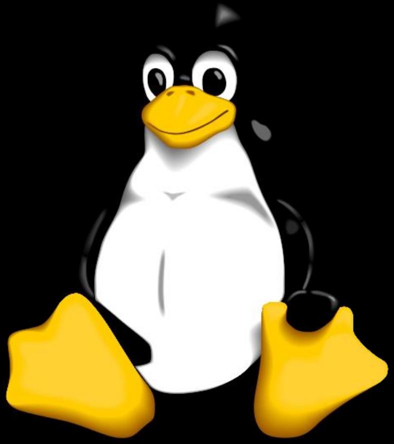
$_{\text{v1}}$Intro to Linux



208d50cbfc9f959a5c4321a80b329187

Introduction

- Install VMware Player and Ubuntu Server
- Welcome to Linux
- Basic Operations
- Directory Operations
- File
  - Creation
  - Viewing
  - Comparison
  - Binary, Octal, & Hex…oh my
  - Properties
  - Location
  - Manipulation
  - Compression and Packaging
- Filesystems
- Processes
- Parting Thoughts

What to Expect

- What this tutorial is…
  - Content based off Ubuntu Server 12.04.4 LTS (64 bit)
  - A basic intro to Linux OS commands
  - It is encouraged to try and emulate all the commands within this presentation
    - Commands are designated by the color green
  - Some commands have little explanation
    - Intended for you to discover what they do!
  - Remember to check out the reference slides at the end for more information
- What this tutorial is not…
  - A definitive guide to the Linux OS
  - An explanation for every command or best practices
    - Some commands you may never use again!
    - Other commands you may find better uses for in the future
    - This is simply an introduction

Hardware Used

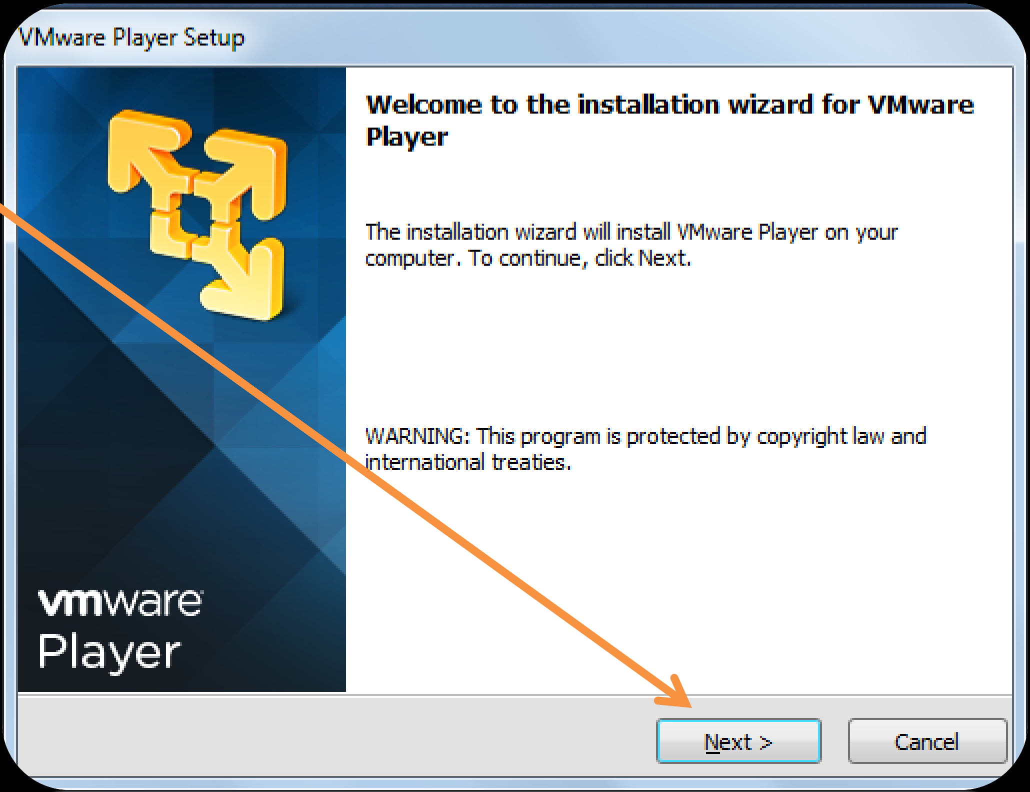
- I used the following when creating this tutorial:

  **versions may differ as updates are constantly released*
  - VMware Player 6.0.1 build-1379776
    - https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0
  - Ubuntu 12.04.4 Server (amd64)
    - http://www.ubuntu.com/download/server
  - Host System: Windows 7 Professional

- My Windows 7 Host Specifications:
  - Windows 7 Professional Service Pack 1
  - Intel i5 CPU M 480 @ 2.67 GHz
  - 8.00 GB installed RAM
  - 64-bit Operating System
  - Minimum of 2.0 GB hard drive space

Installing VMware Player

- Select “**Next >**”



VMware Player Setup

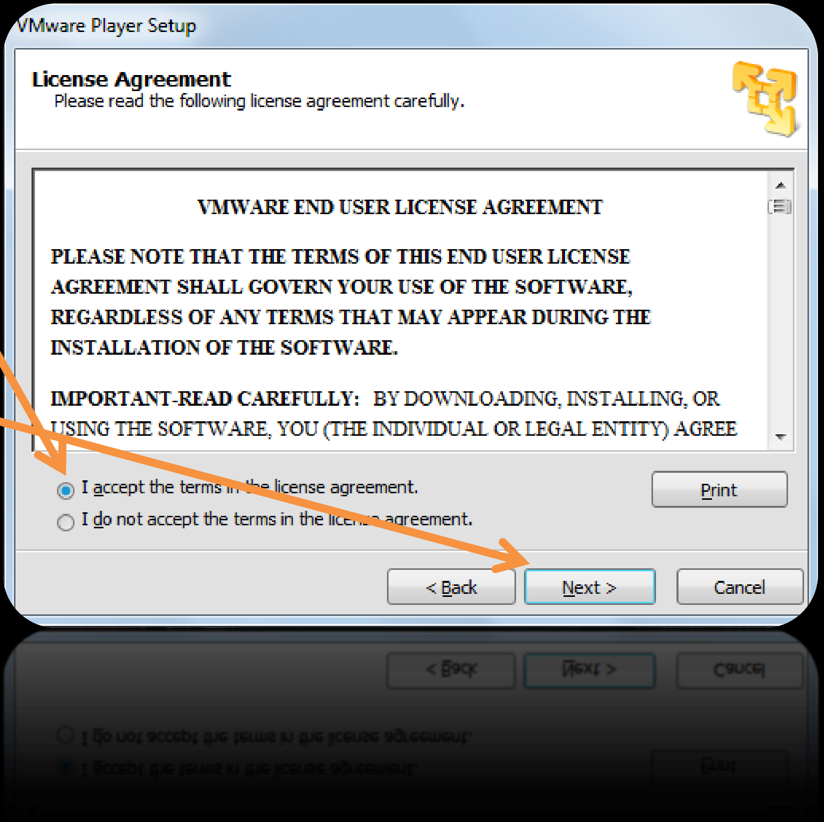
Welcome to the installation wizard for VMware Player

The installation wizard will install VMware Player on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

Cont.

- First, select “**I accept the terms in the license agreement**” and then select the “**Next >**” button to proceed.



VMware Player Setup

License Agreement

Please read the following license agreement carefully.

VMWARE END USER LICENSE AGREEMENT

PLEASE NOTE THAT THE TERMS OF THIS END USER LICENSE AGREEMENT SHALL GOVERN YOUR USE OF THE SOFTWARE, REGARDLESS OF ANY TERMS THAT MAY APPEAR DURING THE INSTALLATION OF THE SOFTWARE.

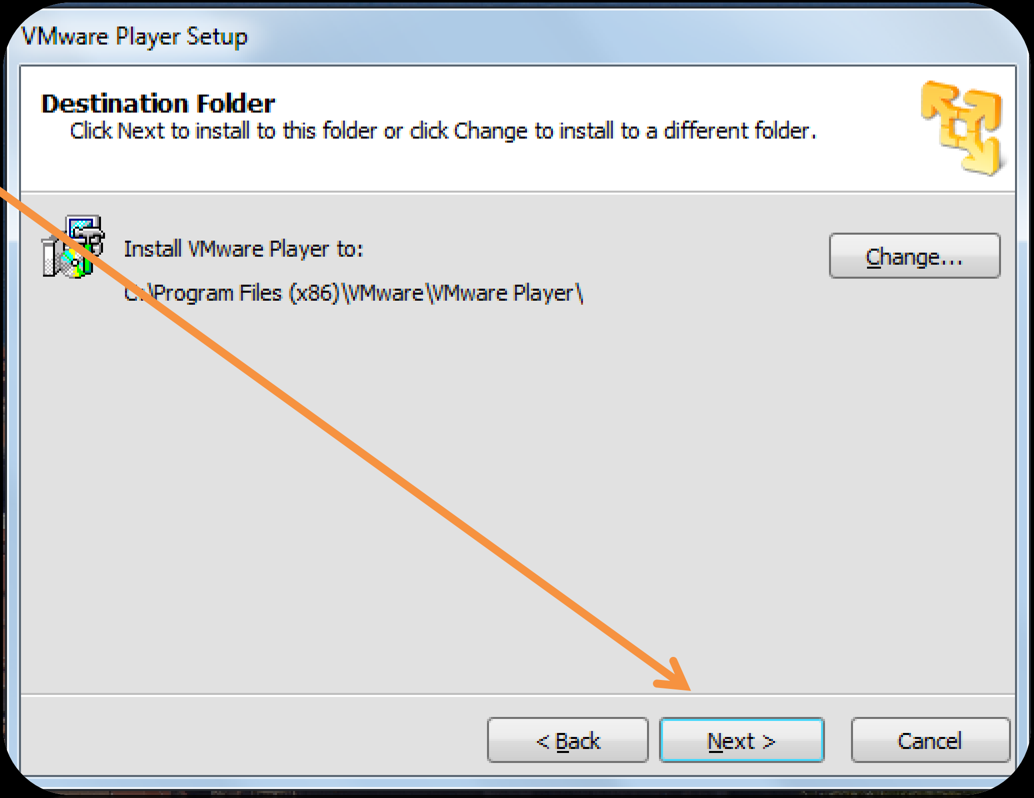
IMPORTANT-READ CAREFULLY: BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE, YOU (THE INDIVIDUAL OR LEGAL ENTITY) AGREE

● I accept the terms in the license agreement.

