

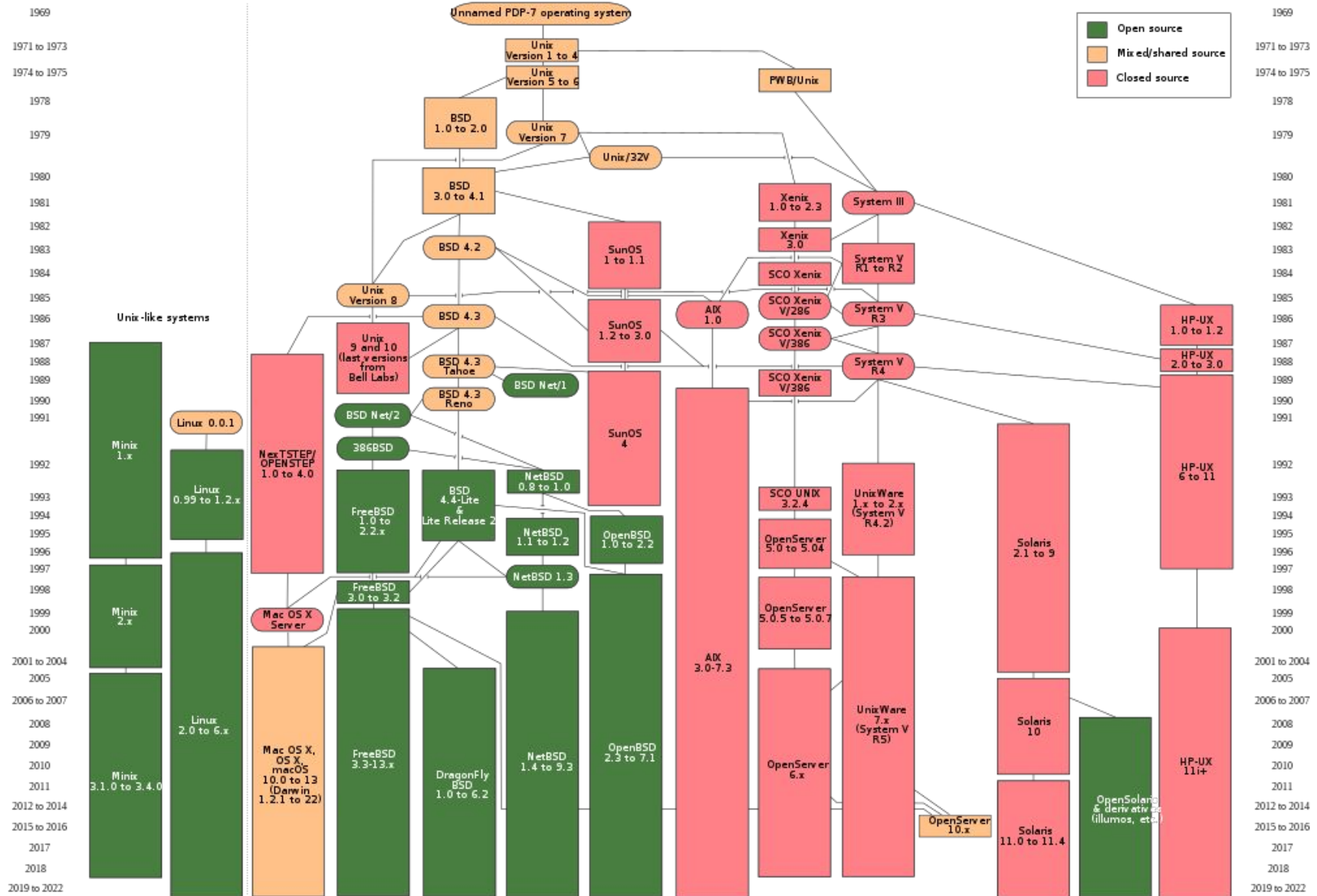
Linux-Basic Commands

Dr. Suchintan Mishra

Contents

- Introduction to Linux
 - History of Linux
 - Basic Commands in Linux
-

History of Linux



Linux Directory Structure

TOP LEVEL DIRECTORIES

- **/bin** – binary or executable programs.
- **/etc** – system configuration files.
- **/home** – home directory. It is the default current directory.
- **/opt** – optional or third-party software.
- **/tmp** – temporary space, typically cleared on reboot.
- **/usr** – User related programs.
- **/var** – log files.

