Welcome

Welcome to Oregon State University Linux Users Group documentation on Linux. If you find any errors, or wish to help expand this documentation please see contributing.md

Goals of This Documentation

- **Inform** newcomers can figure out how to get started.
- **Guide** newcomers can get an overview of what Linux is, as well as some areas in which they may want to learn more.
- **Showcase** useful applications that may be otherwise hard to find.
- **Reference** users to help them understand how the components operate together.
- **Refer** readers to existing external documentation when possible, but not being afraid of explaining if no such useful documentation exists.
- **Learn** from each other

Using This Documentation

Generally the more basic topics will be listed first. Feel free to skip around.

Sections

**Home**

This has general information about how to get started. As well as how to get connected and involved with the club. Highly recommended that you read these.

**Install**

How to install various distributions of Linux, in various different ways (bare metal install, dual-boot, virtualized).
Applications

Collection of applications, such as word processors, audio editors, media viewers, etc.

Tools

Similar to applications, but with more of a utility goal. Such as ping, dig, ncdu, etc.

Linux

Covers a wide range of Linux related topics. Such as how to use bash, components of Linux, virtualization, distributions, backups, etc.

History

Some history on how Open Source went from nothing but a dream, to the widespread phenomena it is today.

OSU

Helpful information about using OSU services on Linux, as well as any notes on using Linux in specific classes.

Terminology

- **LUG** - Linux Users Group
- **distro/distribution** - a particular packaged configuration and software with Linux. More detail

Thank You

Huge shout out to the amazing people at the Open Source Lab who are providing the resources for hosting.
Useful Resources

Arch Wiki

Useful for a overview and medium detail explanation about configurations of Linux components as well as packages. Although package names vary from distro to distro, nearly all of the information transfer.

https://wiki.archlinux.org/

Gentoo Wiki

Useful for a more technical lower level detail of Linux components.

https://wiki.gentoo.org/wiki/Main_Page

Kali

https://www.kali.org/docs/

LinuxDocs.org

http://linuxdocs.org/

Debian Docs

https://www.debian.org/doc/

Linux Foundation

https://www.linuxfoundation.org/projects/linux/

Tutorials Point

https://www.tutorialspoint.com/unix/index.htm
ibiblop.org

I collection of projects and operating systems with links to their wikis. May not be limited to Open Source, but does include them.

https://www.ibiblio.org/catalog/items/browse

die.net

Has a cache of hundreds of manpages, as well as some basics

https://linux.die.net/

The Linux Documentation Project

This is rather old, (most recent article dates from 2012), but may have some useful things

https://tldp.org/
Connect With Us!

Meetings
Weekly on Tuesdays at 6pm on Jitsi (link) Join one of our social platforms for the meeting password.

Mailing List
Join our mailing list here

Communication
Matrix is our primary communication method, however Discord and IRC are bridged, so feel free to join whichever you feel like.

Matrix
- Go to app.element.io
- Create an account if you do not already have an account on a Matrix server
- Join our community
- Join our channels at #general:osulug.club and #announcements:osulug.club

Discord
- Download discord (or run in browser)
- Join a server by hitting the plus in the lower left hand corner
- Join our server with the invite link: rwNeYB2

IRC
- Open an IRC Client (ex. freenode's in-browser)
- Put in a unique nickname and go to the channel osu-lug
- Register your nickname by following the instructions here More comprehensive documentation found here
Contributing

Documentation

1. Fork Our Gitlab repo
2. Make your changes
3. Make a PR (See Git for help using Git)

Markdown

See here for reference.

Markdown Standards

Although we are using mkdocs, and that does support markdown extensions, only use standard markdown, so these documents can be portable.

Including

• Headers
• Lists (both numbered and unnumbered)
• Fenced Code blocks (triple backtick)
• Command (single backtick)
• Images
• Links
• Emphasis (Bold, Italics, Underline, Strike-Through)

Excluding

• Horizontal Rules (use headers instead) (if you do find a reason to use these, use *** and specify in the PR why it was needed)
• Tables (not standardized) (if you find a case to need these, feel free to do so in the PR)
• Other fancy elements
Spacing Standards

- Do not enforce a max width.
- Space after the # in a header before the text.

Table Of Contents

mkdocs only generates a Table of contents if you have one top level header.
Pre-Installation

**IF YOU ARE EVER UNSURE OF HOW TO PROCEED, FEEL FREE TO ASK!**

This section will occasionally reference alternate configurations, if you want that particular option, go to the associated section.

Pick a Distro

Which distro you choose will be based on:

- The speed of your computer
- How stable you want it (normally older software is more stable)
- The Desktop Environment
- And of course personal preference See a brief list of distros here

Select a Target Platform

Bare Metal

This is installing directly onto your computer. Dual Booting instructions at the bottom.

Virtualbox

This is for installing Linux in a Virtual Machine

Windows Subsystem for Linux

This is for those interested in being able to run a Linux terminal within Windows. This does not give you the full Linux experience.
Bootable USB

A Bootable USB is a USB drive that has a mini operating system on it. Just enough of one to allow you to boot.

Creating

1. Download an iso to ISO-PATH. An iso is the raw 1s and 0s of the entire operating system: you can find it by searching the internet for "\<distro> download".

2. Flash the iso to a USB drive. NOTE: etcher is easier than the below methods, but it sends tracking requests to various sites including google-analytics (source).
   - If on Windows use a program called "rufus"
   - If on Mac or Linux
     a. Open Terminal as administrator
     b. Find the drive path of the drive you wish to write (sudo fdisk -l can help with this)
     c. MAKE SURE YOU HAVE THE CORRECT DRIVE
     d. MAKE SURE YOU HAVE THE CORRECT DRIVE AGAIN
     e. Execute `sudo dd if=ISO-PATH of=DRIVE-PATH status=progress`

Terminology Notes:

Installation Media

A way to install another operating system, sometimes a Live USB.

