# Learn Linux in 5 Days:

Day 1

Background and history of Linux

What Linux is and what distributions are

Place to practice your Linux Skills

How to install Linux

Ways to connect to Linux Systems.

Day 2  
Linux filesystem hierarchy

The command line interface

Getting Help, finding documentation

Root, the superuser

Linux directories and navigating

Day 3

How to read, set, and change file and directory permissions.

Sharing files and directories with others.

Finding files and directories

Viewing and editing files

Day 4

Delete, copy, move and rename operations

Input, Output and Redirection

Searching for data

Customizing your prompt and aliases

Transferring files over the network

Day 5

Processes and job control

Schedule and automate jobs

Switching users

Command line efficiency tips

Software package management

### Linux for Beginners Day 1:

What Linux is

Brief history of Linux

What a Linux distribution is

Reasons Linux is used

What is Linux:

An operating system

A Kernel – refers to just the Linux Kernel (core or heart of the operating system) it’s the layer between hardware and applications. intermediary between hardware and applications

History of Linux:

Created by Linux Torvalds in 1991

he wanted to run a Unix type operating system

First version was released in 1994

FOSS Free/Open Source Software)

Unix-Like

Linux Distribution:

Linux Kernel Plus Additional Software

Each distribution can have a different focus

Many distributions available to choose from

Server use, desktop, research and science, multimedia production etc

DistroWatch.com

RedHat Enterprise Linux (RHEL) not free

Fedora

Ubuntu

Debian

SuSe Linux Enterprise Server (SLES)

OpenSuSe

Why Linux?

Runs on many hardware platforms

can run on HP, IBM any platform

Small Footprint (older hardware or embedded

Stable, Reliable, Secure

Great for servers

good operating system for everyday desktop

Free – FOSS

Free Software Ecosystem

Choice/Free

### Linux Distributions:

### Linux is an Operating System

Linux OS = Linux Distribution with curated software and applications

Curated Collection of software that works best together

Distro/Flavors = Distribution

### Linux Kernel:

the kernel is the core = intermediary between applications and hardware

Linux Kernel + apps = Distro

Most Popular Distros: Redhat Enterprise Linux or Ubuntu

RedHat:  
popluar in:  
Banks, Airlines, Telecoms, Healthcare

Must pay for RedHat Enterprise / CENTOS Redhat that is Free

Ubuntu:  
Startups

Saas

Social Networks

Cloud Based

Fun just to check out?

Linux Mint

Debian

Mageia

Fedora

OpenSuse

ArchLinux

Slackware

personal use is your preference

Each distro is slightly different

same goals on all Linux distros

You can’t make a wrong choice

Linux Distro = kernel + software

RHEL and UBUNTU

CentOS = RHEL – branding/logos

Installing VirtualBox on Windows

Virtual Box

7zip

Unarchiver (mac)

Downloading of CentOS vdi

<https://www.linuxtrainingacademy.com/vdi/>

Username: adminuser  
Password: adminuser  
Root Password: adminuser

Downloading UBUNUT:

**username/password:**

adminuser/adminuser

**Root Password:**

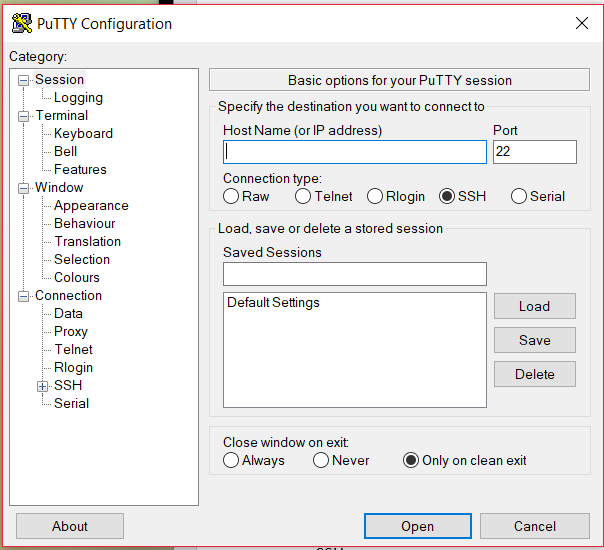
adminuser

Login with username and password on any system

Conecting over the Network:  
SSH

Windows – PuTTY

Mac – Terminal/ssh command line utility

Putty:  


