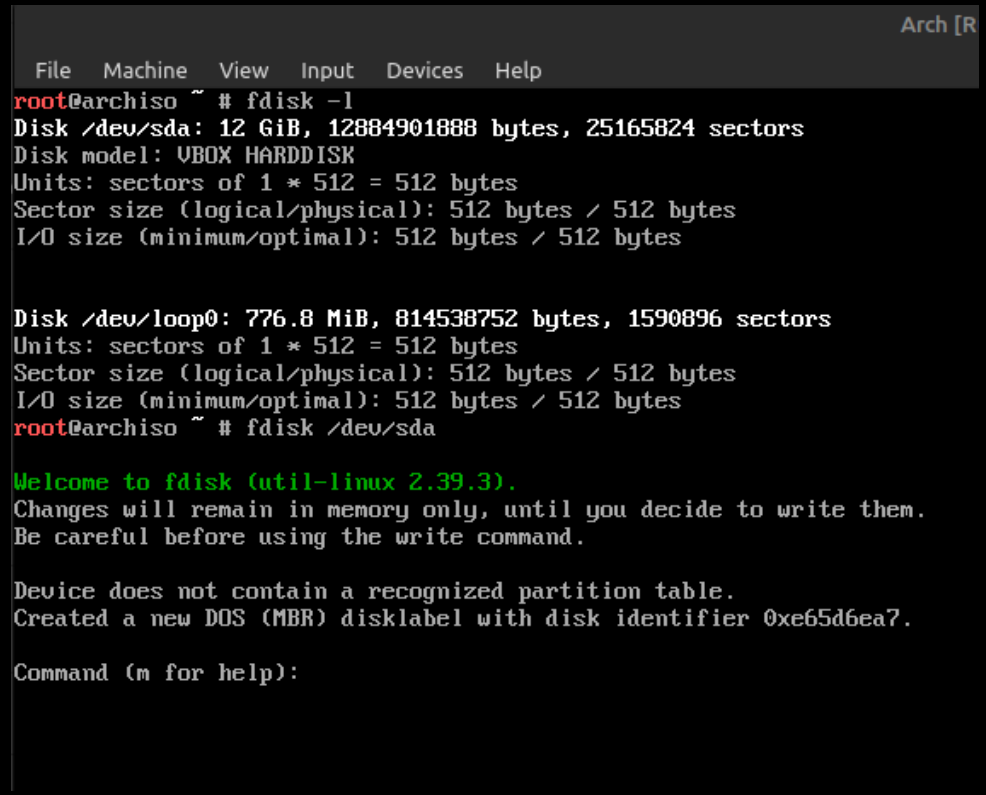
UEFI



- UEFI stands for Unified Extensible Firmware Interface. It does the same job as a BIOS, but with one basic difference: it stores all data about initialization and startup in an .efi file, instead of storing it on the firmware.
- This .efi file is stored on a special partition called EFI System Partition (ESP) on the hard disk. This ESP partition also contains the bootloader.
- UEFI was designed to overcome many limitations of the old BIOS, including:
- UEFI supports drive sizes upto 9 zettabytes, whereas BIOS only supports 2.2 terabytes.
- UEFI provides faster boot time.
- UEFI has discrete driver support, while BIOS has drive support stored in its ROM, so updating BIOS firmware is a bit difficult.
- UEFI offers security like "Secure Boot", which prevents the computer from booting from unauthorized/unsigned applications. This helps in preventing rootkits, but also hampers dual-booting, as it treats other OS as unsigned applications. Currently, only Windows and Ubuntu are signed OS (let me know if I am wrong).
- UEFI runs in 32bit or 64bit mode, whereas BIOS runs in 16bit mode. So UEFI is able to provide a GUI (navigation with mouse) as opposed to BIOS which allows navigation only using the keyboard.

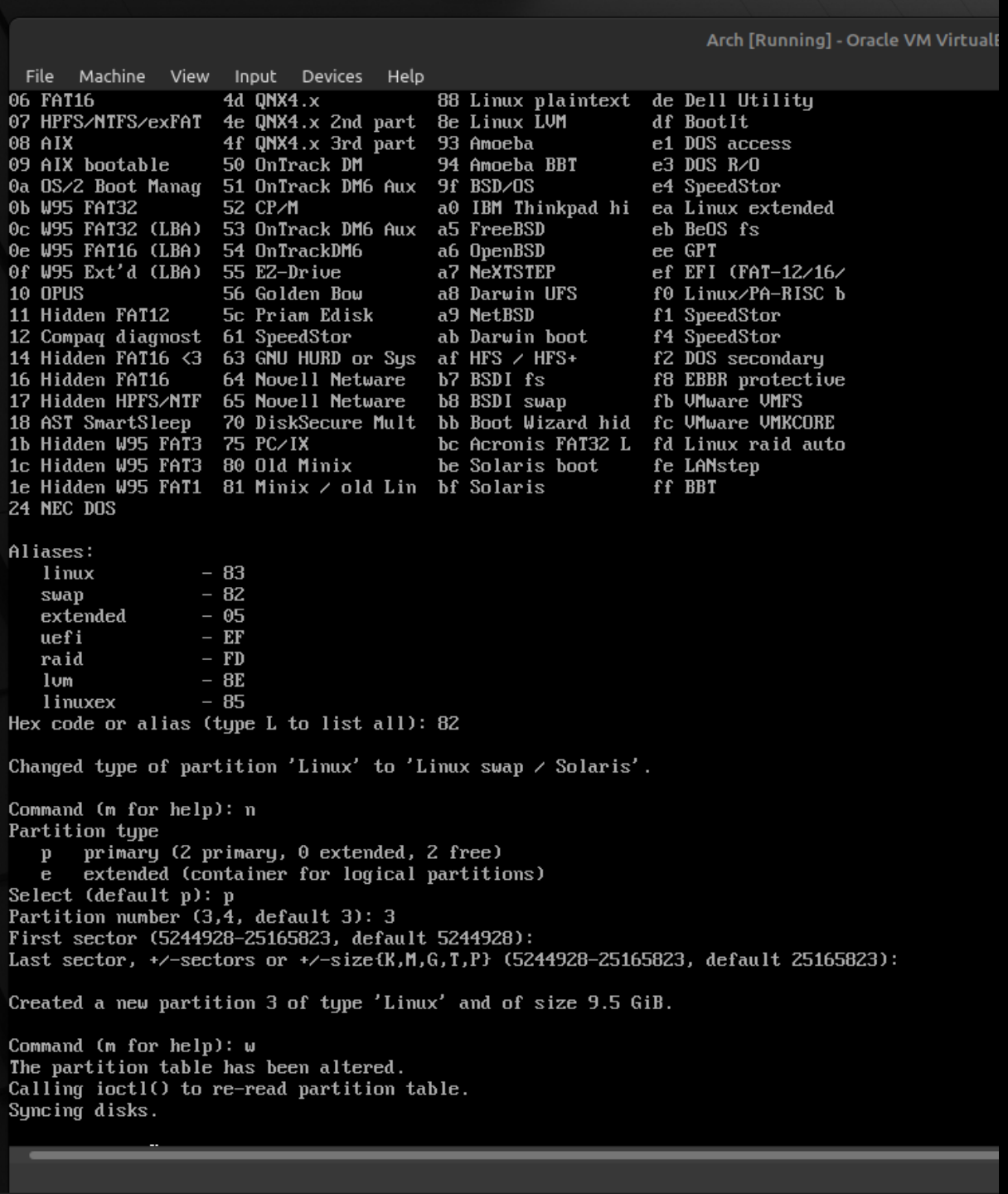
Need for Partition the disks !

- One of the most initial steps in installing and OS is doing partition of disks.
- Before installing Any OS (say Arch Linux in my case), you need to divide your computer's storage into sections called partitions.
- This step is done to prepare hard drive for new OS
- In arch linux we use cli based disk partition manager called fdisk.

