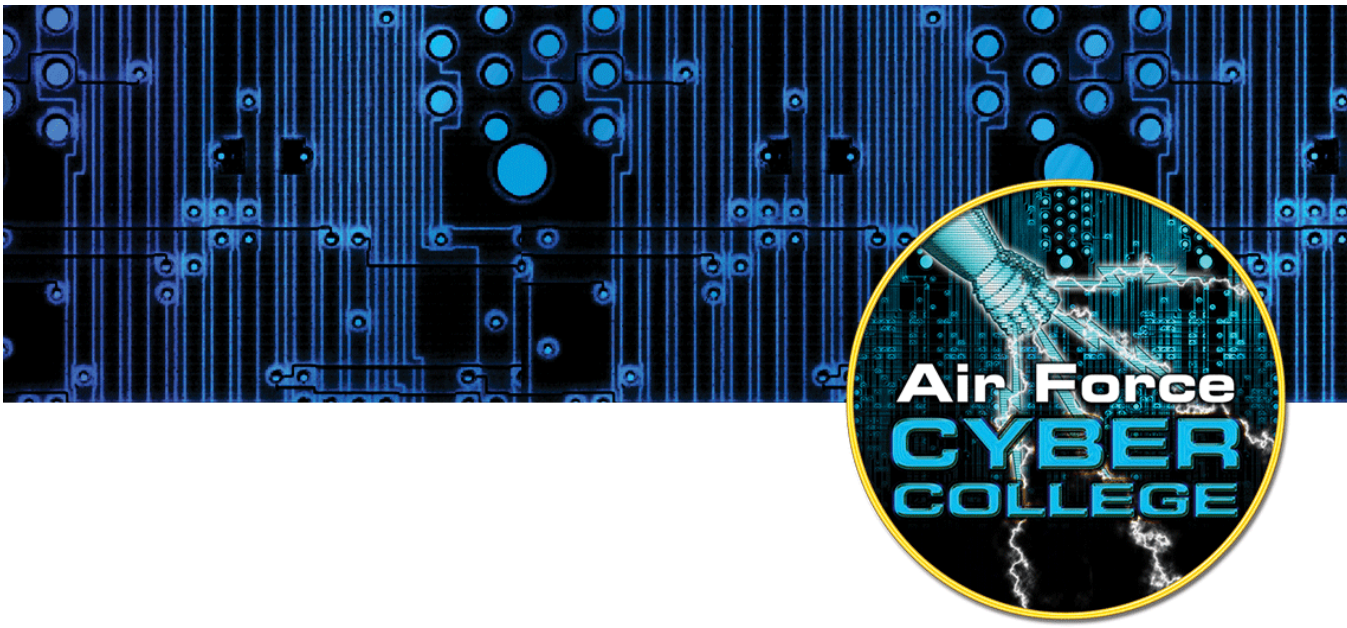


Hands-on Keyboard: Cyber Experiments for Strategists and Policy Makers

Review of the Linux File System and Linux Commands

1. Introduction

Becoming adept at using the Linux OS requires gaining familiarity with the Linux file system, file permissions, and a base set of Linux commands. In this activity, you will study how the Linux file system is organized and practice utilizing common Linux commands.

1.1. Objectives

- Describe the purpose of the /bin, /sbin, /etc, /var/log, /home, /proc, /root, /dev, /tmp, and /lib directories.
- Describe the purpose of the /etc/shadow and /etc/passwd files.
- Utilize a common set of Linux commands including ls, cat, and find.
- Understand and manipulate file permissions, including rwx, binary and octal formats.
- Change the group and owner of a file.

1.2. Materials

- Windows computer with access to an account with administrative rights

The Air Force Cyber College thanks the Advanced Cyber Engineering program at the Air Force Research Laboratory in Rome, NY, for providing the information to assist in educating the general Air Force on the technical aspects of cyberspace.

- VirtualBox
- Ubuntu OS .iso File

1.3. Assumptions

- The provided instructions were tested on an Ubuntu 15.10 image running on a Windows 8 physical machine. Instructions may vary for other OS.
- The student has administrative access to their system and possesses the right to install programs.
- The student's computer has Internet access.

2. Directories

2.1. /

The / directory or root directory is the mother of all Linux directories, containing all of the other directories and files. From a terminal users can type ***cd/*** to move to the root directory.