○ I do not accept the terms in the license agreement.

Cont.

- Select “**Next >**”



VMware Player Setup

Destination Folder

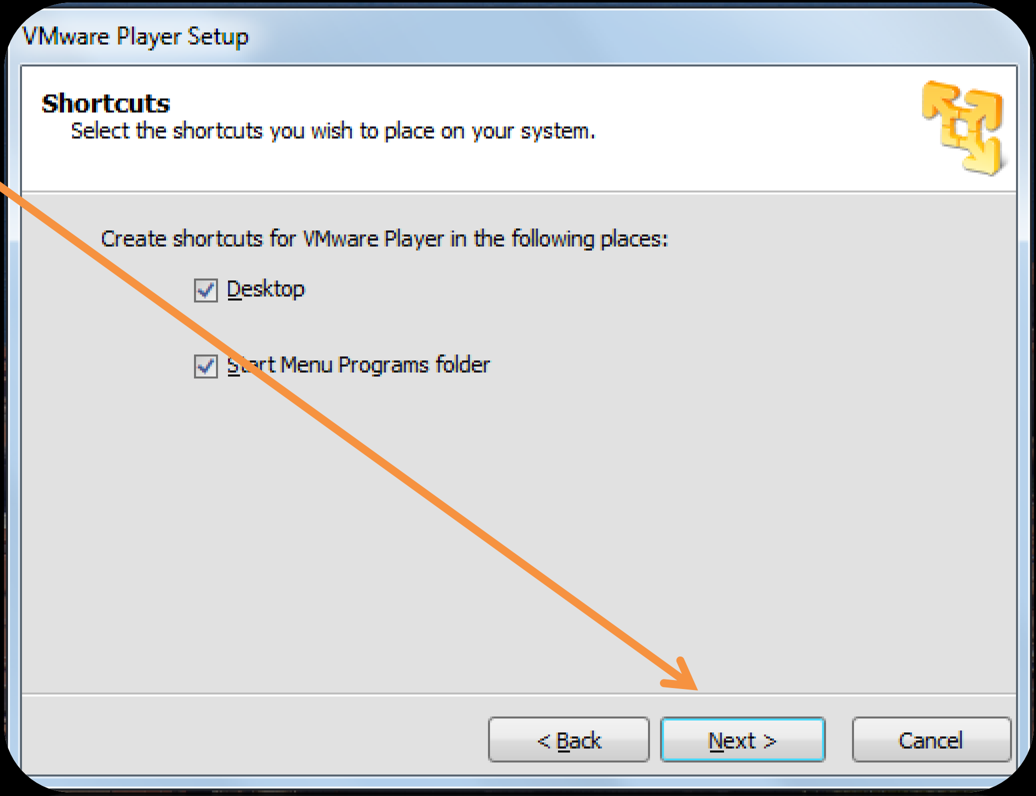
Click Next to install to this folder or click Change to install to a different folder.

Install VMware Player to:

C:\Program Files (x86)\VMware\VMware Player\

Cont.

- Select "**Next >**"



VMware Player Setup

Shortcuts

Select the shortcuts you wish to place on your system.

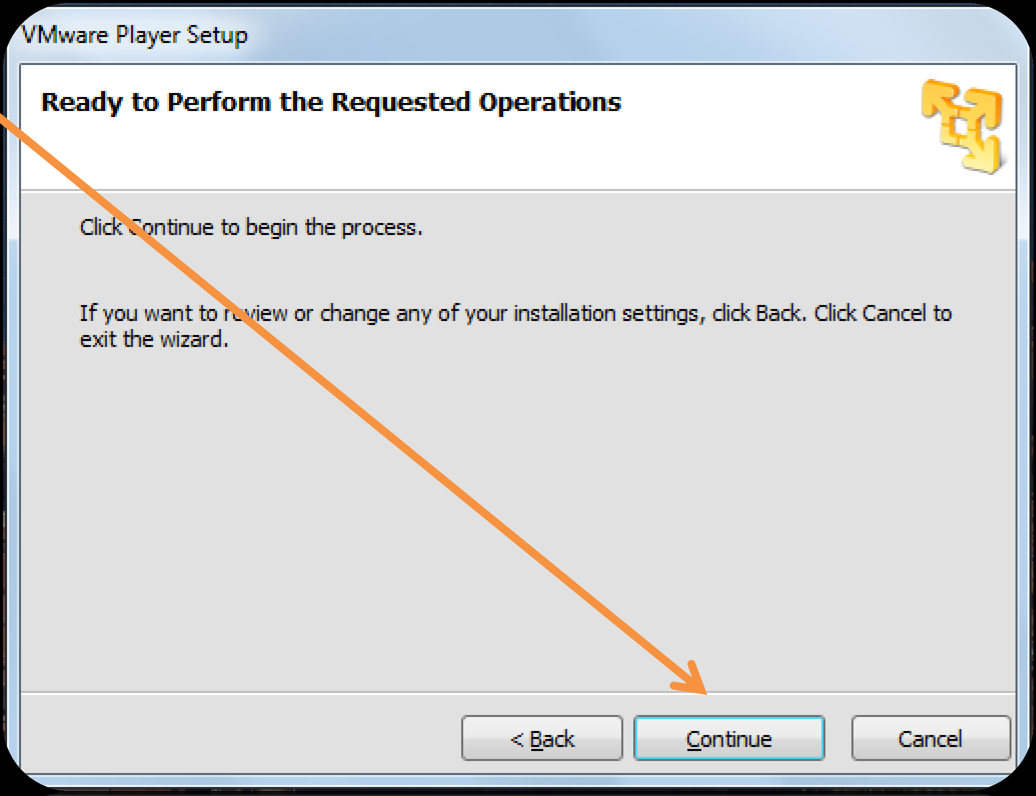
Create shortcuts for VMware Player in the following places:

☑ Desktop

☑ Start Menu Programs folder

Cont.

- Select “**Continue**”



VMware Player Setup

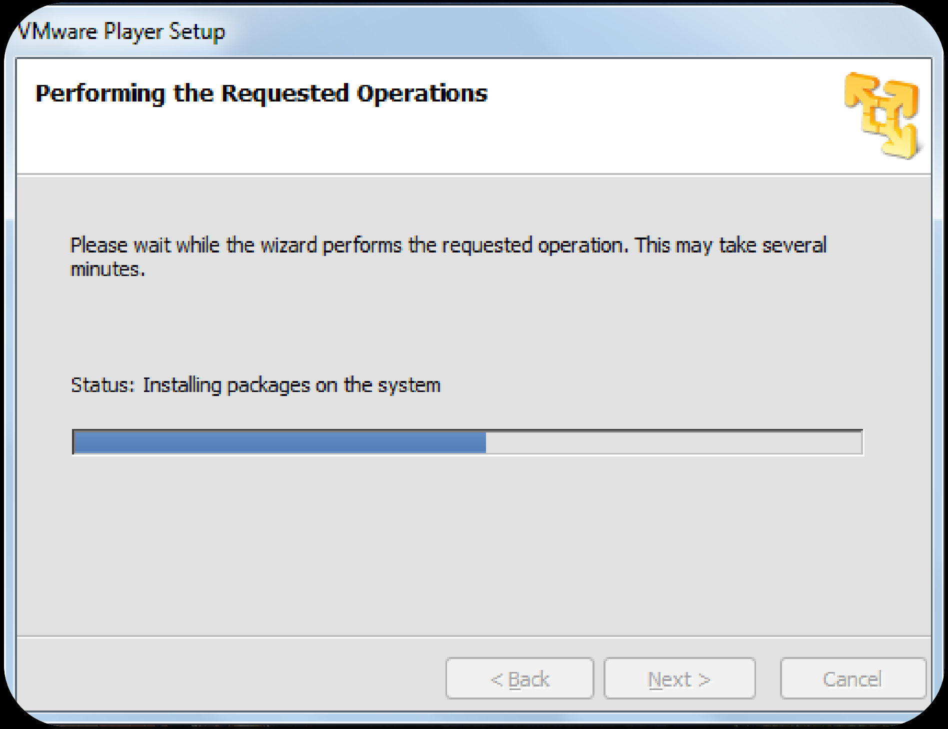
Ready to Perform the Requested Operations

Click Continue to begin the process.

If you want to review or change any of your installation settings, click Back. Click Cancel to exit the wizard.

Cont.