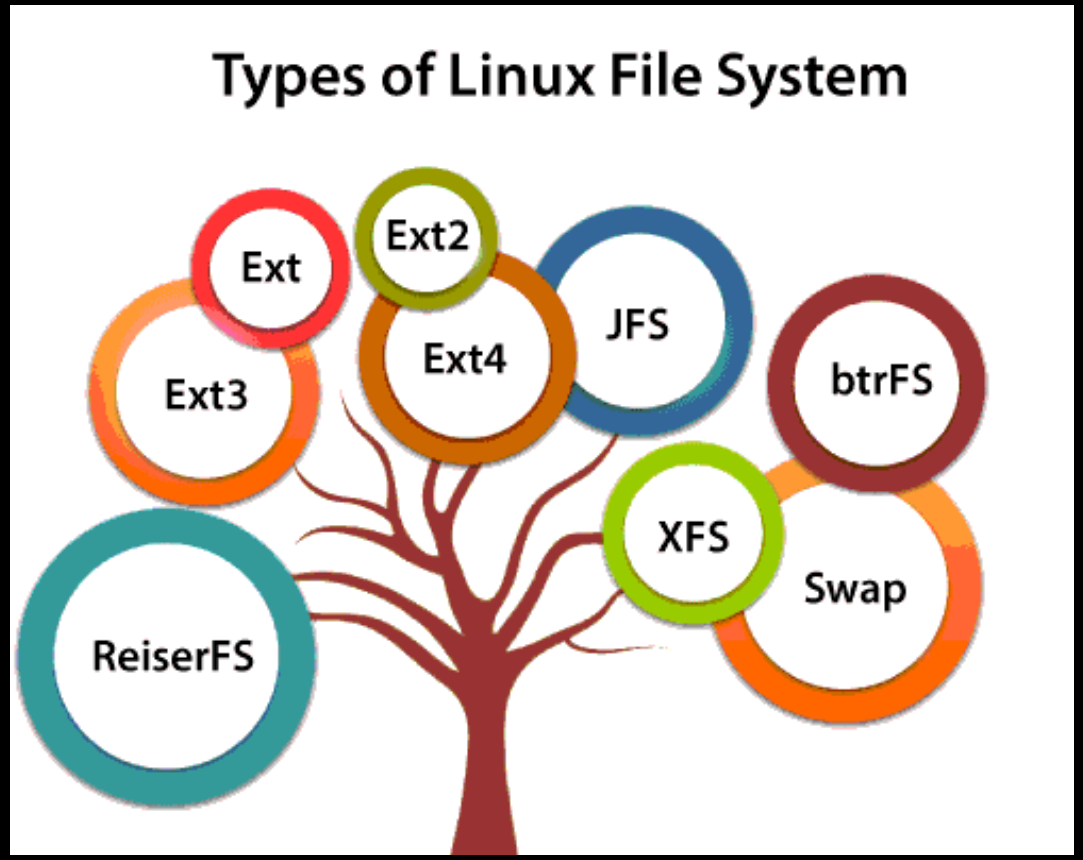


Disk partition processes

- Deleting and formatting disk and creating partitions in hard disk for different purposes.



- We need to check in UEFI mode is enabled if yes then we need to create ESP partition for EFI system which is must in order to boot into our systems
- After creating the partition we are required to change the type of the EFI partition to EFI System (instead of Linux system)
- Which is important as we are assigning memory for bootloader it should be of type which is readable by every type of OS and bootloader



Let's talk about file systems

A file system is a method an operating system uses to store, organize, and manage files and directories on a storage device. Some common types of file systems include:

- **FAT (File Allocation Table):** An older file system used by older versions of Windows and other operating systems.
- **NTFS (New Technology File System):** A modern file system used by Windows. It supports features such as file and folder permissions, compression, and encryption.
- **ext (Extended File System):** A file system commonly used on Linux and Unix-based operating systems.
- **HFS (Hierarchical File System):** A file system used by macOS.
- **APFS (Apple File System):** A new file system introduced by Apple for their Macs and iOS devices.

Creating file systems in Arch Installation after partitions of disk are ready

```
Created a new partition 3 of type 'Linux' and of size 9.5 GiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.

root@archiso ~ # mkfs.fat -F32 /dev/sda1
mkfs.fat 4.2 (2021-01-31)
root@archiso ~ # mkfs.ext4 /dev/sda3
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 2490112 4k blocks and 622592 inodes
Filesystem UUID: b6ca1e3f-0c6b-41f6-86d6-3fc9d989d808
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

Creating filesystem for UEFI system

- So, you have two disk partitions and the first one is EFI type. Create a FAT32 file system on it using the mkfs command:

Cmd: mkfs.fat -F32 /dev/sda1

- Now create an Ext4 filesystem on the root partition:
- mkfs.ext4 /dev/sda2

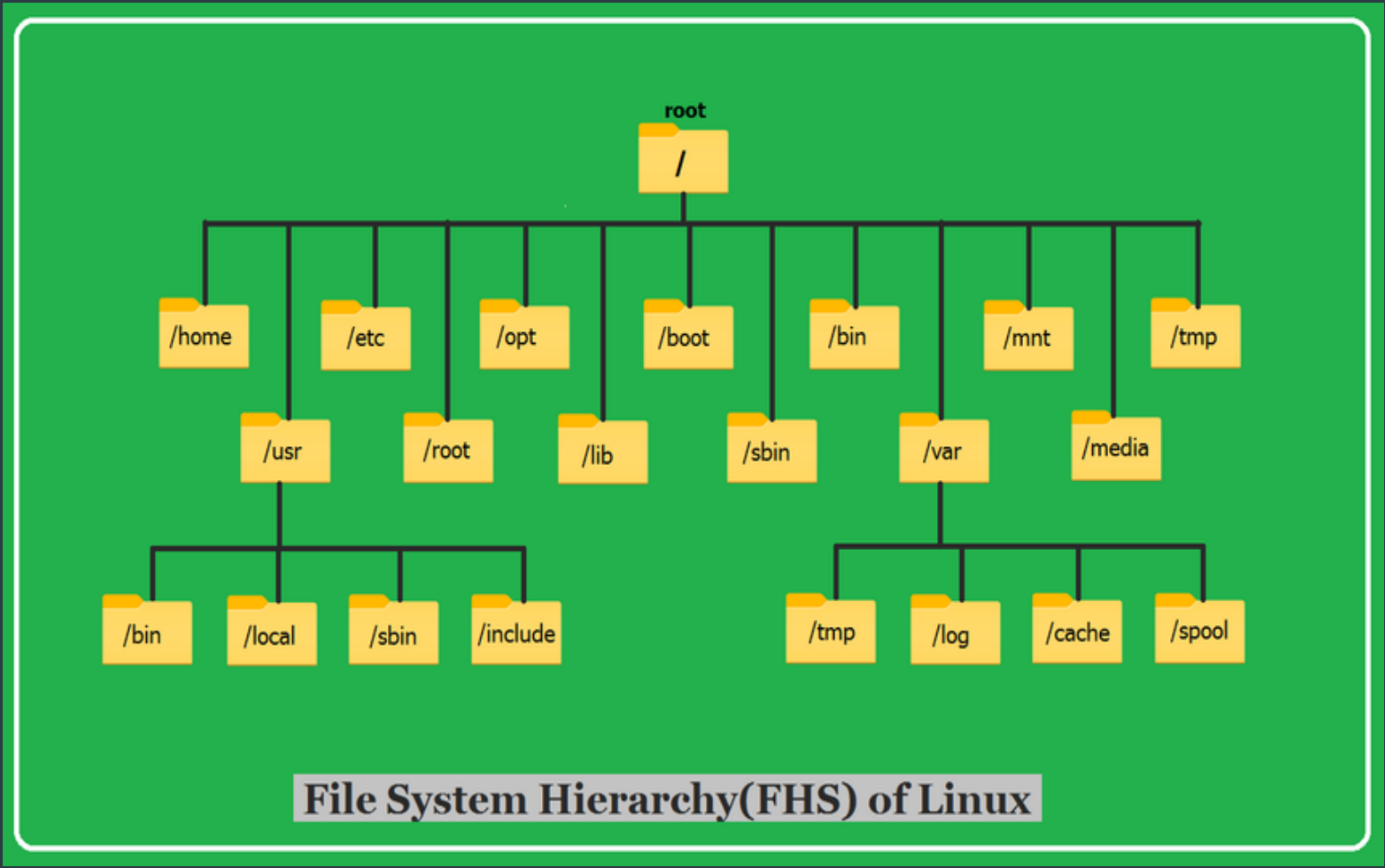
Creating filesystem for non-UEFI system

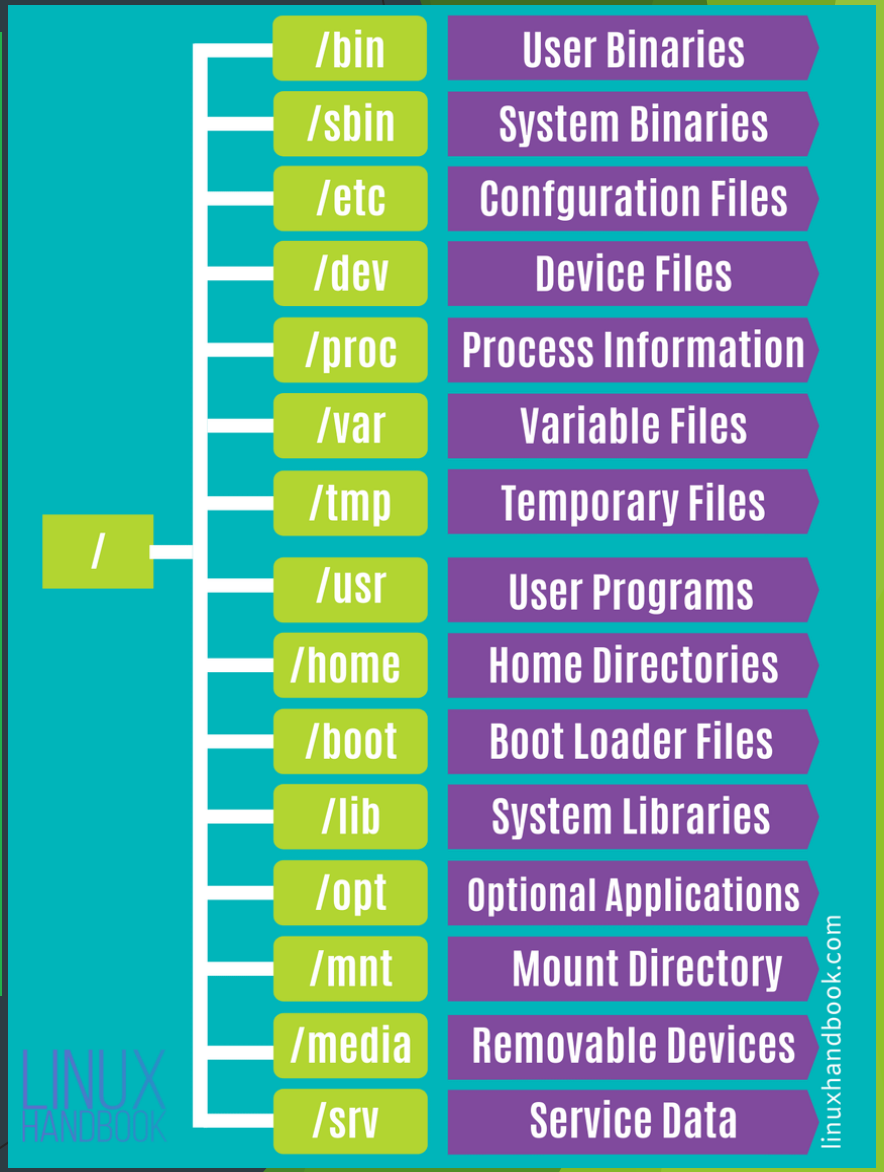
- For non-UEFI system, you only have one single root partition. So just make it ext4:

Cmd: mkfs.ext4 /dev/sda1

- We will create SWAP memory partition (it's basically assigning some amount of hard disk memory as RAM alt in case of RAM overflow
- Also its important for communication between different kinds of memories.
