

Linux

Linux is an OS, but also a kernel, can be used as a server or a desktop.

Linux Distributions:
① Linux kernel + additional Software
② Each distribution have a different focus.

i.e: Ubuntu, Debian, Fedora, Red hat, SuSE, etc.

Redhat = popular in banks, airlines, telecoms, healthcare,
Need to pay for a licence.

CentOS : free

Ubuntu : popular in startups, SaaS, Social Networks, Cloud Based

Connecting over the Network:

- ① ssh
- ② windows putty, then ssh.

the Linux Directory Structure

Common Directories:

/ : root, the top of file system hierarchy.

/bin : binary files and other executable programs

/etc : System configuration files.

/home : home directory

/opt : optional or third party software.

/tmp : temporary files.

/usr : user related programs.

/var : variable data, log files.

/boot : files needed to boot the OS.

/dev : device files, controlled by the OS and system administrator.

/export : shared file system.

/lib, /lib64 : system libraries, 64 bit

The Shell

Shell is the default interface to Linux, also called a command line interpreter.

Command Line Interface vs a GUI

- ① The command line is more powerful.
- ② Server distributions do not include GUIs.
- ③ Desktop distributions have GUIs and CLIs

Root, the superuser, (system administrators).

is all powerful.

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

Root access is required to install/start, or stop an application.

~ expansion (tilde) : can be used to represent the home directory of any active current user.

For example : for root user, `cd ~` would be root (/).

for user jason, `cd ~` would be `./home/jason`.

Basic Linux Command.

ls, cd, pwd (display the present working directory), exit, clear.

cat (concatenates and displays files).

echo (display arguments to the screen)

man . (display the online manual/documentation).

ls option :

`ls -l` (long listing format).

`ls -l --author`

`ls --sort` {
none
time
size
extension
version}

`ls -a / --all` : do not ignore entries starting with '.'

`ls -A / --almost-all` : do not list implied '.' and '..'

Environmental Variables :

- ① storage location that has a name and a value.
- ② typically uppercase.
- ③ Access the content by executing: \$ VAR NAME.

PATH : is an environment variable , controls the command search path, contains a list of directories. i.e: \$PATH

which : find the location of a program.

i.e: which python

Searching Man Page:

man -k search-term

Ask command for help: command_name --help/-h

Working with Directories.

Directory Shortcuts:

- the current directory
- .. the parent directory

cd - : change to the previous directory

•/command : execute command in this dir.

Creating and Removing Directories:

mkdir [P] → Create any necessary parent directory.

mkdir directory_name : Create a directory

rmdir directory_name : Remove a directory

rm -rf directory_name : recursively removes directory

Listing Files./ Understanding ls output

the format of output of "ls -l" is:

permissions number-of-links owner-name Group-name number-of-bytes time file-name
Last modified

hidden files begin with '.', sometimes called "dot files"

hidden files are not displayed by default,

The optional commands can be combined, and the order doesn't matter.

i.e: ls -l -a == ls -la == ls -al

Use ls -F to reveal file types: or "file" file-name

/ : directory @: link #: executable

A link is a point to the actual file or directory.

A link can be used to create a shortcut.

ls -r : display in reverse order.

ls -R : display recursively (show directory by directory).

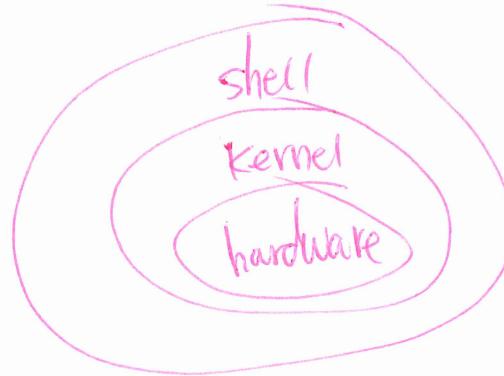
ls -t : sort by time (most recent first).

Shell prompts are slightly different on different Linux system.

Difference between shell, console, and terminal
shell: is the program which actually processes commands and return outputs.
cmd in windows, bash in Linux, can run in a terminal or console.

terminal: is a wrapper program which runs a shell.

Console: a special sort of terminal, interacting with the console is done by using a shell program.



File and Directory Permissions Explained

- regular files.
- d directory
- l symbolic link

r read : allow files to be read
w write : allow files to be modified
x Execute : allow the execution of a file.