- Allow a few minutes for VMware Player to install



VMware Player Setup

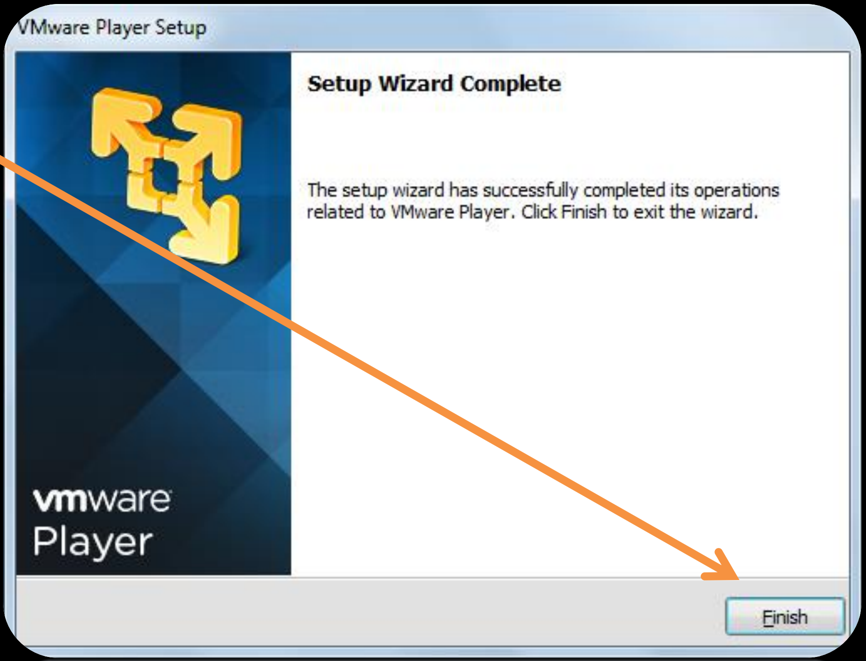
Performing the Requested Operations

Please wait while the wizard performs the requested operation. This may take several minutes.

Status: Installing packages on the system

Cont.

- Select "**Finish**" and move onto the next section!



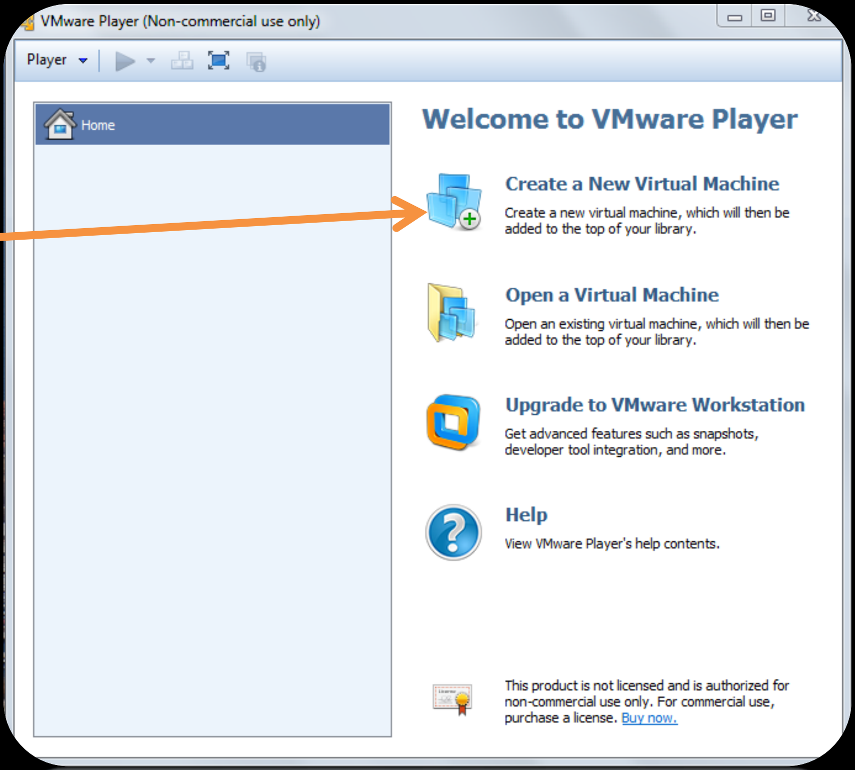
VMware Player Setup

Setup Wizard Complete

The setup wizard has successfully completed its operations related to VMware Player. Click Finish to exit the wizard.

Installing Ubuntu Server

- Load VMware Player
- Select “**Create a New Virtual Machine**”



Welcome to VMware Player

Create a New Virtual Machine
Create a new virtual machine, which will then be added to the top of your library.

Open a Virtual Machine
Open an existing virtual machine, which will then be added to the top of your library.

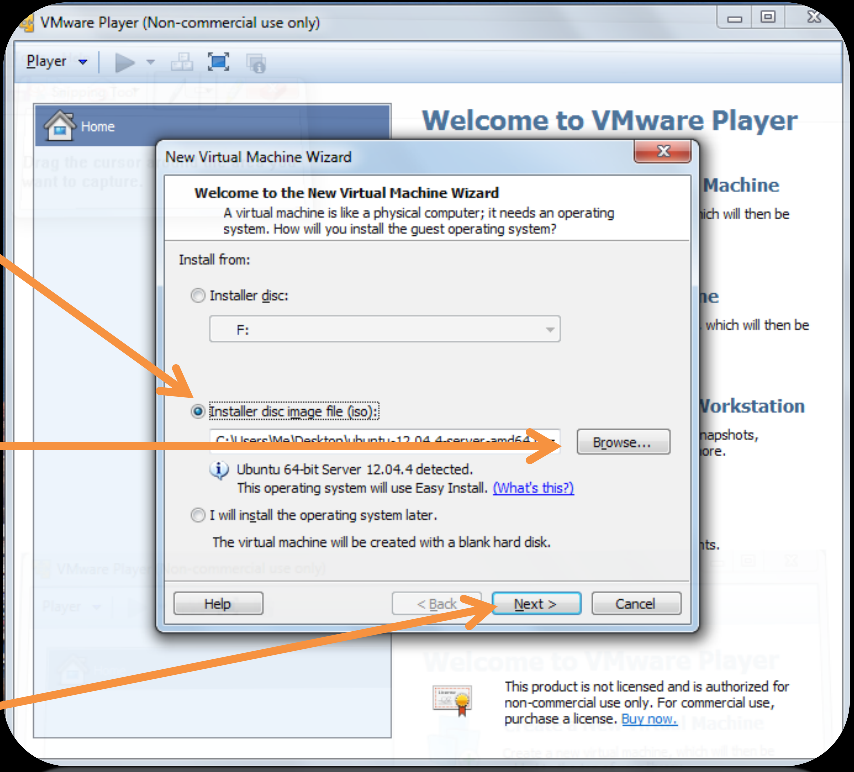
Upgrade to VMware Workstation
Get advanced features such as snapshots, developer tool integration, and more.

Help
View VMware Player's help contents.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- Select the radio button for “**Installer disc image file (iso):**”
- Then, select “**Browse…**” and go to your “**ubuntu-12.04.4-server-amd64.iso**” location
  - For this example, my ISO was located on my desktop
- Once your finished, select “**Next >**”



VMware Player (Non-commercial use only)

New Virtual Machine Wizard

Welcome to the New Virtual Machine Wizard

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

○ Installer disc:

F:

● Installer disc image file (iso):

C:\Users\Me\Desktop\ubuntu-12.04.4-server-amd64

Ubuntu 64-bit Server 12.04.4 detected.
This operating system will use Easy Install. (What's this?)

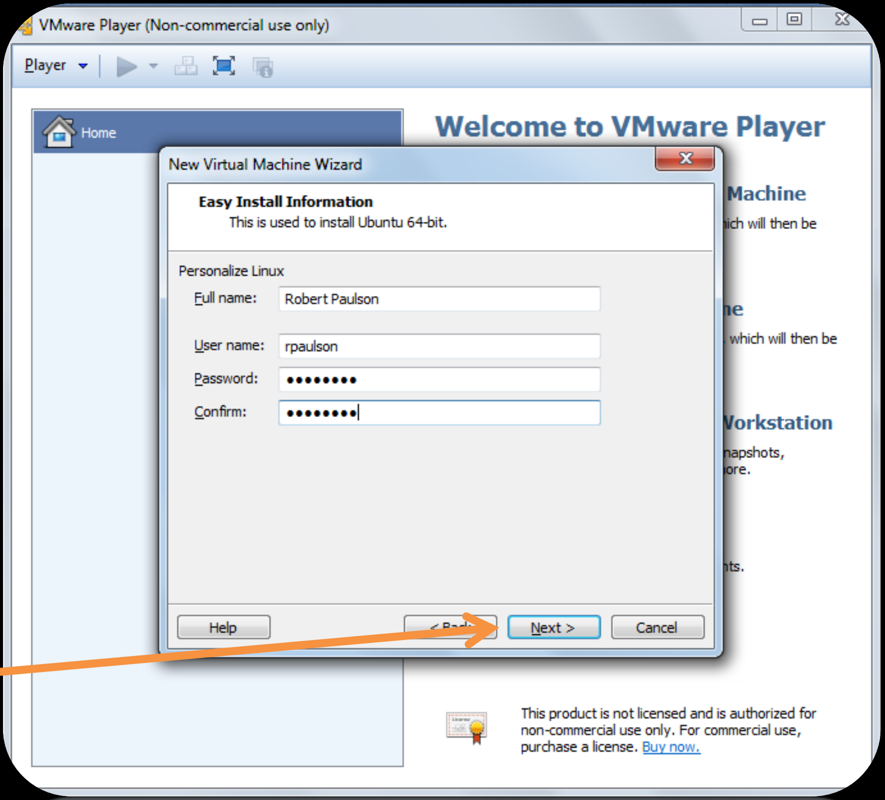
○ I will install the operating system later.