- SWAP memory is slow as compared to RAM
- Next step is to create root partition where all the system files go its similar to locale disk C in Windows only difference is in Linux we follow FHS(file system hierarchy)

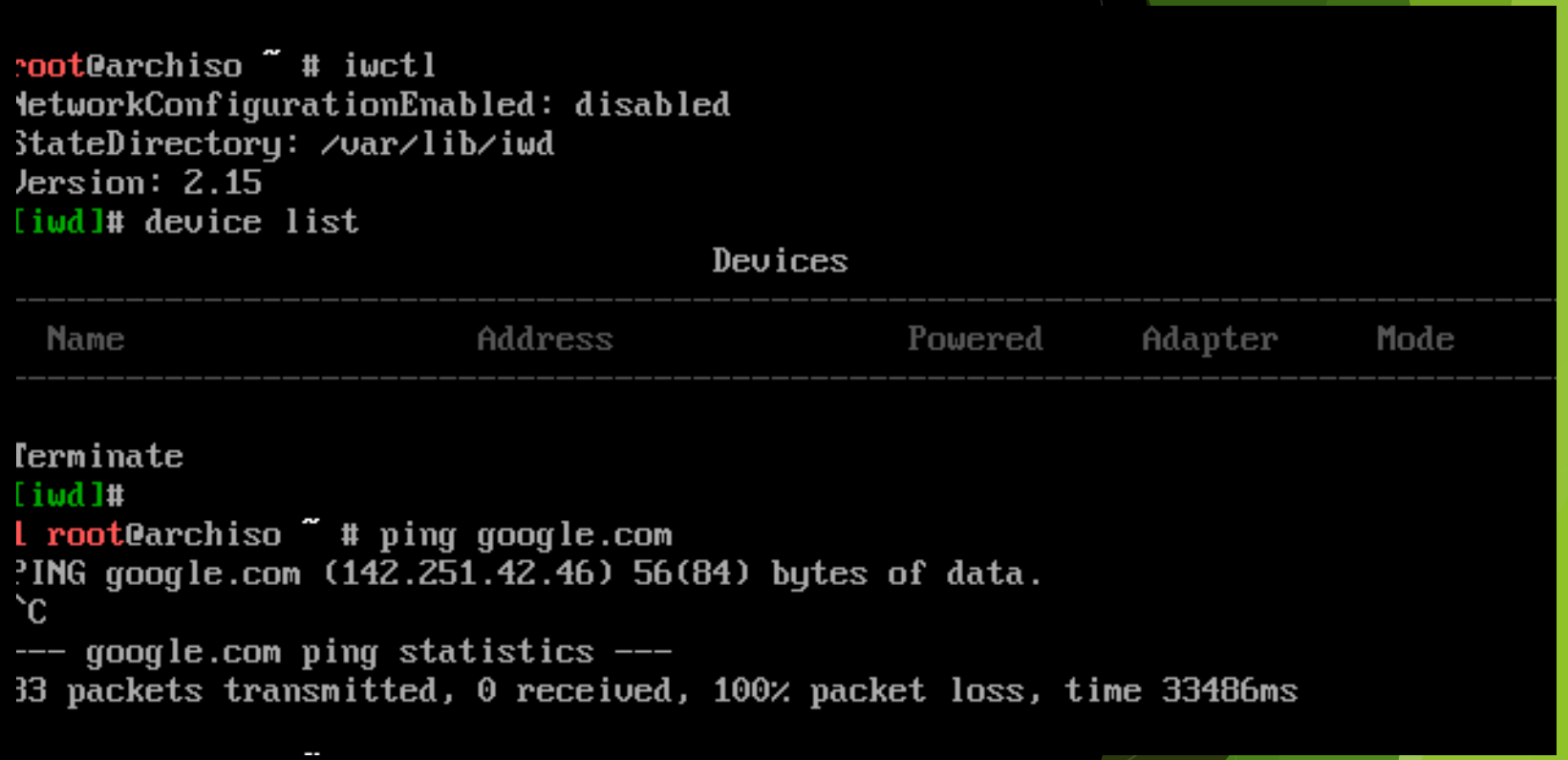
File system hierarchy in Linux based OS

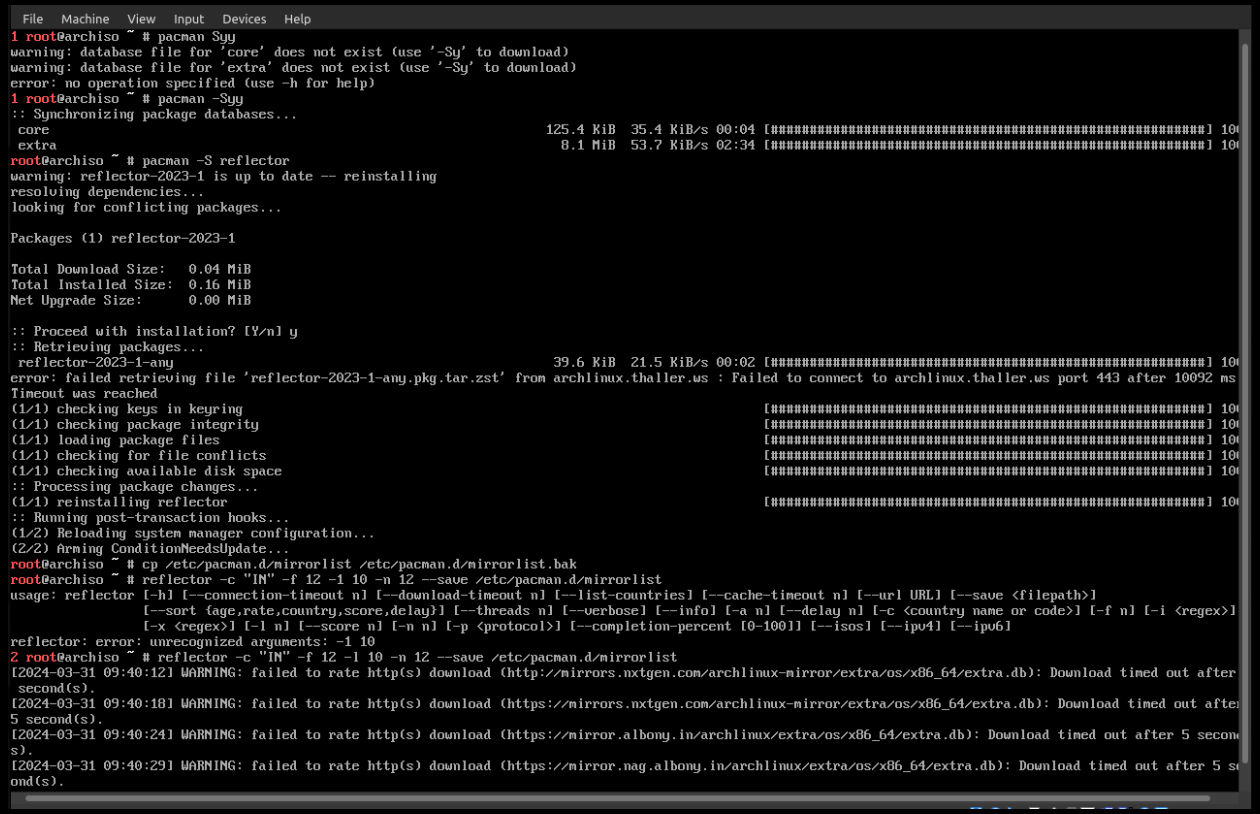




Connect to WIFI

- If you're using a wireless internet connection, you'll need to connect to it during the installation process.