2.2. /home

The home directory stores user-specific directories and files. For instance, during the Ubuntu installation process, you created an account ACEIntern. Creating the account though the GUI automatically creates a directory named /home/aceintern to store your personal files. This directory is known as your home directory, as specified in the /etc/shadowfile (see section 2.6). From a terminal, users can type ***cd~*** to move to the home directory of the current user.

2.3. /root

The /root directory is the home directory for the root account. (Like /home/aceintern is the home directory for your account).

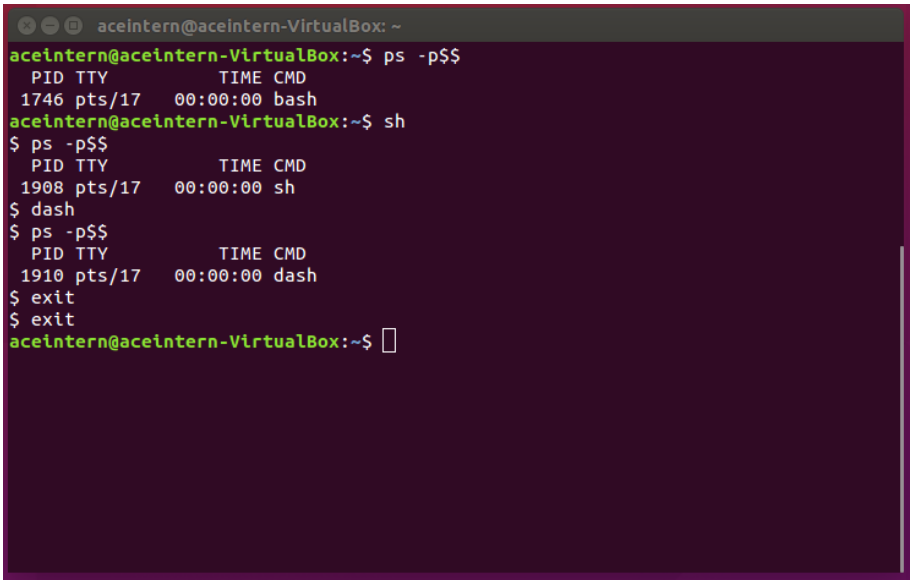
2.4. bin

2.4.1. Purpose

The term ***bin*** stands for binaries or executable files. The /bin directory typically holds commands executable by any user (e.g. ls, pwd, cat, rm).

2.4.2. Linux Shells

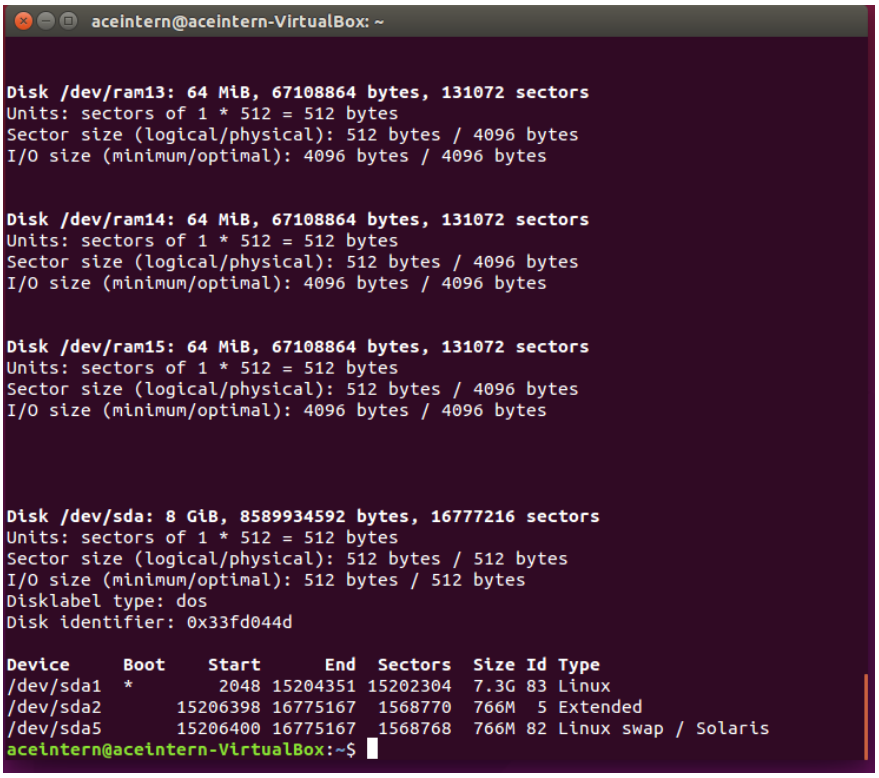
The /bin directory also contains shell binaries such as /bin/bash and /bin/sh. A shell is a special executable that starts an interactive, text-based window and enables users to enter commands. The shell executes the commands and displays the output back to the user. There are different shells (bash, sh, dash, and others) that offer slightly different functionality and sets of commands. By default, Ubuntu assigns users the /bin/bash shell. To determine your current shell in a terminal, type ***ps- p$$***. Try the commands in the screen shot below:



Note that the user in the screenshot example begins in a bash shell, starts a sh shell, and then starts a dash shell. To return to the bash shell, the user types exit (return to sh) and types exit again (return to bash).

2.5. /sbin

The term ***sbin*** stands for *system binaries* which are files executed by the system or privileged users. Try the ***fdisk*** command, which resides in /sbin — first as ***aceintern***, then with ***root*** privileges:



With ***aceintern*** account privileges, the command executes without an error but does not display any results. With root privileges, it lists information about the partition table.

2.6. /etc

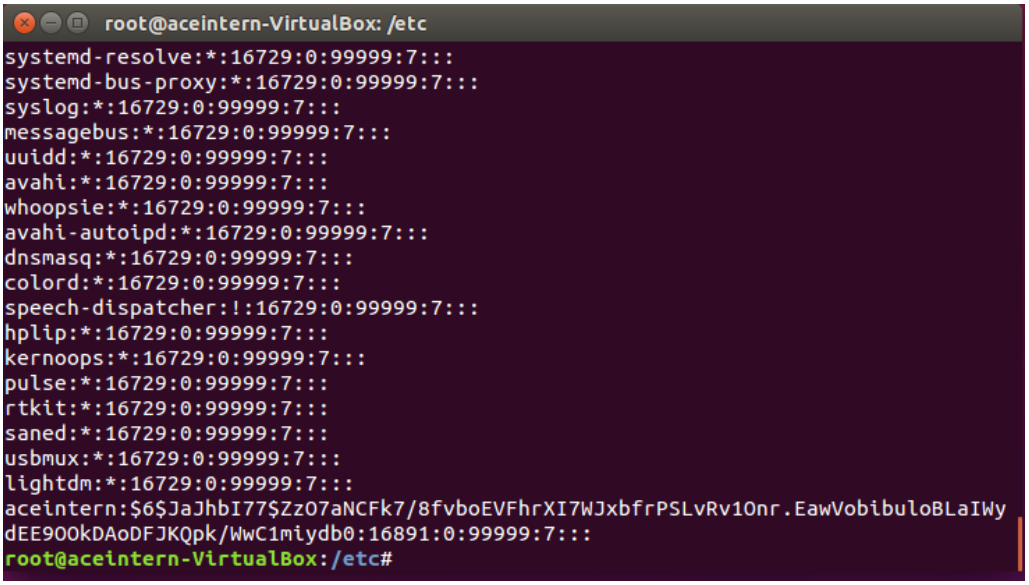
2.6.1. Introduction

The /etc directory holds configuration files for the operating system and applications. For example, the /etc/addusers.conf file specifies options for the adduser command. The adduser.conf file allows system adminstrators to specify the default home directory for new users. The default is /home/<user>, but it could be changed to another existing directory.

2.6.2. /etc/passwd and /etc/shadow

The /etc/shadow file did not exist on early Linux distributions. Originally, only root could access the /etc/passwd file, which stored user names, user configuration information, and passwords. However, when common programs such as ls running under reduced privileges needed access to user names, passwords were moved to the shadow file. Now almost any account can view the passwd file, but only root or administrators can view the shadow file. Open a terminal and execute the commands below to view the /etc/passwd and /etc/shadow files.