Live USB

A bootable operating system that you can preform normal tasks in. This does not store the files you create, much like a guest session. Can be used for recovery and trying out distros.
Bare Metal

For installing directly on hardware.

Entering the BIOS

For more information about BIOS see [here](#).

**BE CAREFUL IN THE BIOS.**

1. Turn off your computer
2. Plug in your Installation USB (for more information about bootable usb see [here](#))
3. Turn on the computer
4. Before the laptop manufacturers logo appears press F2 repeatedly. You should see something like one of the images below, if not see [here](#).
<table>
<thead>
<tr>
<th>Configuration Item</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOS Version</td>
<td>2.21 (7LETC1WW)</td>
</tr>
<tr>
<td>BIOS Date (Year-Month-Day)</td>
<td>2008-07-01</td>
</tr>
<tr>
<td>Embedded Controller Version</td>
<td>1.00</td>
</tr>
<tr>
<td>System-unit serial number</td>
<td>7659C0L3L9169</td>
</tr>
<tr>
<td>System board serial number</td>
<td>VFL67B61T2</td>
</tr>
<tr>
<td>CPU Type</td>
<td>Intel (R) Core (TM) 2 Duo CPU T7500</td>
</tr>
<tr>
<td>CPU Speed</td>
<td>2.20GHz</td>
</tr>
<tr>
<td>Installed memory</td>
<td>4096MB</td>
</tr>
<tr>
<td>UUID</td>
<td>b6181a01-48e3-11cb-a2e7-c97da1a6ced4</td>
</tr>
<tr>
<td>MAC Address (Internal LAN)</td>
<td>00 1E 37 1C 66 04</td>
</tr>
</tbody>
</table>
Allowing the USB to Boot

Unfortunately due to the vast array of layouts of BIOS, all I can do is describe what you are looking for. While you are looking, make sure not to change any settings.

1. First we need to turn off Secure Boot
   - Sometimes this setting is alongside a booting option
   - Sometimes this is under a security tab
   - Secure Boot will always be labeled "Secure Boot" exactly
2. Move the USB to the top of the boot order sequence
   - The boot order sequence is normally under a section on booting
See the bottom of your screen for which keys move items up and down in the list, normally it is F5 for Up, F6 for down

3. Exit the BIOS
   - Normally this is F10
   - Make sure you save and exit

Dual Boot

If you want to dual-boot with existing data on your disk continue following the instructions, otherwise move on.

Windows - Free Disk Space

Before we can install Linux we have to make some space on your hard drive.

1. Boot into Windows (if you are unable to do this see Below)
2. Login
3. Open Disk Management
4. Click on C: in the top pane
5. Right click on the highlighted partition in the bottom pane and choose shrink volume
6. In "Enter the amount of space to shrink in MB" put the number of MB that you want to give to Linux there.
   - See here if you are unsure how much to give.
   - Note that this is in MB. If you want to give it 10 GB, that would be 10*1024=10240MB.
7. Hit Shrink and wait.

Linux - Free Disk Space

1. Boot into a Live USB (see here if need to configure your BIOS) (see here for more about bootable USBs)
2. Start Live environment
3. Start gparted (install it if it is not installed)
4. If you don't see your hard drive see here
5. Right click on the largest partition and hit "Resize/Move"
6. Enter "Free space following" that you want to give to Linux
   - See here if you are unsure how much to give.
   - Note that this is in MB. If you want to give it 10 GB, that would be 10*1024=10240MB.
7. Hit "Resize/Move"
8. Ensure that any existing EFI partition is at least 512MB (see here)
9. Hit the checkmark near the top, then hit Apply
10. If the operations fails, ask for help.
Virtualbox

Initial setup

1. Install Virtualbox
2. Enabling Virtualization
   a. Boot into the BIOS (see here)
   b. Look for item(s) that say virtualization, vt-d, vt-x or amd-v
   c. Make sure they are on
3. Reboot
4. Go to the website of the distro you want to install, and install the iso.
5. Open Virtualbox

Creating the VM

1. Hit "Machine" in the menu bar
2. Hit "New"
3. Choose a name for you VM
4. Make sure the Type is Linux (if it wasn't change version to generic linux)
5. Choose the amount of RAM given to your VM (more is better, but make sure you still have enough to run your normal operating system)
6. Create a virtual disk with 12-16GB if you can afford it
7. Right click on the name of your VM on the left and Hit settings
8. Go to the Storage tab
9. Hit the drive (should be under "Controller IDE")
10. Hit the disk icon to the right of "Optical Drive" on the right pane
11. Browse to where you downloaded the iso, then hit OK
12. See here to install.
Windows Subsystem for Linux

This installation method is fundamentally different than others as WSL is a tool to run a Linux terminal within Windows. This is good for how to use the command line, but does not feature a full Linux Desktop.

NOTE: Some software will not work as intended within WSL compared to a normal Linux distribution.

https://docs.microsoft.com/en-us/windows/wsl/about

https://docs.microsoft.com/en-us/windows/wsl/install-win10
Launching the Installer

Much like the BIOS section, each installer varies slightly, including a different order. I will include a list of the general steps, although each installer may do them in a different order.

1. Boot into the Installation USB by restarting your computer with the USB plugged in.
2. Start the installation process, using the following as a selection guide.

Language and keyboard

In my experience this flawlessly picks the correct default. If it doesn't, manually select the option you want.

Networking

If possible, connect to a network, although it is not explicitly required for most installers. If prompted, download updates while installing.

Partitioning

- If you are dual-booting select "Install Alongside ..." if the option exists. If not, see Linux partitioning
- If you want to wipe the entire drive and just install Linux select "Erase disk and install..."
- If you need a more advanced configuration, see Linux partitioning

User

Create your user.

