

SDE: System Design and Engineering

Lecture – 01 (A)

Introduction to

Linux Crash Course.

From Zero to Google: Architecting the Invisible Infrastructure

by

Aatiz Ghimire

Learning Objectives

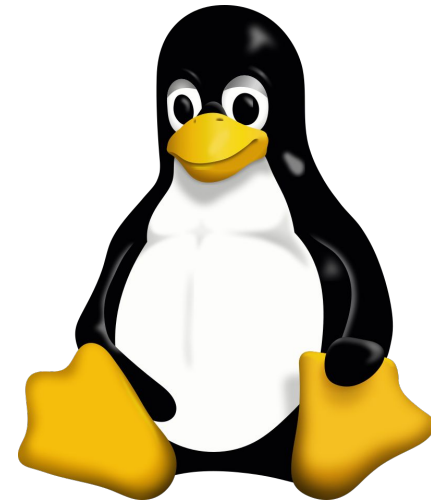
- Become acquainted with the Linux OS
- Get to know some desktops
- Explore Linux and how it handles hardware
- Learn about working with linux

Table of contents

1. Overview
2. Linux Desktop
3. Linux System
4. Compiling Software

What is a Linux

- Originally developed by Linus Torvalds in 1991
- Open Source operating system
<https://github.com/torvalds/linux>
- Available under GPL-2.0 license
- Commonly bundled as Linux Distributions
(Ubuntu, Debian, Red Hat, Arch, ...)
- Omnipresent in HPC and servers



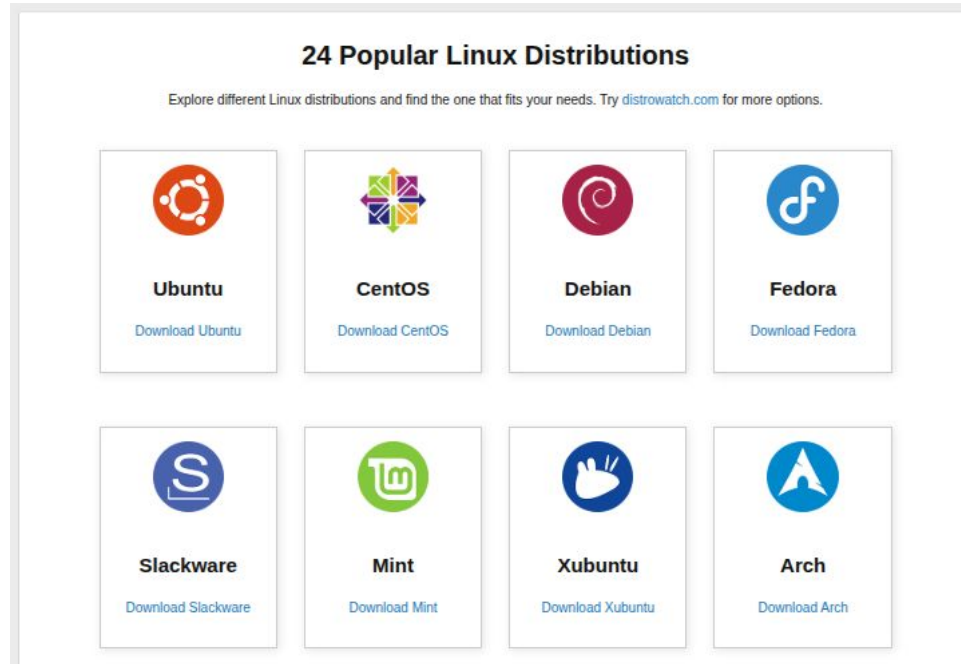
Tux the penguin, the mascot of Linux

Image Source:
<https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/800px-Tux.svg.png>

Desktop Environment (DE)

- Unlike Windows or Mac, multiple DEs supported
- Most popular: GNOME, KDE
- DE (mostly) independent of Linux distribution
- Highly customizable
 - Window management and alignment
 - Replace file explorer, login manager, ...