The virtual machine will be created with a blank hard disk.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- Enter the following information
  - Full name:
  - User name:
  - Password:
  - Confirm:
- Once your finished select “**Next >**”



New Virtual Machine Wizard

Easy Install Information

This is used to install Ubuntu 64-bit.

Personalize Linux

Full name: Robert Paulson

User name: rpaulson

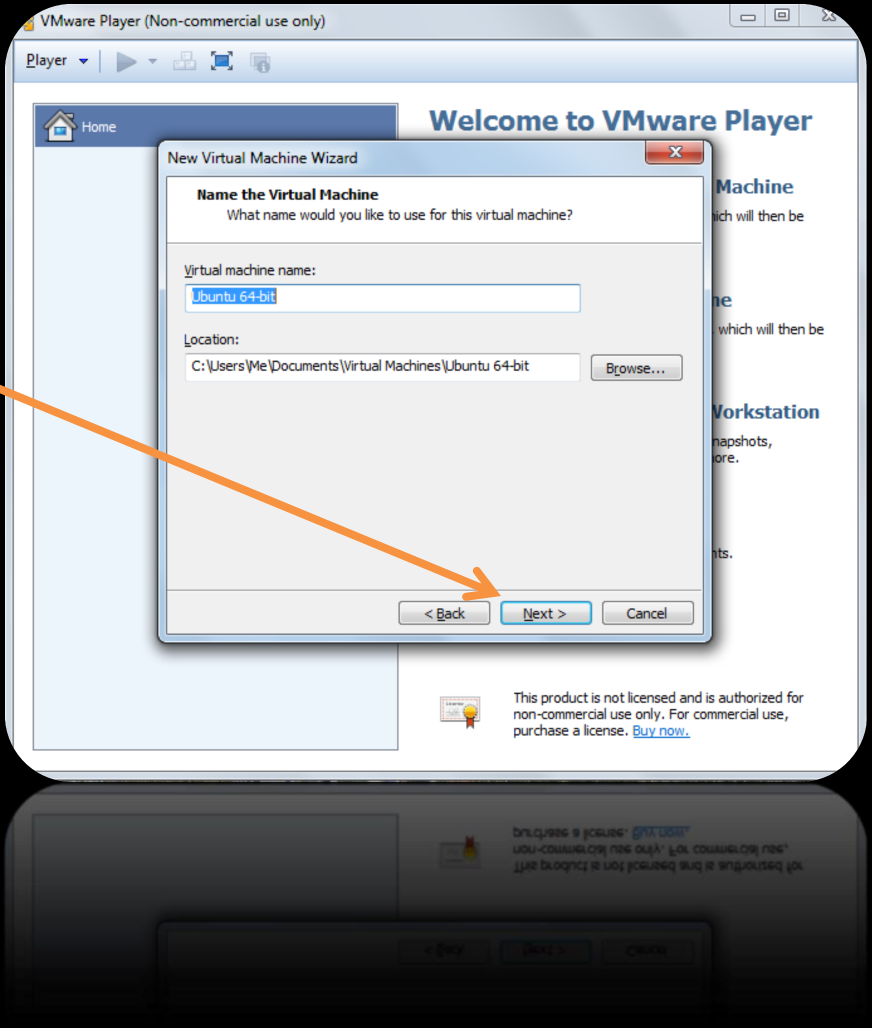
Password: ••••••••

Confirm: ••••••••

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- Leave the following screen as default and when ready select “**Next >**”



VMware Player (Non-commercial use only)

Welcome to VMware Player

New Virtual Machine Wizard

Name the Virtual Machine

What name would you like to use for this virtual machine?

Virtual machine name:

Ubuntu 64-bit

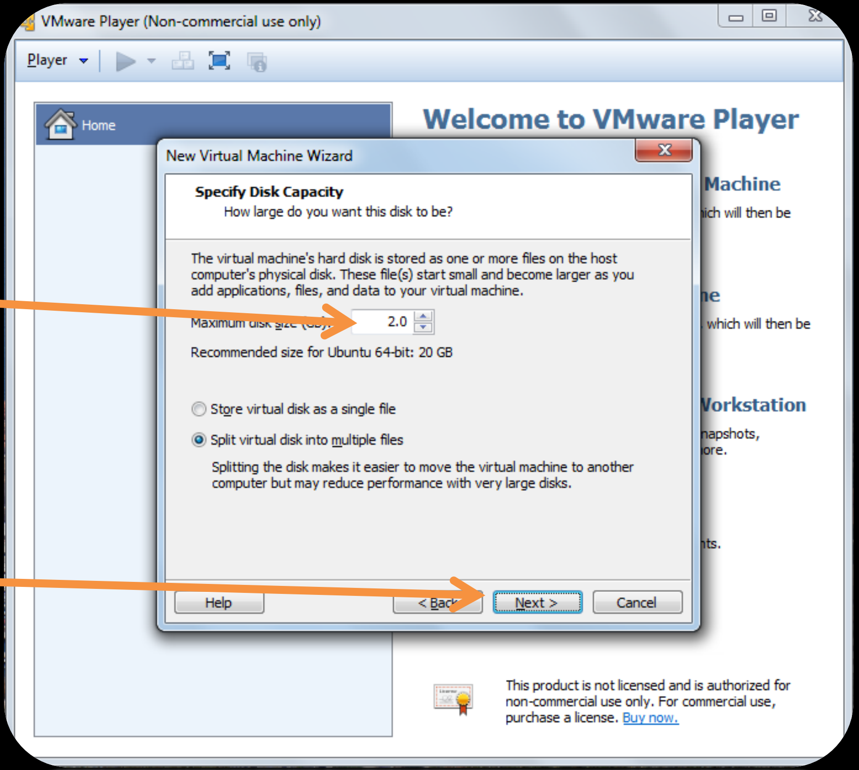
Location:

C:\Users\Me\Documents\Virtual Machines\Ubuntu 64-bit

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- Change “**Maximum disk size (GB):**” to “**2.0**”
- Leave everything else as default, and select “**Next >**”



New Virtual Machine Wizard

Specify Disk Capacity
How large do you want this disk to be?

The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.

Maximum disk size (GB): 2.0

Recommended size for Ubuntu 64-bit: 20 GB

○ Store virtual disk as a single file

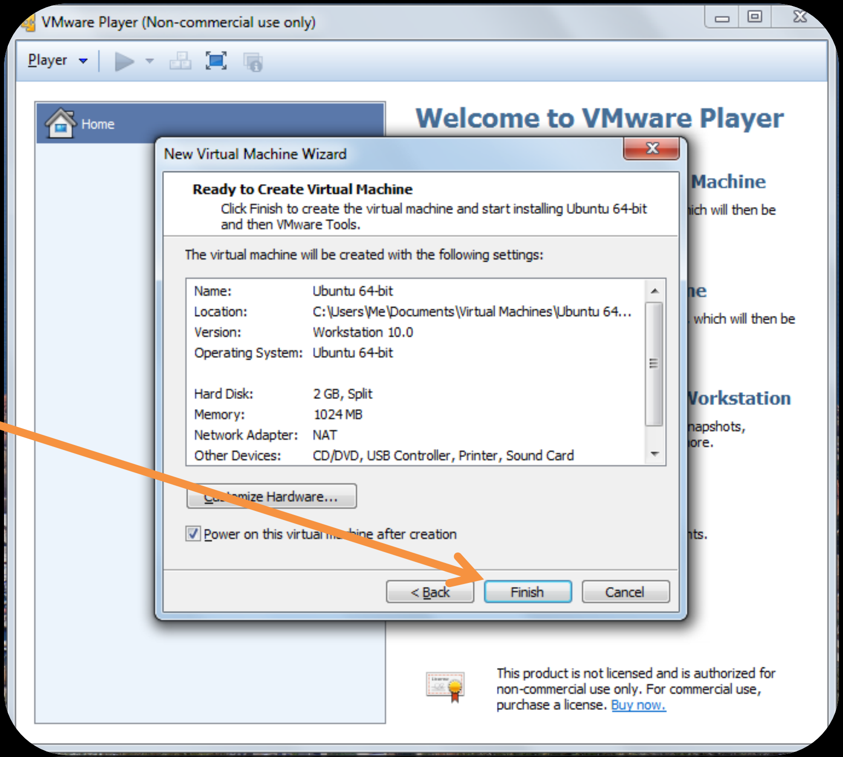
● Split virtual disk into multiple files

Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- Leave the following screen as default and when ready select “**Finish**”



VMware Player (Non-commercial use only)

New Virtual Machine Wizard

Ready to Create Virtual Machine

Click Finish to create the virtual machine and start installing Ubuntu 64-bit and then VMware Tools.