- Arch Linux provides a tool called iwctl to help you do this.
- As I was installing it in virtual machine I don’t need to connect to Wifi as its already working as in form ethernet connection using my system internet.
- This is imp step as we will be downloading many packages during our installation process.





Mirror location is a server from where we download packages required for Arch Installation.

Selecting appropriate mirror location

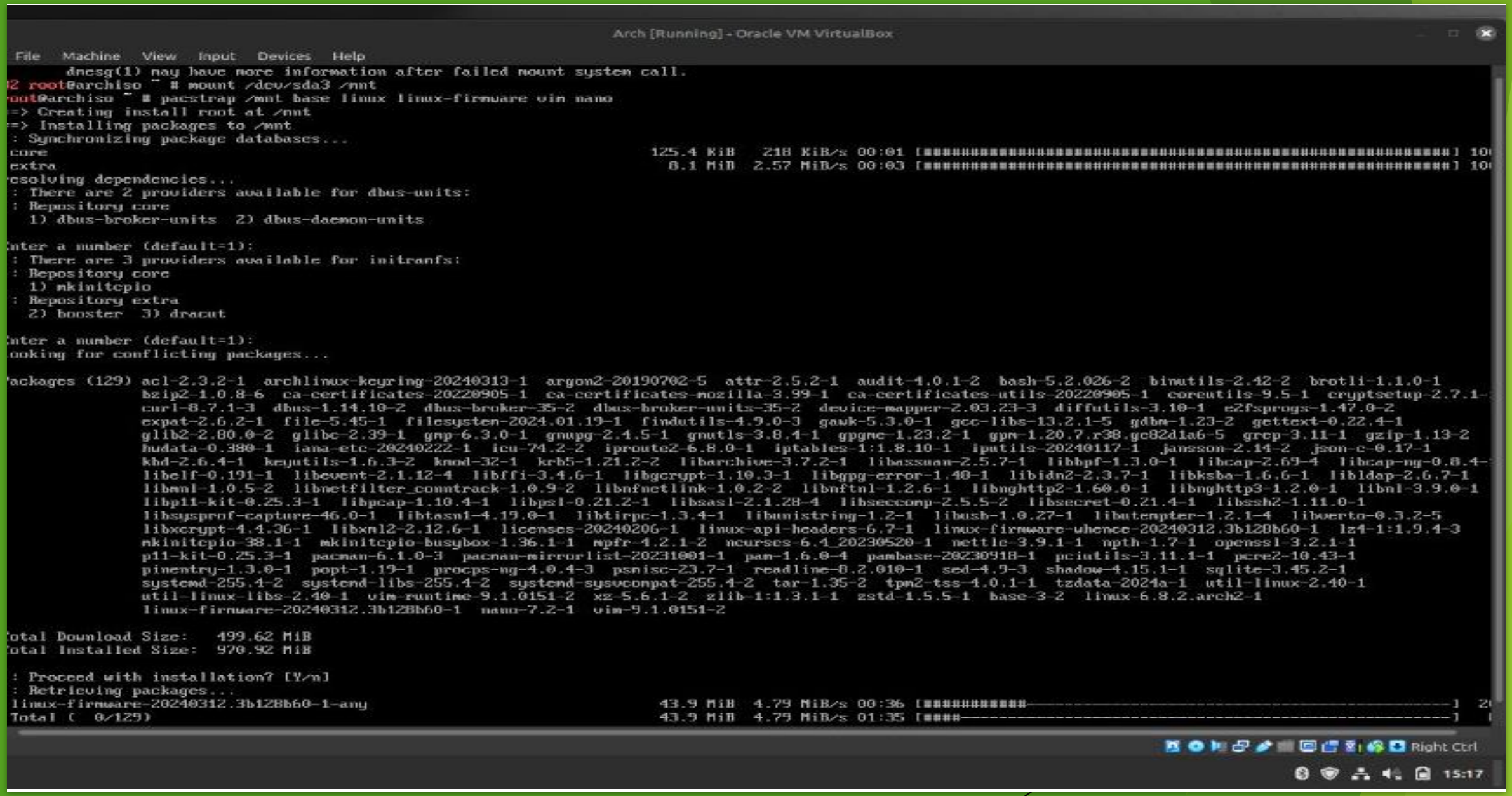
- When installing Arch Linux, you need to choose a server from which to download the installation files.
- It's because the mirrorlist (located in /etc/pacman.d/mirrorlist) has a huge number of mirrors but not in a good order. The top mirror is chosen automatically and it may not always be a good choice.
- Now, install reflector too that you can use to list the fresh and fast mirrors located in your country.
- Now, get the good mirror list with reflector and save it to mirror list.
- Also got this error idk why KGP don't allow Arch servers that's why downloading was failed.

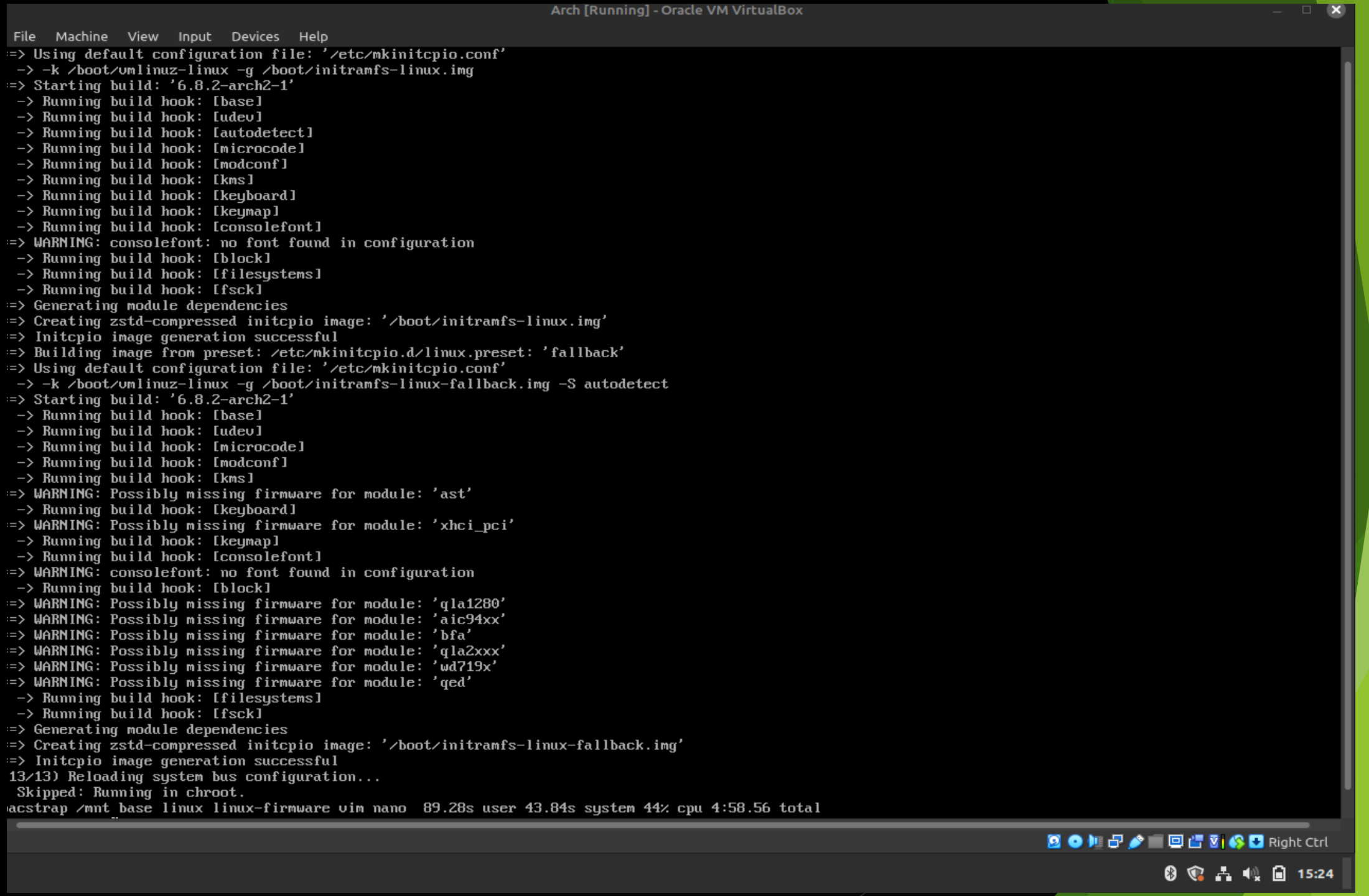
What is meant by mount?

- All files in a **Linux** filesystem are arranged in form of a big tree rooted at '**/**'. These files can be spread out on various devices based on your partition table, initially your parent directory is mounted (i.e attached) to this tree at '**/**', others can be mounted manually using GUI interface(if available) or using **mount** command.
- **mount** command is used to mount the filesystem found on a device to big tree structure(**Linux** filesystem) rooted at '**/**'. Conversely, another command **umount** can be used to detach these devices from the Tree.
- When you 'mount' something you are placing access to the file system contained within onto your root file system structure. Effectively giving the files a location. This is similar to the C:/D: drive labels in windows, but more flexible.