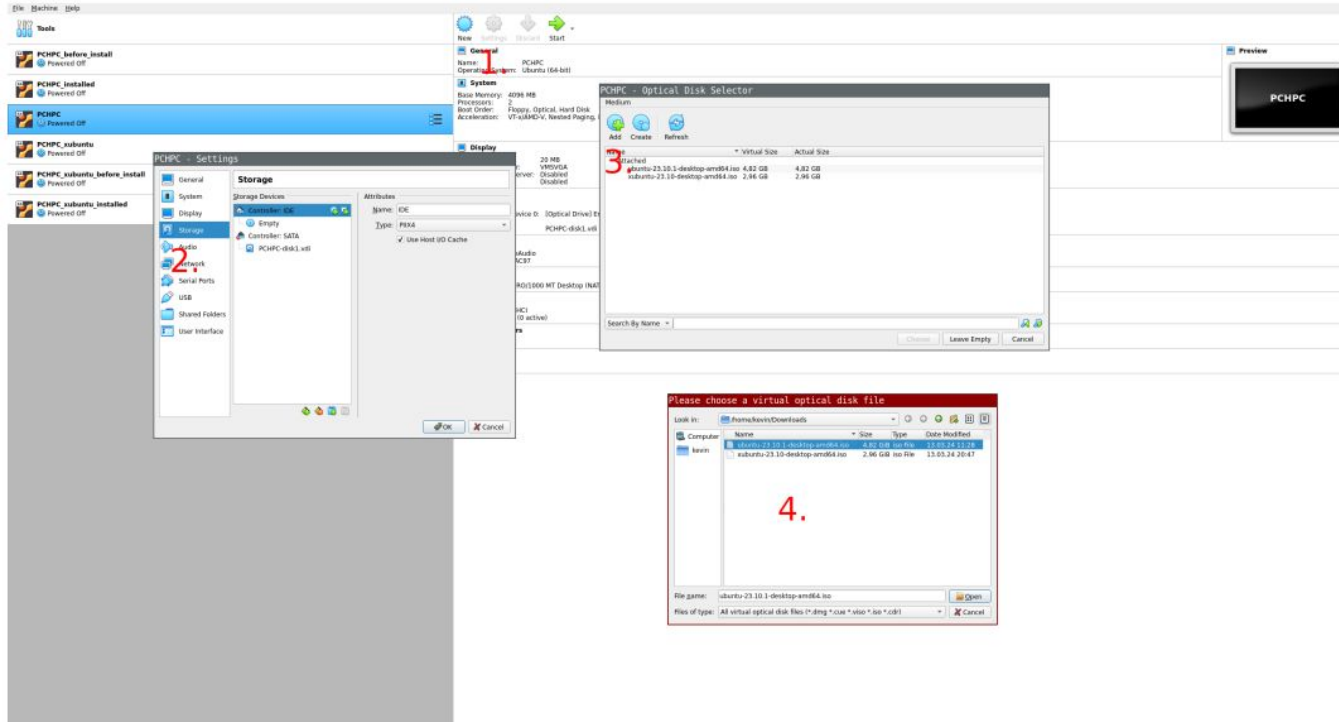
Linux Desktop - Distributions



Linux Desktop - VirtualBox

- Add new - Enter name, Linux, flavor
- Recommended is 2048MB but better is 4096MB RAM
- Create new HDD file now
 - Choose VDI
 - Dynamics allocation
 - At least 20GB of free space
- Later you may want to increase the number of Cores

