HPS
https://hps.vi4io.org

Kevin Lüdemann

# Linux Crash Course

The joy and pain of freedom

Overview
oo

Linux Desktop
0000000000000000

Linux System
0000000

Compiling Software
00000000

# 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

## 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
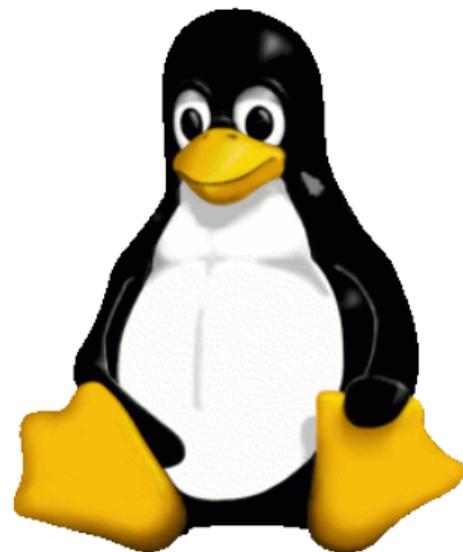


Tux - Linux mascot

Image source: https://en.wikipedia.org/wiki/Tux_(mascot)#/media/File:Tux.png

**Overview**
○●

Linux Desktop
○○○○○○○○○○○○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Debian+Ubuntu Family tree
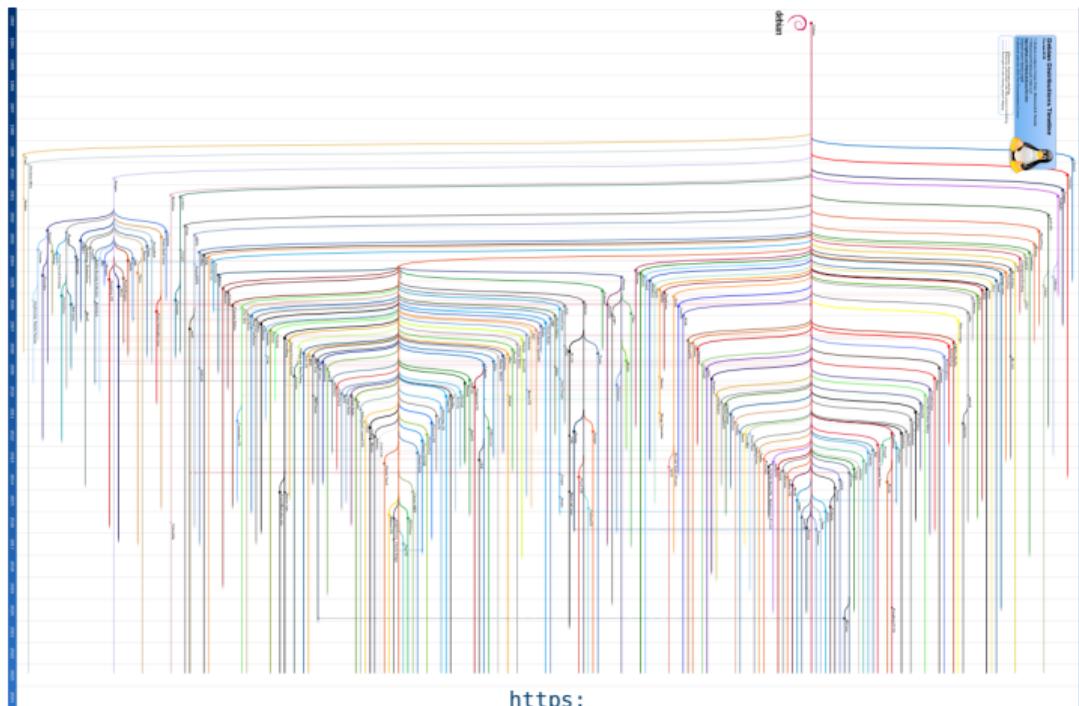


https:
//en.wikipedia.org/wiki/List_of_Linux_distributions

Overview
○○

**Linux Desktop**
●○○○○○○○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# 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, ...
- Checkout what others use: https://www.reddit.com/r/unixporn/

