

Linux Kung Fu



Introduction

What is Linux?

Why Linux?

What is the difference between a client and a server?

What is Linux?

- Linux generally refers to a group of Unix-like free and open-source operating system distributions that use the Linux kernel.
- Examples of Linux distributions:
 - Arch Linux
 - CentOS
 - Debian
 - Fedora
 - Linux Mint
 - Red Hat Enterprise Linux (RHEL)
 - Slackware Linux
 - Ubuntu

Why Linux?

- Because you are taking this course ;)
- Actual benefits
 - **SECURE** (if you know what you're doing)
 - Customizable
 - **The terminal** (more on that in a bit)
 - Power to control OS
 - Open Source
 - Can install/update what you want



Servers versus Clients

- Servers provide services
- Examples:
 - Database servers
 - DHCP servers
 - DNS servers
 - File servers
 - Mail servers
 - Web servers
- Clients consume services
- Examples:
 - Laptops and personal computers
 - Cellular phones
 - Tablets and iPads

The Terminal



The Terminal

- Your shell prompt can be a useful source of information.
- The shell prompt can be customized.
 - This can be done by changing the variable \$PS1.
- You enter commands in the terminal.

The Terminal

- `you@ubnetdef:~$`
 - Username: you
 - Host name: ubnetdef
 - Current working directory: ~
 - Superuser: No (\$)
- `root@universe:/etc/init.d#`
 - Username: root
 - Host name: universe
 - Current working directory: /etc/init.d
 - Superuser: Yes (#)

The Terminal - Commands

- Commands are your way of communicating with your computer
- There are up to 3 components within each command. These are:
 - utility (required)
 - flag
 - argument

[Andrews-MacBook-Pro:~ amavs\$ ls -al website

Basic Linux Commands

```
$ pwd
```

- The `pwd` command prints the name of the current working directory.
 - Essentially, it tells you where you are.
 - Very good command when shell prompt isn't configured to say where you are.
- `$ amavs@ubnetdef:~$ pwd`
 - `/home/amavs`

\$ echo

- The echo command echoes (or displays) text.
 - \$ echo "I love the terminal!"
- The text is sent to standard output by default, but can be redirected.
 - \$ echo "Why did you redirect me?" > redirect.txt
 - \$ echo "Why did you redirect me?" >> redirect.txt
 - Replace vs Append ? you decide
- Very useful command to figure out environment variables
 - Environment variables determine how specific processes behave on an OS
 - Usually in all caps, all of them start with a "\$"

```
$ clear
```

- The clear command clears the terminal's screen if possible.

\$ ls

- The ls command lists the contents of a directory.
 - \$ ls
 - \$ ls /etc
- To include hidden entries:
 - \$ ls -a
 - \$ ls -A
- Did you want more information?
 - \$ ls -l
- They can even be used together!
 - \$ ls -Al /var

```
$ cd
```

- The cd command can be used to change your current working directory.
 - \$ cd ..
 - \$ cd /var/log
- Special directory paths:
 - . - the current directory
 - .. – the parent directory
 - ~ - the current user's home directory
- Useful: “cd -” will change back to your last working directory.
- See also: pushd and popd

\$ history

- Yes, your commands are being logged!
- Essentially, the history command allows you to see the last commands used by the current user.
- When using Bash, the file `~/.bash_history` will be updated at the end of your session.
 - You can bypass this by clearing the current session's history:
 - `$ history -c`

```
$ cat
```

- The cat command concatenates files and/or standard input, printing the result to standard output (by default).
 - \$ cat file1.txt
 - \$ cat file1.txt file2.txt file3.txt

\$ more

- The more program is a file pager.
- It allows you to read files, with support for scrolling down.

\$ less

- Less is the opposite of more. Seriously, it is.
- The less program has more features than the more program.
- The less command gives a terminal pager that allows you to view files with support for scrolling up and down.
- \$ less filename.txt

\$ mkdir

- The mkdir command can be used to make directories.
- To make a directory:
 - \$ mkdir <directory-name>
 - \$ mkdir test
- To make a directory (creating parent directories if needed):
 - \$ mkdir -p <directory-name>
 - \$ mkdir -p this/is/a/test

\$ rm

- The rm command can be used to remove files or directories.
- To remove a file:
 - \$ rm random_file.txt
- To remove any directory:
 - \$ rm -rf random_dir/
- Note: The rm command will remove without warning, so be careful (especially with -rf). Ex: “sudo rm -rf /” → This command will destroy your computer
- To remove an empty directory:
 - \$ rmdir empty_dir

```
$ grep
```

- Grep command in Unix/Linux is the short form of 'global search for the regular expression'
- Used to search for lines matching a specified pattern and print the matching lines to standard output.

```
$ touch
```

- Command that can be used to create new files.
- If file already exists can be used to change time stamp.
- Touch thelist.txt
 - If file thelist.txt doesn't exist a new file will be created
 - If it does exist, time stamp will be changed.