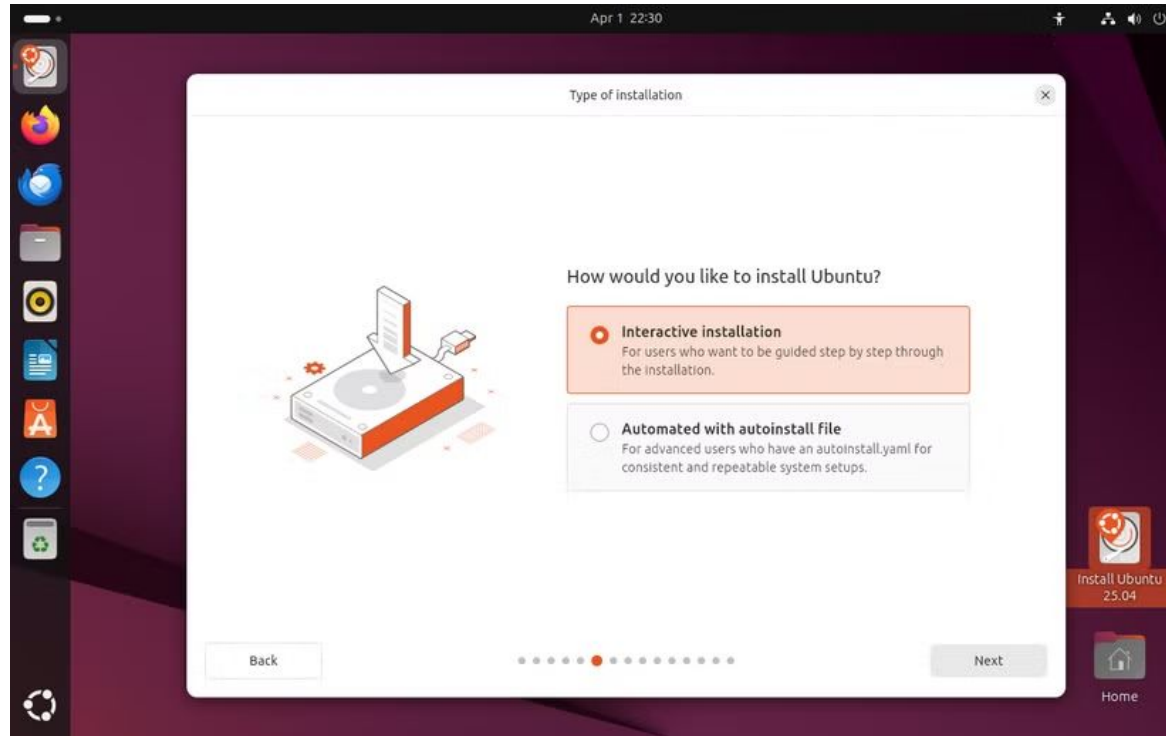
Linux Desktop - VirtualBox



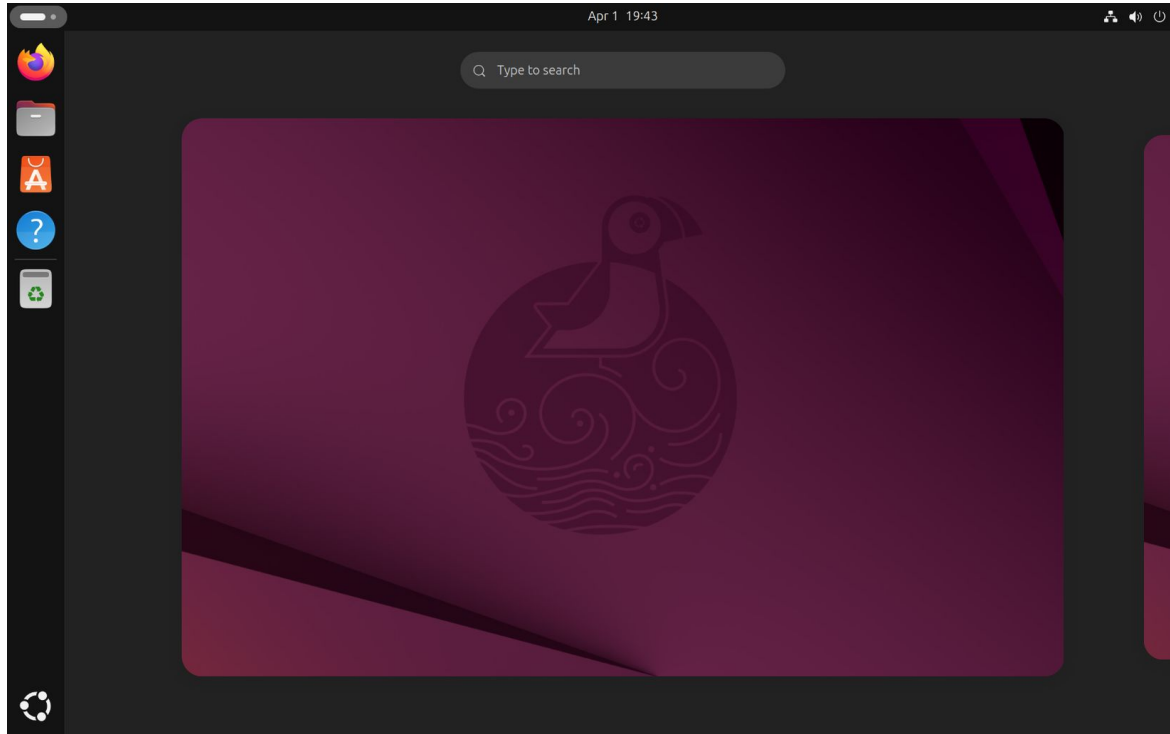
Linux Desktop - Install

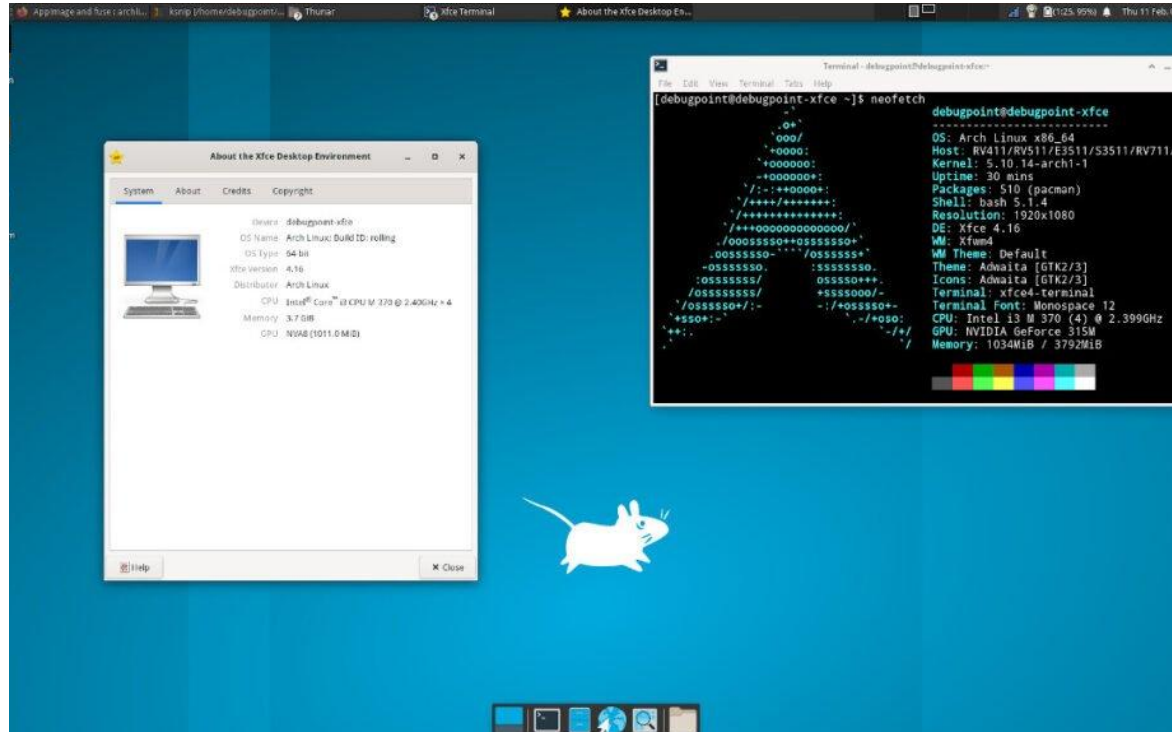


