HPS
https://hps.vi4io.org

Ruben Kellner

Linux Crash Course

What even is a Linux

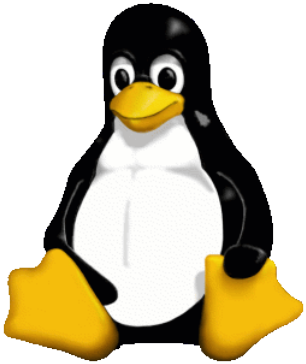
Learning Objectives

- Become acquainted with the Linux OS
- Receive an overview of the Linux history
- Understand the range of usages of Linux
- Learn about Linux system concepts

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 High-Performance Computing
- Most commonly used on servers
  also available for desktops



Tux - Linux mascot

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

History of Linux

- *1960s* IBM develops OS for their Hardware
- *1970s* Unix is developed and becomes popular in academics
- *1980s*
  - First Disk Operating Systems (DOS), home computers start to gain traction
  - First Operating Systems with a GUI pop up
  - *1987* Andrew S. Tanenbaum writes Minix as a free open source Unix for educational purposes
- *1990s*
  - *1991* Linus Torvalds releases the first Linux based on Minix (free of Minix code)
  - Linux was supposed to be named Freax, and was only named Linux, after an Administrator uploaded it under this name
  - The first Linux kernel had a size of 65KB
  - It is released under the GNU Public License (GPL)

Linux Today

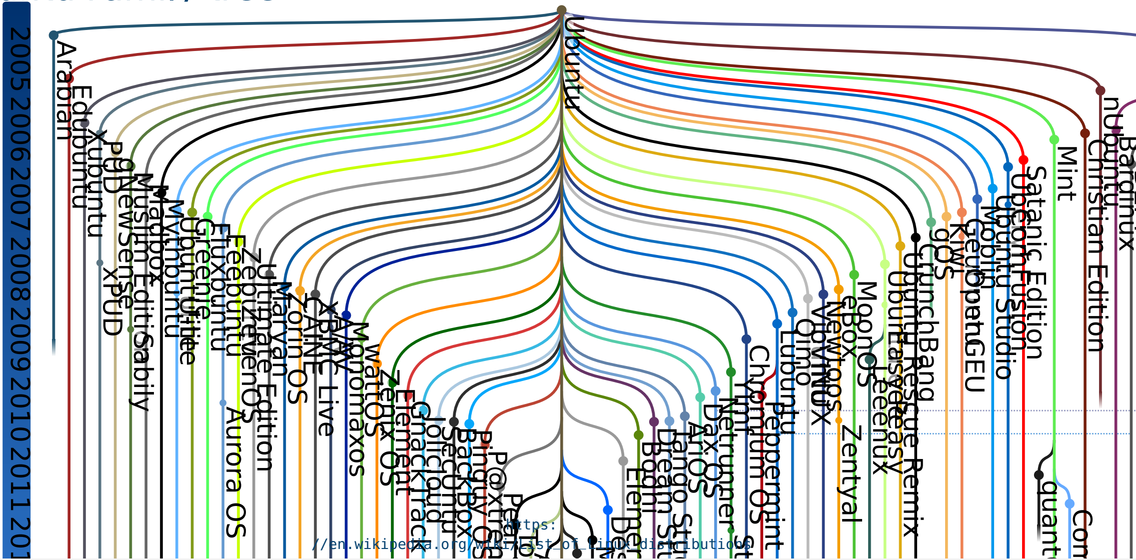
- There are over 1000 different Linux distributions
- Over 300 distributions are actively maintained
- Over 30 million lines of code and over 1 million commits
- Provides an LTS and stable version
- Linus Torvalds is still project lead

Linux OS market share - June 2022

- User Desktop 2.9%
- Smartphones and similar 71%
- HPC and Supercomputing ~100%
  - 48% Linux
  - 16.6% CentOS
  - 9.6% Cray Linux
  - 3.4% SUSE Linux Enterprise Server
  - 2% TOSS
  - 9.6% Other

https://en.wikipedia.org/wiki/Usage_share_of_operating_systemshttps://www.statista.com/statistics/565080/distribution-of-leading-supercomputers-worldwide-by-operating-system-family/