- Open a terminal
- ***cd /etc*** (move to the etc directory)
- ***ls*** (list files)
- ***cat passwd*** (view the /etc/passwd file)
- ***cat shadow*** (Error- regular user cannot view the etc/shadow file)
- ***sudo -s*** (Escalate to root privileges)
- ***cat /etc/shadow*** (view the /etc/shadow file)
  Sample /etc/shadow entry for the Kayla account:



- ***aceintern*** - username
- ***6...db0*** - SHA hash of password
- ***16891*** - Number of days since (Jan 1970) the password was last changed
- ***0*** - Number of days before password can be changed (0 means it can change at any time)
- ***99999*** - Number of days until required password change
- ***7*** - Number of days to warn user of password expiration
- ***:::*** - Number of days after which password is disabled, number of days since 1 Jan 1970 that account has been disabled, reserved field for future use

2.7. /lib

The /lib or library directory holds shared libraries used by the system and its applications. Library files are typically named <library name>.so.<version number>. This directory also holds kernel models, which if loaded add functionality to the base Linux kernel.

2.8. /dev

The /dev or device directory contains special files for devices such as the hard drive, printers, external drives, and ports.

2.9. /proc

The /proc directory holds operating system state information on running processes and hardware. It is a virtual file system stored in memory (volatile). Directories named after process IDs store information about processes.

2.10. /tmp

The /tmp or temporary directory is a place to store temporary files. The system typically deletes temporary files upon every boot-up.

2.11. /var/log

The /var/log directory holds the log files for the system and applications, including web servers, ftp servers, etc. Below see ***syslog*** (system log) output showing a restart, the system requesting an IP address, and some commands executed as the root account.

3. Twenty-Three Useful Linux Commands

This section covers a set of 23 Linux commands. ***Switches*** are command modifiers that often take the syntax of a dash followed by a letter (-h) or two dashes followed by a word (--help). Open a terminal and type the commands in order as listed.

3.1. List Files (ls)

- ***ls*** (List files)
- ***ls -a*** (List all files including hidden files (file name prefaced by a dot))
- ***ls -l*** (List files in long format)
- ***ls -al*** (List all files in long format)
- ***ls -il*** (List the index number or inode of each file in long format)
- ***ls --help*** (To get more information about the ls command)
- ***man ls*** (To get more information about the ls command)
- ***apropos ls*** (To get more information about the ls command)

3.2. Change directory (cd)

- ***cd/*** (Move to the root directory)
- ***cd/home/aceintern*** (move to the home directory of user aceintern)

- ***cd ..*** (Move up a directory (to /home))
- ***cd ..*** (Move up a directory (to /))
- ***cd~*** (Move to the home directory of the current user aceintern)

3.3. Present Working Directory (pwd)

- ***cd/*** (Move to the root directory)
- ***pwd*** (List the current working directory)
- ***cd~*** (Move to the home directory of the current user aceintern)
- ***pwd*** (List the current working directory)
- ***cd..*** (Move up a directory)
- ***pwd*** (List the current working directory)
- ***cd/home/aceintern*** (Move to the home directory of user aceintern)
- ***pwd*** (List the current working directory)

3.4. Create a File, Change the File Timestamp (touch)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***touch test.doc*** (Create an empty file)
- touch test1.doc test2.doc (Create two empty files)
- ***touch -d '1 May 2005 10:22' test.doc*** (Change the last access time on a file to the time specified)
- ***ls -al*** (Verify the time change on file test.doc)

3.5. Display the Contents of a File, Display Backwards (cat, tac)

- ***cd~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***touch test.doc*** (Create an empty file)
- ***cat test.doc*** (Since the file is empty, displays nothing)
- Move to the Desktop GUI and double-click on test.doc to open it in an editor
- Add some text and save the changes
- Return to the terminal
- ***cat test.doc*** (Displays contents of test.doc)
- ***tac text.doc*** (Displays contents of test.doc last line first and first line last)

3.6. Echo Back Text or Variable Values to the Terminal (echo)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***echo Hello World!*** (Echoes “Hello World!” back to you on the terminal)

- ***echo Hello World! > hello.txt*** (Writes "Hello World!" into a new file called hello.txt)
- ***echo $PATH*** (Echo the current user's path [lists the directories where the system automatically looks for commands])

3.7. Copy a File (cp)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***echo hi > star.doc*** (Create file named star.doc containing "hi")
- ***cat star.doc*** (Verify text "hi" is in star.doc)
- ***cp star.doc venus.doc*** (Copy star.doc to new file venus.doc)
- ***cat venus.doc*** (Verify text "hi" is in venus.doc)

3.8. Remove a File (rm)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***ls*** (List files)
- ***rm star.doc*** (Remove (delete) star.doc)
- ***rm test**** (Remove all files start with test. The * is a wild card.)