The virtual machine will be created with the following settings:

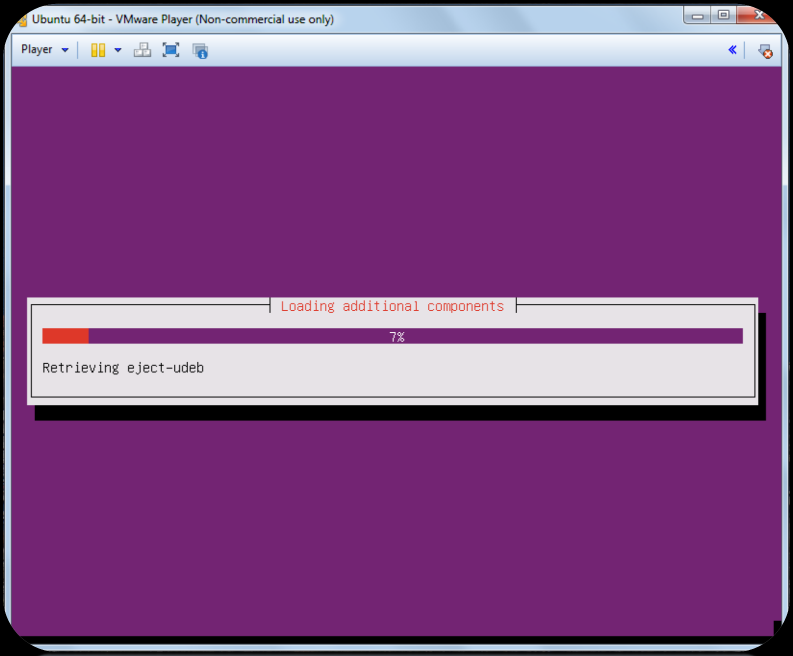
Name:	Ubuntu 64-bit
Location:	C:\Users\Me\Documents\Virtual Machines\Ubuntu 64...
Version:	Workstation 10.0
Operating System:	Ubuntu 64-bit
Hard Disk:	2 GB, Split
Memory:	1024 MB
Network Adapter:	NAT
Other Devices:	CD/DVD, USB Controller, Printer, Sound Card

☑ Power on this virtual machine after creation

This product is not licensed and is authorized for non-commercial use only. For commercial use, purchase a license. Buy now.

Cont.

- If any boxes come up, simply select "**OK**"
- Sit back and allow Ubuntu Sever to install itself
- Depending on your system this process may take 10-15 minutes, or longer
- Minimize VMware Player as it installs the OS and continue on with this tutorial



Loading additional components

7%

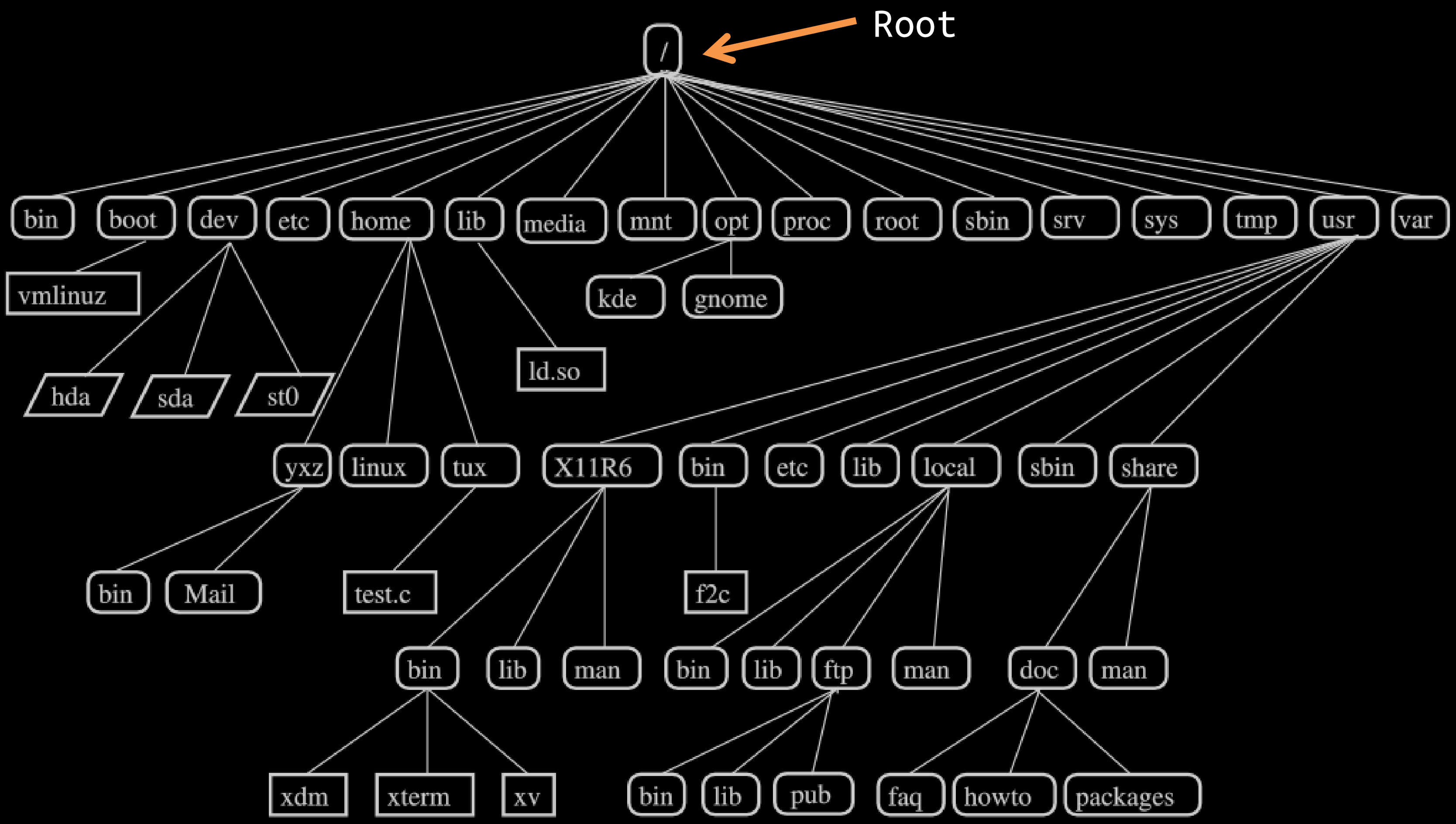
Retrieving eject-udeb

Welcome to Linux

- Kernel
  - "Low level operating system, handling files, disks, networking, and other necessities…" (Linux Pocket Guide)
  - "…the hub of the operating system: it allocates time and memory to programs and handles the filestore and communications in response to system calls." (UNIX Introduction)
- The shell
  - "…acts as an interface between the user and the kernel." (UNIX Introduction)
  - "…command language interpreter." (Linux Bible)

Cont.

- Typical Linux file system structure



Ready…GO!

- Bring your minimized VMware Player back up and proceed to the next slide to begin your hands journey with the Intro to Linux Commands Tutorial!
- If your system is still installing Ubuntu, take a quick break

```
Ubuntu 12.04.4 LTS ubuntu tty1

ubuntu login: rpaulson
Password: _
Last login: Wed Mar 26 12:51:00 PDT 2014 on tty1
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic x86_64)

 * Documentation:  https://ubuntu.com
rpaulson@ubuntu:~$ _
```

Cont.

- Linux ‘Root’ - has multiple meanings
  - The start of the hierarchal file system structure -> ‘/’
  - The default “administrative” account on the Linux OS
    - Similar to Windows SYSTEM access
- The shell prompt (rpaulson@ubuntu:~$)
  - username, system name, and current directory name

rpaulson @ ubuntu: ~ $

  - ‘#’ prompt indicates you are root (hash mark)
  - ‘$’ or another prompt indicates you are not root

**To save room only the prompt will be represented within this presentation from here on out*

Cont.

- Common Linux shortcuts
  - Ctrl + l (clears the screen)
  - Ctrl + c (terminates current process)
  - Ctrl + z (pause current process)

**All future references to ‘Ctrl + n’ in this guide will be ‘^n’, where n designates any character (e.g. ^l, ^c, etc.)*

Cont.

- Typical Linux command structure
  - command [option(s)] [parameter(s)]
- Combining commands
  - ‘;’ - perform commands sequentially
    - `$ ls -al ; date ; time`
  - ‘&&’ - perform sequentially, but stops when one fails
    - `$ ls -al && dte && time`
  - ‘||’ - perform sequentially, but stops when one is successful
    - `$ ls -al || date || time`

I Need Help!

- No Internet access—no problem!
- How to get and find help?
  - $ man [command]
    - Man -> manual page
    - Provides further information for commands
  - $ info [command]
    - Provide an in-depth reference manuals for commands
    - Written in the early 90's to replace the aging man pages
  - $ apropos [keyword]
    - Keyword search in man page's names and descriptions
  - $ [command] --help

Basic Commands

- $ clear
  - Clears the terminal screen (same as ^l)
- $ exit
  - Logout of the current terminal
- $ reset
  - Reset current terminal (try $ cat /usr/bin/who)
- $ su [username]
  - Temporarily switch to another user
  - If no username is supplied, default is root
- $ sudo [command]
  - Execute a single command as another user (generally root)
  - Current user must be a member of the ‘sudo’ group

Cont.

- $ who
  - Show who (all users) that are currently logged on
    - -H -> add headers to each column
- $ whoami
  - Print current userid (username) logged on
- $ echo
  - Display a line of text
    - -n -> Do not output the trailing newline
    - -e -> enable interpretation of backslash escapes
      - ‘\n’ - print a newline
      - ‘\t’ - print a vertical tab
      - ‘\b’ - print one backspace

Cont.

- $ uname
  - Display various system information
    - -a -> print all information
- $ hostname
  - Displays the system’s hostname
- $ id
  - Prints real/effective user and group IDs
- $ logname
  - Print current user login name (similar to whoami)
- $ users
  - Prints all users currently logged in
- $ last

Cont.