Finishing Up

Confirm that you choose all of the options that you want. Especially confirm that you set the partitions up correctly.
That's it! Congrats! For some basic post-install recommendations, see here.
Post-Install

Dual-boot
If you setup a dual-boot, make sure that you install os-prober.

Installing Software
Generally searching for "Software" or "Program" on the Beginner Distros will pull up a nice interface for you to search for and install packages. For more see here.

Updating
In general, search for "Update". Some distros have a nice interface for updates. If not, see here.

Security

• With Windows you needed an Anti-Virus program, but not in Linux. (see here if you want one anyway). However, it is recommended that you have a firewall to block other computers on the network from making connections into your machine.

• To set up a firewall:
  a. Install the package gufw.
  b. Open gufw
  c. Click on the status slider to turn it on.
How Much Space Does Linux Need?

Linux needs at least 16GB to be usable, although it is recommended to use 32 or 64 if you have enough space. If you plan on having large files in Linux, increase this number to 128. You are always welcome to increase this if you wish.

Linux - Resizing EFI

1. Move each partition 512MB left after the EFI partition going right to left (by putting 512 in the space preceding)
2. Hit the checkmark near the top, then hit Apply
3. Note the name of the EFI partition (EFI-PARTITION) (something like /dev/sda1 or /dev/nvme0n1)
4. Backup the files in the EFI partition by executing `sudo mount EFI-PARTITION /mnt; mkdir efi; sudo cp -R /mnt/* efi/; sudo umount /mnt/` in a terminal
5. Delete the EFI partition
6. Create a new fat32 partition of 612MB where the old EFI was with a label of "EFI"
7. Hit the checkmark near the top, then hit Apply
8. Right click on the new EFI partition and "Manage Flags" then add the "esp" and "boot" flags.
9. Copy the files back by executing `sudo mount EFI-PARTITION /mnt; sudo cp -R efi/* /mnt/; sudo umount /mnt` in a terminal

Linux - Partitioning

The interface varies per distro, so I will describe what you are looking for. **BE CAREFUL ON THIS STEP**

1. Create an EFI partition if you don't already have one
   a. Find the + or the add partition button
   b. Select a size of 512MB
2. If you already had an EFI partition, mount it as /boot/efi (but **DO NOT** format it)

3. If you want, make a 8GB SWAP partition (more info [here](#))

4. Create your root partition formatted as ext4 with how much space you want to give Linux ([more info about partitions](#)). See [here](#) if you are unsure how much to give.

5. If you want a separate /home (your files, see [here](#) for more information), which is recommended, create one now with Ext and the rest of the space.

6. Hit Apply

### Alternate BIOS Keys

Rather than pressing F2, try the key below that matches the manufacturer of your laptop.

- ASRock: F2 or DEL
- ASUS: F2 for all PCs, F2 or DEL for Motherboards
- Acer: F2 or DEL
- Dell: F2 or F12
- ECS: DEL
- Gigabyte / Aorus: F2 or DEL
- HP: F10
- Lenovo (Consumer Laptops): F2 or Fn + F2
- Lenovo (Desktops): F1
- Lenovo (ThinkPads): Enter then F1.
- MSI: DEL for motherboards and PCs
- Microsoft Surface Tablets: Press and hold volume up button.
- Origin PC: F2
- Samsung: F2
- Toshiba: F2
- Zotac: DEL

(Source: Tom's Hardware)
Hard drive not detected

1. Go into your BIOS (instructions [here](#))
2. Look through settings relating to hard drives (sometimes under configuration)
3. Turn off Optane
4. If "SATA Controller Mode" is set to "RT" and you have an existing Windows install:
   a. Save and Exit
   b. Boot into Windows
   c. Open Command prompt as Administrator
   d. Execute `bcdedit /set {current} safeboot minimal`
   e. Enter BIOS
   f. Turn "SATA Control Mode" to "AHCI" (don't worry your data won't be lost)
   g. Save and Exit
   h. Boot into Windows
   i. Open Command prompt as administrator
   j. Execute `bcdedit /deletevalue {current} safeboot`
5. Save and exit if you are in the BIOS
6. See if the hard drive is detected now
Basic Commands

Overview

Generally speaking you execute a command with some arguments and flags.

Argument

An argument is just some string given to the binary. For example in the command to change to your Documents directory `cd Documents`, `Documents` is the argument.

Flags (or OPTIONS)

Flags are used to modify the behavior of a command. For example `ls -a`, `-a` tells the `ls` executable to list all files in the directory, including hidden ones. Flags are also called OPTIONS.

Commands

`cd`

Change Directory, changes your "working directory" to the specified folder. For example `cd Documents` changes to your documents folder

`pwd`

Print Working Directory. Outputs your current working directory to console. Something like `/home/user/Documents/`.

`ls`

List files. Lists the files in your current directory. Use the `-a` flag to see hidden folders.

`echo`

Echos the string given as an argument.
Under The Hood

Command

A command is really just an executable found in your \$PATH. PATH is the environment variable that holds where to look for executables. You can see the value of this by executing `echo \$PATH`.

Environment Variables

Environment Variables are variables used to pass into executables. PATH is an environmental variable, so is HOME. You can see them all with `env`, and set them with `export NAME=VALUE`.

Getting Home

`~` in most shells is replaced with your home directory (normally /home/USERNAME). You can get the absolute path of your home directory with `echo \$HOME`, or just go home with a quick `cd` (no arguments needed).
Distros

This is a distribution, or a specific configuration on top of the base GNU/Linux system. See here for more detail.

See DistroWatch for a full list of Linux Distributions.