Overview
○○

**Linux Desktop**
○●○○○○○○○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Ubuntu flavors

# 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

Overview
○○

**Linux Desktop**
○○○●○○○○○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - VirtualBox

## Linux Desktop - Install

Overview
○○

**Linux Desktop**
○○○○○●○○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

## Linux Desktop - Install

Overview
○○

**Linux Desktop**
○○○○○○●○○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Example desktop: GNOME 3
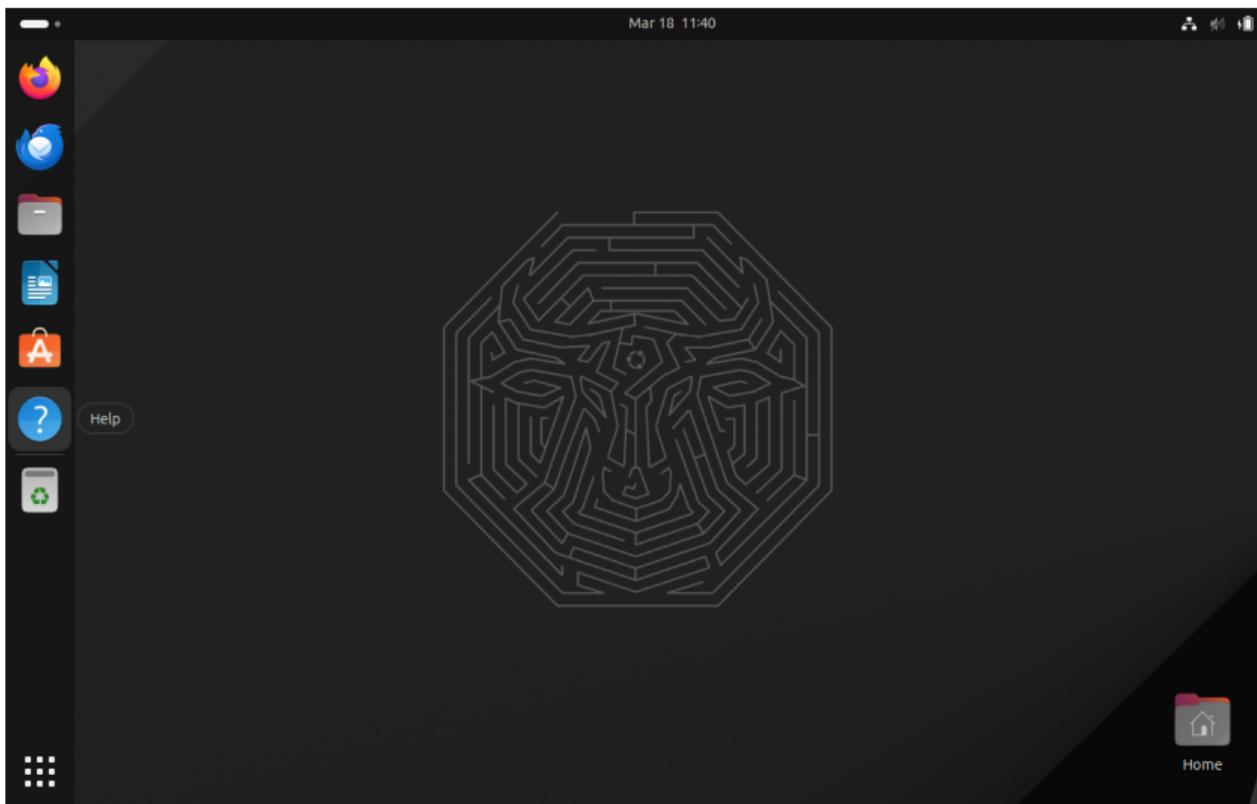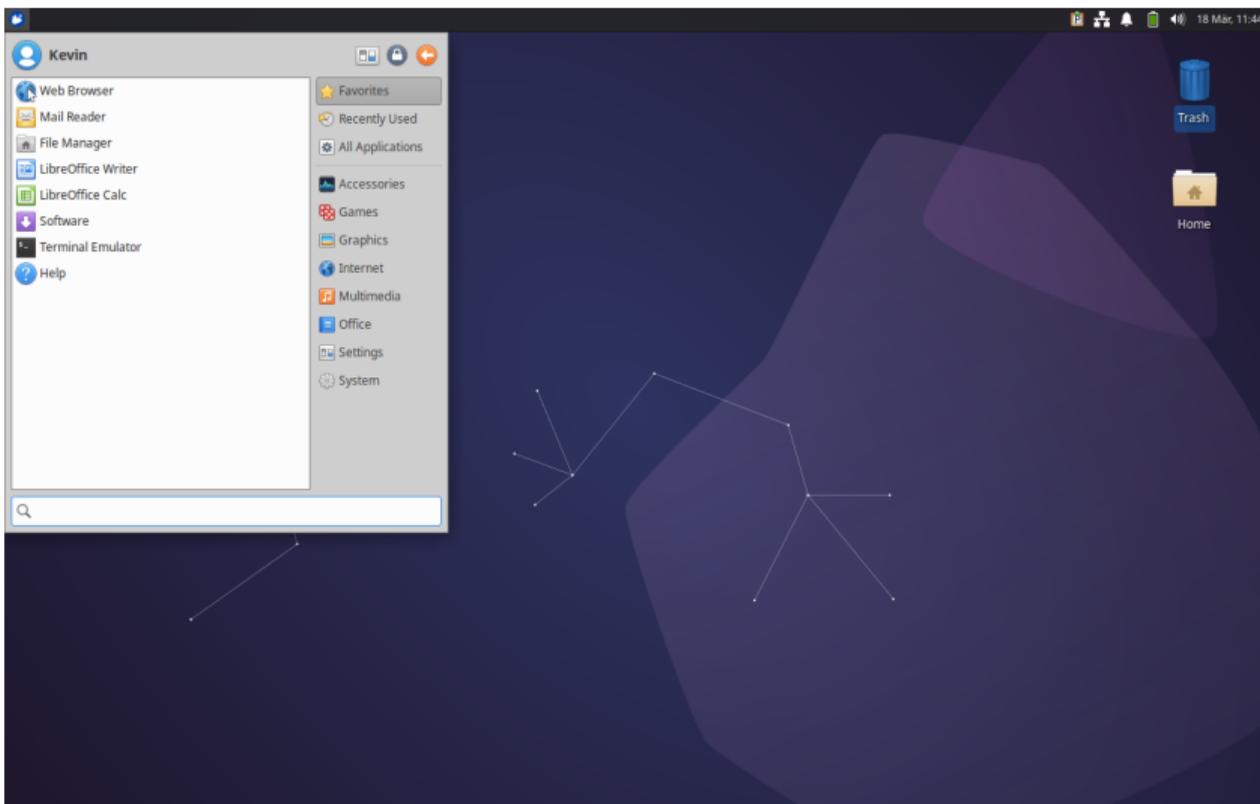
Overview
○○