\$ man

- The man command provides an interface to reference manuals.
 - \$ man pwd
 - \$ man man
- For a shorter response, you can often use the --help flag:
 - \$ touch --help

User Management

Users and Groups

- Create a user account:
 - # adduser <username>
- Create a group:
 - # addgroup <groupname>
- Add a user to a group:
 - # adduser <username> <groupname>
 - # usermod -G <groupname> -a <username>

Users and Groups

- See all groups a user is in:
 - `$ groups`
 - `$ groups <username>`
- See more information about a user:
 - `$ id`
 - `$ id <username>`
- See the following files:
 - `/etc/passwd`
 - `/etc/groups`

```
$ passwd
```

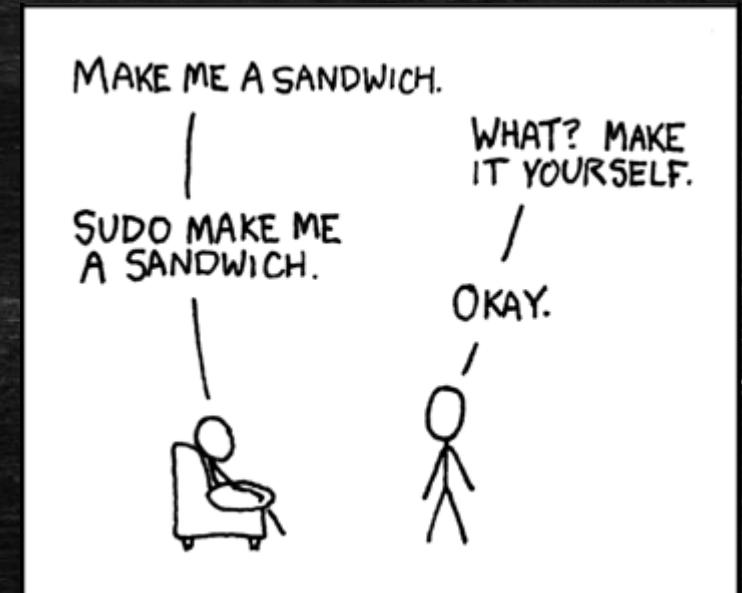
- The passwd command allows changing the passwords of user accounts.
- Changing user passwords:
 - \$ passwd
 - # passwd <username>
- Locking and unlocking user accounts:
 - # passwd -l <username>
 - # passwd -u <username>
- The passwords are stored as hashes in the file /etc/shadow

```
$ su
```

- The su command allows you to switch user.
- If no username is specified, the superuser account (root) will be used by default.

\$ sudo

- Allows permitted users to execute a command as the superuser (i.e. “superuser do”) or another user (if specified).
- Configured in the file /etc/sudoers (can be edited with the visudo command)
 - # visudo
- Examples:
 - \$ sudo whoami
 - \$ sudo cat /etc/sudoers



\$ chmod

- Changes permissions of files/folders for users
- Helpful command to consider when running into configuration/installation issues (or when someone has access to something they shouldn't)
- 3 digit binary number is the first argument for chmod
- Read/Write/Execute for owner/group/all others
 - # chmod *number* *file/directory*
 - White board time!

owner	group	others
r w x	r w x	r w x
(4)(2)(1)	(4)(2)(1)	(4)(2)(1)

Processes and Networking

```
$ ps
```

- The ps command provides a process snapshot.
- \$ ps
- \$ ps aux

```
$ top
```

- Similar to ps, but is interactive and updates every second.
- A similar utility, htop, provides a similar function, but usually needs to be installed first.