- Mounting /dev/sdb1 to /mnt/disk1 places all the files and folders contained within the device standard disk B partition 1 into the directory /mnt/disk1 where you can access them.
- Modern systems have ways to auto-mount drives just as windows auto-mounts drives to drive letters, but the location mounting system in Unix is much more flexible. And unmounting is obviously the removal of the access to those files/folders from that location. You can find out what file systems are mounted by running the command:

  mount

- Also mount, umount, fstab (for fixed mounts), udev, gvfs (for automatic mounting)

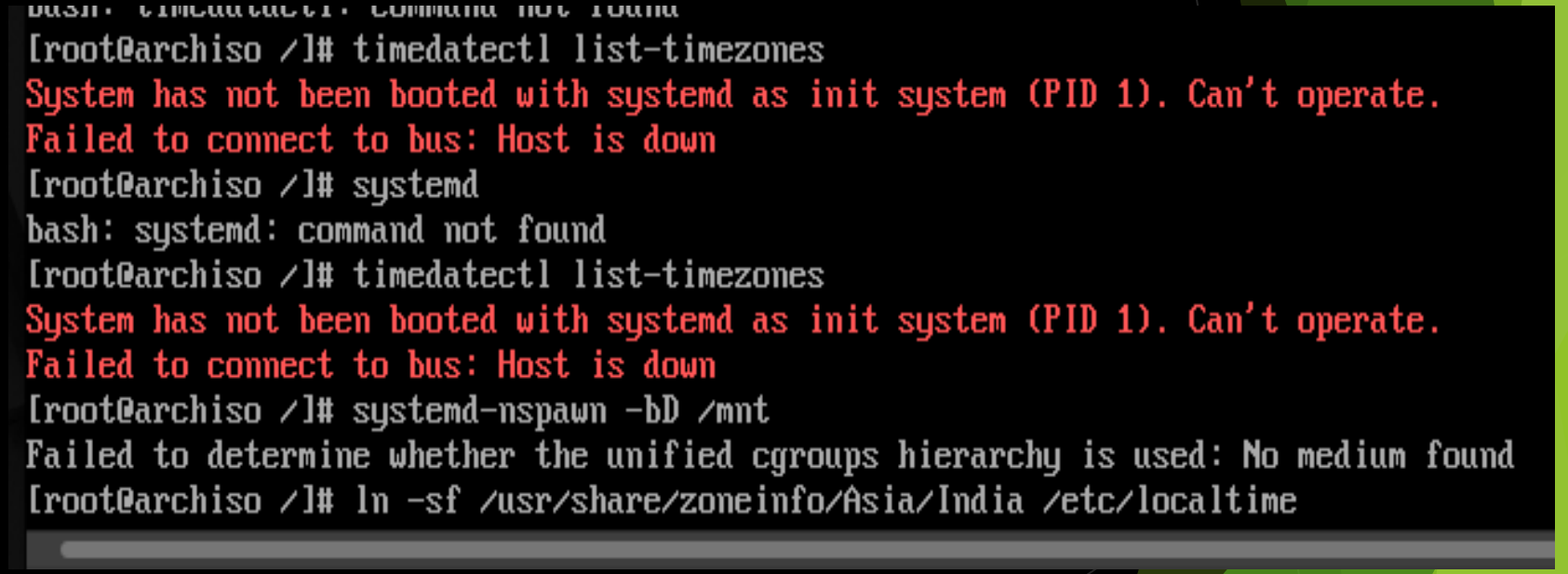

Arch [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
dmesg(1) may have more information after failed mount system call.
root@archiso ~ # mount /dev/sda3 /mnt
root@archiso ~ # pacstrap /mnt base linux linux-firmware vim nano
==> Creating install root at /mnt
==> Installing packages to /mnt
:: Synchronizing package databases...
core 125.4 KiB 218 KiB/s 00:01
extra 8.1 MiB 2.57 MiB/s 00:03
resolving dependencies...
:: There are 2 providers available for dbus-units:
:: Repository core
1) dbus-broker-units 2) dbus-daemon-units
Enter a number (default=1):
:: There are 3 providers available for initramfs:
:: Repository core
1) mkinitcpio
:: Repository extra
2) booster 3) dracut
Enter a number (default=1):
looking for conflicting packages...
Packages (129) acl-2.3.2-1 archlinux-keyring-20240313-1 argon2-20190702-5 attr-2.5.2-1 audit-4.0.1-2 bash-5.2.026-2 binutils-2.42-2 brotli-1.1.0-1
bzip2-1.0.8-6 ca-certificates-20220905-1 ca-certificates-mozilla-3.99-1 ca-certificates-utils-20220905-1 coreutils-9.5-1 cryptsetup-2.7.1-
curl-8.7.1-3 dbus-1.14.10-2 dbus-broker-35-2 dbus-broker-units-35-2 device-mapper-2.03.23-3 diffutils-3.10-1 e2fsprogs-1.47.0-2
expat-2.6.2-1 file-5.45-1 filesystem-2024.01.19-1 findutils-4.9.0-3 gawk-5.3.0-1 gcc-libs-13.2.1-5 gdbm-1.23-2 gettext-0.22.4-1
glib2-2.80.0-2 glibc-2.39-1 gmp-6.3.0-1 gnupg-2.4.5-1 gnutls-3.8.4-1 gpgme-1.23.2-1 gpm-1.20.7.r38.gc82d1a6-5 grep-3.11-1 gzip-1.13-2
hwdata-0.380-1 iana-etc-20240222-1 icu-74.2-2 iproute2-6.8.0-1 iptables-1:1.8.10-1 iputils-20240117-1 jansson-2.14-2 json-c-0.17-1
kbd-2.6.4-1 keyutils-1.6.3-2 kmod-32-1 krb5-1.21.2-2 libarchive-3.7.2-1 libassuan-2.5.7-1 libbpf-1.3.0-1 libcap-2.69-4 libcap-ng-0.8.4-
libelf-0.191-1 libevent-2.1.12-4 libffi-3.4.6-1 libgcrypt-1.10.3-1 libgpg-error-1.48-1 libidn2-2.3.7-1 libksba-1.6.6-1 libldap-2.6.7-1
libmnl-1.0.5-2 libnetfilter_conntrack-1.0.9-2 libnfnetlink-1.0.2-2 libnftnl-1.2.6-1 libnghttp2-1.60.0-1 libnghttp3-1.2.0-1 libnl-3.9.0-1
libp11-kit-0.25.3-1 libpcap-1.10.4-1 libpsl-0.21.2-1 libsasl-2.1.28-4 libseccomp-2.5.5-2 libsecret-0.21.4-1 libssh2-1.11.0-1
libsysprof-capture-46.0-1 libtasn1-4.19.0-1 libtirpc-1.3.4-1 libunistring-1.2-1 libusb-1.0.27-1 libutempter-1.2.1-4 libverto-0.3.2-5
libxcrypt-4.4.36-1 libxml2-2.12.6-1 licenses-20240206-1 linux-api-headers-6.7-1 linux-firmware-uhence-20240312.3b128b60-1 lz4-1:1.9.4-3
mkinitcpio-38.1-1 mkinitcpio-busybox-1.36.1-1 mpfr-4.2.1-2 ncurses-6.4_20230520-1 nettle-3.9.1-1 npth-1.7-1 openssl-3.2.1-1
p11-kit-0.25.3-1 pacman-6.1.0-3 pacman-mirrorlist-20231001-1 pam-1.6.0-4 pambase-20230918-1 pciutils-3.11.1-1 pcre2-10.43-1
pinentry-1.3.0-1 popt-1.19-1 procps-ng-4.0.4-3 psmisc-23.7-1 readline-8.2.010-1 sed-4.9-3 shadow-4.15.1-1 sqlite-3.45.2-1
systemd-255.4-2 systemd-libs-255.4-2 systemd-sysvcompat-255.4-2 tar-1.35-2 tpm2-tss-4.0.1-1 tzdata-2024a-1 util-linux-2.40-1
util-linux-libs-2.40-1 vim-runtime-9.1.0151-2 xz-5.6.1-2 zlib-1:1.3.1-1 zstd-1.5.5-1 base-3-2 linux-6.8.2.arch2-1
linux-firmware-20240312.3b128b60-1 nano-7.2-1 vim-9.1.0151-2
Total Download Size: 499.62 MiB
Total Installed Size: 970.92 MiB
:: Proceed with installation? [Y/n]
:: Retrieving packages...
