Complete Linux Training Course to Get Your Dream IT Job 2023 -Udemy

21st January 2023

INSTRUCTOR: Imran Afzal

https://hdfcbank.udemy.com/course/complete-linux-training-course-to-get-your-dream-it-job/

horizontal line

# Section 1:

Date - 21st Jan 2023

## Linux vs Unix

* Linux is free and open source.
* Linux can be installed on many hardware and has many flavours.

24th Jan 2023

**Linux Distributions**

* Redhat - not open source but has technical support.
* CentOS - Exactly same as Redhat. Also managed by redhat, but no Technical support. Completely free.
* Fedora - free
* Suse - Highly customisable.
* Debian.
* Ubuntu.

Date - 6th Feb 2023

**Linux vs Windows**

* Linux is free, but not user friendly, has enterprise softwares only, reliable(does not require many reboots), very secure, has multi tasking ability,and is open source.

# Section 3:

* Root is the most powerful users, it can delete the OS as well
* Linux is mostly CLI and not GUI like windows
* Linux Kernel is not an operating system. It is a small software with any Linux operating system that takes commands from you, from users and passes them through the system, hardware and peripherals.
* Linux is very flexible post the learning curve.

**Access to Linux system -**

1. By console i.e. connect a hdmi cable to the machine and get the console of the linux system
2. Remote - We need a putty client to access the remote linux machine using the IP address.

**Filesystem -**

* Filesystems are used to organize and structure the files.
* Linux file systems - ext3,ext4,xfs
* Windows file systems - NTFS,FAT , etc.

Date - 7th Feb 2023

**Filesystem structure -**

1. /boot - Contains file used by boot loader (grub.cfg)
2. /root - root user home directory. It is not same as /
3. /dev - System devices created as files (eg. disk, cdrom, speakers,keyboard)
4. /etc - Configuration files for all system applications like DNS, NTP,,.....[always take backup before changing]
5. /bin ->(mapped to) /usr/bin - Everyday user commands like cd , pwd , are stored here.
6. /sbin ->(mapped to) /usr/sbin - System/filesystem commands used to extend or configure filesystems.
7. /opt - Optional add-on applications (not part of OS apps) . eg - oracle , SAP.
8. /proc - Creates files of Running processes when they are running (only exists in memory)
9. /lib -> usr/lib - C programming library files needed by commands and apps. Eg. ls command has a code associated with it to what action needs to be performed. “strace -e open pwd” will give the c libraries used for running a linux command
10. /tmp - directory for temporary files.
11. /home - Directory for user
12. /var - System Logs and sometimes application logs as well
13. /run -To store files of System daemons[processes] that start very early (eg. systemd and udev) to store temporary runtime files like PID files (process IDs)
14. /mnt - To Mount external filesystem (eg.NFS)
15. /media - For CD roms.

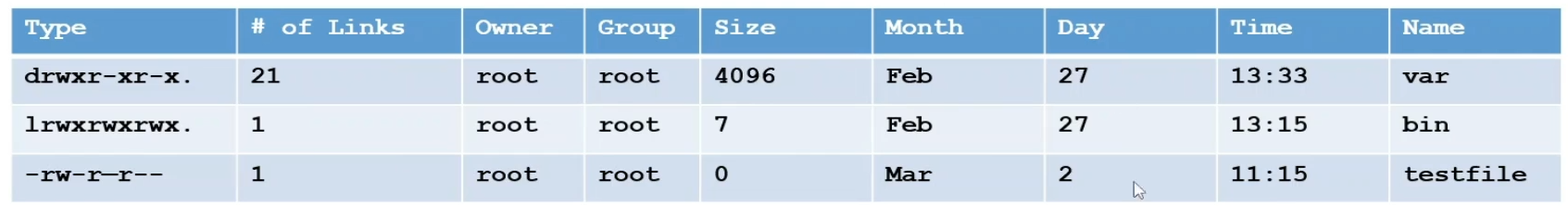
**Filesystem Commands -**

1. cd - change directory

cd /home or cd .. , etc ..