\$ kill

- To ask a process to ~~commit suicide~~ terminate (but it could choose to ignore you):
 - \$ kill <pid>
- To ask the kernel to ~~commit homicide~~ kill a process (this cannot be ignored):
 - \$ kill -9 <pid>
 - \$ kill -KILL <pid>
 - \$ kill -SIGKILL <pid>

\$ ping

- The ping command sends an ICMP ECHO_REQUEST to network hosts.
- Pinging IP addresses is usually a simple way to check if your internet connection is working.
- Example:
 - \$ ping 8.8.8.8

\$ ifconfig

- The ifconfig command can be used to view or configure network interfaces.
- View all interfaces:
 - \$ ifconfig
- View specific interface:
 - \$ ifconfig <interface-name>
 - \$ ifconfig lo
- Bring an interface online or offline (respectively):
 - # ifconfig <interface-name> <up | down>
 - # ifconfig ens32 up

\$ ip

- The ip command has subcommands for showing and manipulating routing, network devices, interfaces, and tunnels.
 - Commands such as arp, ifconfig, and route are actually deprecated, and might not be on some newer systems by default.
- Some subcommands:
 - \$ ip address
 - Can be shortened to “ip a”
 - Replacement for “ifconfig”
 - \$ ip neigh
 - Can be shortened to “ip n”
 - Replacement for “arp -a”

Services

Services

- In Linux, services are applications or processes that run in the background.
- They are sometimes referred to as daemons.
 - Many of their names will end with “d” out of convention (e.g. sshd, httpd, mongod).

Services

- There are two main ways to control services:
 - System V (older; also called SysV)
 - systemd (newer)
- System V
 - `# service <name> <start | stop | restart | reload | status >`
 - `# service sshd status`
- systemd
 - `# systemctl <start | stop | restart | reload | status > <name>`
 - `# systemctl reload nginx`

Package Managers

Package Managers

- Package managers can help with automating common tasks such as installing, upgrading, and uninstalling programs or *packages*.
- Examples:
 - apt (Advanced Packaging Tool)
 - apt-get
 - aptitude
 - dpkg (Debian Package)
 - dnf (Dandified Yum)
 - pacman
 - rpm (RPM package manager)
 - yum (Yellowdog Updater, Modified)
 - installpkg (Cucumber Linux)

\$ apt

- Update the local package index:
 - # apt update
- Upgrade a package:
 - # apt upgrade <package-name>
- Upgrade all packages:
 - # apt upgrade

\$ apt

- Install a package:
 - # apt install <package-name>
- Uninstall a package (leave configuration):
 - # apt uninstall <package-name>
- Uninstall a package (remove configuration):
 - # apt purge <package-name>
- Uninstall unneeded dependencies:
 - # apt autoremove

Useful Tips and Tricks

Useful Tips and Tricks

- Pressing the up arrow recalls the previous commands.
- Pressing tab while typing a command can sometimes help to autocomplete a command's name or a file/directory path.
- If you need to stop a currently-running command, use Ctrl+C.
- Typing “!!” in the terminal will re-run the last command. Ie) sudo !!
- If you accidentally print the contents of a binary file to the terminal, it may affect the terminal's display. The “reset” command can be used to resolve that issue.
- **\$cd** - to go back to the most recent directory

Directory Structure

Directory Structure

- /bin: Contains all binaries that are necessary for the system to function
- /boot: Contains the Linux bootloader
- /dev: Contains all raw DEVices
- /etc: Contains configuration files for the system
- /home: Contains user's home directories
- /mnt: Typically used when mounting devices (like a cdrom)
- /opt: Optional, addon packages

Directory Structure

- /proc: A pseudo-file system containing information about processes
- /root: User root's home directory
- /sbin: Contains all the binaries that are necessary for the system to run; only super users can use these
- /tmp: Temporary files
- /usr: User-installed files
- /var: Variable data that change as the system is running normally, such as log files, lock files, cache, spool, and temporary files

Want More Commands?

```
$ lsof
```

- The lsof command provides a way to list open files.
- All open files:
 - \$ lsof
- All open files by a specific user:
 - \$ lsof -u <username>
- All open network connections:
 - # lsof -i

\$ route

- The route command provides a way to view and manipulate the routing table.
- To view the routing table:
 - \$ route
- To specify 10.0.0.1 as the default gateway:
 - # route add default gw 10.0.0.1

```
$ dig
```

- The dig command is a DNS lookup utility.
- This is useful for verifying that DNS records are being properly resolved.
- This utility disregards the /etc/hosts file.
- Example:
 - \$ dig ubnetdef.org
 - \$ dig +short ubnetdef.org