linux-firmware-20240312.3b128b60-1-any 43.9 MiB 4.79 MiB/s 00:36
Total (0/129) 43.9 MiB 4.79 MiB/s 01:35
Right Ctrl
15:17


Arch [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
==> Using default configuration file: '/etc/mkinitcpio.conf'
-> -k /boot/vmlinuz-linux -g /boot/initramfs-linux.img
==> Starting build: '6.8.2-arch2-1'
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [autodetect]
-> Running build hook: [microcode]
-> Running build hook: [modconf]
-> Running build hook: [kms]
-> Running build hook: [keyboard]
-> Running build hook: [keymap]
-> Running build hook: [consolefont]
==> WARNING: consolefont: no font found in configuration
-> Running build hook: [block]
-> Running build hook: [filesystems]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: '/boot/initramfs-linux.img'
==> Initcpio image generation successful
==> Building image from preset: /etc/mkinitcpio.d/linux.preset: 'fallback'
==> Using default configuration file: '/etc/mkinitcpio.conf'
-> -k /boot/vmlinuz-linux -g /boot/initramfs-linux-fallback.img -S autodetect
==> Starting build: '6.8.2-arch2-1'
-> Running build hook: [base]
-> Running build hook: [udev]
-> Running build hook: [microcode]
-> Running build hook: [modconf]
-> Running build hook: [kms]
==> WARNING: Possibly missing firmware for module: 'ast'
-> Running build hook: [keyboard]
==> WARNING: Possibly missing firmware for module: 'xhci_pci'
-> Running build hook: [keymap]
-> Running build hook: [consolefont]
==> WARNING: consolefont: no font found in configuration
-> Running build hook: [block]
==> WARNING: Possibly missing firmware for module: 'qla1280'
==> WARNING: Possibly missing firmware for module: 'aic94xx'
==> WARNING: Possibly missing firmware for module: 'bfa'
==> WARNING: Possibly missing firmware for module: 'qla2xxx'
==> WARNING: Possibly missing firmware for module: 'wd719x'
==> WARNING: Possibly missing firmware for module: 'qed'
-> Running build hook: [filesystems]
-> Running build hook: [fsck]
==> Generating module dependencies
==> Creating zstd-compressed initcpio image: '/boot/initramfs-linux-fallback.img'
==> Initcpio image generation successful
13/13) Reloading system bus configuration...
Skipped: Running in chroot.
acstrap /mnt base linux linux-firmware vim nano 89.28s user 43.84s system 44% cpu 4:58.56 total
Right Ctrl
15:24

Facing issue while setting time zone

The systemd I was booted with was only responsible for current live environment and was not the same systemd as the one that i have installed in /mnt. This mismatch caused the error .

To fix this I needed to link manually or run this after booting into new OS



Proceeding ahead with Arch Installation

Install Arch Linux

- With everything prepared, you can now install Arch Linux on your computer.
- This step installs the basic components of the operating system onto your hard drive.

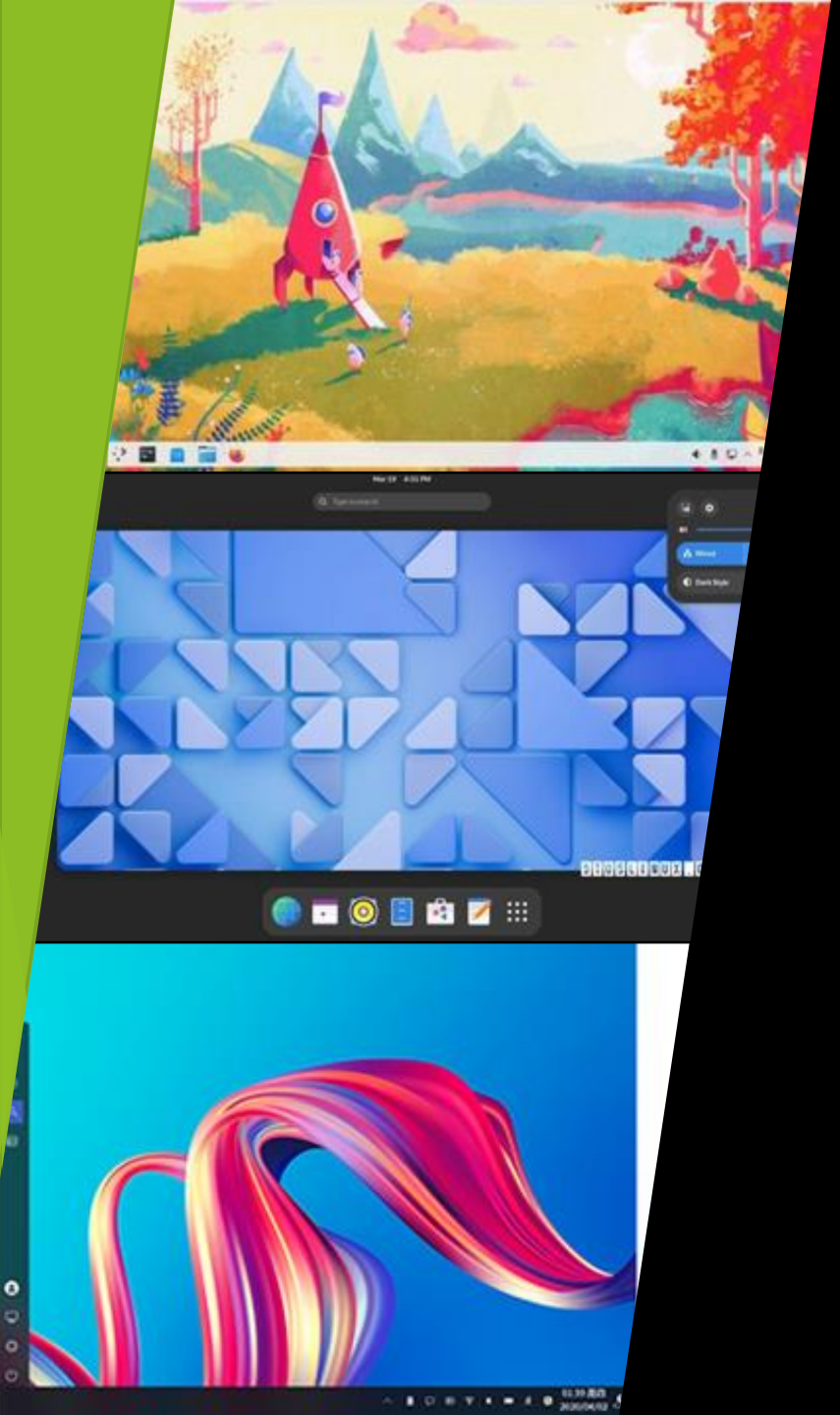
Configure the installed system

- After the installation, you'll need to set up some basic settings like time zone, language, and network configuration.
- This makes sure your new operating system works properly.

Install Grub bootloader

- The bootloader is a program that starts your computer and tells it to load the operating system.
- This step installs a bootloader called Grub onto your computer.

GRUB bootloader
in action

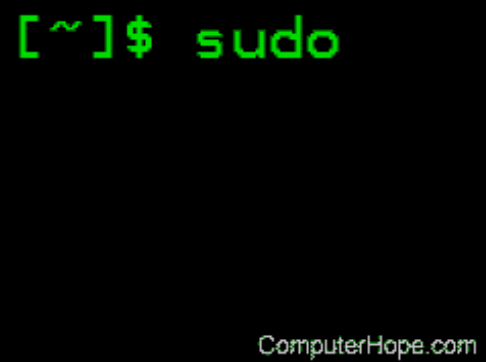


What is desktop environments?

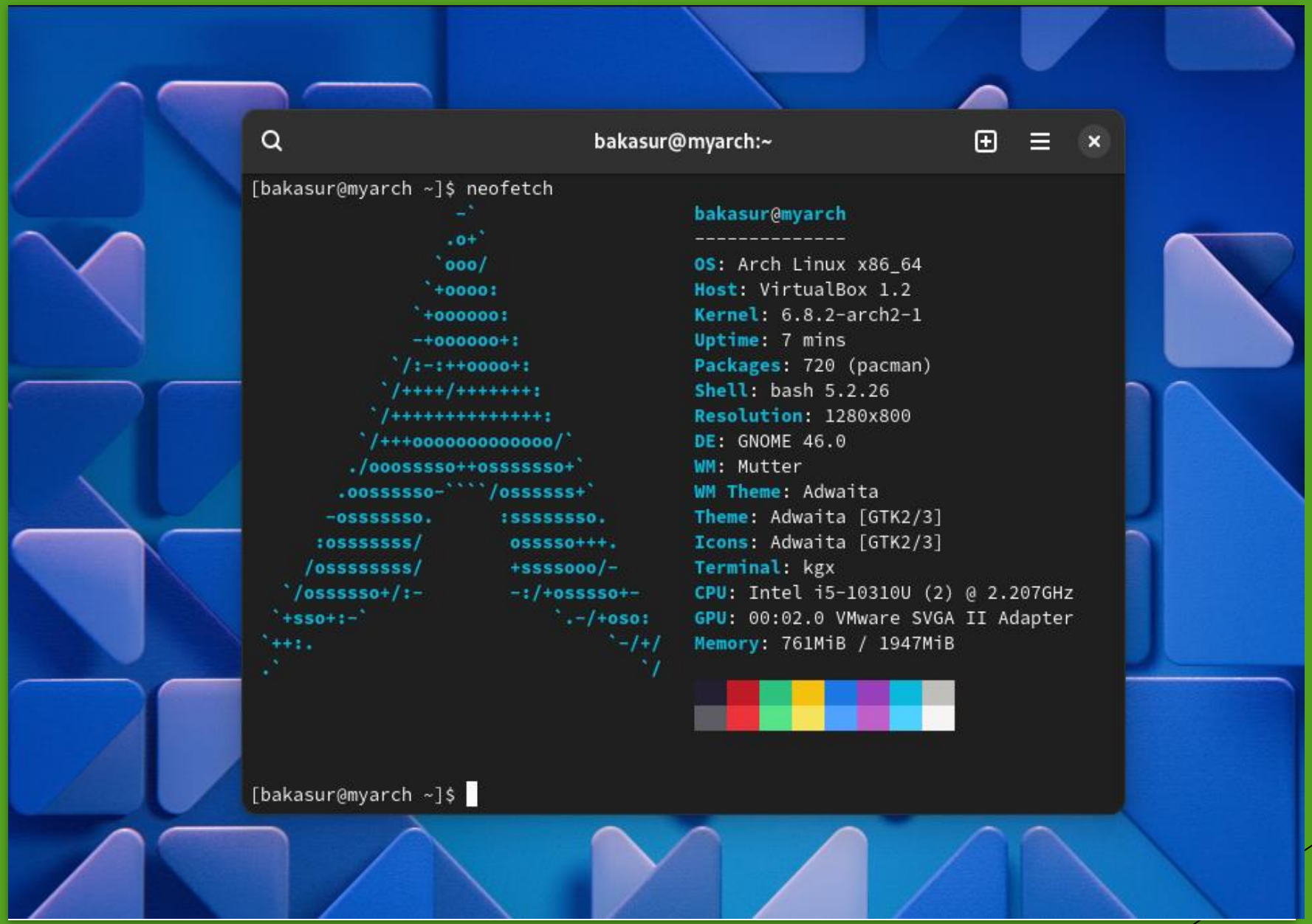
A desktop environment (DE) is **a bundle of software running on top of a computer operating system that provides a common graphical user interface (GUI).** It consists of various elements such as icons, toolbars, wallpapers, and widgets, and supports user-friendly functionalities such as file management, application launching, and system settings configuration.

