



Intro to Linux Workshop

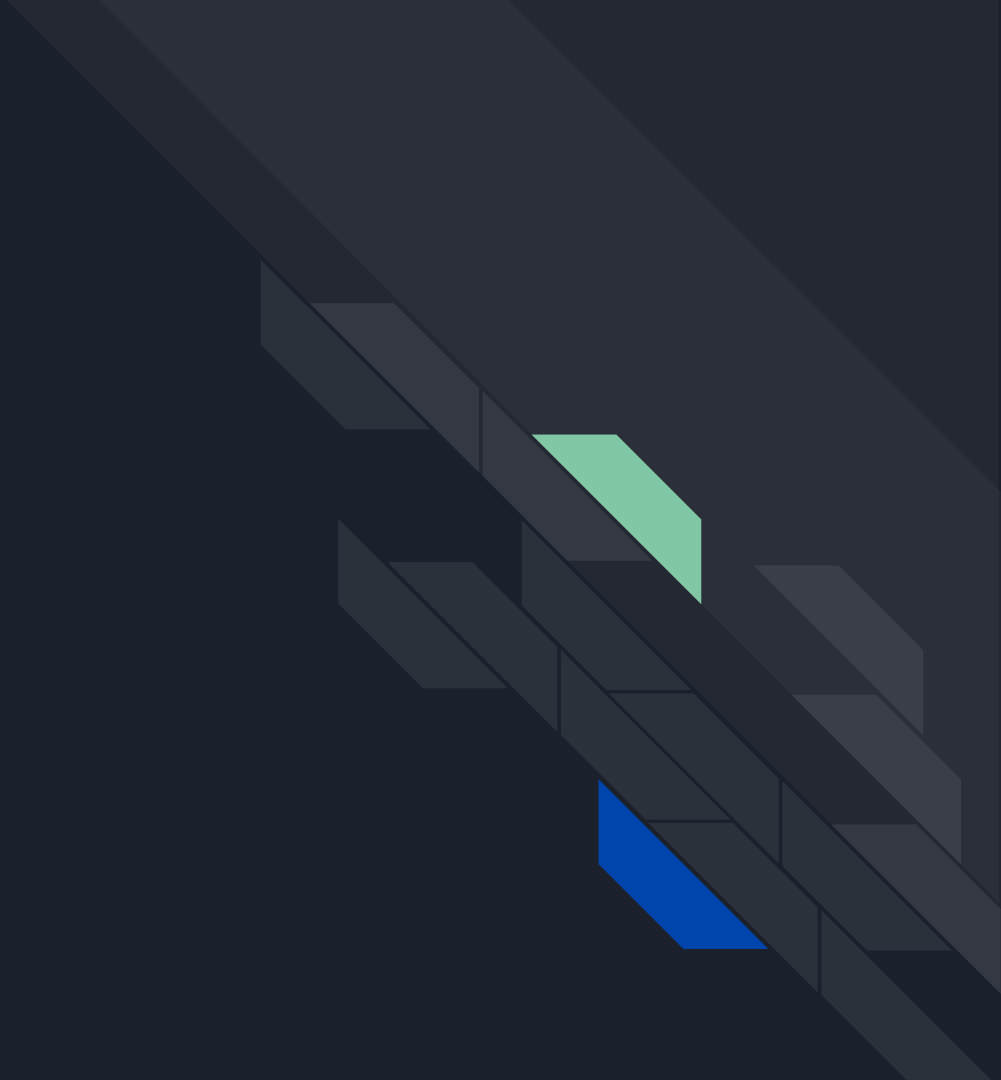
Presented by Lizzy Presland
Co-written by Aidan Hahn
Spring 2019
UWB ACM



Workshop Goals:

- Familiarize you with an industry standard tool
- Introduce you to the values of the Libre Software Movement
- Enable your success in UWB curriculum
- Encourage use of UWB Linux resources

Background



What Is Linux?



Three Tips for New Users:

1. Don't Panic.
2. When in doubt, read the documentation.
3. There is no such thing as a stupid question.



What is Linux *Actually*?

- Part of an Operating System
- Used in a “Distribution”
- Maintained by a team
- The vast majority of web servers
- Steam OS
- Android (Cell Phones)



What is FLOSS?