- $ history
  - Print entire command history
    - $ history *n*
      - Print the last *n* commands
    - $ history -c
      - Clear history
    - $!!
      - Re-run last command
    - $!*n*
      - Run the numbered command
- Six virtual consoles (remember the exit command?)
  - Alt + [F1, F2, F3,… F6]
  - Where tty[1-7] come from

Practice

- Check out $ info root
  - Outstanding quick reference for most commands
- What is your current kernel version?
- What does the command ‘grep’ do?
- Play around with the commands introduced in this section and if you have a question–research it!
  - Google will be your best friend!

Basic Operations

- $ ls [option(s)] [path]
  - List directory contents
    - -a -> list everything (including dotfiles)
    - -l -> use long listing format
  - Default path is your current directory
  - Dotfiles
    - Files with a ‘.’ preceding their name
    - Hidden files
- $ cp [current file(s)] [copy to here]
  - Copy files and directories
    - -i -> interactive mode: prompt before overwrite

Cont.

- Wildcards
  - * - Match any number of characters
    - *.txt would return: ex1.txt, ex2.txt, HAL.txt, etc.
  - ? - Match any single character
    - b?t would return: bat, bit, etc. NOT bt or bait
  - [] - Match any characters inside of brackets
    - [fbc]all would return: fall, ball, call, etc.
  - [^] or [!] - Any characters *not* in brackets
    - [!t] would not return anything containing a t
  - {} - Expand into multiple arguments
    - test{1,2,3,4}.txt would return:
      - test1.txt, test2.txt, test3.txt, test4.txt

Cont.

- $ mv [current file(s)] [move to here]
  - Move/rename files
    - -i -> interactive mode: prompt before overwrite
- $ rm [file(s)]
  - Remove files or directories
    - -i -> interactive mode: prompt before removal
    - -r, -R -> recursive: remove directories and their contents

USE -r or -R WITH CAUTION

- $ ln [option(s)] [file to link] [new file]
  - Makes links between files
    - -s -> make a symbolic link instead of a hard link
    - -i -> interactive mode: prompt whether to remove destinations

Hard Links vs. Soft Links

- Hard links
  - An additional pointer to the same inode
  - A second name for a file
  - Editing the original file will change the linked file
  - Deleting the original file will not affect the linked file
- Symbolic links (Soft links)
  - New file and inode pointing to the original inode
  - Allows multiple files to associate with a single file
    - E.g. Microsoft Windows shortcuts
  - Separate files storing a file path
  - Editing/deleting the original file will affect the linked file

Practice

- Take time to review and understand Linux wildcards, they will save you loads of time in the future!
- List the contents of ‘/usr/share/’ and see if you can discover any symbolic links
- How many files do you have in your current directory?
  - Hint: it should be four!
- Try the following:
  - Copy /etc/passwd to your current directory (./)
  - Change the name of passwd to myPasswd.txt
  - Create a hardlink ‘pass’ for myPasswd.txt
  - Remove myPasswd.txt
- The following sections we will utilize these commands in a greater manner

Directory Operations

- $ pwd
  - Print absolute path of your current working directory
  - pwd -> Print Working Directory
- $ mkdir [directory name]
  - Creates directories
    - -m -> set permissions (more on this later)
    - -p -> make parent directories as needed
- $ rmdir [directory name]
  - Remove directories
    - -p -> remove directory and its ancestors

What about rm -r?

Cont.

- $ cd [directory path]
  - Change directories
    - $ cd or $ cd ~
      - Brings you to your home directory
    - $ cd .
      - Stay in current directory
    - $ cd ..
      - Change to parent directory
- Absolute path:
  - $ cd /var/log/
- Relative path (located in the current path):
  - $ cd news

Practice

- Move around the Linux operating system utilizing the previous commands you have learned
- Create, delete, and change into new directories
- Pay special attention to the ‘pwd’ command and try to change directories with both absolute and relative
  - Did you run into any problems?

File Creation

- $ touch [name of file(s)]
  - Changes file timestamps
  - Creates an empty file if one does not exist
- Output redirection ‘>’
  - $ echo “Hello World!” > first.txt
  - Warning: will overwrite same file names unless ‘>>’ is used, which appends the current file
- File editors
  - vi, emacs, nano (upgrade from pico)

Nano

- $ nano [file name]
- A small, free and friendly editor
  - ^G - help
  - ^O - write out (save file)
  - ^X - exit (ask to overwrite if changes occur)
  - ^C - cancel
  - ^C - curser position (line number, percentage, etc.)
  - ^Y - previous page
  - ^V - next page

Vi

- Powerful Linux text editor
- $ vi [file name]
- Command mode (default mode or by pressing ‘ESC’):
  - Navigate:
    - ‘h’ -> left
    - ‘j’ -> up
    - ‘k’ -> down
    - ‘l’ -> right
  - ‘d*n*d’ will delete the current and *n* lines below
  - ‘dd’ will delete the current only
  - ‘x’ delete current character
  - ‘u’ will perform an undo

Cont.

- Command mode (cont.):
  - ‘o’ will insert a newline below the current line
  - ‘y*n*y’ will copy the current and *n* lines below
  - ‘yy’ will copy the current only
  - ‘pp’ will paste a copied line below the current line
  - ‘/*n*’ will result in a search, where *n* is your string
  - ‘ZZ’ quit and write to (save) file
  - ‘:’ followed by the following
    - ‘:q!’ -> quit without saving
    - ‘:qw’ -> quit and write to file
- Insert mode (press ‘i’):
  - Works like a regular test editor in this mode

File Viewing

- $ cat
  - Prints file contents to the standard output (computer screen)
    - -n -> number all output lines
- $ less
  - Similar to the more command, but much *more* powerful
    - -N -> add numbers to each line
    - -m -> Percentage of file displayed
  - Controls
    - Enter -> moves one line at a time
    - Spacebar -> moves one screenful forward
    - b -> moves one screenful backwards
    - /*n* -> search function where *n* is a string
    - Up/down arrow keys -> one line up/down

Cont.

- $ head
  - Output the beginning of a file (default: first 10 lines)
    - -N -> print the first N lines
- $ tail
  - Output the end of a file (default: last 10 lines)
    - -N -> print the last N lines
- $ nl
  - Prints out the file contents while numbering the lines

More on Redirection

- Output redirection ‘>’
  - $ ls -l / > list.txt
  - Warning: will overwrite same file names unless ‘>>’ is used, which appends the current file
- Input redirection ‘<’
  - $ sort < list.txt
- Pipes ‘|’
  - Redirecting the standard output of one command as the standard input for another
  - $ ls -l /usr/share/ | less
  - Lists the contents within ‘/usr/share/’ in *long format* as input for the less command

Practice

- Practice with your newly learned commands
  - Be careful with ‘nl’ and ‘ln’!
- Try and construct a command utilizing all three of the redirection techniques taught
  - Hint: use the example already provided, but redirect the standard output to a file instead of the screen
- Get comfortable with VI
  - Although nano is more user friendly VI will only serve you more as you progress further into Linux
  - *Trust me*, I hated VI and refused to learn it at first
    - Now its my favorite editor!

Binary, Octal, & Hex…oh my

- Decimal notation (dec or base_{10})
  - 1999_{10}
- Binary notation (bin or base_{2})
  - $(0111\ 1100\ 1111)_2$
- Octal notation (oct or base_{8})
  - 3717_8
- Hexadecimal notation (hex or base_{16})
  - $0x7CF_{16}$

$1999 \Leftrightarrow 11111001111 \Leftrightarrow 3717_8 \Leftrightarrow 7CF_{16}$

Binary Conversion

- Reads from right to left
- Based off the power of 2

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
1	0	0	1	1	1	0	0
128	0	0	16	8	4	0	0

Total = 128 + 16 + 8 + 4

Total = 156

Decimal Notation

- You already think this way…without realizing it

10^3	10^2	10^1	10^0
1,000	100	10	1

1	9	9	9
1,000*1	100*9	10*9	1*9

Total = (1000*1) + (100*9) + (10*9) + (1*9)
Total = 1000 + 900 + 90 + 9
Total = 1999

Revisit Binary

2^3	2^2	2^1	2^0
8	4	2	1

12 in binary?

1	1	0	0
8	4	0	0

$$\text{Total} = (2^3 * 1) + (2^2 * 1) + (2^1 * 0) + (2^0 * 0)$$

$$\text{Total} = 8 + 4 + 0 + 0$$

$$\text{Total} = 8 + 4$$

$$\text{Total} = 12$$

Octal Notation

- 1999 ÷ 8 = 249 r 7
- 249 ÷ 8 = 31 r 1
- 31 ÷ 8 = 3 r 7
- 3 ÷ 8 = 0 r 3

$\boxed{7}$

$\boxed{1}$

$\boxed{7}$

$\boxed{3}$

$$1999_{10} = 3717_8$$

	$\boxed{3}$			$\boxed{7}$			$\boxed{1}$			$\boxed{7}$	
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0
0	1	1	1	1	1	0	0	1	1	1	1

$$3717_8 = (0111\ 1100\ 1111)_2$$

Hex Notation

- Each segment is made up of 1 byte (4 bits)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- Take 1999 and convert it to binary
  - $(0111\ 1100\ 1111)_2$
  - 1111 = 15 = F
  - 1100 = 12 = C
  - 0111 = 7 = 7

$$1999_{10} = 7CF_{16}$$

What you need to remember

Binary

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Octal

2^2	2^1	2^0
4	2	1

Hex

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Practice

- 1) Convert **156_{10}** to binary, octal, and hex
- 2) Convert **2771_8** to decimal, binary, and hex
- 3) Convert **0x7DE** to decimal, binary, and octal
- 4) convert the following into decimal (4 octets)