1. pwd - present working directory
2. ls - list all the directory and files

**Linux file and Directory properties -**



1. If it starts with - in -rw-r- - r- -, it is a file and

if it starts with d like drw-r- -r- - , then it is a directory and

if it starts with l like lrw-r- -r- - , then it is a link to another file

1. No of links - Second column is the number of hard links to the file for a directory. The number of hard links is the number of immediate subdirectories it has, plus its parent directory and itself.
2. Owner
3. Group
4. Size
5. Date and time
6. Name

Date - 9th Feb 2023

**Linux File types -**

| File Symbol | Meaning |
| --- | --- |
| - | Regular file |
| d | Directory |
| l | link |
| c | Special file or device file |
| s | socket |
| p | Named Pipe |
| b | Block device |

**Root -**

1. root account - most powerful and privileged account.
2. Root directory - the very first or main directory i.e /
3. Root home directory - /root is the home directory of root user.

**Change Password -**

passwd [userid]

If you run only passwd without any userid, it will change password for root.

**File system paths -**

1. Absolute path - it begins with “/” eg. cd /var/log
2. Relative path - it is relative to your current working directory. Eg. cd /log

**Creating Files and Directories -**

1. Creating files
   1. touch
   2. cp
   3. vi

**Writing to files-**

using echo - echo “abcd” > testfile.txt

Append something to existing file echo “continued” >> testfile.txt

1. Creating Directories
   1. mkdir

**Copying Directories -**

1. cp -R <source\_folder> <destination folder>

cp -r config /tmp - this will copy the config folder to /tmp with the same name.

cp -r config /tmp/config-bak - this will copy the config folder to /tmp and rename it as config-bak.

**Finding files and directories -**

1. find -> find . -name ”kramer” - this finds kramer in the current directory (.)
2. locate -> locate “kramer”

Date - 13th Feb 2023

**Difference between find and locate -**

locate uses a pre-built database(can be seen as cache) while find iterates over the filesystem. Hence, find is more accurate as it not limited to the knowledge of the updated database, but locate is faster.

To update the local database for locate command, run ‘updatedb’. System updates the db at some interval.

**Wildcards-**

Characters used as a substitute for character or class of characters in a search

1. \* - represents zero or more characters

-example -> ls -ltr abcd**\***

1. ? - represents a single character

-example -> find . -name abcd**?**-xyz

1. {} - represents a range of character

- example -> touch abcd**{**1..9**}**-xyz #creates 9 files from 1 to 9

1. [] - represents a set of characters, will be true if any one character matches -

- example -> ls -l \***[**cd**]**\* #gives all files containing either c or d

1. \ - used to represent an escape character - There are some special characters in Unix for ex :" $ & \ ~" .Putting a \ before a special character inverts the special meaning of a character. so it is getting a simple ascii text

-example ->

[karan@localhost ~]$ echo ~

/home/karan

[karan@localhost ~]$ echo \~

~

Also can be used to type scripts on multiple lines.

1. ^ - used to represent the beginning of the line

-example -> list only all the directories -

[karan@localhost /]$ **ls -ltr | grep ^d**

drwxr-xr-x. 2 root root 6 Apr 11 2018 srv

drwxr-xr-x. 2 root root 6 Apr 11 2018 opt

drwxr-xr-x. 2 root root 6 Apr 11 2018 mnt

drwxr-xr-x. 2 root root 6 Apr 11 2018 media

drwxr-xr-x. 13 root root 155 Feb 10 00:24 usr

drwxr-xr-x. 3 root root 19 Feb 10 00:29 home

1. $ - used to represent end of line

One of the example -

echo $?

- to get exit code status of previous command -If status shows **'0'** then command was successfully executed and if shows **'1'** then command was a failure.