- Free Libre Open Source Software
- Unrestricted access. No Paid licenses or subscriptions
- Access to source code
- Widespread volunteer effort

Read about Open Source software and the Four Essential Freedoms here: <https://www.gnu.org/philosophy/free-sw.html>



Accessing and Using Linux

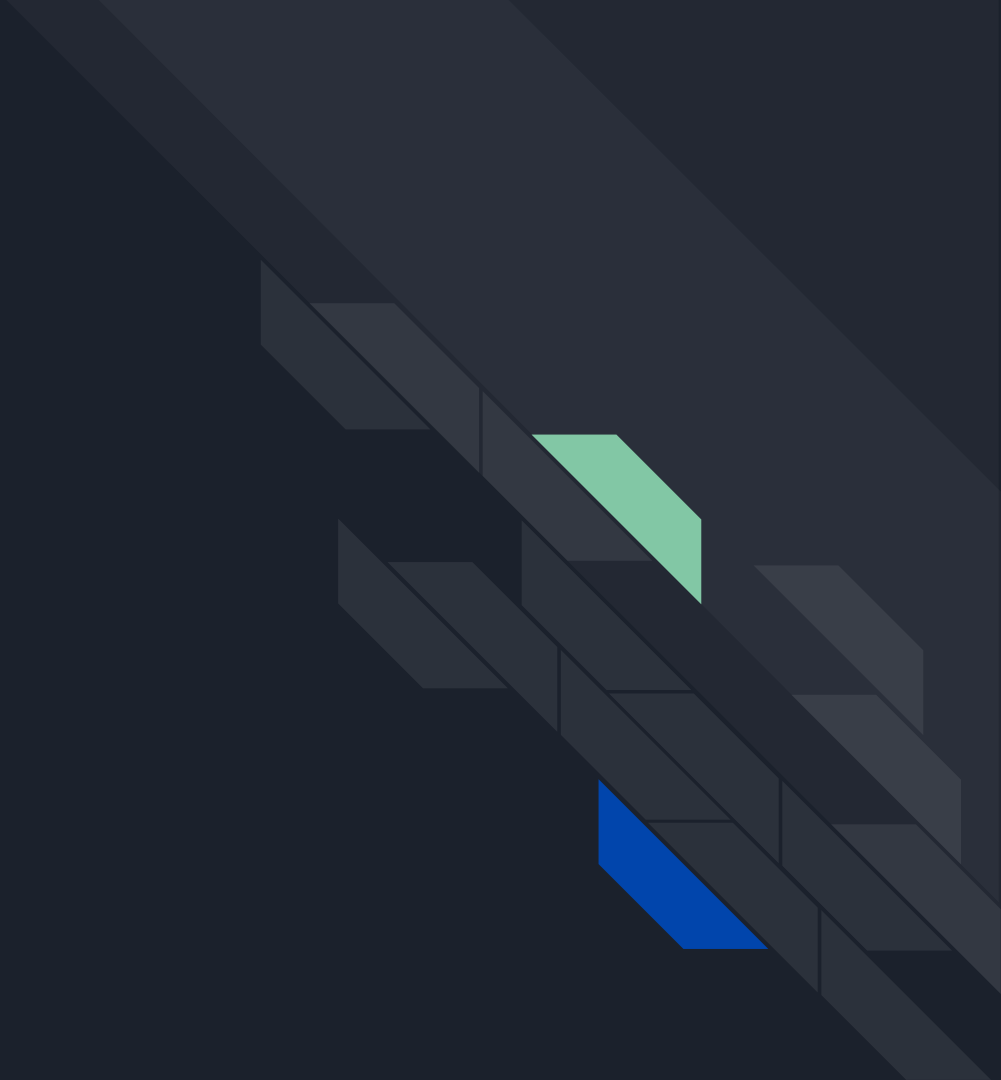
- Live Booting
- Dual Boot
- Virtual Machine
- Main install
- SSH into a server



Where to go for more help

- Man pages
- Distribution wikis
- Tutors at the Linux lab
- Linux Users
- Presentation Slides
- Stack Overflow (as needed)

Usage





Terminal Basics

Anatomy of a shell command:

```
$ command --flag <argument>  
$ command -f      <argument>
```

- **Command:** the name of the executable you want to run
- **Flag:** an optional “switch” which specifies a special execution method or setting
- **Argument:** a (usually optional) piece of information the process or flag requires (such as a file name, a directory, a port number, etc)

“But how do I know what to use?”, you ask?



Terminal Basics (continued)

“But how do I know what to use?”, you ask?

...