**Linux Desktop**
○○○○○○○●○○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Example desktop: xfce

Overview
○○

**Linux Desktop**
○○○○○○○○●○○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Example desktop: mate

Overview
○○

**Linux Desktop**
○○○○○○○○○○●○○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Example desktop: lxde

Overview
○○

**Linux Desktop**
○○○○○○○○○○●○○○○○○

Linux System
○○○○○○○
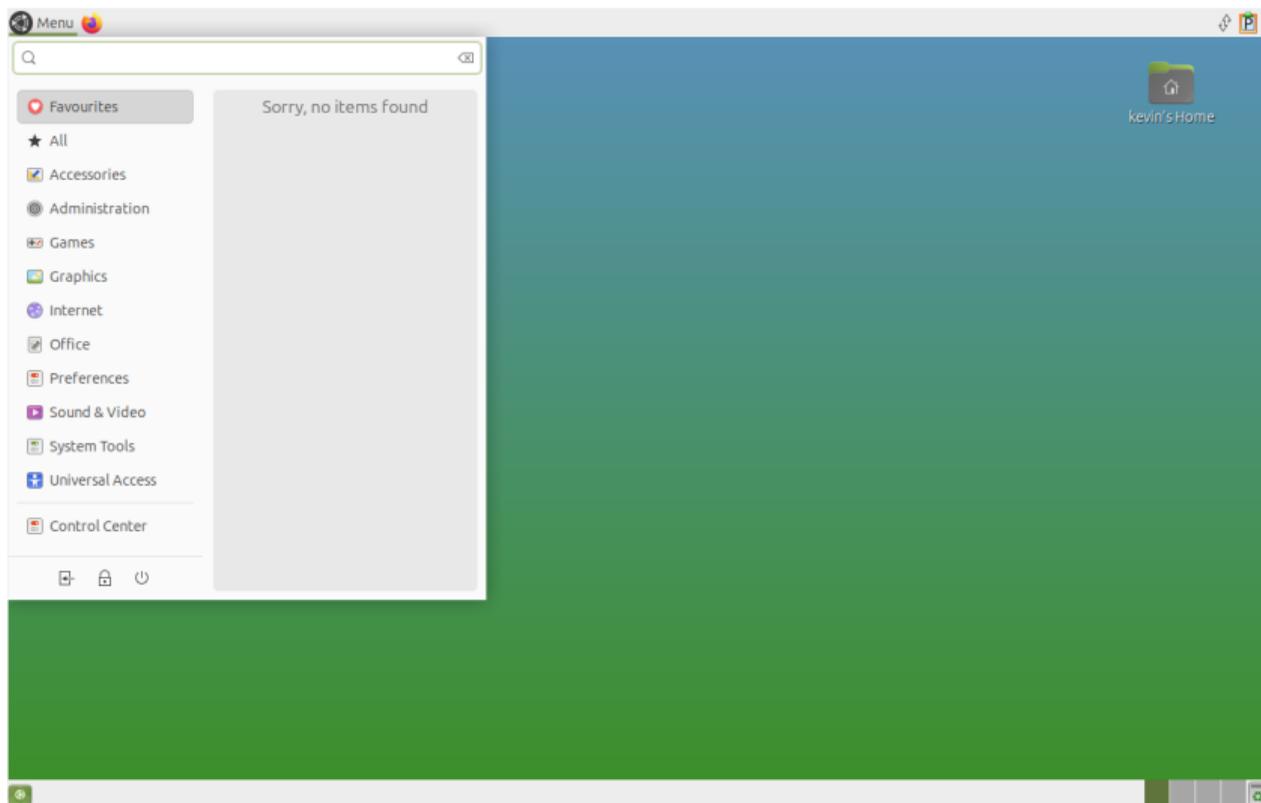
Compiling Software
○○○○○○○○○
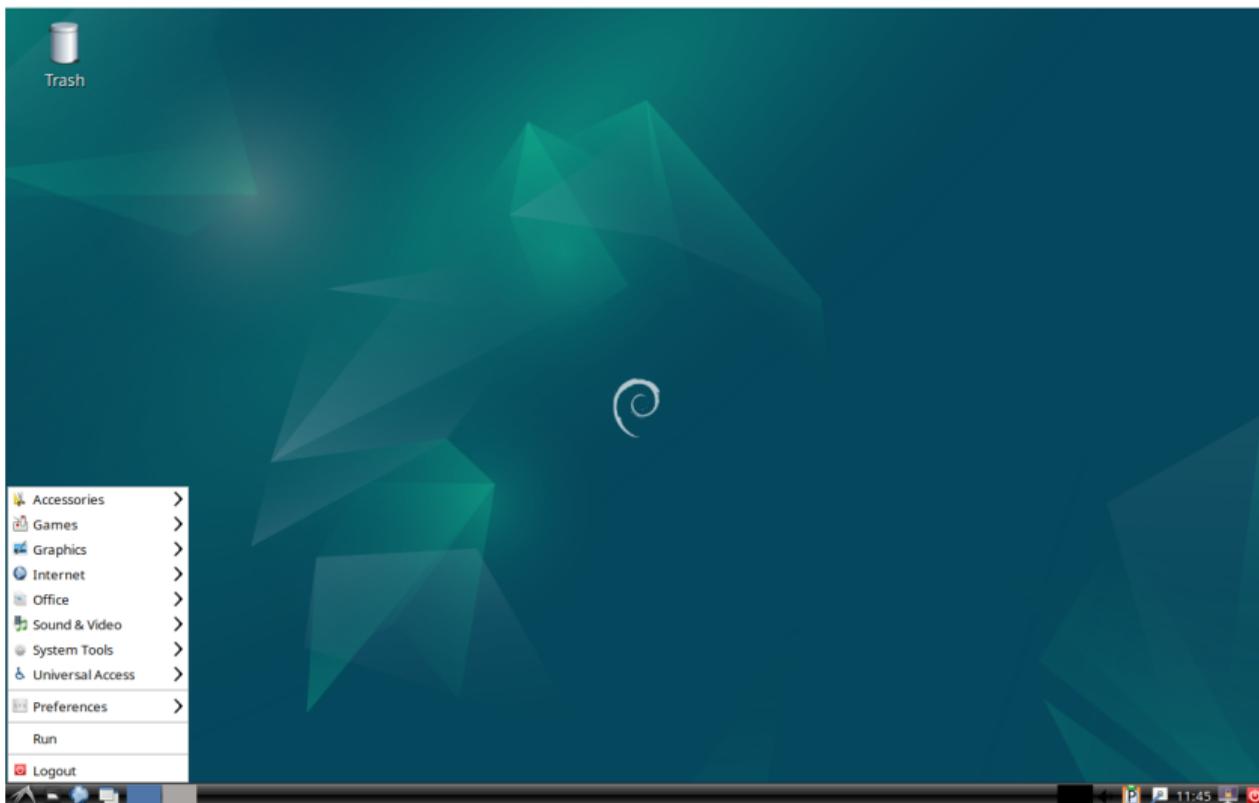
# Linux Desktop - Example desktop: Cinnamon