**Soft Links and Hard Links-**

* inode - pointer or number of a file on hard disk.
* Soft Link - link will be removed when the source file is removed or renamed

ln -s

* Hard link - deleting , renaming or moving the original file will not affect the hard link.

ln

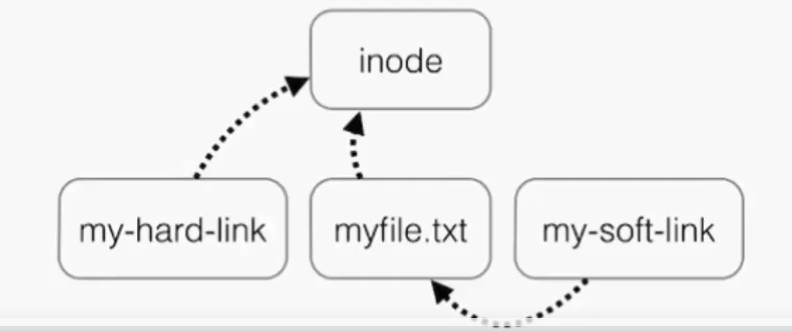
Note:

1. we cannot create soft and hard links with the same name in the same directory as the original file.

2.Soft links does not have same inode number as the source file

3.Hard links have the same inode number as the source file

4.Hard links works within the same partition only.



Date - 14th Feb 2023

Section 4

**Linux command syntax-**

command option(s) argument(s)

-example -> ls -ltr

-example with argument-> ls -l file1.txt

**File and Directory permissions-**

3 types of permission -

* r -> read
* w -> write
* x -> execute

Each permission can be controlled at 3 levels -

u - user → yourself

g - group → can be people in the same project

o - other → everyone on the system.

To change permissions use command -> chmod

-example

chmod g-r file.txt [for the group ‘-’ remove the permission to read]

chmod u+rw file.txt [for the user ‘+’ add the permission to read and write]

Directory has executable permission as it may contain some script which needs executable permission.

Permission using numeric mode

| Number | Symbol |
| --- | --- |
| 0 | * - - |
| 1 | * - x |
| 2 | * w - |
| 3 | * w x |
| 4 | r - - |
| 5 | r - x |
| 6 | r w - |
| 7 | r w x |

chmod 746 file.txt [7 is for user, 4 is for group , 6 is for others]

**File Ownership -**

There are 2 owners of a file -

User and group

chown - change ownership of the file

chgrp - change group ownership of the file

-R option means recursively change ownership in the directory

Date - 16th Feb 2023

**Access Control List (ACL) -**

ACL provides an additional and more flexible permission mechanism for file systems.

In a scenario where root has created a file, but you don’t want to make the other user part of the group root , and also you don’t want to change the permission of the file for other users[as it will allow every other user to access it]. In this case we use ACL.

Commands -

1. To add permission for user -> setfacl -m u:user:rwx /path/to/file
2. To add permission for group -> setfacl -m g:group /path/to/file
3. To allow all files or directories to inherit ACL entries from the directory it is within → setfacl - rm “entry” /path/to/file
4. To remove a specific entry → setfacl -x u:user /path/to/file
5. To remove all entries → setfacl -b path/to/file [for all users]
6. To get permissions of a file → getfacl path/to/file

As we assign ACL permissions , it adds + sign to the permissions in ls -ltr, indicating the file has ACL permissions.

Setting w[write] permission with ACL does not allow that assigned user to remove a file by other that the owner.

**Help Commands-**

1. whatis command
2. command –help
3. man command

**Tab completion and up arrow key-**

1. Hitting tab key completes the available commands,files or directories.
2. Up arrow key gets the last run command.

**Adding text to files(Output redirects)-**

[stdout] standard output has file descriptor number 1

1. Using vi editor
2. Using redirect command output > , >> [this can add the output of a command directly to a file] - example → ls -ltr > listofdir.txt
3. echo > [write to the file] or >> [append to the file]

**To read file contents and use it in a command(Input Redirects)-**

[stdin] standard input has file descriptor number 0

‘<’ is used for input redirect.

