Introduction to Linux
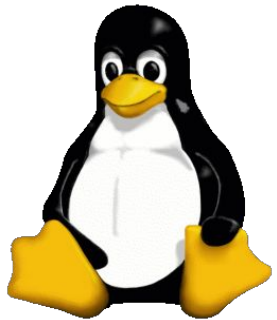
Justin W. Flory

UNIX 101

- To understand Linux, you need to understand what UNIX is, 30 years ago...
- **1969**: Team of Bell Labs developers begin working on solution to address software problems with regards to compatibility of older systems
  - They wanted a new operating system that was **simple and elegant**, written in the **C programming language** (instead of Assembly), and be able to **recycle code**
  - The developer team named the project: UNIX
- UNIX set forth a series of standards that would be followed for operating systems for years to come (even to today)

POSIX Standards

- Set of standards derived by the Institute of Electrical and Electronics Engineers (IEEE) for maintaining compatibility among operating systems
  - Influenced by UNIX
- Defines the API, along with command line shells and utility interfaces, for software compatibility with variants of Unix and other operating systems
- Fundamental for design of future operating systems, kernels, and other applications to all work well with one another

College student with too much time

- Introducing **Linus Torvalds**
    - **1991**: 21 year old student at the University of Helsinki (Finland)
    - UNIX was big and proprietary - as a student, it was difficult to get a copy and gain experience with it without having to pay large amounts of money
    - **August 25, 1991**: Torvalds announces a (hobby) project to develop a free and open source operating system
        - Follows POSIX standards, which UNIX users loved! Easy to port applications!
- Spawns a hobby project that *accidentally* becomes a big project that *accidentally* triggers a global open source movement that *accidentally* sets Torvalds' net worth to $150 million
- But what is it exactly?

Linux is just a kernel, seriously.

- **Kernel**: Central component of most computer operating systems; *bridge* between applications and the actual data processing done at hardware level
  - Some responsibilities of kernel include: managing system's resources and communicating between hardware and software components
  - Does not handle or deal with things like the graphical user interface or defining the user experience as an operating system
- Or in other words… Linux is just a kernel. Seriously!
  - Linux is the base or foundation for creating a full operating system

Where's the "operating" in my operating system?

- If Linux is just a kernel, how do I use it as an operating system?
- Introducing **distributions**
  - Distributions are individual projects that offer a Linux-based operating system implementation - in other words, they're all Linux on the inside, but the outside is what makes them different
  - *Well-known Linux distributions*: Ubuntu, Fedora, Debian, Mint, CentOS/RHEL, countless others
- Distributions are what you, the user, would see and interact with instead of "Linux"

So then what are they distributing? Linux?