Linux Desktop - Install

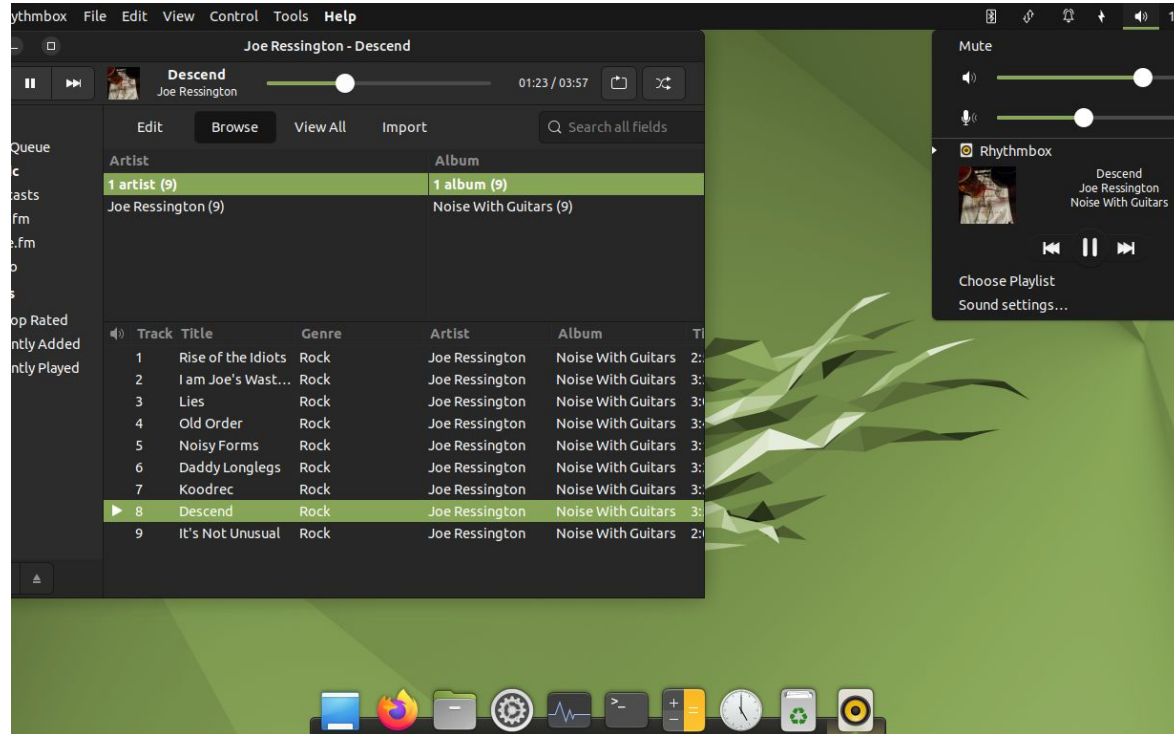


Linux Desktop - Example desktop: GNOME 48

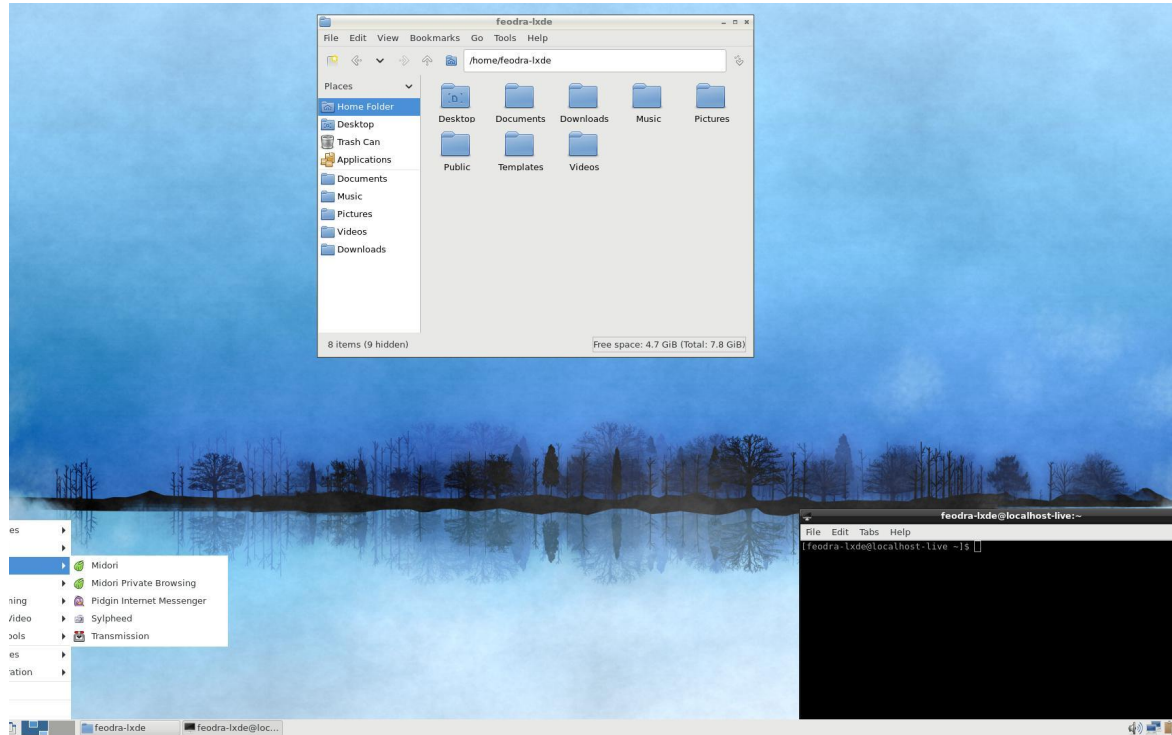