Example - file listings has a file name which we want to display the ls -ltr output.

ls -ltr < listings

cat < listings

**To redirect only errors to a certain file-**

[stderr] standard error has file descriptor number 2

If your are in user - karan and you try to access /root , it gives error, but we want only the error to be stored in a file.

Example → ls -ltr /root **2>** errorfile

**tee command [store and view the output of a command at the same time]**-

example →

echo "printing and saving output at the same time" | tee teecommandout.txt

If you want to append the output

echo "appending and printing and saving output at the same time" | tee -a teecommandout.txt

**Pipes ( | ) -**

Pipe is used by shell to connect the output of one command directly to the input of other command

command1 [args] | command2 [args]

Example → ls -ltr | more

**File maintenance commands-**

1. cp
2. rm
3. mv
4. mkdir
5. rmdir or rm -r
6. chgrp
7. chown

**File display commands-**

1. cat
2. more
3. less
4. head
5. tail

**Filters / Text processor commands-**

1. cut

cut is a command line utility that allows you to cut parts of lines from specified files or piped data and print the result to standard output. It can be used to cut parts of a line by delimiter, byte position, and character

• cut filename = Does not work

• cut --version = Check version

• cut –c1 filename = List one character

• cut –c1,2,4 = Pick and chose character

• cut –c1-3 filename = List range of characters

• cut –c1-3,6-8 filename = List specific range of characters

• cut –b1-3 filename = List by byte size

• cut -d: -f 6 /etc/passwd = List first 6th column separated by :

• cut -d: -f 6-7 /etc/passwd = List first 6 and 7th column separated by :

• ls –l | cut –c2-4 = Only print user permissions of files/dir

1. awk

awk is a utility/language designed for data extraction. Most of the time it is used to extract fields from a file or from an output

• awk --version = Check version

• awk ‘{print $1}’ file = List 1st field from a file

• ls –l | awk ‘{print $1,$3}’ = List 1 and 3rd field of ls –l output

• ls –l | awk ‘{print $NF}’ = Last field of the output

• awk '/Jerry/ {print}' file = Search for a specific word

• awk -F: '{print $1}' /etc/passwd = Ouput only 1st field of /etc/passwd

• echo "Hello Tom" | awk '{$2="Adam"; print $0}‘ = Replace words field words

• cat file | awk '{$2=“Imran"; print $0}‘ = Replace words field words

• awk 'length($0) > 15‘ file = Get lines that have more than 15 byte size

• ls -l | awk '{if($9 == "seinfeld") print $0;}‘ = Get the field matching seinfeld in /home/iafzal

• ls -l | awk '{print NF}‘ = Number of fields.

1. grep and egrep

The grep command which stands for “global regular expression print,” processes text line by line and prints any lines which match a specified pattern

• grep –-version OR grep --help = Check version or help

• grep keyword file = Search for a keyword from a file

• grep –c keyword file = Search for a keyword and count

• grep –i KEYword file = Search for a keyword ignore case-sensitive

• grep –n keyword file = Display the matched lines and their line numbers

• grep –v keyword file = Display everything but keyword

• grep keyword file | awk ‘{print $1}’ = Search for a keyword and then only give the 1st field

• ls –l | grep Desktop = Search for a keyword and then only give the 1st field

• egrep –i “keyword1|keyword2” file = Search for 2 keywords.

1. sort and uniq

What are sort and uniq commands?

• Sort command sorts in alphabetical order

• Uniq command filters out the repeated or duplicate lines

• sort –-version OR sort --help = Check version or help

• sort file = Sorts file in alphabetical order

• sort –r file = Sort in reverse alphabetical order

• sort –k2 file = Sort by field number

• uniq file = Removes duplicates

• sort file | uniq = Always sort first before using uniq their line numbers

• sort file | uniq –c = Sort first then uniq and list count

• sort file | uniq –d = Only show repeated lines.