- The DE is a default interface provided by virtually all modern operating systems, including Windows, Linux, Mac, and more. It was developed to replace the command-line interface, which was used in legacy operating systems such as DOS and Unix.[1] Some well-known desktop environments include Windows, macOS, GNOME, and KDE

Need to Create Additional user and enforce privileges



- We should not boot into a system, which has only a root user account. This way, every change we make will happen without any authentication required, and we might end up messing up your system.
- Of course, we can still choose to do it, but it is not the recommended solution for a stable and secure experience.
- So there should be an additional user, who gets root privileges using sudo.
- For that we must first download a package called sudo using pacman –S sudo.
- After that create a user with password and add him/her in the group of user with special previllages which can act as superuser using sudo cmd.


bakasur@myarch:~
[bakasur@myarch ~]$ neofetch
bakasur@myarch
OS: Arch Linux x86_64
Host: VirtualBox 1.2
Kernel: 6.8.2-arch2-1
Uptime: 7 mins
Packages: 720 (pacman)
Shell: bash 5.2.26
Resolution: 1280x800
DE: GNOME 46.0
WM: Mutter
WM Theme: Adwaita
Theme: Adwaita [GTK2/3]
Icons: Adwaita [GTK2/3]
Terminal: kgx
CPU: Intel i5-10310U (2) @ 2.207GHz
GPU: 00:02.0 VMware SVGA II Adapter
Memory: 761MiB / 1947MiB
[bakasur@myarch ~]$

Troubleshooting Linux Issues

1

Identify the Issue

Gather information about the problem, such as error messages, system logs, and unexpected behavior.

2

Diagnose the Problem

Analyze the collected information to determine the root cause of the issue.

3

Research Solutions

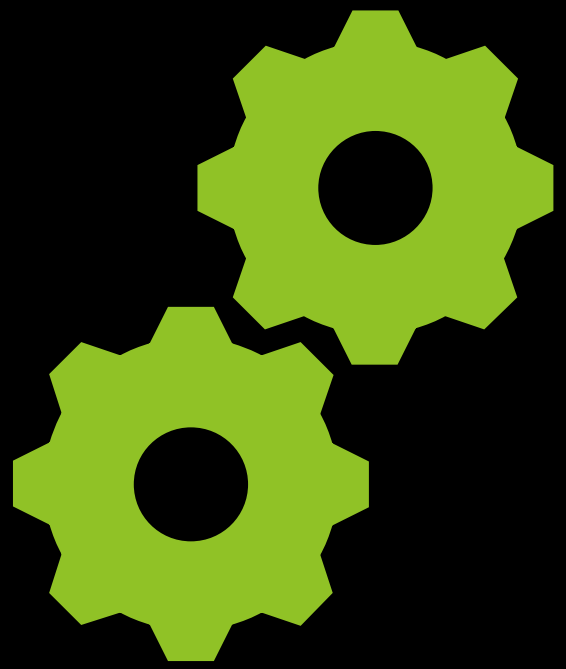
Search online forums, documentation, and community resources for potential solutions to the problem.

4

Implement the Fix

Carefully apply the appropriate solution, testing it to ensure the issue is resolved.

Troubleshooting Linux issues requires a methodical approach. Start by gathering relevant information about the problem, then analyze the data to pinpoint the root cause. Next, research potential solutions and implement the most appropriate fix, verifying that the issue has been resolved.



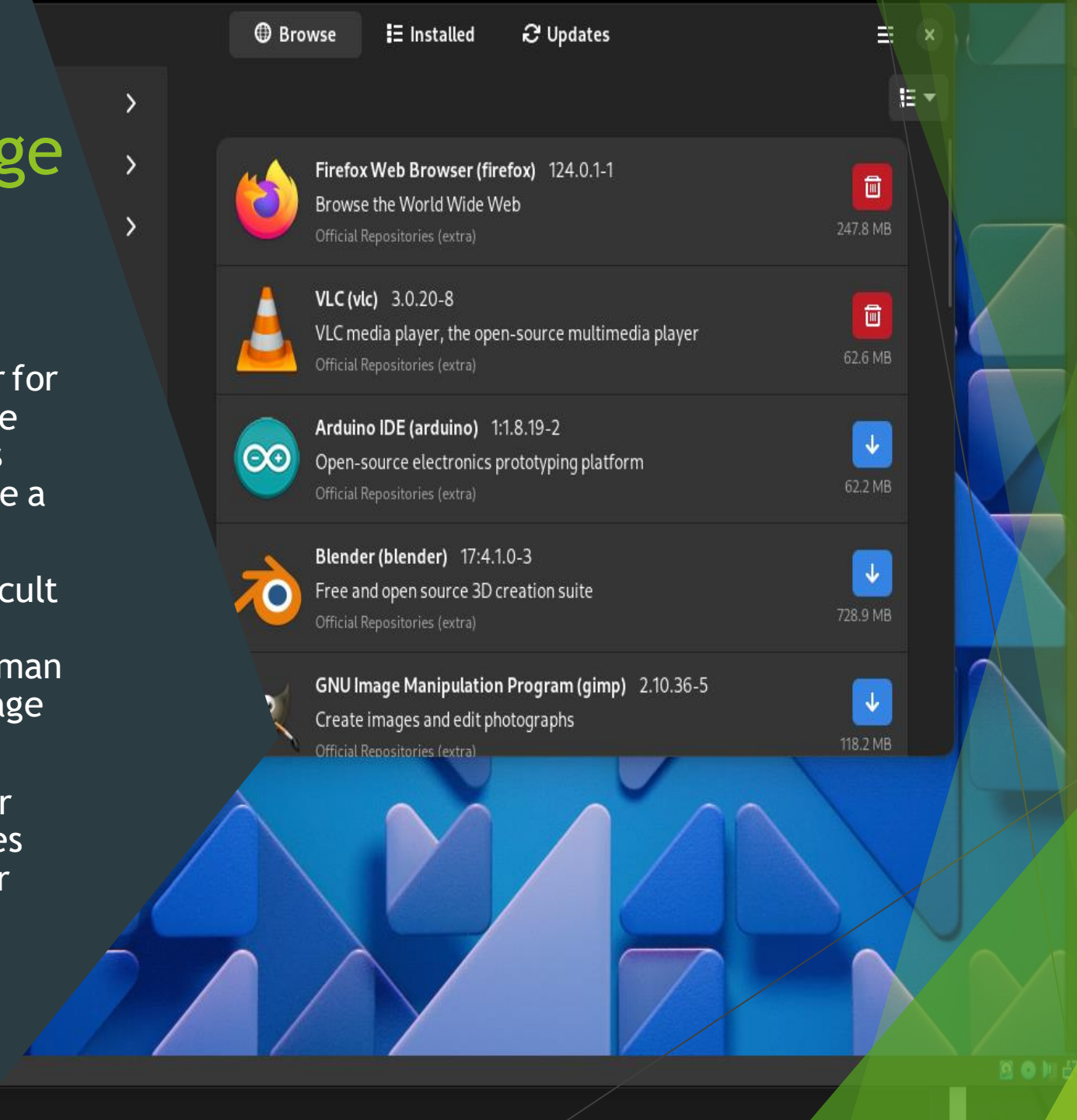
Customizations

Install an LTS kernel

- Installing an LTS kernel means you have a more stable kernel with better support to older hardware. Also, the LTS kernels are supported for at least 2 years with bug fixes and performance enhancements.
- If you rather choose to use the latest Linux kernel, you may find regression and bugs introduced by the latest kernel updates to your existing software and system. It's not a certainty but it is definitely a possibility. For example, a Kernel update broke GNOME in Arch based Linux some time back.