Overview
○○

**Linux Desktop**
○○○○○○○○○○○●○○○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Example desktop: budgi

Overview
○○

**Linux Desktop**
○○○○○○○○○○○○○●○○○○

Linux System
○○○○○○○
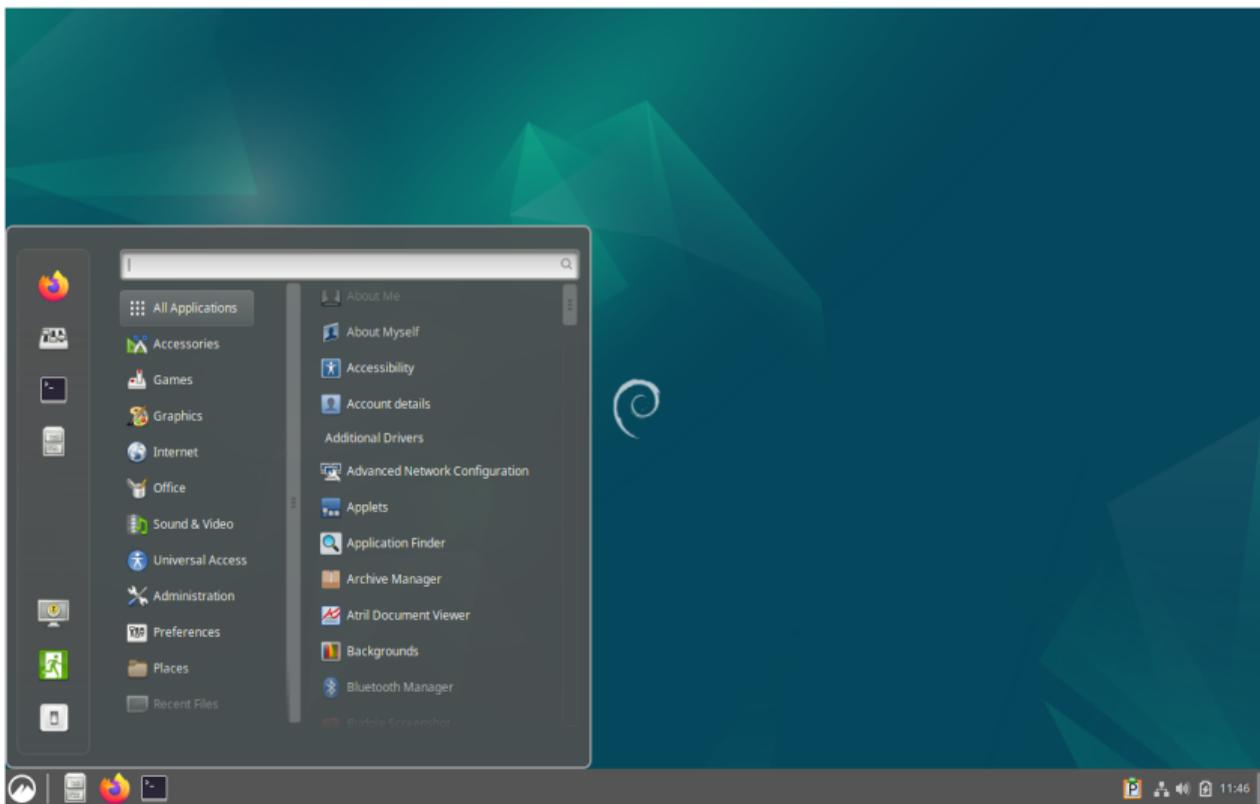
Compiling Software
○○○○○○○○○
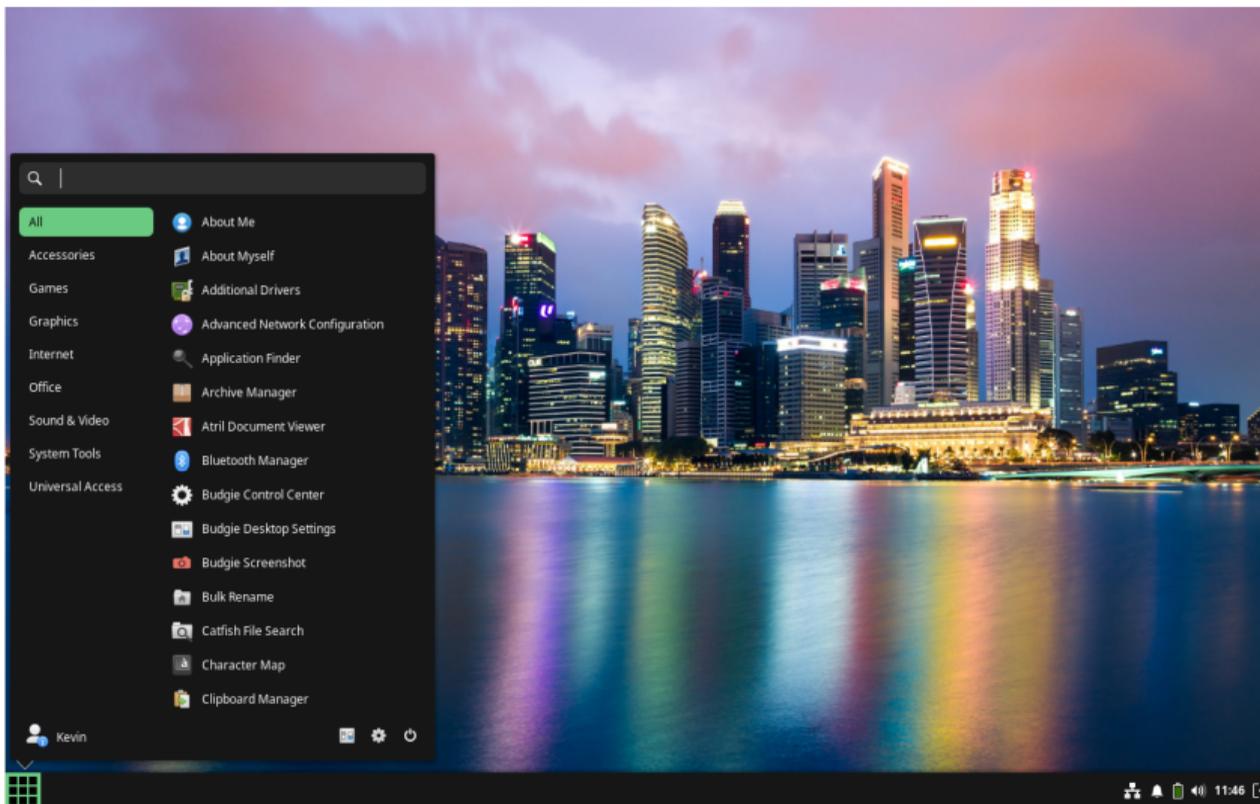
# Linux Desktop - Example desktop: KDE plasma