Beginner

Mint

Parent Distro: Ubuntu

Pros:
- Built on Ubuntu
- Doesn't have built-in shortcut to Amazon

Cons:
- None

Ubuntu

Parent Distro: Debian

Pros:
- This is the most common user friendly choice for computers

Cons:
- Has a built-in shortcut to Amazon's store

Manjaro

Parent Distro: Arch

Pros:
- Accessible to both newcomers and has extensive options for more experienced users
- Easier to install than Arch
Built on Arch
Cons:
• Not as user friendly as Ubuntu

Pop OS
Parent Distro: Ubuntu
Pros:
• Gaming
• Made by system76
Cons:
• Only has support for one desktop environment

Intermediate
OpenSUSE
Parent Distro: Independent
Pros:
• Some rare pieces of scientific software will focus on supporting openSUSE
Cons:
• Few people use comparatively
• Lack of Pros

Debian
Parent Distro: Independent
Pros:
• Less bloat that would normally come with Ubuntu
• Works on a Raspberry Pi
Cons:
• Lack of Pros
Fedora
Parent Distro: Independent Pros:
• Less bloat
Cons:
• Lack of Pros
• Less user friendly

Advanced
CentOS
This is normally a server OS, yet it can also be used as a normal operating system as well. Parent Distro: Fedora Pros:
• Made for servers
• Stability
Cons:
• Less user friendly
• Older software (takes time to get feature updates)

Arch
Parent Distro: Independent Pros:
• Just what you need to run a system
• Practically no bloat
• Small
• Newest software
Cons:
• Less user friendly
• Can be pain to install, as you do everything manually
• Manjaro is easier to install
Crazy Qubes OS

Parent Distro: Fedora

Everything in this distro runs in VMs, so you can run Fedora/Debian/Arch applications nearly seamlessly. Pros:

- Security
- Custom-ability
- Multiple distros simultaneously, so you can handpick the distro for the job

Cons:

- Can just break
- Smaller project
- Less support
- Takes time and knowledge of how the VMs are arranged to setup properly
- Does not run in a VM
- Requires your CPU to have certain technologies
- Takes a lot of RAM

Gentoo

You have to compile this yourself. Parent Distro: Independent Pros:

- Just what you need to run a system
- Practically no bloat
- Can run on strange architectures
- Can be installed beside Android

Cons:

- Is a pain to install
Desktop Environment

This defines how the windows are arranged, and other workflow related tasks (like the

Choosing a Desktop Environment

Beginner

**GNOME**

- Defaults on Fedora, Debian, Ubuntu, CentOS, etc.
- One of the larger contributor networks
- Many familiar features to MacOS and Windows
XFCE

- Lightweight and performant
- "Aims to be fast and low on system resources"
Xfce is a desktop environment that offers the following features:

**Window Management**
Handles window management for applications.

**Panel (xfce4-panel)**
Programs can be added to the panel.

**Desktop Interface**
Sets the background image and applies it to all.

**File Manager**
A basic file management tool with file browser features.
Cinnamon

- Fork of GNOME 3, for Linux Mint
- Created after disagreement of GNOME 3 design direction, follows more traditional patterns
KDE Plasma

• Most Windows-like
• Multiple virtual desktops, with custom layouts for each
• Also a very large contributor and support network

MATE

• The continuation of GNOME 2
• Interested in "preserving a traditional desktop experience"
Intermediate

i3

- vi-like control system, mostly with no mouse control
- New windows maximize screen size by default, in a tiling-manner.
- Minimal interface and features, just enough to provide access to program-specific GUIs.
This is tiling window manager (windows split the screen as shown) \[ \text{image}\{\text{width}=0.7\text{\linewidth}\}\]

**bspwm**

- Tiling window manager like i3, but represents windows as leaves of a full binary tree.
- Focuses on configuration and standardization through X11 events, and uses separate program (bspc) to manage them.

**LXDE**

- Performance as a priority
- Highly modular, with few inner dependencies making it easier to install only what a user wants.

**Advanced**

These are highly configurable, but they don't work well out of the box

**Openbox**

- Stacking window manager, meaning windows drawn in specific order, allowing windows to display on top of each other.
- Programs run in boxes, and openbox allows for full control on how boxes interact, run, etc.

**Awesome**

- Aims to provide a "framework window manager", where there is very minimal features.
- Provides standardized API to interact with, with out-of-the-box scripting available.
AWESOME(1)

NAME
awesome - awesome window manager

SYNOPSIS

DESCRIPTION
awesome is a window manager for X. It manages windows in different layouts, like floating, tiled, and stacking, and applied dynamically, optimizing the environment for the application in use and the task at hand.

In a tiled layout, windows are managed in a master and stacking area. The master area contains all windows except the ones currently managed. Dialog windows are always managed as floating, regardless of the layout. Floating layouts are special cases of the tiled layout where the stacking area is arranged as a rectangular fractal for the later.

Windows are grouped by tags in awesome. Each window can be tagged with one or more tags. Multiple windows can be managed using these tags.

awesome can contain small window groups which can display anything you want: all available tags, the visible windows, text, etc.

OPTIONS
-v, --version
Print version information to standard output, then exit.

-h, --help
Print help information, then exit.

-c, --config FILE
Use an alternate configuration file instead of $XDG_CONFIG_HOME/awesome/rc.lua.

-k, --check
Check configuration file syntax.

--search
Add a directory to the library search path.

-a, --no-arcb
Don’t use ARGB visuals.

-r, --replace
Replace an existing window manager.

DEFAULT MOUSE BINDINGS

Navigation
Button1 on tag name
  View tag.

Button4, Button5 on tag name
  Switch to previous or next tag.

Button4, Button5 on root window
  Switch to previous or next tag.

Button1, Button3, Button4, Button5 on layout symbol
  Switch to previous or next layout.

Layout modification
Mod4 + Button1 on tag name
  Tag current client with this tag only.

Mod4 + Button3 on tag name
  Toggle this tag for client.

Button3 on tag name
  Add this tag to current view.