Manual pages!

```
$ man command
```



Terminal Basics (continued)

Permissions

```
$ su
```

```
$ sudo command --flag argument
```



Simple commands

- `echo`
 - Echoes standard input to standard output
- `ls`, `pwd`, `cd`
 - Basic file system navigation
- `mkdir`, `touch`
 - Directory and file creation



Simple commands (continued)

- `nano, vim, emacs`
 - Text editors in the command-line interface (CLI)
- `cat`
 - Print file contents to standard output
- `apt, apt-get`
 - Installs new software on your machine
- `which`
 - Determines if software is installed



Git

- Use apt to install git
- Use git to download this presentation (we will do this later :D)

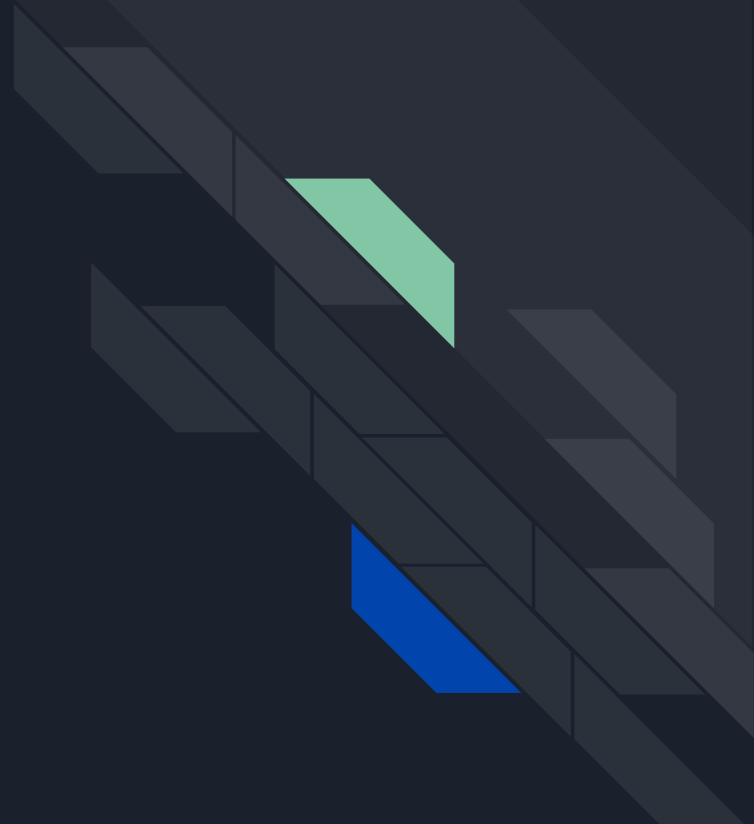
(its here -> <https://gitlab.com/whom/linux-workshop-fall-2018>)

(and here! -> <https://github.com/UWB-ACM/Linux-Crash-Course>)

More Git documentation is available here:

- <https://git-scm.com/>
- <https://towardsdatascience.com/getting-started-with-git-and-github-6fcd0f2d4ac6>

Try It Out





Lets Get Started!

- Boot int your machine
- Install g++, default-jdk, default-jre

```
$ sudo apt-get install g++
```

```
$ sudo apt-get install default-jdk
```

```
$ sudo apt-get install default-jre
```

- Make sure it installed

```
$ which g++
```

```
$ which javac
```

```
$ which java
```



Let's Do Stuff!

```
$ pwd
```

```
$ ls
```

```
$ ls /tmp
```

```
$ ls /
```

```
$ ls ~
```

```
$ echo "Hello, World!"
```

```
$ man echo
```

- Use j to scroll down
- Use q to exit out of the man page



Lets Hello, World!

- Open nano

```
$ nano Hello.java
```

- Write hello world in java
 - Ctrl + X to save and close the file in nano
- Compile and run java hello in the terminal

```
$ javac Hello.java
```

```
$ java Hello
```



Lets Git!

- Check if git is installed

```
$ which git
```

```
$ sudo apt-get install git
```

- Clone the workshop repository

```
$ git clone
```

```
https://github.com/UWB-ACM/Linux-Crash-Course
```



Hello, World! 2: Electric Boogaloo!

- Find our newly-cloned repository

```
$ ls
```

- Change directory to the Git repository

```
$ cd Linux-Crash-Course
```

- Edit hello.cpp to include your name

- Use the same text editor you used for the Java file

- Compile and run

```
$ g++ hello.cpp
```

```
$ ls
```

```
$ ./a.out
```



Introducing the Linux Lab!

- SSH and SCP allow us to securely use another computer to work and share files
- Linux Lab has set up remote machines for students to use via SSH and SCP



Let's Linux Lab!

Steps:

- SCP code over to your machine of choice (or a randomly selected one)

```
$ scp hello.cpp NETID@uw1-320-lab.uwb.edu:~
```

- SSH into the machine you were pointed to (check your prompt for the machine number and replace XX with the number)

```
$ ssh NETID@uw1-320-XX.uwb.edu
```

- Recompile, retest



In Conclusion

- Linux Lab in person tutors
- LFNW, SeaGL, ToorCamp
- Shout out to the UWB ACM!