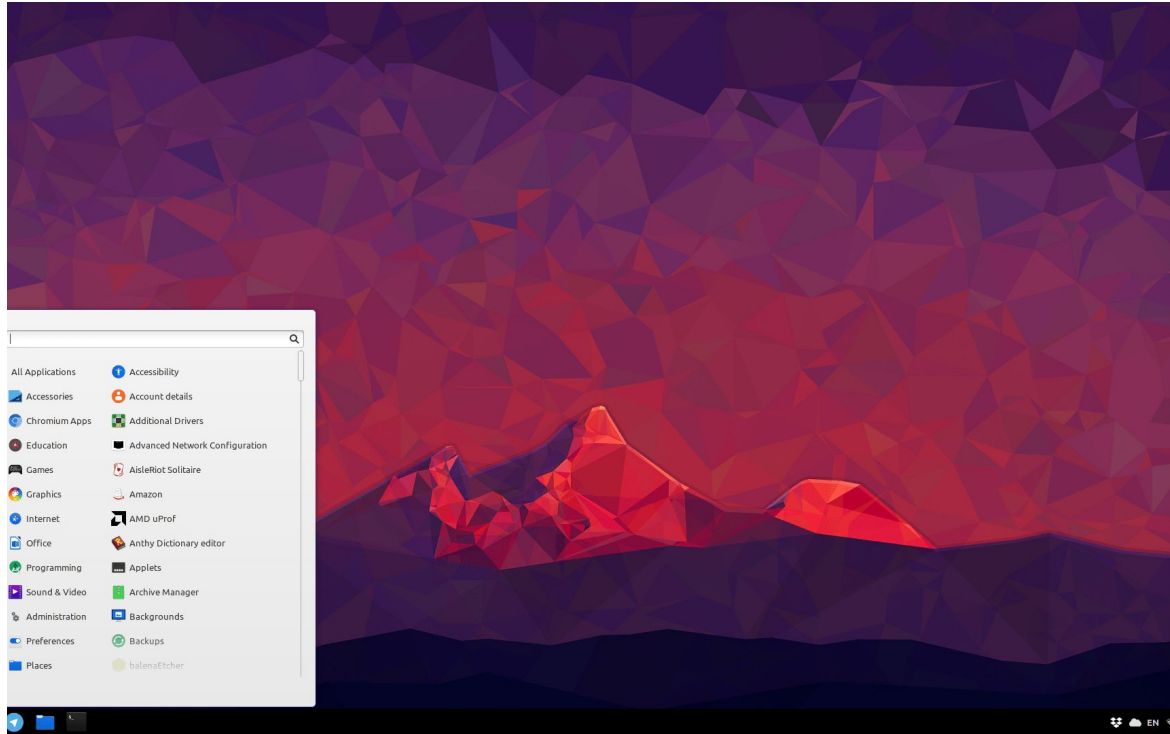
Linux Desktop - Example desktop: mate



Linux Desktop - Example desktop: lxde



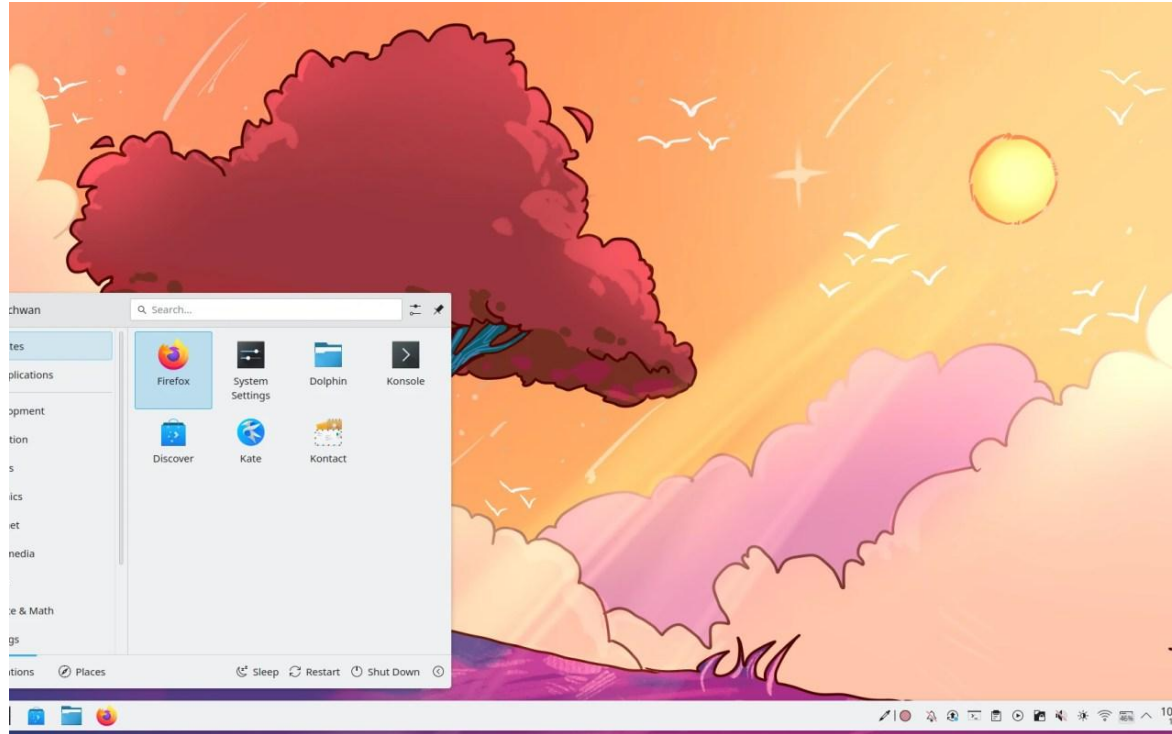