Mod4 + Button1 on client window
  Move window.
Folder Layout

bin
For Binary files

boot
Basic files to start the boot process

dev
A "virtual" file system

/dev/urandom
Outputs a stream of random numbers

/dev/nvme
Nvme hard drives can be found starting with this path. Normally
/dev/nvme${DRIVEINDEX}p${PARTITION_NUMBER} (for example /dev/nvme0p0)

/dev/sd
Normal hard drives are found here, /dev/sd${LETTER}$({NUMBER}), where
letter is the drive letter, and number is the partition number (for
example /dev/sda1).

/dev/zero

etc
Additional libraries and configuration files needed to complete the
boot process

home
Where your files are stored, and your configurations

lib
Library files. Needed for most binaries.
lib64
Library files for 64bit systems. Needed for most binaries.

lost+found
Files that for some reason got partially corrupted go here. Almost always empty (and you want it that way)

media
A place to mount USBs/other external media automatically

mnt
A place to mount external media manually

opt
A place to install proprietary and other software that is self-contained

proc
A "virtual" file system that has process information

root
A home for the "root" user

run
General information about the system that is valid until reboot.

sbin
Super User (root) binaries. These binaries normally modify your system in some way, and are not needed for normal use.

sys
A "virtual" file system that represents your physical hardware.
tmp

A place for temporary files that do not persist across reboots.

usr

Additional software, normally managed by your package manager. Some configuration files are here as well.

usr/local

A place for you to add additional software, outside of the package manager. Not normally needed.

var

Varies state about the system that persists across reboots.
Partitions

Partition/File System Types

ext4
Moving files in ext4 is nearly instant. Can have files up to 16TB.

fat32
This is type of the EFI partition. Can have files up to 4GB.

swap
If you run out of RAM, your system may be able to use this space rather than crashing applications.

Partition Tables
A partition table is a table that stores where each partition is on the disk. MSDOS is one such table, and it has a limit of 4 partitions. GPT is the current standard.
Driver
Networking
Bios
Package Managers

A Package manager keeps track of what "packages" are installed on your system. These packages contain the software and the libraries that are needed to run that software.

Graphical Front Ends by Distro

- Ubuntu: Ubuntu Software
- Debian: synaptic (not built-in)
- Arch: pamac-manager (built-in with Manjaro)

List by Distro

- Debian: apt
- Arch: pacman
- Fedora/CentOS: dnf/yum
- OpenSUSE: zypper
- Gentoo: portage

Using apt

Where NAME is a package name or description (or part of one) and PACKAGE is a package name

- Search: sudo apt search PACKAGE
- Install: sudo apt install PACKAGE
- Update: sudo apt update then sudo apt upgrade

Using pacman

Where NAME is a package name or description (or part of one) and PACKAGE is a package name

- Search: sudo pacman -Q NAME
- Install: sudo pacman -Sy NAME
- Update: sudo pacman -Syu update
Using dnf or yum

Where `NAME` is a package name or description (or part of one) and `PACKAGE` is a package name

- **Search**: `sudo dnf search PACKAGE`
- **Install**: `sudo dnf install PACKAGE`
- **Update**: `sudo dnf upgrade`
Display server
Permissions
Boot loader
File input
Tty
Login manager
Kernel
Kernel module
Window manager
Firewall

Generally a firewall allows finer grain control about the connections made in and out of your computer. For a normal computer, this normally means block all incoming connections, and allow outgoing. For servers, this normally means block all incoming except for these ports from these IPs.

ufw

A simpler firewall manager. More information here

gufw

Graphical Front end for the ufw firewall 1. Install gufw 2. Open gufw 3. Click on the status slider to turn it on. 4. This will block other computers on the network from making connections into your machine.

iptables

"traditional userspace utility for managing a firewall" - Arch Wiki

Some docs here here and here for some explanation on how to use.

Other

See more here
SELinux
Overview

Goals of Security

Risk Management
GPG
Privacy

https://www.privacytools.io/
PGP
Anti-Virus

ClamAV is an Open Source virus scanner. It does not scan automatically, nor update automatically. The Arch Wiki shows how to update virus definitions as well as scan.

Frontends

Clam-gtk is a simple frontend that gives you a one-click way to update, or scan a directory. NOTE: This does not search folders recursively by default.
Checksum

Overview

A Checksum is a normally shorter value calculated from a larger input. For example a 500K file is reduced to simply 79922b4f359d2ad1281956ad878453a487d4476a1b08c315e99caf24c39a8e6a. While there is some loss of data, it is still a valuable tool to check for modification, or see if two files are the same at a bit level. Any bit change will result in a fundamentally different string. I changed one character in the file and it changed to 85ef5f1c6f53849ba696f02402f30468a18c269ff205b6048c782fe6a3f433d. More Information here.

Choosing an algorithm

There are various executables that perform this operation, some have been found to be insecure, see a list here (try to avoid using red). At this time, sha256 is widely used, however sha512 is also out.

Usage

`sha256sum FILE`
Encryption

Public vs Private

On Linux


Disk

Disk Encryption is useful when you are worried about someone stealing your physical drive. Once your computer is on, and you have provided the password, the Operating System (and all programs running), can see the decrypted version. So if you get a virus, or other malicious software, they can still read your files on an encrypted drive. However, if your drive is stolen, they will be unable to simply plug your drive into their computer and read everything. Disk Encryption makes it slightly harder to perform recovery, but not by much. If you forget your password, that’s it. The files are locked, even from you.

https://wiki.archlinux.org/index.php/Data-at-rest_encryption#Why_use_encryption?