  0111 1111 . 0000 0000 . 0000 0000 . 0000 0001

- 5) convert the following to octal

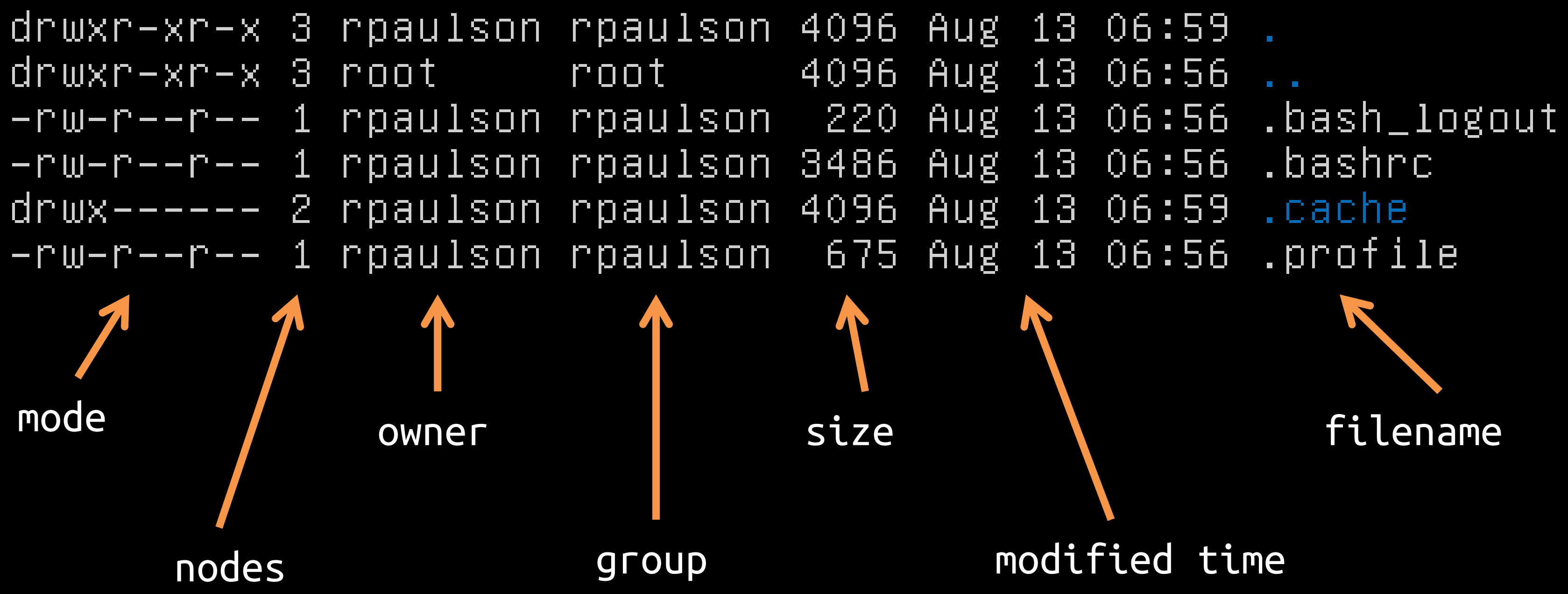
  110 111 100

File Properties

- $ stat
  - Display file or file system status
- $ wc
  - Print newline, word, and byte counts for each file
    - -l -> print newline counts only
    - -w -> print word counts only
    - -c -> print byte counts only
- $ du
  - Estimate file space usage
    - -a -> print count of all files, not only directories
    - -b -> print in bytes
    - -k -> print in kilobytes (default)
    - -m -> print in megabytes
    - -h -> print in human readable

chmod

- Change final mode bits (huh?)
- In English:
  - Changing the permissions for the file



```
drwxr-xr-x 3 rpaulson rpaulson 4096 Aug 13 06:59 .
drwxr-xr-x 3 root     root     4096 Aug 13 06:56 ..
-rw-r--r-- 1 rpaulson rpaulson  220 Aug 13 06:56 .bash_logout
-rw-r--r-- 1 rpaulson rpaulson 3486 Aug 13 06:56 .bashrc
drwx------ 2 rpaulson rpaulson 4096 Aug 13 06:59 .cache
-rw-r--r-- 1 rpaulson rpaulson  675 Aug 13 06:56 .profile
```

chmod – modes

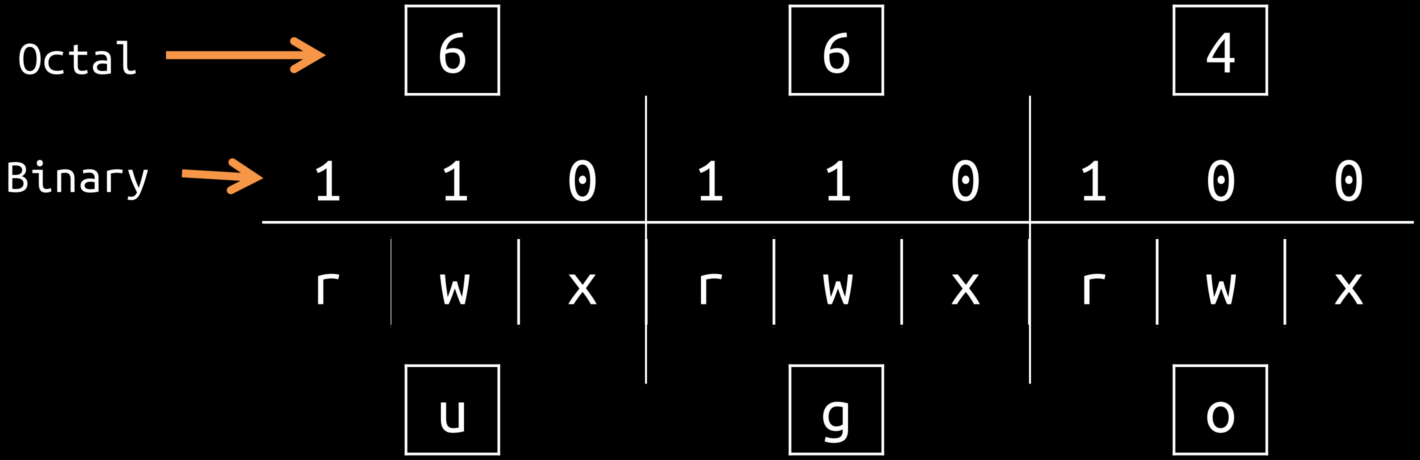
- Ten positions

1	2	3	4	5	6	7	8	9	10
-	r	w	x	r	w	x	r	w	x

- 1st position
  - ‘d’ (directory) or ‘-’ (file)
- 2-4 positions - user
  - (r)ead) (w)rite e(x)ecute
- 5-7 positions - group
- 8-10 positions - other

Cont.

- Each section represents its own octal value
- E.g. user
  - (r)ead = 100_2
  - (w)rite = 010_2
  - e(x)ecute = 001_2



Octal		6			6			4	
Binary	1	1	0	1	1	0	1	0	0
	r	w	x	r	w	x	r	w	x
		u			g			o	

Cont.

- `$ chmod 777 [file name]`
  - Would grant:
    - User: read, write, execute
    - Group: read, write, execute
    - Other: read, write, execute
- `$ chmod 640 [file name]`
  - Would grant:
    - User: read, write
    - Group: read
    - Other: nothing!

Practice

- Create a random file
  - Grant only the user read and execute permissions
  - Now remove execute permissions
  - Grant user and group only read/write permissions and other only read permissions
- Play around with ‘chmod’
- Before moving on, are you comfortable with the binary/octal system?
  - Review it if your not!

More chmod

- Another (easier) way to change file permissions
  - User = u
  - Group = g
  - Other = o
  - All = a
- Change file permissions by ‘+’, ‘-’, or ‘=’
- $ chmod ug+r,a-x [file(s)]
  - User/group read right; removes execute from everyone
- $ chmod a=r [file(s)]
  - Remove all other permissions and only grant read rights

More Practice

- Perform the same exercise as before, but use the new technique
- Which way do you prefer?
  - Make sure you understand both ways!

Cont.

- $ chown [option(s)] [user] [files]
  - Change file ownership
- $ chgrp [option(s)] [user] [files]
  - Change file group
- $ lsattr
  - List file attributes
    - -a -> list all
- $ chattr
  - Change file attributes; performed ‘+’, ‘-’, ‘=’
    - -a -> file can only be appended
    - -i -> file cannot be changed or deleted
    - -s -> secure deletion; data is written over with zeroes

Cont.

- $ umask
  - Sets the default mode when creating files/directories
    - i.e. if they are readable, writeable, executable
    - -S -> displays in English
  - Opposite of chmod
    - $ umask 777
      - Would result in no permissions
    - $ umask 000
      - Would result in full permissions
  - Take 777 and subtract umask for permission setting
    - E.g. $777_8 - 614_8 = 163_8$ (u=x, g=rw, o=wx)

Practice

- Create a file
  - $ `touch random`
- Change your default umask to 417 and create a new file
  - $ `umask 417 ; touch random2 ; ls -l`
  - Do you notice the difference?
- Change your umask back to the default (112)
- Change the owner of one of your files to root
  - Hmmm…did you figure out how?
- Play around with the various attribute settings

File Text Manipulation

- $ grep [pattern] [file]
  - One of the most useful commands in Linux
  - Prints lines matching a pattern
    - -v -> print lines that do not match the pattern
    - -l -> print only the names of the files matching the pattern
    - -i -> case insensitive matching
    - -w -> match only complete words
    - -r -> recursively search all files in a directory/subdirectory
- $ sort
  - Prints a files contents sorted (you already have seen this command!)
    - -n -> sort numerically

Cont.