- This is why it is advisable to use an LTS kernel if you prefer a more stable system and/or have an older one.

Install GUI Package Manager Pamac

- The default package manager for Arch Linux is Pacman (Package Manager) and using Pacman is quite easy to install or remove a software.
- However, it's sometimes difficult to talk in commands. Pamac provides a GUI option for Pacman and works like Synaptic Package Manager or GNOME Software.
- Pamac serves as a GUI tool for installing or updating packages and works well with Arch User Repository AUR.



"Yay" an AUR helper

- Yay is a popular AUR (Arch User Repository) helper written in Go. It simplifies the process of installing packages from the AUR on Arch Linux

Install

Yay (AUR Helper)

on Arch Linux

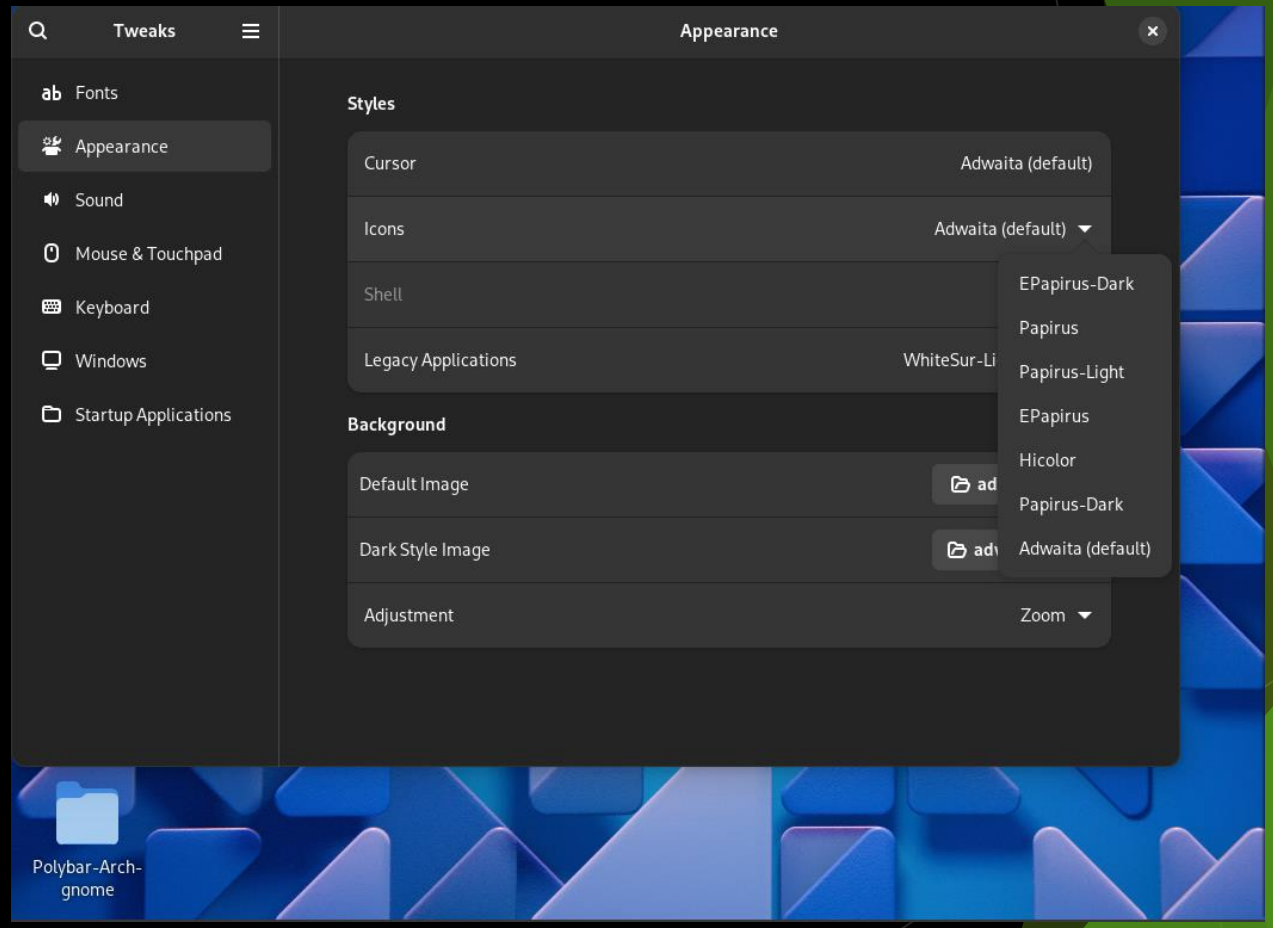
Installing Codecs and plugins

As going ahead in any system we do many such works involving media like video and audio images all these require their necessary plugins and codecs to run on our system.

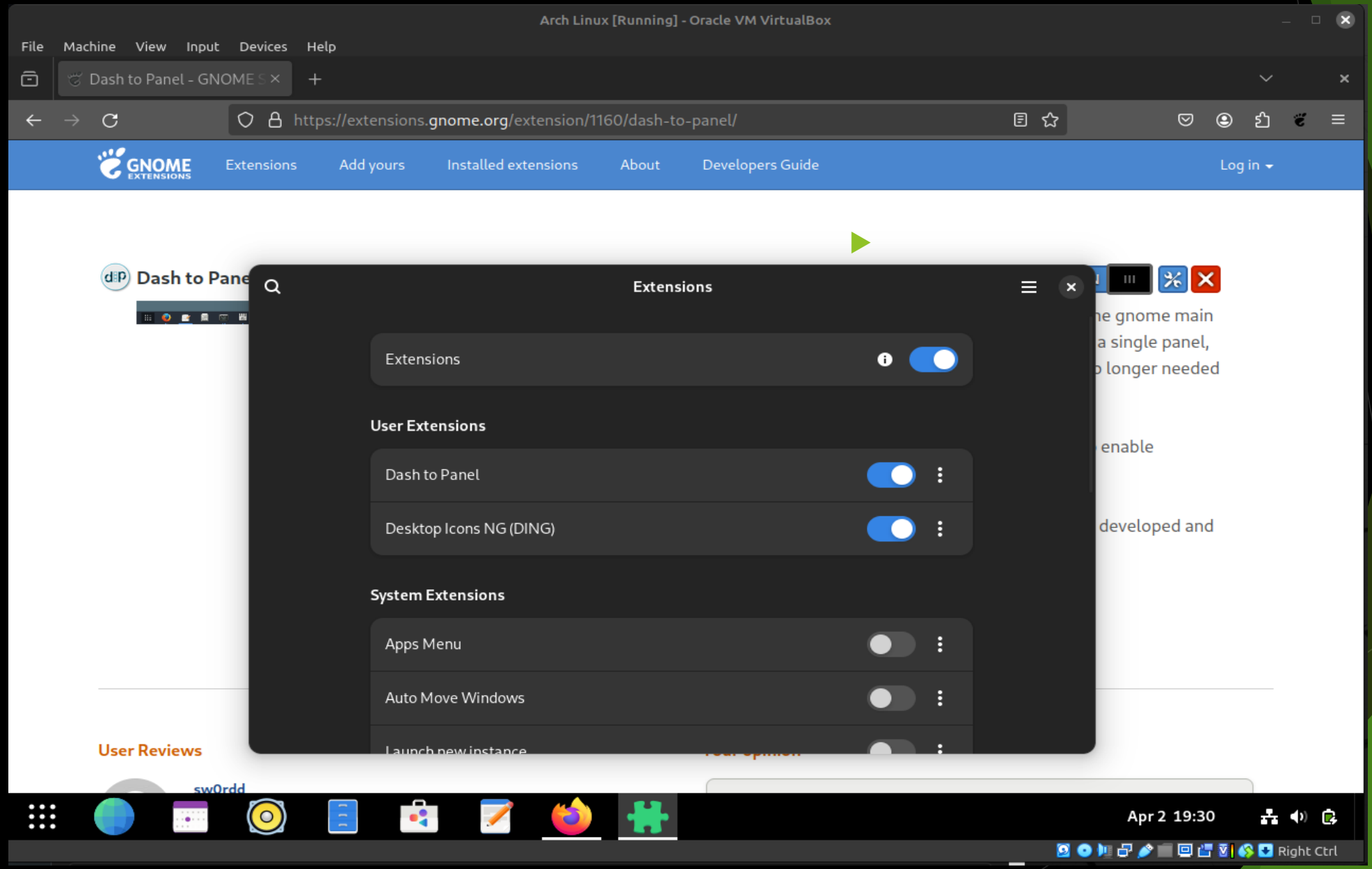
However, installing a media player like VLC imports all the necessary codecs and installs it.

Gnome tweak tool

- Gnome Tweak Tool (Tweaktools) is **a powerful tool that allows users to customize every aspect of the Gnome desktop environment.** It allows users to change themes, icons, system fonts, menus, cursors, extensions, and other settings and customization of the Gnome Shell interface.



Gnome browser shell extension



Terminal Customization

Installing themes and monitoring tool

Window Compositors

File manager and PDF reader

Windows Managers & Screen Resolution

Spotify & Spicetify