## DAY 2:

Linux directory Structure

Location of operating system components

Application directory structures

### Linux directory structure:



Common Directories:  
most common:

* / “Root” the top of the file system hierarchy
* /bin Binaries and other executables programs
* /etc System configuration files
* /home Home directories
* /opt Optional or third party software
* /tmp Temporary space, typically cleared on reboot
* /usr User related programs
* /var Variable data, most notably log files
* /var/logs Where log files are typically found on Linux system
* /boot Files needed to boot the operating system
* /cdrom Mount point for CD-Roms
* /cgroup Control Grou hierarchy
* /dev Device files, typically controlled by the OS and the system administrators
* /export Shared file system
* /lib System Libraries
* /lib64 System Libraries, 64-bit
* /lost+found Used by the files system to store recovered files after a file system check has been performed
* /media Used to mount removable media like CD-ROMs
* /mnt Used to mount external file systems
* /proc Provides info about running processes
* /root Home directory for the root account
* /sbin System administration binaries
* /selinux Used to display information about SELinux
* /srv Contains data which is served by the system
* /srv/www Web Server files
* /srv/ftp FTP files
* /sys Used to display and sometimes configure the devices known to the Linux Kernel

#### Application Directory Structures:

* /usr/local/crashplan/bin
* /usr/local/crashplan/etc
* /usr/local/crashplan/lib
* /usr/local/crashplan/log

AVG software:

* /opt/avg/bin
* /opt/avg/etc
* /opt/avg/lib
* /opt/avg/log

Some applications follow:

* /etc/opt/myapp
* /opt/myapp/bin
* /opt/myapp/lib
* /var/opt/myapp

Sometimes applications get installed are not given their own directory structure (shared):

* /usr/local/bin/myapp
* /usr/local/etc/myapp.conf
* /usr/local/lib/libmyspp.so

Using Opt/ Organization Name or Company Name:

* /opt/acme
* /opt/acme/bin
* /opt/acme/etc

or

* /opt/web-team
* /opt/acme/web-team
* /usr/local/acme/web-team

Using CompanyName/AppName:

* /opt/google
* /opt/google/chrome
* /opt/google/earth

DAY 2: THE SHELL  
  
What is the shell?

How to access the shell

how to access super user account

The shell is the default user interface to a linux system

when you log in over the network the shell is activated…

Terminal gives you access to the shell.

UBUNTU click on the icon looking like UBUNTU Icon and type terminal

The shell is a program that accepts your commands and executes those commands

Also called the command line interpreter.

The SHELL can be more powerful

There will always be a command line

Server Distributions do not include a GUI

Desktop distributions have GUIs and SHELL/Terminal/CLI

You can rename 100 documents at once, where GUI you must do it to each file.

The Shell prompt:  
[adminuser@localhost ~]$

waits for you to do something

gives you information:  
username, and system attached to localhost

SUPER USER:  
Super User uses # and is the root account.

[root@localhost: ~]#

Root is all powerful (similar to administrator account on Windows)

Normal accounts can only do a subset of the things root can do.

Root access is typically restricted to system administrators

Root access may be required to install, start or stop and application

Day to day activities will be performed using a normal account.

Prompt:  
~ = home directory (current account)

~jason = /home/Jason

~root = /root

services have their own accounts:  
~ftp = /srv/ftp

### BASIC LINUX COMMANDS

Used often

Case Sensitive!!!

ls = lists all items in the current working directory

cd = changes the current directory, cd without any arguments takes you back to your home directory

pwd = Displays the present working directory

cat = Concatenates and displays files

echo = Displays arguments to the screen

man = Displays the online manual

exit = exists the shell or your current session

clear = Clears the screen

which = locate a command (full path)

tac = displays files in reverse order

type = The type command is used to find out if command is builtin or external binary file. It also indicates how it would be interpreted if used as a command name.

man = command itself is extremely easy to use. Its basic syntax is

man [option(s)] keyword(s)

ls Options:

-l = long listing format shows read/write

How to Navigate man pages

How the $PATH environment variable is used

what the which command does

how to ask commands for help

how to search man pages

Navigating Man Pages

Enter Move down one line

Space Move down one page

g Move to the top of the page

G Move to the bottom of the page

q Quit

Environmental Variables:

* Storage location that has a name and value
* Typically, uppercase
* Access the contents by executing

echo $VAR\_NAME

PATH

* An environment variable
* Controls the command search path
* Contains a list of directories

When you do a search, it searches the directories in the path for commands.