- $ tr
  - Translates or deletes characters
- $ uniq
  - Filters out adjacent repeated lines in a file
    - -c -> count adjacent repeated lines
    - -u -> print unique lines only
    - -i -> ignore case
- $ cut
  - Cuts text from lines
    - -d -> delimiter
    - -f -> field to display
- $ paste

Practice

- Play with the grep command and become comfortable with it
  - This is one of the most used command in Linux!
- Experiment with the other command, do you know what they can be used for?
- `$ cat /etc/passwd | cut -d : -f 1,5 | sort`
  - Explain exactly what the command above is doing

File Comparison

- $ diff [options] [file1] [file2]
  - Compares two files and reports differences
- $ comm [options] [file1] [file2]
  - Compares two sorted files and outputs
    - 1: lines unique to file1
    - 2: lines unique to file2
    - 3: lines that appear in both file1 and file2
- $ cmp
  - Compares two files byte by byte
  - Returns first difference

Cont.

- `$ md5sum`
  - Generates an MD5 message digest
- `$ sha256sum`
  - Generates an SHA256 checksum

Practice

- Did you notice the hash at the beginning of this tutorial?
- Play around with the hash examples on the files you have created
  - Why are these important?
- Create two files that are only slightly different
  - Use the various compare commands and see how they work

File Location

- $ `find / -type f -name [file name] -print`
  - Search for files
  - The above is just one example of the find command
    - Review the man page for find–LOTS of good information!
  - Append ‘`2>/dev/null`’ to eliminate displaying errors to the standard output (computer screen!)
- $ `locate [file]`
  - Find files by name
  - May display to much information
- $ `whereis [command]`
  - Display the command and command man files locations

Cont.

- $ which [command]
  - Where command executable are stored
  - The program itself on the disk
- $ type [command]
  - Similar to ‘which’ command
  - Only available with bash shell

Practice

- Experiment with the various ways of searching for files and commands in Linux
  - Where is the python command located?
  - Where is the man page for cat located?
- Find the file ‘mail.log’?

File Packaging

- $ tar [options] [tar file] [files to tar]
  - Most common file packaging with Linux
  - Stands for “tape archive”
    - Previously used to backup files to a tape drive
  - Packs multiple files and directories into a single file
  - Note: tar is NOT a compression tool

Cont.

- $ `tar -czvf myTar.tar.gz *`
  - Main options
    - -c -> create archive
    - -t -> list the archive
    - -x -> extract contents from the archive
  - Additional options
    - -z -> use gzip compression
    - -j -> use bzip2 compression
    - -Z -> use Unix compression
    - -v -> verbose mode (print out additional information)
    - -f *file* -> read/write the archive from/to *file*
  - .tar and .gz are NOT mandatory for the file name
    - But highly recommended for obvious reasons!

File Compression

- $ gzip
  - GNU zip format (suffix .gz)
  - Only compresses single files (cannot pack files)
  - Original file is deleted after compression
- $ gunzip
  - Used to uncompress a gzip file
- $ bzip2
  - Burrows-Wheeler format (suffix .bz2)
  - Only compresses single files (cannot pack files)
  - Original file is deleted after compression
- $ bunzip2
  - Used to uncompress a bzip2 file

Cont.

- $ zip
  - Windows Zip format (suffix .zip)
  - Can compress multiple files into one file
- $ unzip
  - -r -> recursive compression
  - -l -> list .zip contents

Practice

- Tar all of your files within your home directory into one file using gzip compression
- Now un tar the file
- Play around with the various compression commands
  - Get a feel for how they work
  - What happens to your old files after compressing them?
- Create a zip file that is password protected

Filesystems

- $ df
  - Filesystem disk space usage
    - -T -> print filesystem type
    - -h -> print in human readable
- $ sudo mount [options] [device] [where to mount]
  - Mount a filesystem (usually in ‘/tmp/’ or ‘/mnt/’)
  - $ sudo fdisk -l
    - Displays all filesystem locations
  - -t [type] -> specify the type of filesystem to mount
- $ umount
  - Remove a mounted filesystem
  - $ umount /mnt/usb

Processes

- $ ps
  - Display a snapshot of the current running processes
    - -aux -> display ALL running processes in BSD syntax
    - -ejH -> print a process tree
- $ uptime
  - Shows how long the system has been running
- $ w
  - Display who is logged into a system and on what terminal
- $ top
  - Display a “real time” view of current processes

Cont.

- $ kill
  - Terminate a currently running process
  - For example:
    - $ top
    - Press ^z to send the process to the background
    - $ ps
    - $ kill -9 *PID*

Final Commands

- $ sleep [number(suffix)]
  - Delay the system for the specified timeframe
    - s -> seconds
    - m -> minutes
    - h -> hours
  - $ sleep 5s
    - Will suspend the system for five seconds
    - Seconds are default, so ‘s’ is not mandatory
- $ shutdown [option(s)] [time] [message]
  - -r -> reboot the system after shutdown
  - -c -> cancel shutdown
  - -P -> power off the system after shutdown

Practice

- Look at the various processes commands
- Play around with the kill command
  - Try it without the ‘-9’ option, what do you notice?
- Restart your Linux using the shutdown command
  - Is there another command that can do this?
  - Is there yet another command that can accomplish this?

Parting Thoughts

- Congratulations, you have completed the Intro to Linux Commands Tutorial!
- I hope you enjoyed this tutorial and learned along the way
  - If you encountered <u>any</u> errors or was confused by the presentation, no matter how small, please let me know!
- Continue researching and practicing what you learned, like anything perfecting your skills with Linux takes PRACTICE, PRACTICE, PRACTICE!

…and remember…**GOOGLE IS YOUR BEST FRIEND!**

Resources

- Linux Pocket Guide, 2nd Edition (Slide 19, etc.)
  - http://shop.oreilly.com/product/0636920023029.do
- Linux Bible, 8th Edition (Slide 19, etc.)
  - http://eu.wiley.com/WileyCDA/WileyTitle/productCd-111821854X.html
- Linux Man/Info Pages (Throughout)
  - http://www.linuxmanpages.com/
- UNIX Introduction (Slide 19, etc.)
  - http://www.ee.surrey.ac.uk/Teaching/Unix/unixintro.html
- Ask Ubuntu (Throughout)
  - http://askubuntu.com/
- (Slide 1) - Linux Image:
  - http://en.wikipedia.org/wiki/Tux
- (Slide 20) - Image of Linux Filesystem:
  - http://www.linuxtopia.org/online_books/suse_linux_guides/SLES10/suse_enterprise_linux_server_installation_admin/graphics/verzeichnisse_baum.png
- (Slide 26) - Linux Man/Info Pages:
  - http://askubuntu.com/questions/9325/what-is-the-difference-between-man-and-info-documentation
  - http://www.gnu.org/

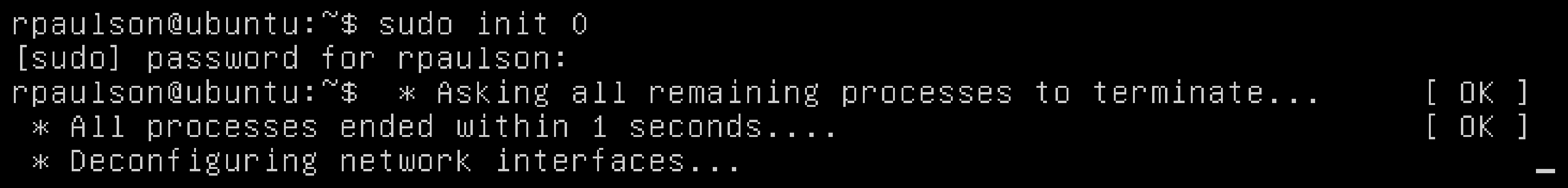
Cont.

  - http://manpages.ubuntu.com/manpages/trusty/en/man5/info.5.html
  - http://www.gnu.org/prep/standards/standards.html#Man-Pages
- (Slide 35) - Hard and Soft Links:
  - http://www.geekride.com/hard-link-vs-soft-link/
  - http://www.thegeekstuff.com/2012/01/linux-inodes/
  - http://www.linux.org/threads/intro-to-inodes.4130/
- (Slide 46) - Redirection:
  - http://linuxcommand.org/lts0060.php
- (Slide 48) - Binary Conversion:
  - http://www.asknumbers.com/BaseNumberConversion.aspx
- (Slide 49) - Decimal to Octal:
  - http://www.wikihow.com/Convert-from-Decimal-to-Octal
- (Slide 57) - File Permissions:
  - http://www.shopsite.com/help/12.0/en-US/sc/pro/index.htm?page=/help/12.0/en-US/install/install.permissions.html
- (Slide 81) - Mounting Filesystems:
  - http://askubuntu.com/questions/37767/how-to-access-a-usb-flash-drive-from-the-terminal-how-can-i-mount-a-flash-driv
  - https://help.ubuntu.com/community/Mount/USB
- And last, but not least, GOOGLE.COM!!!!

Version Updates

- v1
  - Initial release on 15 August 2014
  - Finally after four months in draft, 90 slides, and sitting six hours in the Dubrovnik airport waiting on my plane to arrive, I completed the first release!

Thank you…and Good-bye!



```
rpaulson@ubuntu:~$ sudo init 0
[sudo] password for rpaulson:
rpaulson@ubuntu:~$  * Asking all remaining processes to terminate...     [ OK ]
 * All processes ended within 1 seconds....                              [ OK ]
 * Deconfiguring network interfaces...
```

Until next time…WRS