Overview
○○

**Linux Desktop**
○○○○○○○○○○○○○○●○○○

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○
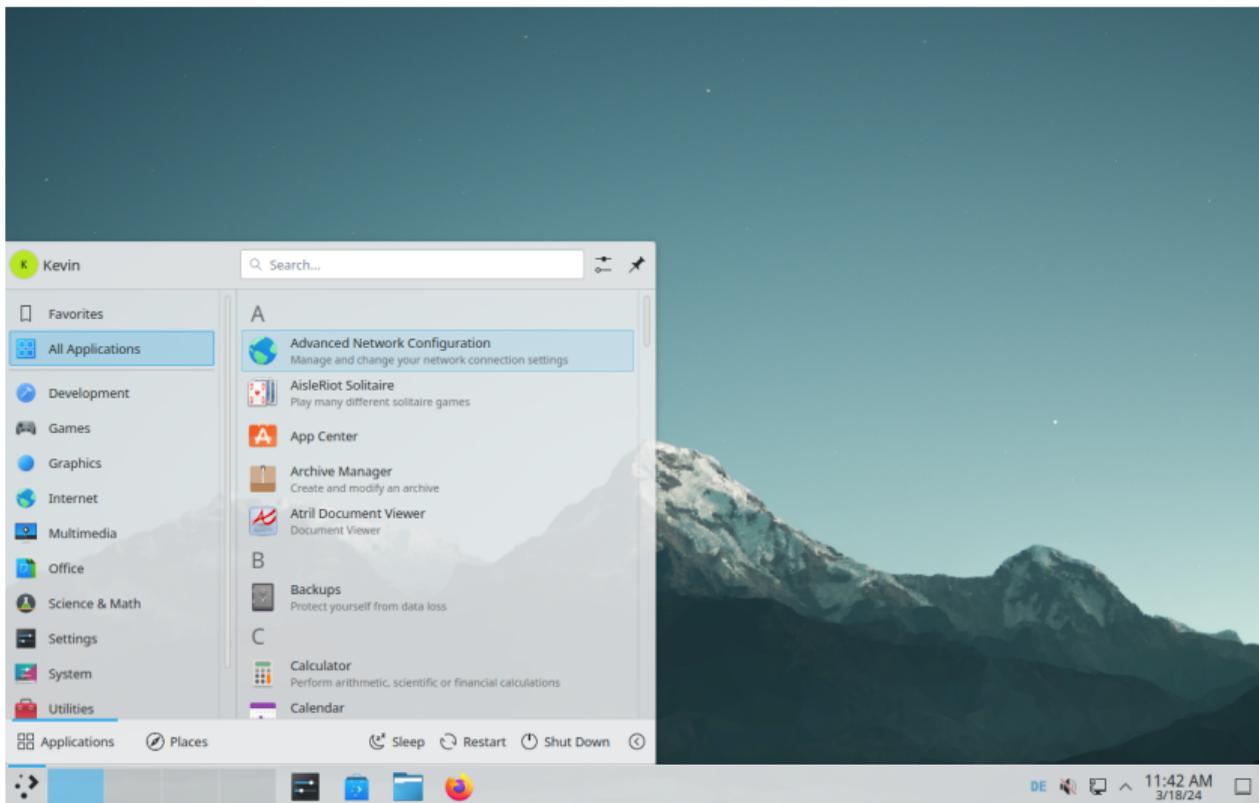
# Linux Desktop - Example desktop: i3 WM
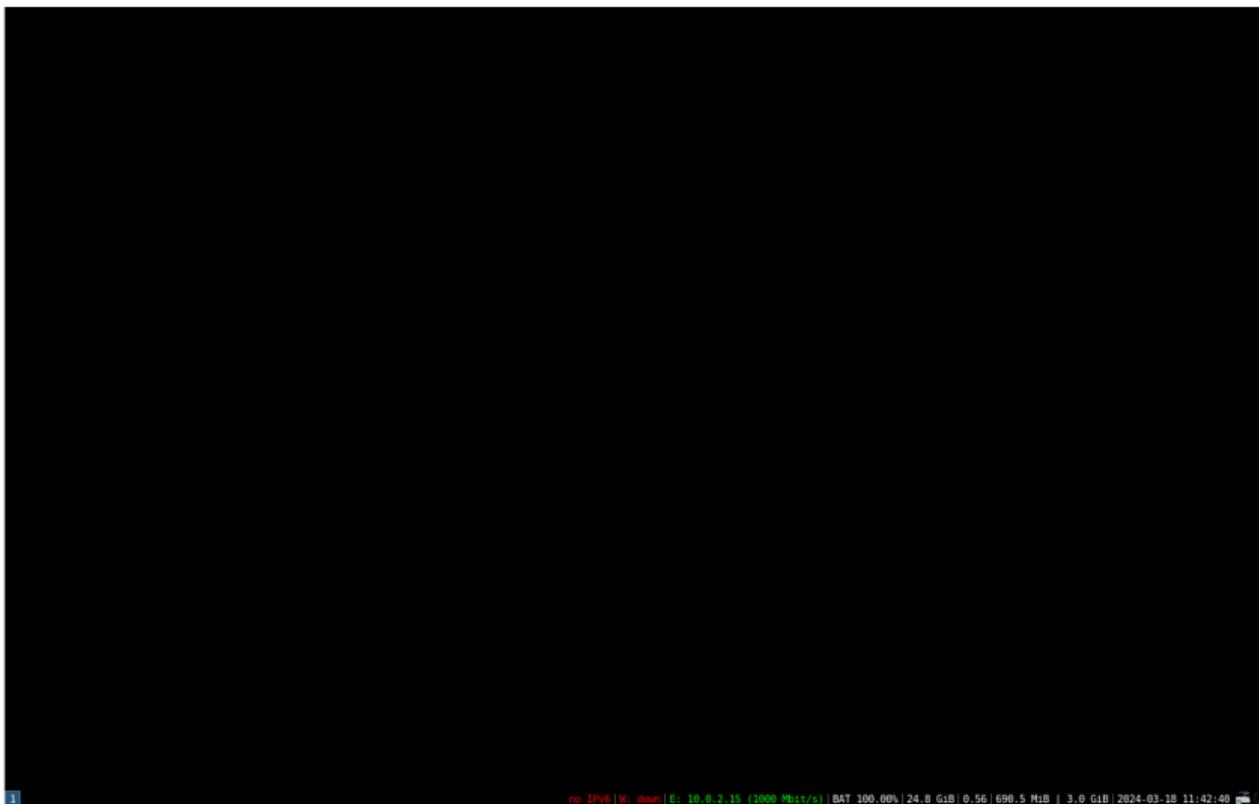
# Linux Desktop - Installing software

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

# Linux Desktop - App/Software center

Overview
○○

**Linux Desktop**
○○○○○○○○○○○○○○○○●

Linux System
○○○○○○○

Compiling Software
○○○○○○○○○

# Linux Desktop - Want to play games

Overview
○○

Linux Desktop
○○○○○○○○○○○○○○○○

**Linux System**
●○○○○○○

Compiling Software
○○○○○○○○

# The Shell

- ■ What is the Shell used for?
  - ► Your gateway to HPC power
  - ► 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