3.9. Make and Remove Directories (mkdir, rmdir, rm -rf)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***mkdir emptydir*** (Make an empty directory called emptydir)
- ***mkdir clowndir*** (Make a directory called clowndir)
- ***cd clowndir*** (Move to directory clowndir)
- ***touch bozo.txt*** (Create an empty file in clowndir called bozo.txt)
- ***echo big nose > curly.txt*** (Create a file curly.txt containing "big nose" in clowndir)
- ***cd ../*** (Move up one level to Desktop)
- ***rmdir emptydir*** (Delete the emptydir directory)
- ***rmdir clowndir*** (Try to remove clowndir, but get error due to contents)
- ***rm -rf clowndir*** (Recursively remove clowndir and its contents. Be very careful using the rm -rf command!)

3.10. Switch User (su)

- ***sudo -s*** (Switch to root access)
- ***su aceintern*** (Switch back to aceintern user account)

3.11. List Current User Name (whoami)

- ***whoami*** (List current user aceintern)
- ***sudo -s*** (Switch to root access)
- ***whoami*** (List current user root)
- ***su aceintern*** (Switch back to aceintern user account)

3.12. Text Editors (nano, gedit)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***nano nano.txt*** (Open a new file for writing in the terminal text editor nano)
- Write some text into nano.txt
- Press ***control-X*** to exit the program
- Type ***Y*** and hit ***enter*** to confirm exit and save
- ***gedit gedit.txt&*** (Open a new file for writing in the GUI text editor gedit. Note: The & opens a process not tied to the terminal.)
- Type some text, save the file and exit

3.13. List Where Program Is Installed (which)

- ***which ls*** (List location of ls command)
- ***which fdisk*** (List location of fdisk command)
- ***which perl*** (List location of perl)

3.14. List Current Users (w)

- ***w*** (List the users currently logged onto your system)

3.15. Search File System (find)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***find / ssh**** (Starting at the / directory, search for files with names that start with ssh. Recall that * is a wild card.)
- ***sudo -s*** (Switch to the root user)
- ***find / ssh**** (I typically search as root user to ensure I find all files)
- ***find ssh**** (Searches only the current directory, returns nothing)
- ***find *.txt*** (Searches for all .txt files in the current directory, returns gedit and nano files)

3.16. List Processes (ps)

- ***ps*** (List minimal information about running processes)

- ***ps -ef*** (List full information about processes running)

3.17. Create Archive (tar)

- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***tar -cvf arch.tar nano.txt gedit.txt*** (Put nano and gedit text files into an archive called arch.tar)
- Return to the Desktop GUI
- Right click on ***arch.tar*** and select ***Extract Here***
- Note that this extracts nano and gedit to a directory named arch

3.18. Search a File or Terminal Output (grep)

- ***ps -ef | grep gnome*** (Takes the output of ps and pipes it to grep for searching, return lines containing gnome)
- Use gedit (Section 3.12) to create a file named limerick.txt and paste the following text onto it (from poetry-online.org):
  There was an Old Person whose habits,
  Induced him to feed upon rabbits;
  When he'd eaten eighteen,
  He turned perfectly green,
  Upon which he relinquished those habits.
- Return to the terminal
- cat limerick.txt | grep rabbits
- cat limerick.txt |grep habits
- cat limerick.txt |grep e