Linux Desktop - Example desktop: Cinnamon



Linux Desktop - Example desktop: budgi



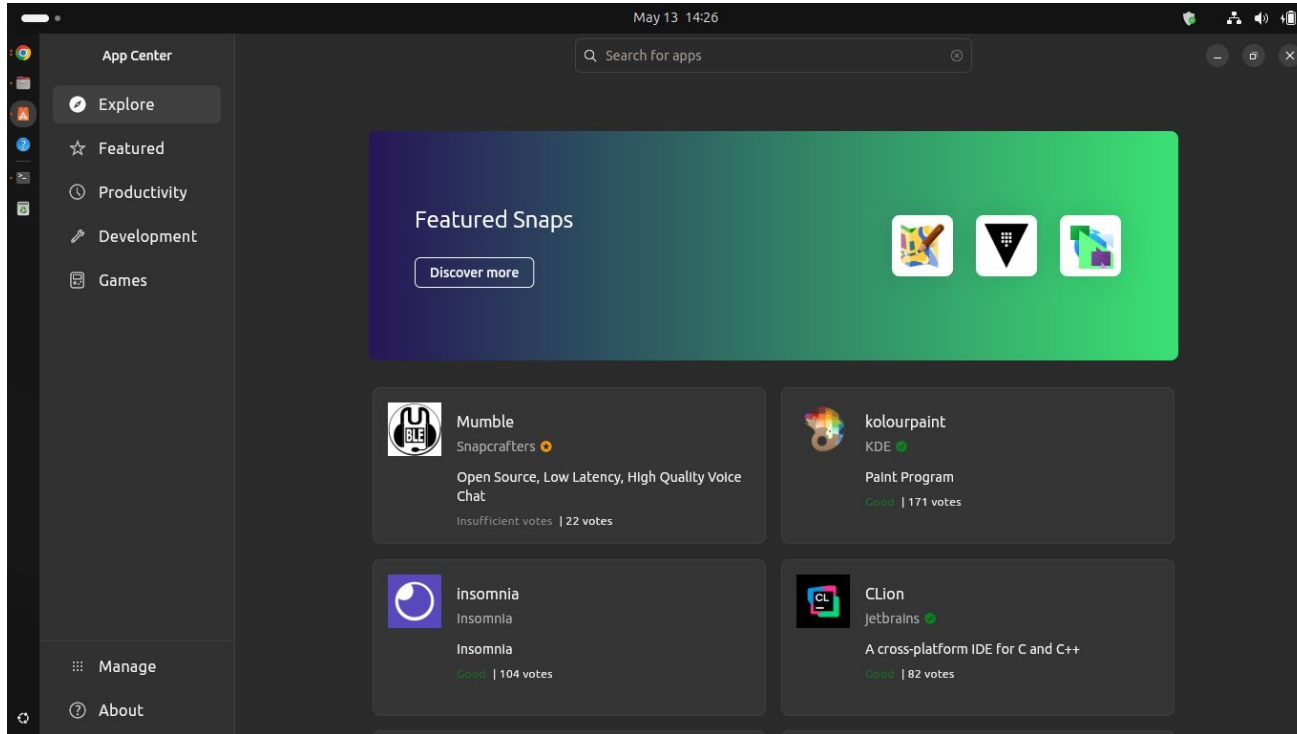
Linux Desktop - Example desktop: KDE plasma