{ u user
g group
o other
a all

} Every user is in at least one group, and can belong to many groups.
Group are used to organize users.
To display a user's group: {① use "groups"
② "id -Gn"

The file permission format:

-rwx-r--r--
① ② ③ ④

{ ① file type
② user permission
③ group permission
④ other permission

Changing permissions:

{ chmod change mode
ugo a user group other all
+ - = Add/subtract/set permission
| (rwx) read, write, Execute.
| 0-7
i.e: chmod g-wx file-name

Numeric Based Permissions

r	w	x
0	0	0
1	1	1
4	2	1

value for off
Binary value for on
Base 10 value for on.

Therefore: Octal

0

Binary

String

Description

No permission

Execute only

write only

execute and write

read only

1

--x

2

-w-

3

-wx

4

r--

5

r-x

6

rw-

7

rwX

Working with groups.

New files belong to your primary group.

Use "chgrp" command changes the group.

Permissions on a directory can affect the files in the directory.

File creation Mask : determine default permissions.

The "umask" command:

umask [S] [mode] or umask 022
 \u \u \u
For symbolic notation.

i.e:

	Directory	File	
Default permission	777	666	
Subtract Umask	-022	-022	can subtract any permission
Creation Permissions	755	644	

Displaying the contents of Files.

cat : display the contents of file.

more : display through a file.

less : more features than "more"

{ head : output the beginning portion of file.

tail : output the ending portion of file.

an interactive process

top is

viewer.

head/tail only display 10 lines by default, to change the number :

tail -15 file.txt

tail -f file : follow the file, display the data as it is being written to the file.

The VI editor

Vi file : edit/create file.

Vim file : same as vi, but more features.

View file : in read-only mode.

Vi insert mode: i, I, a, A

i : insert at the cursor position

I : Insert at the beginning of the line.

a : Append after the cursor position.

A : Append at the end of the line.

Vi line mode:

:w writes/saves the file.

:w! force the file to be saved.

:q quit

:q! quit without save

:wq! write and quit without save

:wq/:x write and quit.

:n positions the cursor at line n.

:\$ positions the cursor on the last line

:set nu turn on line numbering

:set nonu turn off line numbering

Vi repeating commands : repeat a command by preceding it with a number.

i.e.: 80i<Text><ESC> : insert <Text> 80 times

Do this before in the file directly, not in ":"

vi - deleting Text

x : delete a character

dw : delete a word

dd : delete a line

D : delete from the current position.

vi - change the text

R : replace the current character.

CW : change the current word

cc : change the current line

C/c\$: change the text from current position.

v : reverse the case of a character (a→A, A→a)

Vi - copy and paste :

yy : copy the current line

y<position> : copy the <position>

p : paste the most recent deleted or copied text.

Vi - Undo / Redo

u : undo

ctrl + R : Redo

Vi - Searching :

/ <pattern> : start a forward search . i.e: /hello

? <pattern> : start a reverse search . i.e: ?hello

Three modes of operation in Vi.

① Command mode : deleting, changing, copy/paste., undo/redo .

② input mode [i] = insert, changing .

③ Line mode [: or] . save/ quit/ set / search .

Finding Files and Directories

{ find
 | locate

The find command: `find [path] [expression_of_search]`

Find option :
-name pattern : find files/directories that match pattern
-iname pattern : like -name but ignore case.
and more.

The locate command: faster than find, but not in real-time.

locate pattern : list files that match pattern.
(file-name).

search data in file : `grep pattern file`.

search data in vi : `/pattern` or `?pattern`.

which : find the location of a program

Graphical Editors (editors with GUI)

emacs, gedit, vim, kedit, kate.

Remove files : `rm`

Copying files : `cp`

Moving and renaming files : `mv source-file destination`

Sorting Data :

Sort

sort file-name : sort text in file (line by line, not word by word)

options : `-k F` : sort by key, F is the field

`-r` : in reverse order

`-u` : sort unique.

tar

Creating a collection of files : create, extract or list contents of a tar archive using pattern

tar options : { `c` create `tar`

`x` extract

`t` display the table of contents.

`v` be verbose

`z` use compression

`f file` use this file.

`tar [options] tarfile [pattern]`

Compressing files To save Space

gzip : compress files

gunzip : uncompress files

g2cat

zcat } concatenates compressed files.

wildcards : '?' , '*' , [] , [! ..] , [a .. z],
any one of the character
any any characters that not included, exactly
one character.

[t:alpha:] : matches 'alphabet.'

[t:alnum:] : matches alphabet and numbers.

[t:digit:] : matches digits.

[t:lower:] : lower case

[t:upper:] : upper case

[t:space:] : space

\ : escape character

Input , Output, and Redirection

Input/output type : stdin, stdout, stderr

0

1

2

(file descriptor)

Redirection :

> redirects standard outputs to a file, overwrites existing contents.

>> redirects standard outputs to a file, appends to existing contents.