# 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`

Overview
○○

Linux Desktop
○○○○○○○○○○○○○○○○○○

**Linux System**
○○○●○○○○

Compiling Software
○○○○○○○○○

# 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
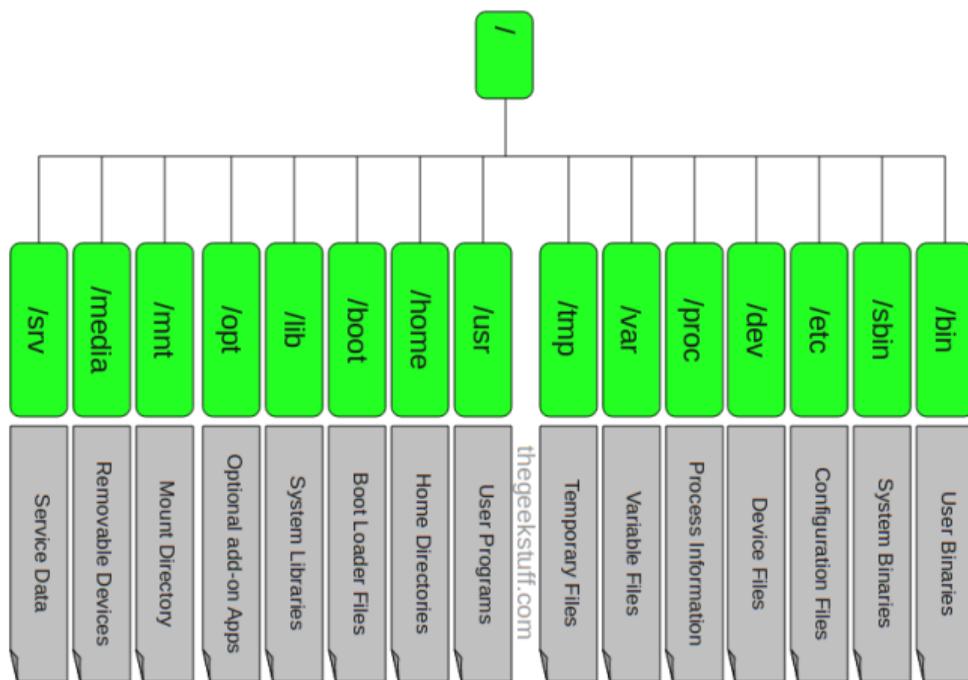
# System Logging

- 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`

Overview
00

Linux Desktop
0000000000000000

**Linux System**
000000●

Compiling Software
00000000

# 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`

Overview
oo

Linux Desktop
oooooooooooooooooo

Linux System
ooooooo

**Compiling Software**
●ooooooo

## 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!

Overview
OO

Linux Desktop
OOOOOOOOOOOOOOOOO

Linux System
OOOOOOO

**Compiling Software**
O●OOOOOO

# 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` "

Overview
○○

Linux Desktop
○○○○○○○○○○○○○○○○

Linux System
○○○○○○○

**Compiling Software**
○○●○○○○○

# Recipe: `wget` and `tar`

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

```
> mkdir $HOME/build
> cd $HOME/build
> wget <tarball URL>
> tar xvzf <name-version>.tar.gz
> cd <name-version>
```

Overview
OO

Linux Desktop
○○○○○○○○○○○○○○○○○

Linux System
○○○○○○○

**Compiling Software**
○○○●○○○○

# Reminder: Connecting with SSH

- Place the SSH key you received per mail in your user folder
- **NN** is the number in the key file name
- In PowerShell or Terminal type the following command

  ```
  ssh -i hpctrainingNN hpctrainingNN@login-mdc.hpc.gwdg.de
   -o ProxyCommand='ssh -W %h:%p hpctrainingNN@login.gwdg.de
   -i hpctrainingNN'
  ```

- Confirm the connection and enter the SSH keys passphrase **twice**
- The passphrase is in the email you received
- If you are already in the GÖNET, you only need the first line

Overview
○○

Linux Desktop
○○○○○○○○○○○○○○○○○

Linux System
○○○○○○○

**Compiling Software**
○○○○●○○○

# Downloading Sourcecode

- ■ create a directory with mkdir
  - ► `apps/install/fftw/`
- ■ switch into the directory
  - ► `cd apps/install/fftw/`
- ■ download fftw
  - ► `wget http://www.fftw.org/fftw-3.3.10.tar.gz`
- ■ you do the extraction with
  - ► `tar xvzf fftw-3.3.10.tar.gz`

Overview
OO

Linux Desktop
000000000000000000

Linux System
0000000

**Compiling Software**
000000●00

# Compile the program

- load up the Compiler on the cluster
  - ► `module load intel-oneapi-compilers`
- Configure the prefix
  - ► `cd fftw-3.3.10`
  - ►
    `./configure CC=icc -prefix=/usr/users/(yourusername)/apps/fftw-3.3.10`
- with the prefix set you can compile the software
  - ► `make -j 4`
- now check the installation, and install the program
  - ► `make check`
  - ► `make install`

Overview
oo

Linux Desktop
oooooooooooooooo

Linux System
ooooooo

**Compiling Software**
oooooooo●o

## Compile the program

- check the installation with
    - ▶ `ls -alh  /apps/fftw-3.3.10/`
- Now we have installed fftw successfully
- Check whether the installation is there
- Check if the permissions to execute are set

Overview
OO

Linux Desktop
OOOOOOOOOOOOOOOOO

Linux System
OOOOOOO

**Compiling Software**
OOOOOOO●

## 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