Ubuntu Family tree



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

Linux License - GPL-2.0

- GNU General Public License (GPL) - Copyleft
- Anybody may redistribute and sell it
- Source must be public
- Any derived product also under same license
  - A company may take and modify Linux source
  - They must make the modified source available
- GPL also called “Virus” license

https://www.oreilly.com/library/view/linux-device-drivers/0596000081/ch01s06.html

Linux Versions

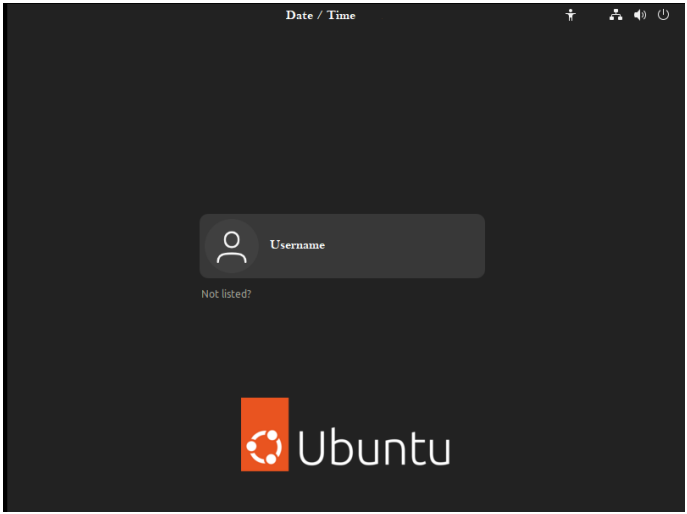
- X.YY.ZZZ (e.g., 6.1.23
  - X.YY signals major version
  - ZZZ is bug fix release
- Increment of X has no special meaning
  - Linus prefers YY to not get *“too big”*
- Current LTS release is 6.1
- Find kernel version with `uname -r`
  - Distributions may append version number for their modifications

https://www.kernel.org/category/releases.html

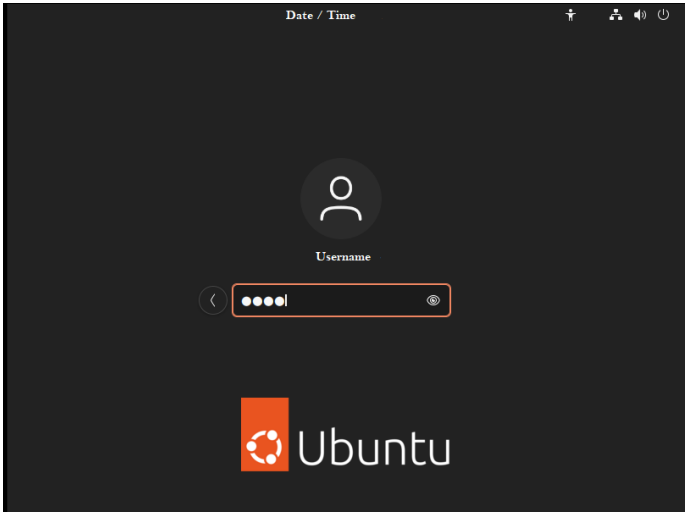
Desktop Environment (DE)