< redirect input from a file to a command.

& used with (>, >>, <) to signal that file descriptor is being used.

Comparing the contents of files. diff, sdiff, vimdiff

diff file1 file2 : compare two files.

sdiff file1 file2 : side-by-side comparison.

vimdiff file1 file2 : highlight difference in vim.

Searching in files:

grep pattern file : display lines matching a pattern

- options :
- i ignore case
 - c count the number of occurrences in a file
 - n precede output with line numbers.
 - v invert match, print lines that don't match

file file-name : display the file type.

strings file-name : display printable strings.

Pipe (|)

command-output | command-input : take output as input.

i.e: cat file | grep pattern.

The cut command: cut sections from each line of files and writing the results to stdout. cut out selected portions of file, if file is omitted, use standard input.

cut [file]

options: -b, byte position : i.e: cut -b [range] file

-d , delimiter

-f N , display the Nth field.

-c [num] , select column number , i.e: cut -d ';' -f 1 file

range

delimiter
↑

Copying files over the network : scp : secure copy

sftp : ssh file transfer protocol.

Aliases : (to be persistent, add this alias to '.bash_profile', using vi `w! .bash-profile`)

alias name = value

unalias name : remove the alias

unalias -a : remove all aliases

Environment variables

name/value pair, case sensitive

Create Environment variables:

no space
export var="value"

Removing environment variables: unset var

Persisting environment variables: add to .bash_profile (vi ~/.bash-profile)

Print environment variable: ① printenv var ② echo \$var

Processes and Job Control.

ps : display process status

option: -e all processes, everything
-f full format listing
-u username user's processes
-p pid display information for pid

pstree : display processes in a tree format.

top / htop : interactive process viewer

Background and foreground processes:

& : start command in background.

ctrl + c : kill the foreground process

ctrl + z : suspend the foreground process.

bg [%num] : background a suspended process

fg [%num] : foreground a background process

kill : kill a process by job number or PID

jobs [%num] : list jobs

where %num is the process id

Scheduling Repeated jobs with Cron.

Cron : a time based job scheduling service.

Crontab : a program to create, read, update, and delete your job schedules.

use cron to schedule and automate tasks.

crontab:

min	h	d	m	weekday	command
①	②	③	④	⑤	

① : minute, from 0 ~ 59

② : hour, from 0 ~ 23

③ : day of the month, from 1 ~ 31

④ : month of the year, from 1 ~ 12

⑤ : day of the week, from 0 ~ 6

i.e: 0 7 * * 1 /weekly-report

Run every monday at 7am.

shortcuts	yearly	monthly	weekly	daily	hourly
yearly	0 0 1 1 *	0 0 1 * *	0 0 * * 0	0 0 * * *	0 * * * *

Crontab command:

crontab file-name : install a new crontab from file

crontab -l : list cron jobs

crontab -e : edit your cron jobs

crontab -r : remove all your cron jobs.

Switching users and Running Commands as others

su [username] : change user ID or become superuser

whoami : display the username

sudo : execute a command as superuser

sudo -u user command : run command as user

sudo su - Username : switch to the username account.

sudo -s : start a shell

sudo -u user -s : start a shell as user

history : display the shell history

!N : repeat command line number N.

!! : repeat the previous command line

!string : repeat the most recent command starting with "string".

!^ : the first argument

!\$: the last argument.

Install package using = yum, rpm on (RPM distributions)

Install packages in DEB (debian, Linux mint, Ubuntu) : apt, dpkg

apt-cache search string :

apt-get install [-y] package

apt-get remove package : remove package, but leave configuration

apt-get purge package : remove package, deleting configuration.

apt-cache show package : show information about the package.

dpkg -l : list installed packages

dpkg -L package : list all files in package

dpkg -i package.deb : install package

File system: user data and metadata (data about other data)

Linux: ext4 (up to 1 exabyte, and files up to 16 tebibytes)

Redhat: xfs (for files > 100 TB).

Windows: NTFS.

df command reports system disk space usage, i.e.: df -Th

lsblk -f, display the file system type on partitions

fdisk

↓
mkfs

↓
mount

① To create new partitions, you can use "fdisk" or "parted";

② Then you can use "mkfs" to create file systems for new partitions.

i.e.: mkfs -t ext4 /dev/sda4 (create ext4 FS on sda4 partition)

③ Then you should use "mount" command to mount the partition into a mount point (directory), to start using the file system.

Why partition: ① to achieve higher data security in case of disaster.

There are two types of major partitions on a linux system:

① Data partition: normal system data, i.e; root partition, data to startup, etc.

② swap partition: extra memory on hard disk.

Use "fdisk" to set partition type,