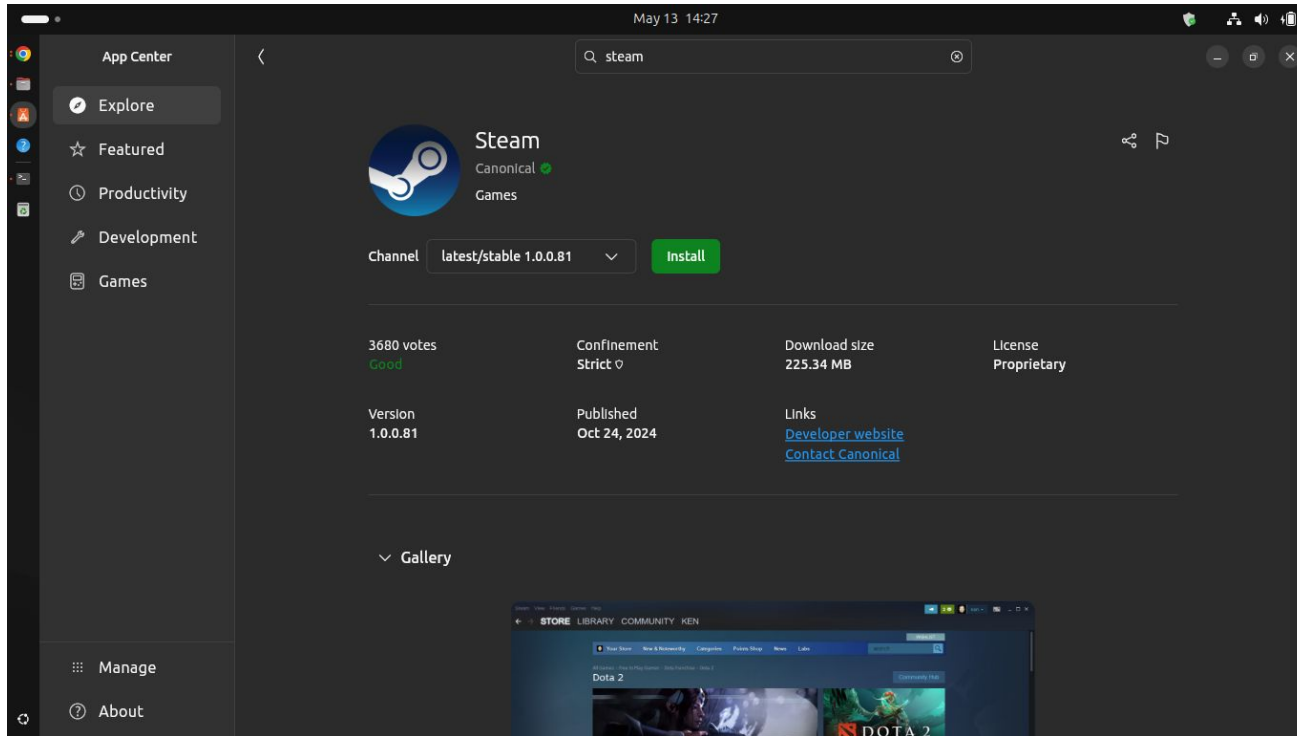
Linux Desktop - Installing software

- Using package manager
 - apt, apt-get, pacman, yum, dnf
 - snap and snap packages
 - flatpack
- Compiling from source (someone said gentoo??)
- Software manager APP

Linux Desktop - App/Software center



Linux Desktop - Want to play games



The Shell

- What is the Shell used for?
 - Your gateway to Linux Machine
 - Managing files and folders
 - Compiling from source
 - Running programs
 - Managing the system even without GUI
- Terminal emulator - improved terminals
- Remove the need for a mouse
- Working with the best editor: VIM / NANO (My preference)

File System

- Many different file system (FS) implementations exist
- Some support **Journaling**
 - FS keeps a log (journal) of file operations
 - Enables consistency in case of crash during write
- Some are better for parallel IO
- NFS for network mounting
- See currently mounted FS via
 - `df -T`

File System Types - Examples

- Ext4
 - Native Linux FS
- XFS
 - High-performance FS
- BeeGFS
 - High-performance parallel File system
- NTFS/FAT
 - Windows FS
 - USB-Sticks, . . .
- HFS+
 - Mac FS
- Tmpfs
 - Linux temporary in-memory FS