- Distributions have unique offerings in the **user experience** - every distribution makes decisions about how to form and create the userspace
  - Applications
  - Software packaging standards (inc. kernel updating policy)
  - Desktop environments
  - And more...
- **Desktop environments** are different flavors of desktops - some are closer to a Windows experience while others are more bold and unique
  - Some don't even have a graphical user interface and work solely with a keyboard (often known as **window managers**)
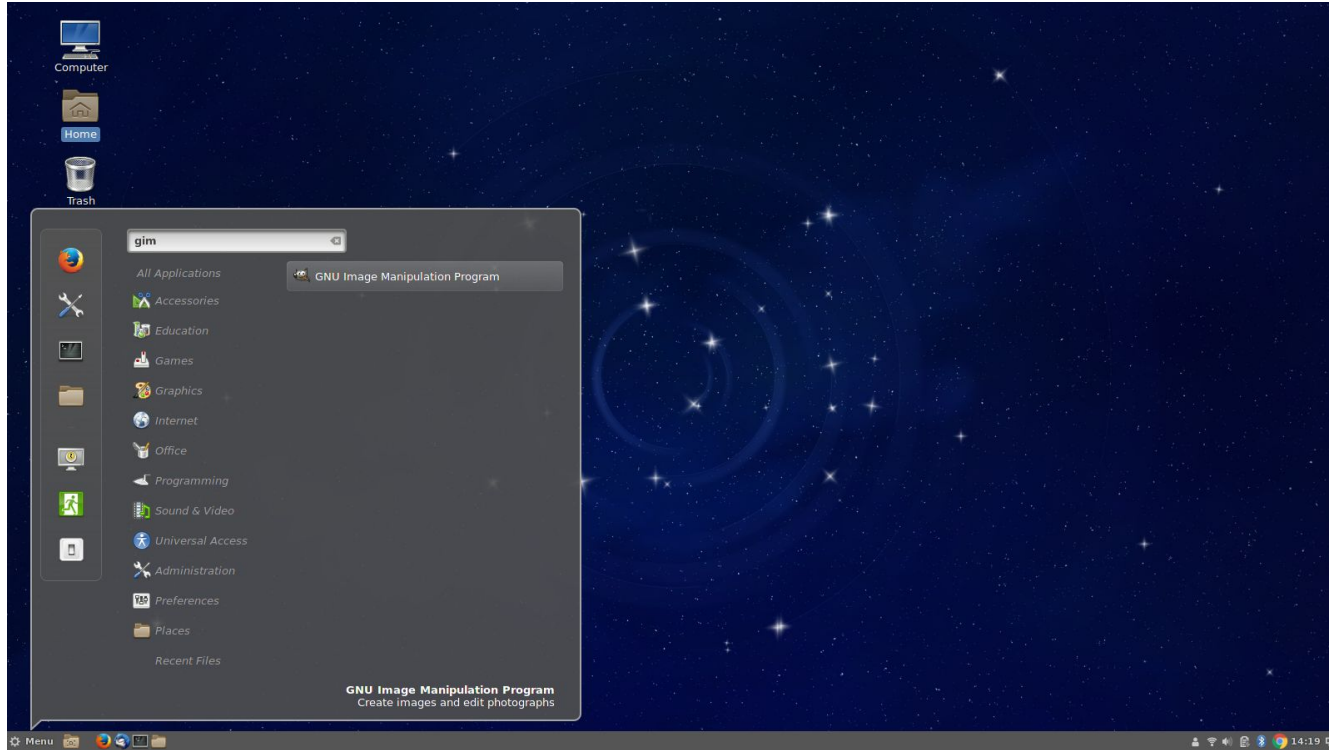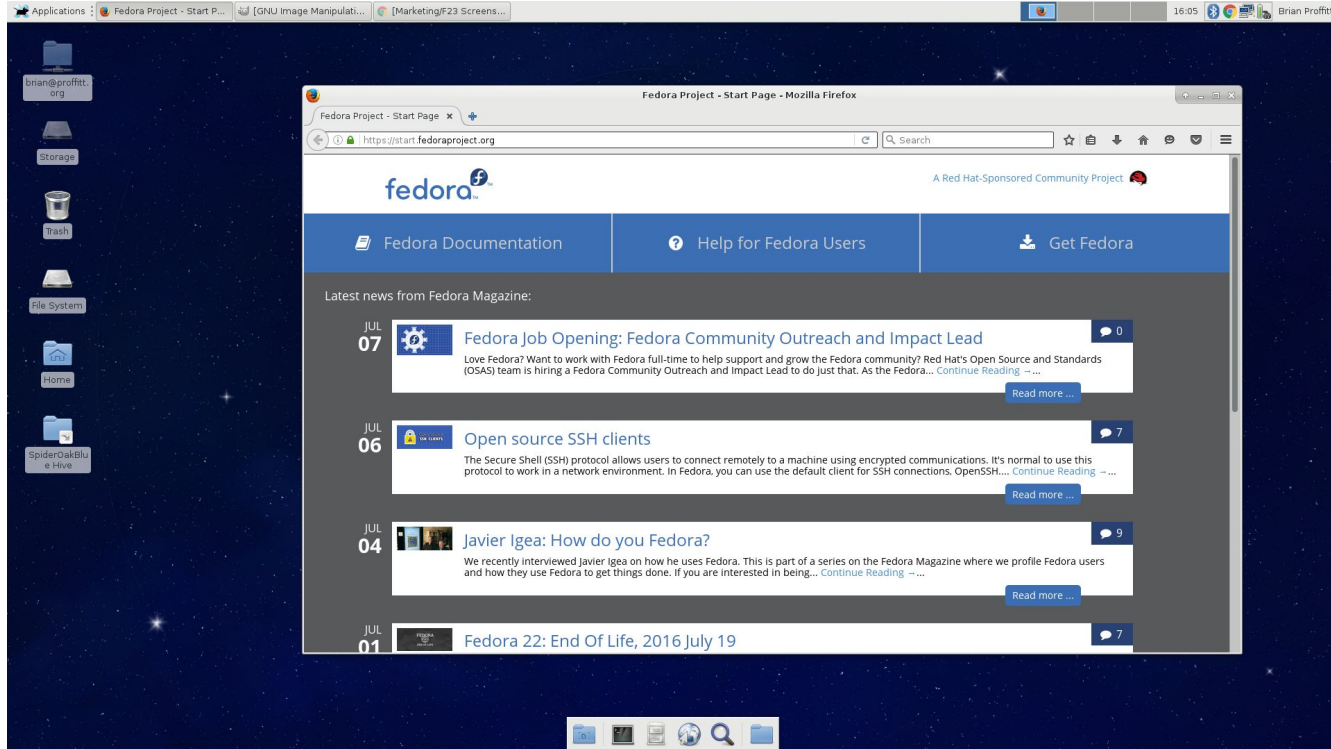
GNOME