GPG

OpenSSH
OpenBSD
FreeBSD
MacOS
NetBSD
Firejail
Virtualbox
Xen
Kubernetes

A tool to orchestrate many containers at once, designed for scale.

https://kubernetes.io/
Docker

https://docs.docker.com/get-started/

Docker Hub

A repository for Docker images. Can also create your own.

https://hub.docker.com/

Dockerfile

https://docs.docker.com/develop/develop-images/dockerfile_best-practices/

https://docs.docker.com/engine/reference/builder/

Docker Compose

Simple YAML for orchestrating groups of containers together https://docs.docker.com/compose/

https://docs.docker.com/compose/compose-file/
General

Troubleshooting

https://hackernoon.com/another-reason-why-your-docker-containers-may-be-slow-d37207dec27f
LXCD

https://linuxcontainers.org/lxd/
Programming Languages

Bash
https://cs.lmu.edu/~ray/notes/bash/

Markdown
https://www.markdownguide.org

Regex
https://www.tutorialspoint.com/unix/unix-regular-expressions.htm
https://regexr.com/
Backups

https://www.redhat.com/sysadmin/5-backup-tips

syncthing

File synchronization with options for file history. Easy to setup.
https://syncthing.net/

duplicati

Backups that can connect and push to many different sources
https://www.duplicati.com/

DejaDup

Simple GNOME backup tool.
https://wiki.gnome.org/Apps/DejaDup
https://www.howtoforge.com/tutorial/ubuntu-backup-deja-dup/

rsync

Raw file sync between two directories.
https://rsync.samba.org/
Rescue

If you are here, something has gone wrong.

My System Won't Boot

Windows

NOTE: This section only applies to problems arising when dual-booting. If you choose Windows in Grub, and it doesn't boot, follow these steps.

1. Boot into a Windows Live Media (also called Recovery Media or Installation Media)
2. Hit next
3. Choose "repair your computer"
4. Hit Troubleshoot
5. Hit Command Prompt
6. Execute `bootrec /fixmbr`
7. Execute `bcdboot c:/windows`
8. Try booting now

Linux

1. Boot into a Live USB
2. Find your root partition with gparted
   a. Launch gparted
      b. Note the hard drive name (DRIVE) and the name of your main partition (ROOT-PATH) (something like /dev/sda1 or /dev/nvme0np1)
3. To re-install grub run these commands
   a. `sudo mount ROOT-PATH /mnt`
   b. `sudo mount -bind /sys /mnt/sys`
   c. `sudo mount -bind /proc /mnt/proc`
   d. `sudo mount -bind /dev /mnt/dev`
   e. If you have a separate boot partition, mount that now to /mnt/ boot
f. `sudo chroot /mnt`

g. `sudo grub-install DRIVE`

h. `sudo grub-mkconfig -o /boot/grub/grub.cfg` (make sure `/boot/grub` exists, it may be under `/boot/grub2`)

Booting a live iso and re-installing grub
Reading Man Pages

Users
General

Unless otherwise stated all applications in this section run on Linux. Some of the software may also run on other Operating Systems.
Spacemacs

A text editor that borrows from both emacs and vim.

https://www.spacemacs.org/
Emacs

One of the two giants (other being vim) when it comes to powerful text editors. Has support for tons of plugins to improve the behavior.

https://opensource.com/article/20/3/getting-started-emacs


https://www.gnu.org/software/emacs/

https://www.gnu.org/software/emacs/manual/
Nano

A simple text editor that doesn't require any special training in order to operate. Has some keyboard shortcuts along the bottom in order to save, typing works as expected.
Vim

One of the giants (other being emacs) of powerful text editors for Linux.
Chromium

Chromium is the Open Source base for Google Chrome, before Google puts some of their integrations and tracking into it. If you need Google Chrome for compatibility reasons, try Chromium first, it will probably work the same. It also comes in the distribution ChromiumOS (rather than ChromeOS).

https://www.chromium.org/
Terminal Based

elinks
A terminal based web browser. Does not use Javascript or CSS.
http://elinks.or.cz/

lynx
A terminal based web browser. Does not use Javascript or CSS.
Tor Browser is a hardened version of Firefox Designed to make you more anonymous when browsing online (see here and here for disclaimers). Read the disclaimers and understand how Tor works before using it for anonymity. Don't use Tor for illegal activities (abuse FAQ).

https://www.torproject.org/

User Manual

Gitlab documentation

Old website
Firefox

An Open Source web browser made by Mozilla. Some privacy related configuration suggestions can be found here.
Search

DuckDuckGo

Default in Tor and Brave. Search Engine that works and doesn't store your searches. Not fully Open Source. Gets money from un-targeted ads at the top of results.

https://duckduckgo.com/about

Searx

Open Source and can be self-hosted. May not have as quality of results as some of the other providers.

https://searx.info/

StartPage

Also doesn't track you. Pays Google to use their search engine. Also earns its money from un-targeted ads (citation).

https://www.startpage.com/

Ecosia

Ecosia's mission is to use the revenue generated by their ads and use them to plant trees. They proxy Bing search results. Has a minimal amount of tracking, but it respects Do Not Track requests.

https://www.ecosia.org/
Git

Gitlab

Hosted git. The server side code for Gitlab is open source, and you can self-host. You can migrate from Github, and you can even login to Gitlab with your github account.

https://about.gitlab.com/ https://gitlab.com

Github

This is more common to see than Gitlab, however their server side code is NOT Open Source, while they advocate for Open Source, their code itself isn't. Another consideration is Github is not owned by Microsoft. https://github.com
Images

Pixel Based
GIMP
https://www.gimp.org/

Vector Based
Inkscape
https://inkscape.org/

Other
ImageMagik
Can convert files between formats, merge two photos together with transparency, and other basic operations on the command line.
http://www.imagemagick.org/
LaTeX

Resources
https://www.latex-tutorial.com/tutorials/first-document/
https://www.latex-project.org/help/documentation/

Templates
https://www.overleaf.com/latex/templates
https://www.latextemplates.com/

texmaker
https://www.xm1math.net/texmaker/

Overleaf
They have an online editor in addition to plenty of documentation about how to use some LaTeX modules.
https://www.overleaf.com/
Fortune

Outputs a quote or joke. Used purely for amusement. You can write your own, and also pair with cowsay and lolcat if you want a rainbow cow saying the quotes.
Figlet

Figlet prints out whatever text you specify to it as ascii art!
Cowsay

Outputs a Cow (or other animals with the -f flag) saying specified text.
Lolcat

Ever wanted to add a splash of rainbow to your terminal output? That is the entire purpose of Lolcat. Source here
Storage Tools

WARNING: DO NOT fill your entire drive on Linux, your system may FAIL to boot if it there is insufficient space. Keep in mind if your current user can't see a directory, these tools can't show you how much disk space those files are using.