Linux File Tree

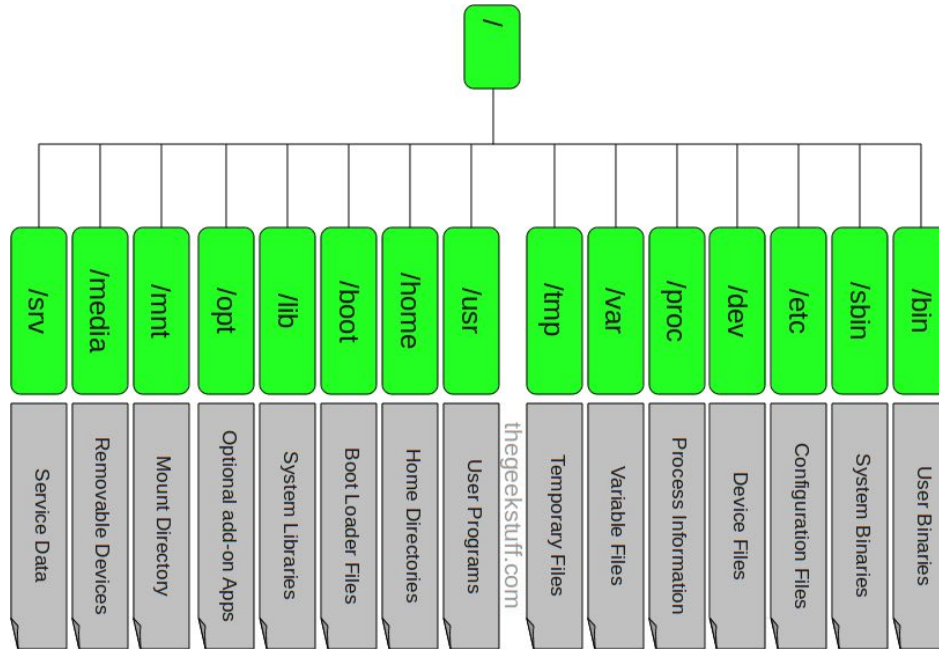


Image Source:
<https://static.thegeekstuff.com/wp-content/uploads/2010/11/filesystem-structure.png>

System Logging

- Logs commonly in `/var/log`
 - Find application and system logs here
 - Use `tail -f file` to follow changes
- `dmesg` print Kernel ring buffer
- `journalctl` for systemd logs

Linux Services and systemd

- Service management software (controversial but works)
- Interaction commands are:
 - `systemctl status`
 - `systemctl start/stop`
 - `systemctl enable/disable`
 - `systemctl --user`
- Try it out for these services
 - `systemctl status sshd`
 - `systemctl status ntpd`

Linux hardware files

- Mounting hard drives and USB Sticks
 - Find devices using `lsblk`
 - Mount a device `mount /dev/sda1 /mnt`
 - Unmount a device `umount /mnt`
- Finding the Battery
 - Could be at `/sys/class/power_supply/BAT0/`
 - Current status `charge_now`
- Finding the CPU `lscpu` / `cat /proc/cpuinfo`
 - Could be at `/sys/devices/system/cpu/cpu0/cpufreq/`
 - Current frequency `scaling_cur_freq`

Compiling own Software

- Compiling means to create an executable – or a library – from the source code
- Scientific software is often only available as source code
- Compiling on the target system often yields better performance
- Prepackaged software typically requires administrator (root) privileges ...
 - (on the Cluster `sudo` or `su` won't work)
 - but you can use Singularity containers!

Getting and Unpacking the Source Code

- Source code is usually packaged as “tarball”
 - Look for file extensions “ `tar.gz` ”, “ `tar.bz2` ”, “ `tgz` ”
 - Naming convention is often `{NAME}-{VERSION}.tar.gz`
- If the tarball is available on the web use “ `wget` ” to download
- Use “ `tar` ” to unpack the tarball
 - Use “ `tar xvzf` ” for “ `tar.gz` ”, “ `tgz` ”
 - Use “ `tar xvjf` ” for “ `tar.bz2` ”

Recipe: `wget` and `tar`

- Using `wget` and `tar` to prepare the source code

```
> mkdir $HOME/build
```

```
> cd $HOME/build
```

```
> wget <tarball URL>
```

```
> tar xvzf .tar.gz
```

```
> cd <name-version>
```

Downloading Source Code

- create a directory with mkdir
 - `python-3.14`
- switch into the directory
 - `cd python-3.14/`
- download python-3.14
 - `wget https://github.com/python/cpython/archive/refs/tags/v3.14.0b1.tar.gz`
- you do the extraction with
 - `tar xvzf cpython-3.14.0b1.tar.gz`

Compile the program

- load up the Compiler on the cluster (skip if not needed) (for HPC Users)
 - `module load gcc`
- Configure the prefix
 - `cd python-3.14`
 - `./configure CC=gcc -prefix=/home/(yourusername)/python-3.14 --enable-optimizations`
- with the prefix set you can compile the software
 - `make -j 4`
- now check the installation, and install the program
 - `make check`
 - `make install`

Compile the program

- check the installation with
 - `ls -alh /home/(yourusername)/python-3.14`
- Now we have installed Python 3.14 successfully
- Check whether the installation is there
- Check if the permissions to execute are set
- Ensure you can execute it
 - `/home/(yourusername)/python-3.14/bin/python3.14 --version`
- Optional: Add it to your path in `.bashrc`
 - `export PATH=/home/(yourusername)/python-3.14/bin/python3.14/bin:$PATH`

Summary

- You learned the beauty of Linux desktops
- You learned how to install software
- You should be able to:
 - Find your way around the file tree
 - Look for hardware and kernel parameters
 - Compile software