

Wiki

arcticinferno

May 4, 2023

Contents

1	Introduction	2
1.1	What is this wiki aiming to do?	2
1.2	What we will document	2
2	Getting Started	5
2.1	Installation	5
2.1.1	Prerequisites	5
2.1.2	Flashing The ISO	5
2.1.2.1	MacOS	5
2.1.2.2	Windows	5
2.1.3	Pop!_OS	6
2.1.3.1	Installing the Operating System	6
2.1.3.2	Should I encrypt my disk?	6
2.2	Getting Started With Your New Operating System	6
2.2.1	Updating Your System Packages	6
2.2.2	Using Your Package Manager	7
2.2.3	What actually is a Linux distribution?	7
2.2.4	Alternative ways to install packages	7
3	System	8
3.0.1	Structure of a Linux Distribution	8
3.0.1.1	File System Structure	8

Chapter 1

Introduction

1.1 What is this wiki aiming to do?

Inside this wiki/document, we intend to document various parts of Linux that you may encur while switching. I have compiled a list of things that I intend to cover, and will put them in a section just ahead. This document will consist of everything we write, but I intend to eventually "port" it to a website format, but for now, it should be fine as a document.

1.2 What we will document

I will include a tree of things that will be documented ahead, but I would first like to thank the people from the following reddit threads for giving me a bunch of things to add.

Reddit. *r/linux*. https://www.reddit.com/r/linux/comments/12lyrw/im_going_to_write_a_full_progression_guide_for/. April 2023

Reddit. *r/linux*. https://www.reddit.com/r/linuxquestions/comments/12lzryp/im_going_to_write_a_full_progression_guide_for/. April 2023

Alright, now that we've got that out of the way, here are the things I'm going to document: **also please note that this is not in order*

1. Hardware
 - (a) To Be Expanded (Supported hardwarem, good brands, etc.)
2. Use Cases
 - (a) Developing Software
 - (b) Gaming
 - (c) System Administration

- (d) To Be Expanded
- 3. Installation
 - (a) What is a Linux Distribution?
 - i. What makes up a "distro"
 - A. The Kernel
 - B. The Shell
 - C. The Applications
 - D. The Package Manager(s)
 - ii. Should I use a small distro? Should I use a large distro?
 - iii. Why you shouldn't use Kali Linux as a desktop.
 - (b) Burning A USB Drive
 - (c) Booting into the BIOS
 - (d) Dual Booting - Benefits & Disadvantages
 - (e) Installing Linux
- 4. System
 - (a) Mounting Drives
 - (b) Automounting Drives
 - (c) File Structure
 - (d) Networking
 - (e) System Configuration
 - (f) System Administration
 - (g) Users
- 5. Useful Resources
 - (a) Arch Linux Wiki - <https://wiki.archlinux.org/>
 - (b) Gentoo Linux Wiki - <https://wiki.gentoo.org/>
- 6. Gaming
- 7. Virtual Machines
 - (a) KVM
 - (b) Xen
 - (c) Virtual Box
 - (d) QEMU + {KVM, Xen}
- 8. Advanced Parts Of A Linux System
 - (a) Types Of Distributions - Source Based / Binary Based

- (b) Init Systems
 - (c) Boot Loaders
 - (d) Kernel(s)
 - (e) Filesystems
9. Trying out more advanced software
- (a) Window Managers
 - (b) Text Editors
 - (c) Terminal Emulators

Chapter 2

Getting Started

2.1 Installation

The majority of these steps will work for any Linux distribution, but I will specifically document the process for pop_os

2.1.1 Prerequisites

You will need the following thing(s) to install Linux:

1. A USB drive with \geq 4GB of space

2.1.2 Flashing The ISO

2.1.2.1 MacOS

In MacOS you won't need to download any additional software to burn the ISO to your USB drive. Simply navigate to the folder in which you downloaded the ISO, and run the following commands:

```
lsblk
```

This will show you the drives that you can install to.

```
sudo dd if=<iso> of=/dev/<drive> status=progress bs=4M
```

After the command fully finishes, simply remove the drive.