Tools to See What is Using Disk

df
Disk Free simply tells you how much of each partition is used vs free

du
Disk Usage lists all of the files in your current directory (and all sub folders)

ncdu
Command line tool for viewing which directories (and all of their contents) are taking the most space. Much easier to follow than du.

boabab
Utility to show graphically how much space each folder is taking.
Remote Desktop Protocols

There are various remote desktop protocols, as well as several clients and servers for each protocol.

VNC

VNC is the standard remote desktop solution for Linux, however it lacks some of the nice features like audio. tightvnc and tigervnc are two implementations of this protocol for Linux. There are countless other clients and servers for other platforms like MacOS, Android, Windows, even web (novnc).

SPICE

"The SPICE project aims to provide a complete open source solution for remote access to virtual machines in a seamless way so you can play videos, record audio, share usb devices and share folders without complications." (source) It is integrated into QEMU.

NX

This is NoMachine's Protocol, it was open source up to version 2. See X2Go for a maintained client/server. The NX protocol does not require nearly as much bandwidth compared to the other protocols.

RDP

This is the Windows Remote Desktop protocol, but there are implementations for it on Linux, like freerdp. Nice thing about RDP is existing Windows integration, as well as a full feature set.
Networking Tools

Connection

curl/wget
Used for downloading files. Depending on how the website redirects, curl or wget may be easier to use.

ping
Used to check to see if a host is up by sending ICMP packets.

traceroute
Shows the different connections established between you and the destination. Useful for debugging routing issues.

DNS
DIG
Prints out DNS records for a specified domain

Packet Capture
tcpdump
Dumps all TCP packets to output. Useful for a quick way to confirm that packets are indeed flowing.

wireshark
Extremely useful tool for analyzing packet captures, does protocol parsing among other things.
Ports

nmap

Port scanner, can tell you what ports are open, as well as what protocol they are using.
Git

Software that keeps track of changes to files to do version control as well as help with multiple concurrent editors.

Basic Overview

https://rogerdudler.github.io/git-guide/

Github's Guide

Explains more in-depth about each command, useful for beginners wanting to learn more.

https://github.com/git-guides

Full Reference

https://git-scm.com/docs

Git Workflows

Ultimately there are many git flows out there, each with their pros and cons, use what is best for your project. Some flows are more complex, but ensure organization, while others keep it simple, but are less strict on keeping it organized.

Example Lightweight Github flow

https://guides.github.com/introduction/flow/

Example Remote Github Workflow

https://guides.github.com/introduction/git-handbook/#github

Another Example Workflow

Another Hugely Popular Git Flow

https://nvie.com/posts/a-successful-git-branching-model/
SSH

SSH is a protocol to remotely execute commands on a target computer. Configured incorrectly it can be a huge security risk (not the fault of ssh, simply remote command execution is that dangerous).

Configuration Tips

- Turn off root login
- Whitelist using iptables if you can
- Use SSH Keys and disable password login

SSH Keys

- Generate a key with `ssh-keygen -b 4096 -t rsa -C YOUR_EMAIL_HERE`
- Add the contents of `~/.ssh/id_rsa.pub` on your computer to `~/.ssh/authorized_keys` on the remote computer.
- Make sure to set the permissions on the remote computer correctly: `chmod 700 ~/.ssh` and `chmod 600 ~/.ssh/authorized_keys`
- Now you should be able to ssh in with `ssh USESR@HOST:22` (if no user is specified it assumes your own) (if no port is specified it assumes 22)

SSH Config

SSH config is located at `~/.ssh/config`. Used to save common hosts that you connect to so that you can connect easier.

```
Host SIMPLE_NAME
    HostName FULL_HOST_NAME_HERE
    User USER_NAME_HERE
    Port CUSTOM_PORT_HERE
```

(See here for more info)
**SSH Agent**

- Start an SSH agent with `eval $(ssh-agent)` (some desktop environments start this for you)
- Add your ssh key `ssh-add PATH_TO_YOUR_KEY` (if no path is specified it assumes ~/.ssh/id_rsa)
- Now you will not need to be prompted for your password every time. If you want to forward your ssh agent to a remote server you can use the -A flag on the ssh command.

**More information**

https://wiki.archlinux.org/index.php/SSH_keys
Contributing

https://opensource.guide/how-to-contribute/

Unix
LUG
Linux History
GNU History

https://www.gnu.org/gnu/gnu-history.html

https://www.gnu.org/gnu/thegnuproject.html
This class uses MASM (the Microsoft Assembler) and Visual Studio for assembling and running code. If you do not want to install a Windows VM and Visual Studio, MASM does work under Wine with some tweaks.

**Note:** Debugging assembly in Wine may prove much harder than in Visual Studio! `winedbg` may work, but there is no editor integration.

**Note 2:** This version of MASM may not support some things that are required by some versions of this class (like 64-bit mode). At time of writing (Winter '21), this class does not require anything extra, but YMMV.

Steps mostly taken from [this guide by Ryan Eberhardt](#).