1. wc

The command reads either standard input or a list of files and generates: newline count, word count, and byte count

• wc –-version OR wc --help = Check version or help

• wc file = Check file line count, word count and byte count

• wc –l file = Get the number of lines in a file

• wc –w file = Get the number of words in a file

• wc –b file = Get the number of bytes in a file

• wc DIRECTORY = NOT allowed

• ls –l | wc -l = Number of files

• grep keyword | wc -l = Number of keyword lines.

**Commands for comparing files [diff and cmp]-**

1. diff → [line by line]
2. cmp → [bytes by bytes]

**Compress and uncompress Files-**

1. tar → does not compress heavily

compress eg. → tar cvf karan.tar .

decompress eg. → tar xvf karan.tar

1. gzip → compresses the data heavily
2. gzip -d or gunzip → decompress the gzip file

**Truncate file size -**

It chops the file and chopped data is lost

truncate –s 10 filename

**Combining and Splitting Files-**

Multiple files can be combined into one and

• cat file1 file2 file3 > file4

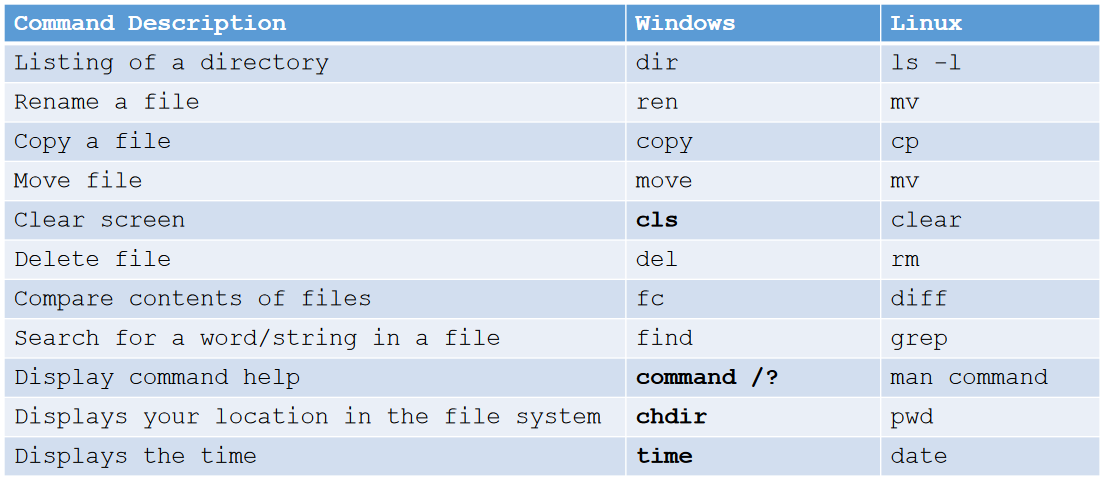
One file can be split into multiple files

• split file4

• e.g. split –l 300 file.txt childfile

Split file.txt into 300 lines per file and output to childfileaa, childfileab and childfileac

**Linux vs Windows commands-**



**Finding System Information-**

1. uname –a

Sometimes it is required to quickly determine details like kernel name, version, hostname, etc of the Linux box you are using.

1. cat /etc/redhat-release

This file provides information about your system distribution and its version

You can also run /etc/\*rel\* for systems that are not on CentOS or Redhat

1. dmidecode

dmidecode is a tool for dumping a computer's DMI (some say SMBIOS) table contents in a human-readable format. This table contains a description of the system's hardware components, as well as other useful pieces of information such as serial numbers and BIOS revision. Thanks to this table, you can retrieve this information without having to probe for the actual hardware.