KDE
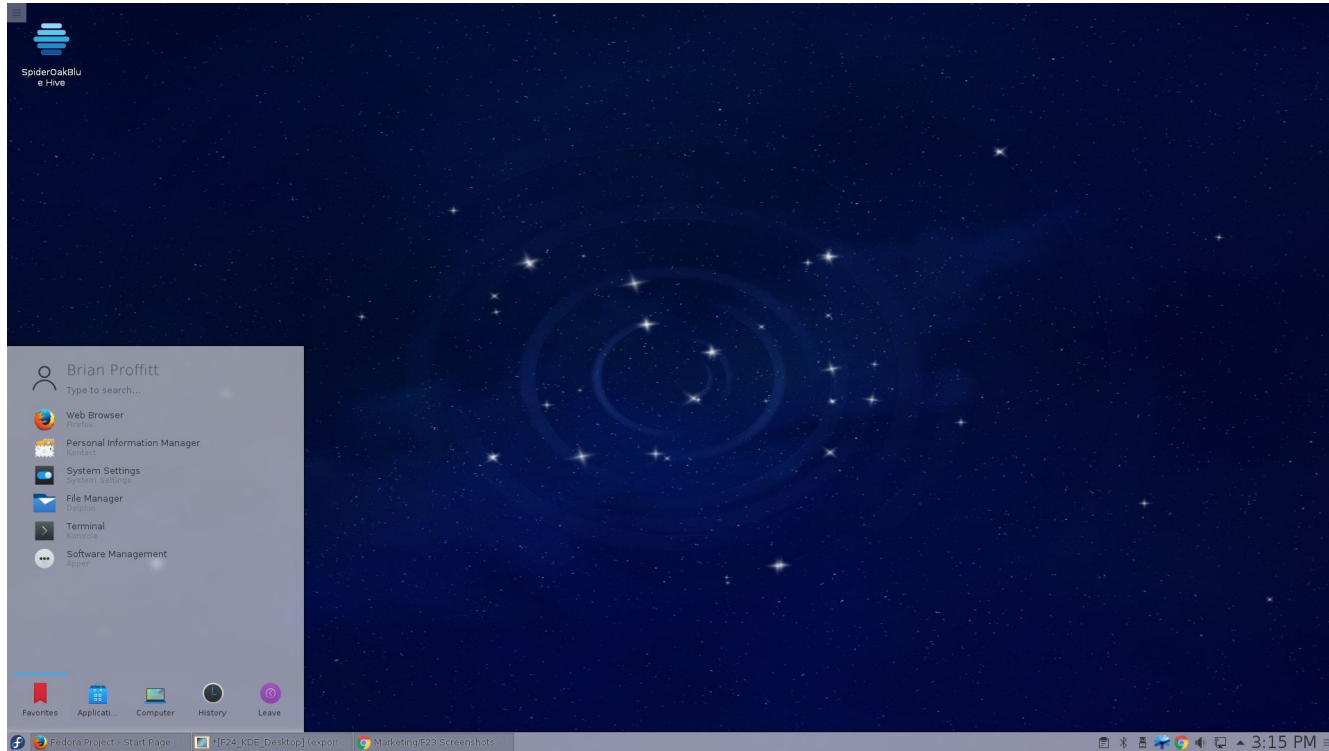
Cinnamon (Windows-like)

Pantheon (OS X-like)

Xfce

Where is Linux? Who uses it?

- Linux… is… the Internet! (only partially exaggerated)
  - February 2014: ~79.3% of public Internet servers are Unix-based, large amount of Linux derivatives
  - ~20.7% Windows Server installations
- **Enterprise computing**: Enterprise-oriented flavors of Linux are common and widely used across the industry – they are everywhere
  - If Windows "wins" on the user desktop, Linux truly "wins" in the server / cloud industry
- Google uses Ubuntu on servers and recommends it for employees
- Even Microsoft loves Linux – seriously!

How can I use Linux?

- Virtual machines
  - Vagrant: For development
  - VirtualBox / VMWare: For your own workstation
- Live boots
  - Boot from a stick
  - Unetbootin
- Dual boots
  - Best of both worlds
  - You don't have to be a rockstar to get the best of both!
- Completely purist
  - Always an option, should you so choose

Example: Using Python in Fedora

- Oh snap! Python assignment due in a few hours!
  - `$ sudo dnf install python python-devel`
- Get your workspace ready! Go grab PyCharm for Linux!
  - https://www.jetbrains.com/pycharm/download/#section=linux
- I need `urllib`! Quick!
  - `$ sudo dnf install python-urllib3`
- Everything is a quick, one-line command away
  - Work with *Python 2.x and 3.x simultaneously*
  - *Jython*: Mix and match your favorite snake with your favorite caffeinated beverage
  - *GitPython*: Interact with git via Python
  - *Winpbd*: Debug problems quickly and easily

Scale of Stability

‹eof›

- Congratulations! You've earned your first badge in mastering Linux!
- Comments, questions, ideas, other things?
  - Have any experiences with Linux to share?