1. Choose a Wine prefix:

   ```bash
   mkdir -p ~/.local/share/wineprefixes/masm/
   alias wine_masm="WINEARCH=win32 WINEPREFIX=~/local/share/wineprefixes/masm/wine"
   ```

2. Install MASM:

   ```bash
   wget http://website.assemblercode.com/masm32/masm32v11r.zip
   unzip masm32v11r.zip
   wine_masm install.exe
   ```

3. Install the Irvine support materials:

   ```bash
   unzip Irvine.zip
   mv ./irvine ~/.local/share/wineprefixes/masm/drive_c/Irvine
   ```

4. Set PATH, LIB, and INCLUDE in the Registry:

   These will add the MASM and Irvine library headers to the system path so `INCLUDE` directives do not need to specify the full path.

   ```bash
   wine_masm regedit
   ```
Create the following String Values:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKEY_CURRENT_USER/Environment/PATH</td>
<td>%SystemRoot%\system32;%SystemRoot%;%SystemRoot%\System32\Wbem;C:\masm32\bin</td>
</tr>
<tr>
<td>HKEY_CURRENT_USER/Environment/INCLUDE</td>
<td>C:\masm32\include;C:\Irvine</td>
</tr>
<tr>
<td>HKEY_CURRENT_USER/Environment/LIB</td>
<td>C:\masm32\lib;C:\Irvine</td>
</tr>
</tbody>
</table>

5. Grab the example projects:

unzip Project32_VS2019.zip
unzip Project64_VS2019.zip

6. Set up the wrapper script:

Put this script somewhere in $PATH, e.g. at ~/.local/bin/masm

```bash
#!/bin/bash
set -e # Stop on errors
shopt -s expand_aliases

# Get the path to the file without a .asm extension
FILENAME="$(basename "$1")"
EXTENSION="${FILENAME##*.}"

shopt -s nocasematch # String case-insensitive comparison
if [[ "$EXTENSION" = "asm" ]]; then
    FILENAME="${FILENAME%.*}" # Remove extension
fi
FILEPATH="$(dirname "$1")/$FILENAME"

# WINEDEBUG flag hides the fixme:ntdll:NtQuerySystemInformation warning (which can be safely ignored)
alias wine_masm="WINEDEBUG=fixme-all WINEARCH=win32 WINEPREFIX=~/.local/share/wineprefixes/masm/ wine"

echo "> Assembling..."
wine_masm ml -nologo -c -coff -Zi "$FILEPATH.asm"

echo "> Linking..."
wine_masm link -nologo -subsystem:console irvine32.lib kernel32.lib user32.lib "$FILEPATH.obj"
```
7. Test it out:

cd path/to/Project32_VS2019
masm AddSub2.asm
Printing
Citrix
Engr servers
Open Source Lab

The Open Source Lab is a nonprofit organization working for the advancement of open source technologies. source. They are responsible for hosting various Open Source software repositories, a list is here. A big thank you to the OSL for providing us hosting resources so we can share this documentation with you!

https://osuosl.org/
Wifi
VPN
Term Overview (2020-09-29)

18:28 <+magical> reminder that lug is having an online meeting right now
18:29 <+magical> https://jitsi.osulug.club/WeeklyLUG pw: "git pull jitsi
weekly_lug"
18:29 <+magical> had some technical difficulties so it's just getting started
18:32 <+magical> i guess i'll try to liveblog this
18:32 <+magical> new documentation website at https://docs.osulug.club/
18:34 <+magical> lug has a matrix server now
18:34 <+magical> if you want to register through the lug's matrix
homeserver, contact nathan or drew
18:35 <+magical> otherwise you can use the official homeserver
18:35 <+magical> we're trying to move more of our communication into matrix
18:35 <+magical> so people have a place to chat outside the weekly meetings
18:36 <+magical> irc discord and matrix will be synced with each other
18:36 <+magical> the matrix/discord bridge is up and running
18:36 <+magical> irc bridge coming soon
18:37 <+magical> sounds like we're moving the LUG repos from github to gitlab
18:37 <+magical> https://gitlab.com/osu-lug
18:38 <+magical> the website and everything is there
18:50 <+magical> nathan has started an installation guide: https://
docs.osulug.club/Install/general/
18:50 <+magical> and other using linux guides
18:50 <+magical> which we could use some help filling in
18:53 <+magical> drew will be reaching out to brewster to see if we can
bring some of the ecampus students into lug
18:54 <+magical> meeting adjourned
x2go Demo (2020-10-06)

Files

Downloads/base/Dockerfile

FROM ubuntu:latest
ENV DEBIAN_FRONTEND=noninteractive
RUN apt-get update
RUN apt-get upgrade -y
RUN apt-get install -y x2goserver openssh-server
RUN useradd -m -U user
RUN mkdir /var/run/sshd
EXPOSE 22
CMD ["/usr/sbin/sshd","-D"]

Downloads/run.sh

#!/bin/bash
sh -c 'cd base;     docker build -t local/x2godocker/base .'
sh -c 'cd daliclock;    docker build -t local/x2godocker/daliclock .'
sh -c 'cd matrix;   docker build -t local/x2godocker/matrix .'
sh -c 'cd firefox;  docker build -t local/x2godocker/firefox .'
wait
docker-compose up -d --force-recreate
x2goclient --session=Matrix --hide --close-disconnect
x2goclient --session=DaliClock --hide --close-disconnect
x2goclient --session=Firefox --hide --close-disconnect
wait
x2goclient

Downloads/matrix/Dockerfile

FROM local/x2godocker/base
RUN apt-get install -y xscreensaver-data-extra
COPY ./matrix.desktop /usr/share/applications

Downloads/docker-compose.yml

version: "3.4"
services:
  matrix:
    image: local/x2godocker/matrix
    ports:
      - "2000:22"
    volumes:
      - "/home/debian/.ssh/id_rsa.pub:/home/user/.ssh/
    authorized_keys:ro"
  firefox:
Downloads/daliclock/Dockerfile

FROM local/x2godocker/base
RUN apt-get install -y xdaliclock

Downloads/firefox/Dockerfile

FROM local/x2godocker/base
RUN apt-get install -y firefox