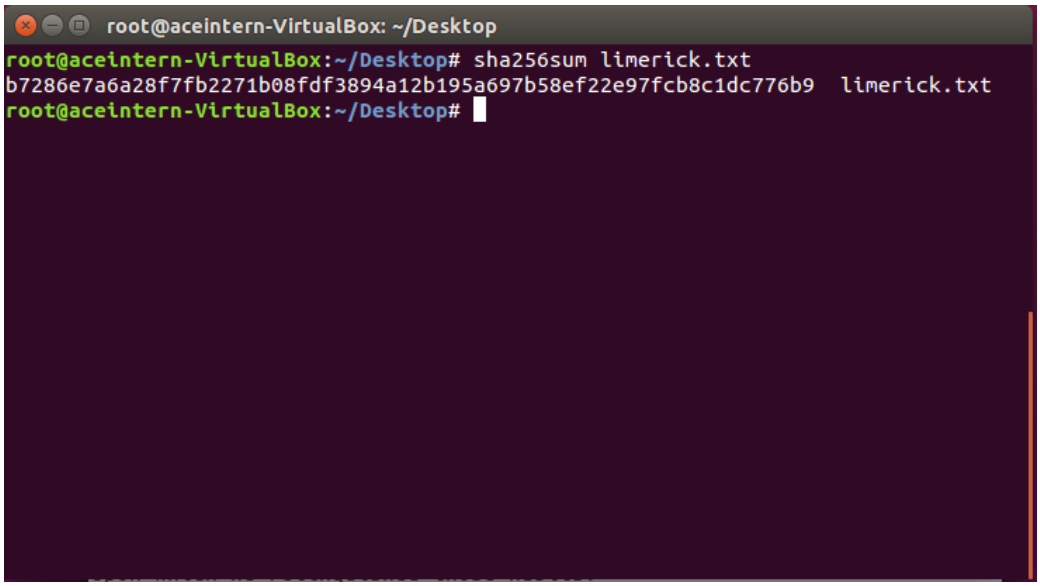
3.19. Display Text a Single Screen at a Time (more)

- Use gedit to edit limerick.txt
- Select all text and copy
- Paste the text into the document 10 times
- Return to the terminal
- more limerick.txt

3.20. Calculate MD5 Hash of FIle (md5sum)

- Sha256sum limerick.txt

- Note: The MD5 checksum of a file serves as a unique identifier for that file. If one bit is changed in a file, its MD5sum changes. The MD5sum of a file is useful for verifying files download correctly.
-

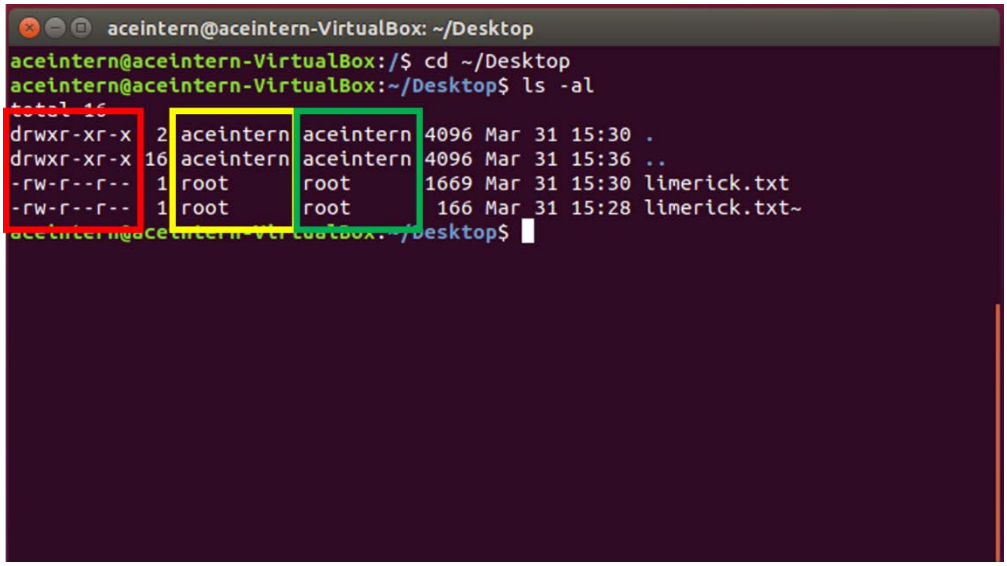
1. 

4. File Permissions

4.1. Introduction

Linux uses discretionary access control (DAC), meaning the owner of a resource dictates rights to the resource. This means the file creator (owner) and root have the ability to control who reads, changes, executes, or deletes the file.

- Open a terminal
- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***ls -al*** (Long listing format including file permissions)
-

2. 

- Circled in **red** are the file permissions, in **yellow** is the file's owner and in **green** is the file group. Nine bits of each file are used to store read, write, and execute file permissions.
    - **r** = Read permission (e.g. view with gedit)
    - **w** = Write permission (e.g. change with nano text editor and save)
    - **x** = Execute permission (e.g. execute a perl binary)
    - - = Unauthorized to read, write and/or execute

The listing specifies permissions for the file owner, members of the file's group, and all other users. See below a close-up and colorized version of the file permissions for gedit.txt.