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

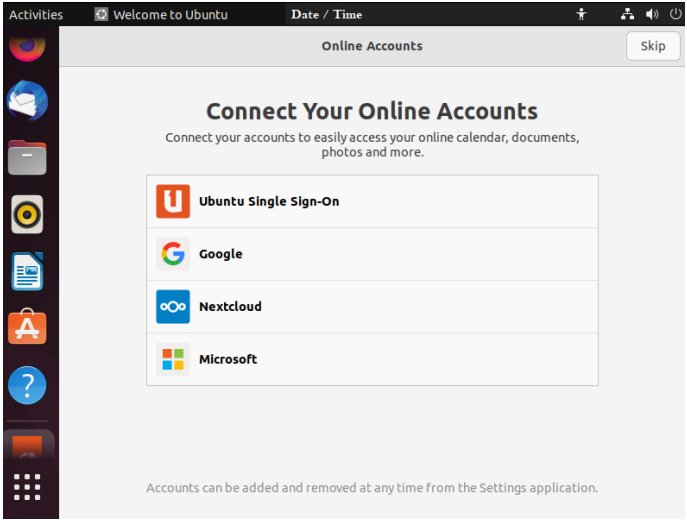
Linux Desktop - Ubuntu Gnome



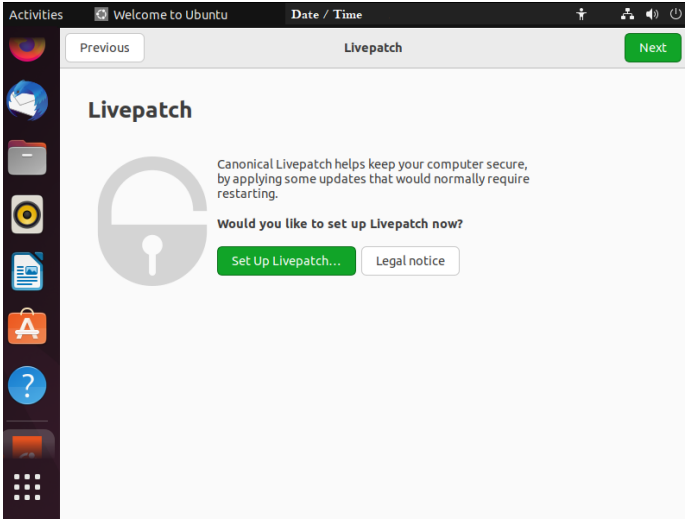
Linux Desktop - Ubuntu Gnome



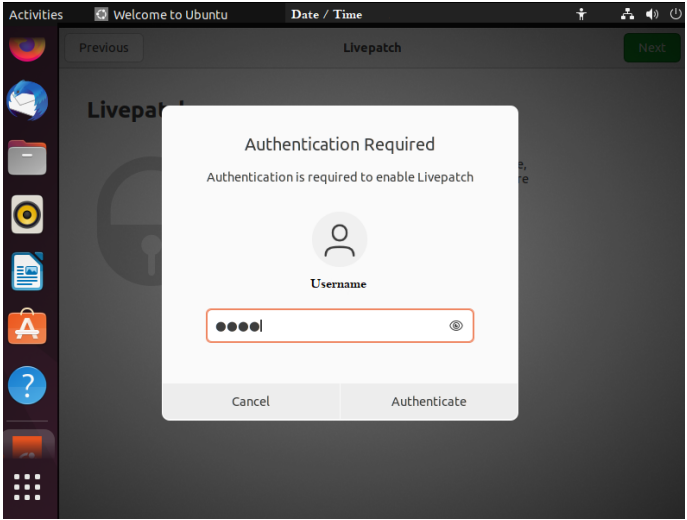
Linux Desktop - Ubuntu Gnome



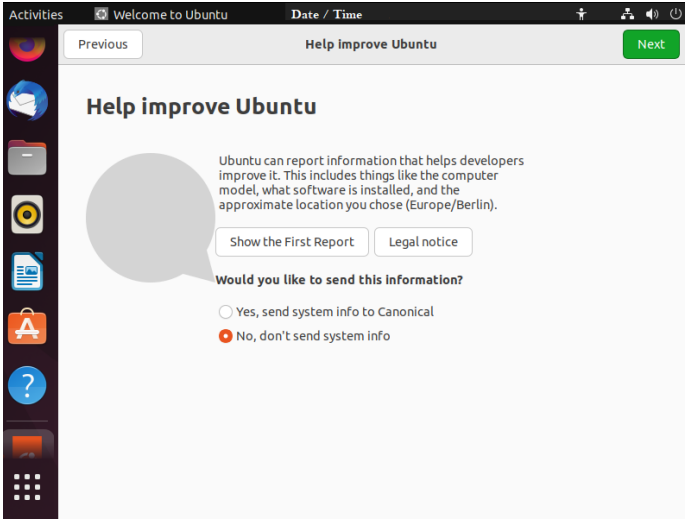
Linux Desktop - Ubuntu Gnome



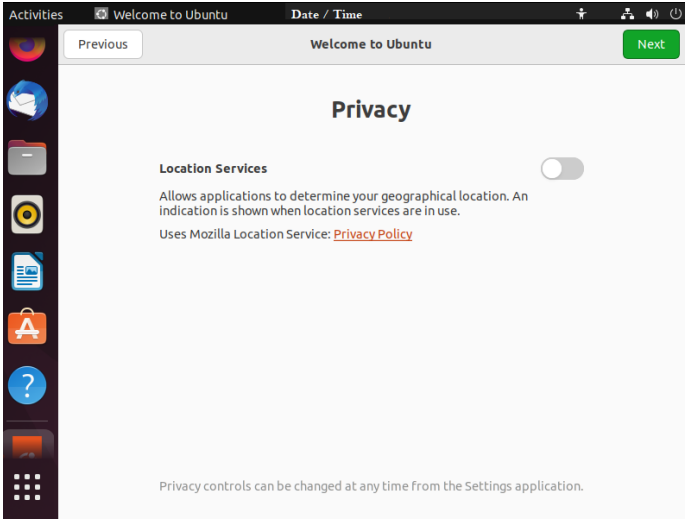
Linux Desktop - Ubuntu Gnome



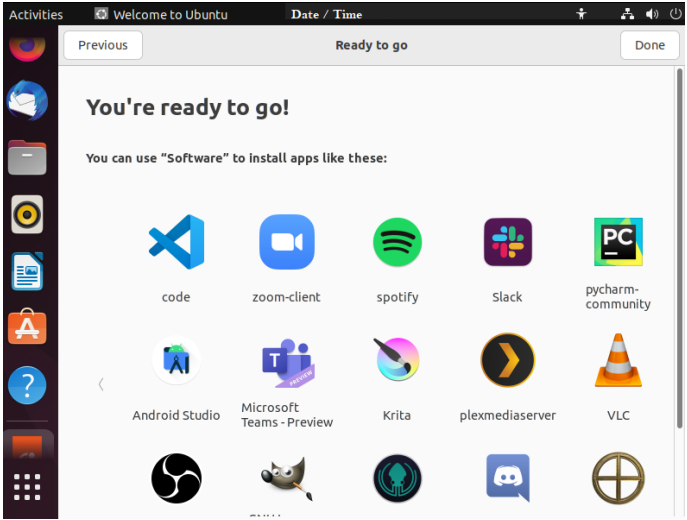
Linux Desktop - Ubuntu Gnome



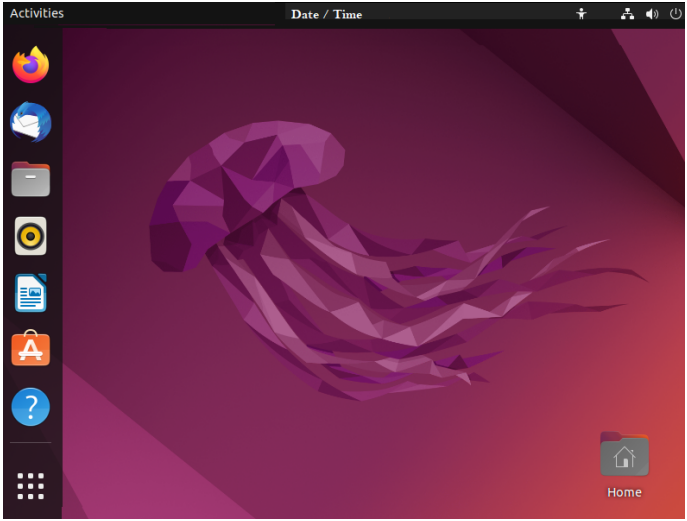
Linux Desktop - Ubuntu Gnome



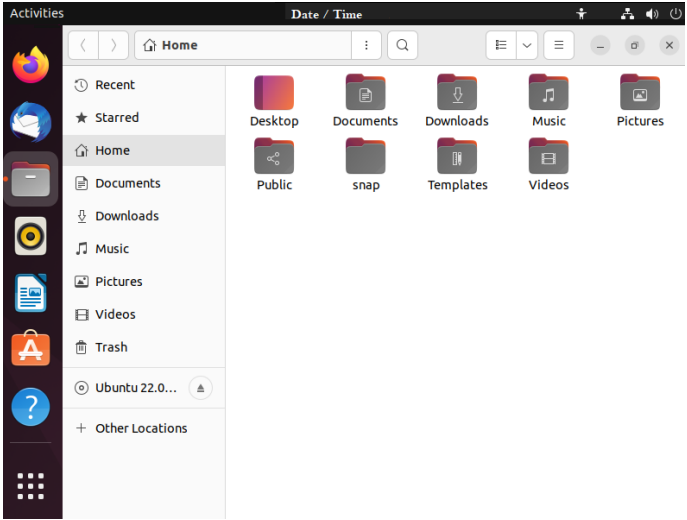
Linux Desktop - Ubuntu Gnome



Linux Desktop - Ubuntu Gnome



Linux Desktop - Ubuntu Gnome



The Shell

- *What is the Shell?*