PATH: /usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/bin:/sbin:/home/adminuser/.local/bin:/home/adminuser/bin:

which cat:  
/usr/bin/cat

which tac:  
/usr/bin/tac

Starting to FISH:

* Look at the directories in $PATH
* Look at the files in each directory
* Use man to learn what the command does.

Get Help with –help or -h

* Add – help to a command to get help
* Try -h if -- help doesn’t work

ls – help

gzip -h

Searching Man Pages:

man -k SEARCH\_TERM

man -k calendar:

cal (l) - display a calendar

cal (lp) – print a calendar

difftime(3p) – compute the difference between two calendar time values.

Summary

* Man is used to display documentation
* $PATH controls your search path
* Learn the full path to commands with which
* Ask commands for help with -- help or -h
* Search man pages by using man -k

Next Steps  
  
Examine your $PATH environment variable

Perform an ls on each directory in your $PATH

Pick a few commands that look interesting

Use man to learn what the command does

Try using - - help and - h on a few commands

### Working with Directories

How to use directory shortcuts

How to execute commands outside of $PATH

How to create and remove directories

Directories:

containers for other files and directories

Provide a tree like structure

can be accessed by name or shortcut

Directory Shortcuts:  
. This directory

.. Parent directory

cd - Change to the previous directory

/ Directory separator (forward slash)

if you cd /home/Jason = /home/Jason/

echo $OLDPWD = holds the directory you were previously in

Executing Commands:  
$PATH determines command search path

You can specify a command with a full path

You can execute command not in $PATH

./command = Execute a command in this dir.

Creating and Removing Directories:  
mkdir[-p] directory – Create a directory -p is optional

rmdir[-p] directory – Remove empty directory -p is optional

rm -rf directory – Recursively removes directory.

removes it an all files everything in that directory and below that directory.

THERE IS NO UNDO! when you delete it is gone.

mkdir dir1/dir2/dir3 will not work unless dir1 and dir2 already exist

unless you use mkdir -p (make parents)

### Working with LS!

What the long listing format components are

How to see hidden files and directories

how to list files by type

how to list files sorted by time

how to handle spaces in files names

What a symbolic link is:

Decoding ls – l (long) output:  
$ ls -l

-rw-rw-r- - 1 jason users 10400 Sep 27 08:52 sales.data

Permissions -rw-rw-r- -

Number of links 1

owner name Jason

Group name users

Number of bytes in the file 10400

Last modification time Sep 27 08:52

File name sales.data

##### Listing all Files, including Hidden files:

Hidden files begin with a period (sometimes called “dot files”.

Hidden files are not displayed by default

To show hidden files with ls use ls -a

Command options can be combined

* ls -l -a is the same as ls -la and ls -al.

##### Listing files by types:

Us ls -F to reveal file types

/ Directory

@ Link

\* Executable

##### Symbolic Links: A link is a pointer to the actual file or directory

Use the link as if it were the file

A link can be used to create a shortcut

Use for long file or directory names

Use to indicate the current version of software

##### Listing Files by Time and in Reverse:

ls -t List files by time

ls -r list files in reverse order

ls -latrF Long listing including all files showing type and reversed sorted by time.

ls -R List files recursively (shows files in directories

ls -d List directory name, not contents

ls --color Colorize the output

The tree Command:  
Like ls -R but creates a visual output

tree -d List all directories only

tree -c Colorize output

Installing Tree in CentOS:  
# yum install tree -y

##### Working with Spaces in Names:

Just say no to spaces!

Alternatives:

Hyphens (-)

Underscores(\_)

CamelCase

Encapsulate the entire file name in quotes

Use a backslash(\) to escape spaces

Quiz