2.1.2.2 Windows

Compared to UNIX systems like MacOS, Windows is a bit trickier when it comes to flashing ISOs. You will need to download a program to do it, which we will briefly cover next. Once you have your ISO, you can choose one of many different ISO flashing utilities, but for the purpose of this guide, we will be covering a program called Balena Etcher. You can download etcher at <https://www.balena.io/etcher>

2.1.3 Pop!_OS

To get started, we will download the official ISO from their website. (<https://pop.system76.com/>) Also, to note, if you have an NVIDIA GPU, you should select the nvidia option whilst downloading. This will make it a lot easier in the future.

2.1.3.1 Installing the Operating System

When you are installing most linux distributions, they will typically offer you a chance to explore the operating system without actually writing to the disk, thus allowing you to get used to what the system looks and feels like. Please note that sometimes performance will not be optimal until it has installed, as it is running off a USB, and you probably won't have many drivers installed yet. When you boot up PopOS, you will be dropped into a live environment, and you should definitely play around with it a bit before downloading.

Now that you've hopefully played around with it for a while, we can start installing the operating system. Navigate to the bottom dock, and click the gray box. You will be put into a Calamares (Installation tool) menu, and there will be very simple questions asking you about what you want your system to look like.

2.1.3.2 Should I encrypt my disk?

Simply put, yes. It is a good idea to encrypt your disk, as it prevents people from pulling data out from it. There are some downsides, though. When you encrypt the disk, things like rescuing a locked account are near impossible, unless you know the decryption key. You also cannot decrypt the drive after.

2.2 Getting Started With Your New Operating System

Firstly, congratulations on successfully installing Linux! You've already overcome most of the challenge of changing operating systems. Now we can progress to the next steps.

2.2.1 Updating Your System Packages

On any new Linux install, you should always update the packages to ensure you are using the newest and most up to date versions.

On Pop!_OS, you can run the following commands to update your packages:

```
sudo apt update && sudo apt upgrade
```

2.2.2 Using Your Package Manager

On Debian based operating systems like Ubuntu and Pop!_OS, you get to use the ‘apt’ package manager. Here is a list of commands you can use to interact with the package manager:

```
sudo apt install <package>
sudo apt remove <package>
sudo apt search <package>
```

There are a lot more commands that go with ‘apt’, but for now that should suffice.

2.2.3 What actually is a Linux distribution?

Simply put, a linux distribution is any system that consists of the Linux kernel, and a set of coreutils. The Kernel is responsible for enabling everything to work (For example, the Linux Kernel enables people to use various pieces of hardware.) But at a higher level, your distro consists of a lot more software than that.

Kernel What enables your system to use programs, hardware, and more.

Core Utilities The programs that let you interact with your computer (e.g. ‘ls’, ‘cp’, ‘mkdir’, etc.)

Init System The program that starts other programs and services.

Boot Loader The program that loads the kernel into memory.

Package Manager The program that installs and updates packages.

Terminal Emulator The program that runs commands in your terminal.

File System The system that allows you to interact with files and create stuff.

Shell The program that wraps your kernel.

On most distributions, you will never really really interact with things like the kernel or init system directly, as there are many wrappers that are in place to prevent you from messing it up.

2.2.4 Alternative ways to install packages

On most modern linux distributions, there will be some sort of graphical way to install packages (typically called a software centre or application manager,) for example, on Pop!_OS, there is a program called ‘pop shop’ which lets you install software graphically. I do not recommend that you use these graphical methods, as they abstract over the actual package manager, and any issues that you incur may not be easily solvable.

Chapter 3

System

3.0.1 Structure of a Linux Distribution

3.0.1.1 File System Structure

On Linux, we have a way of structuring our file system that differs a lot from things like Windows, but is somewhat similar to Unix based operating systems like MacOSX or *BSD. Here is a small tree view of the file system with a summary:

```
→ / (root directory)
  → /bin (user binaries)
  → /sbin (system binaries)
  → /etc (configuration files)
  → /dev (device files)
  → /proc (processes (fake directory))
  → /tmp (temporary files)
  → /usr (user files/programs)
  → /var (variable files)
  → /home (home directory for users other than root)
  → /root (root directory)
  → /boot (boot loader files)
  → /mnt (mount points)
  → /lib (system library files)
  → /sys (system files)
  → opt (optional files)
```