  - a command line interface
  - no GUI
  - you type in commands and parameters
  - steep learning curve
  - easier to implement new functions compared to a GUI
  - fast as no GUI components need to be calculated

Linux File System Tree

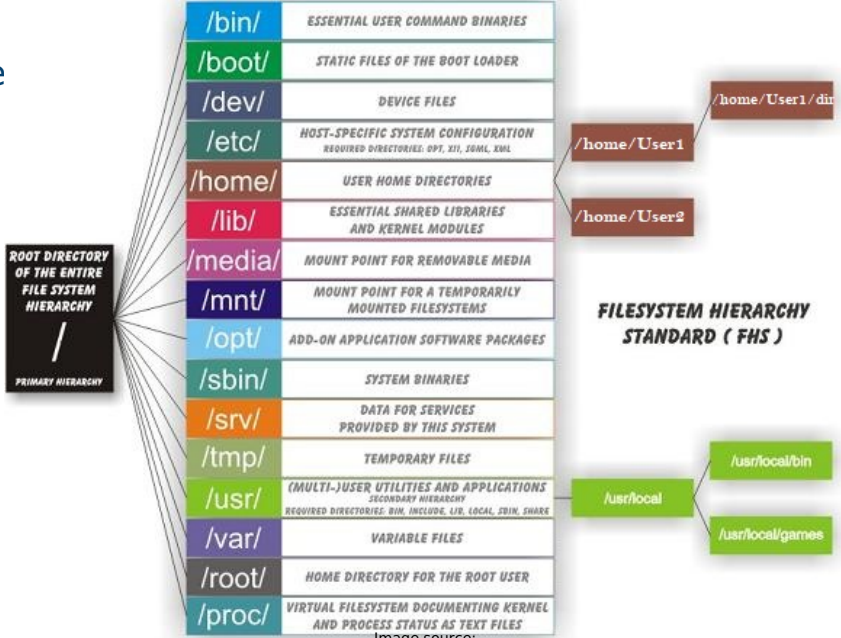


Image source:

File System Types

- 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
- See currently mounted FS via
  - `df -T`

File System Types - Examples

- **ext4**
  - Native Linux FS
- **XFS**
  - High-performance FS
- **NTFS/FAT**
  - Windows FS
- **HFS+**
  - Mac FS
- **tmpfs**
  - Linux temporary in-memory FS

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

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 <name-version>.tar.gz
> cd <name-version>
```

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

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`

Compile the program

- load up the Compiler on the cluster
  - `module load intel-oneapi-compilers/2022.0.1`
- 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 10`
- now check the installation, and install the program
  - `make check`
  - `make install`

Compile the program

- check the installation with
  - `ls -alh  /apps/fftw-3.3.10/`

now we have installed fftw successfully, you can check whether the installation is there by navigating into the folder we chose in the prefix and checking for the files

- `/usr/users/(yourusername)/apps/fftw-3.3.10`

Last Frame