- In **black**, the dash specifies gedit.txt is a file. A *d* in this position specifies a directory.
- In **red** are the permissions for the file's owner. The owner may read, write, or execute the file.
- In **orange** are the file permissions for the file's group. Members of the group may read or write to the file, but may not execute it.
- In **green** are the permissions for all other users. All other users may read the file but may not write to it or execute it.

Note: Set the default file permission for new files in /etc/profile with the umask command. For example, add umask 222 at the end of the profile to set the default file permission to 555.

4.2. RWX Examples

- File T permissions are -rwxrwxrwx (Anyone can read, write, or execute T.)
- File U permissions are drwxrwxrwx (U is a directory.)
- File V permissions are -r--r--r-- (Anyone can read V, no one can write to or execute.)
- File W permissions are -rwx------ (Only the owner can read, write, or execute.)
- File X permissions are ---------- (Root can still read but not write or execute. No one else can read, write, or execute.)

4.3. Numeric Representation of File Permissions

Computers use bits to store file permissions. rwxrwxrwx is stored as 1s and 0s, in this case 111111111.

Recall octal (same as decimal from 0–7) / binary equivalents:

Octal	Binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

4.4. Numeric Representation Examples

Letter version			Binary version			Octal Version		
Owner	Group	All Other Users	Owner	Group	All Other Users	Owner	Group	All Other Users
rwx	rwx	rwx	111	111	111	7	7	7

r w-	r- -	- w-	1 10	1 00	0 10	6	4	2
r wx	-- x	r w-	1 11	0 01	1 10	7	1	6
r- x	- wx	-- -	1 01	0 11	0 00	5	3	0

4.5. Change RWX File Permissions (chmod)

- Open a terminal
- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***ls -al*** (Long listing format including file permissions)
- ***sudo -s*** (Switch to root)
- ***chmod 777 limerick.txt*** (Change file permissions so all users can rwx)
- ***ls -al*** (Long listing format including file permissions)
- ***chmod 000 limerick.txt*** (Change file permissions)
- ***gedit limerick.txt*** (Try to read/write file with root privileges)
    - Can you read the file with root privileges?
    - Can you edit the file with root privileges?
- ***chmod 755 limerick.txt*** (Change file permissions)
- ***ls -al*** (Long listing format including file permissions)

4.6. Change File Owner (chown)

- Open a terminal
- ***su aceintern*** (Ensure aceintern privileges)
- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***touch chown.txt*** (Create file owned by aceintern)
- ***ls -al*** (Long listing format including owner)
- ***sudo -s*** (Escalate root privileges)
- ***chmod 770 chown.txt*** (Change file permissions)
- ***chown root chown.txt*** (Change file owner to root)
- ***ls -al*** (Long listing format including owner)
- ***exit*** (Switch to aceintern privileges)

- gedit chown.txt
    - Can you read/write chown.txt? Why or why not?

4.7. Change File Group (chgrp)

- Open a terminal
- ***cd ~/Desktop*** (Move to the Desktop of the current user aceintern)
- ***ls -al*** (Long listing format including group)
- ***sudo -s*** (Escalate to root privileges)
- ***chgrp root chown.txt*** (Change file group to root)
- ***ls -al*** (Long listing format including group)
- ***su aceintern*** (Switch to aceintern privileges)
- gedit chown.txt
    - Can you read/write to chown.txt? Why or why not?

5. Review Exercises

5.1. Compare and contrast the /bin and /sbin directories.

5.2. Assume you have root privileges. Write the terminal commands to change the owner and group of a file named *file.txt* to *galactica* (owner) and *starbuck* (group).

5.3. Write the binary and octal representations of the following file permissions:

a. -rwxr-x-w-

3.

b. -rw---xr—

4.

5.

5.4. Compare and contrast the information stored by the /etc/shadow and /etc/passwd files.

5.5. Questions a–e refer to the screenshot:

```
root@ace:~/Desktop# ls -al
total 8
drwxr-xr-x  2 ace ace  4096 2009-08-31 11:12 .
drwxr-xr-x 30 ace ace  4096 2009-08-31 10:56 ..
-rwxr----x  1 ace root    0 2009-08-31 11:12 superman.txt
root@ace:~/Desktop#
```

a) Who is the owner of superman.txt?

b) To what group does superman.txt belong?

c) List the owner privileges for superman.txt.

d) List the group privileges for superman.txt.

e) List all other user privileges for superman.txt.