Linux Terminals

ZSH- Zsh is also an environment that can be used as a command-line interpreter for shell scripting or as an interactive login shell. Zsh is built on top of bash thus it has additional features. **Zsh is the default shell for macOS and Kali Linux**

BASH-Bash, or the Bourne-Again Shell, is by far the most popularly used shell and it comes installed as the default shell in the most popular Linux distributions. Bash is the default login shell for most Linux distributions. It is also accessible for Windows and **it is the default user shell in Ubuntu, Linux Mint, Solaris 11, Pop OS, etc.**

ls

ls -a

In Linux, hidden files start with . (dot) symbol and they are not visible in the regular directory. The (ls -a) command will enlist the whole list of the current directory including the hidden files.

ls -l

It will show the list in a long list format.

ls -lh

This command will show you the file sizes in human readable format. Size of the file is very difficult to read when displayed in terms of byte. The (ls -lh)command will give you the data in terms of Mb, Gb, Tb, etc.

ls -lhS

If you want to display your files in descending order (highest at the top) according to their size, then you can use (ls -lhS) command.

ls -l - -block-size=[SIZE]

It is used to display the files in a specific size format. Here, in [SIZE] you can assign size according to your requirement.

ls -d */

It is used to display only subdirectories.

ls -g or ls -lG

With this you can exclude column of group information and owner.

ls

ls -n

It is used to print group ID and owner ID instead of their names.

ls --color=[VALUE]

This command is used to print list as colored or discolored.

ls -li

This command prints the index number if file is in the first column.

ls -p

It is used to identify the directory easily by marking the directories with a slash (/) line sign.

ls -r

It is used to print the list in reverse order.

ls -R

It will display the content of the sub-directories also.

ls -lX

It will group the files with same extensions together in the list.

ls -lt

It will sort the list by displaying recently modified files at top.

ls ~

It gives the contents of home directory.

ls ../

It gives the contents of parent directory.

pwd

The **pwd** command writes to standard output the full path name of your current directory (from the root directory). All directories are separated by a / (slash). The root directory is represented by the first /, and the last directory named is your current directory.

pwd : prints the present working directory

pwd -L: Prints the symbolic path.

pwd -P: Prints the actual path.

cd,MKDIR,RM

cd- **cd** command in linux known as change directory command. It is used to change current working directory.

`cd directory_name`

`cd`

`cd /`

mkdir- **mkdir** command in Linux allows the user to create directories (also referred to as folders in some operating systems). This command can create multiple directories at once as well as set the permissions for the directories.

`mkdir --version`

`mkdir -p first/second/third`

`mkdir directory_name`

RM

rm stands for **remove** here. rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX. To be more precise, rm removes references to objects from the filesystem, where those objects might have had multiple references (for example, a file with two different names). **By default, it does not remove directories.**

Removing one file at a time

```
$ rm a.txt
```

```
$ rm -r *
```

```
$ ls
```

```
b.txt c.txt d.txt e.txt
```

Removing more than one file at a time

```
$ rm b.txt c.txt
```

```
$ rm -i d.txt
```

```
rm: remove regular empty file
```

```
'd.txt'? y
```

Cat, VI, NANO

cat-Cat(concatenate) command is very frequently used in Linux. It reads data from the file and gives their content as output. It helps us to create, view, concatenate files.

```
$cat filename      $cat file1 file2      $cat -n filename
```

```
cat [filename-whose-contents-is-to-be-copied] > [destination-filename]
```

vi/nano-VI/NANO are text editors for linux that can edit any file regardless of its extension.

curl, wget

Wget is the non-interactive network downloader which is used to download files from the server even when the user has not logged on to the system and it can work in the background without hindering the current process.

```
wget http://example.com/sample.php
```

```
wget -b http://www.example.com/samplepage.php
```

```
wget -b http://www.example.com/samplepage.php
```

curl is a command-line tool to transfer data to or from a server, using any of the supported protocols (HTTP, FTP, IMAP, POP3, SCP, SFTP, SMTP, TFTP, TELNET, LDAP, or FILE). *curl* is powered by Libcurl. This tool is preferred for automation since it is designed to work without user interaction. *curl* can transfer multiple files at once.

```
curl -# -O ftp://ftp.example.com/file.zip  
curl --silent ftp://ftp.example.com/file.zip
```

systemctl

The systemctl command is a utility which is responsible for examining and controlling the systemd system and service manager. It is a collection of system management libraries, utilities and daemons which function as a successor to the System V init daemon. The new systemctl commands have proven quite useful in managing a servers services. It provides detailed information about specific systemd services, and others that have server-wide utilization.

systemd Utilities

systemctl journalctl notify analyze cglsgctop loginctl nspawn

systemd Daemons

systemd
journald networkd
logind user session

systemd Targets

bootmode basic multi-user graphical user-session
shutdown reboot dbus telephony display service
dlog logind user-session tizen service

systemd Core

manager unit login namespace log
systemd service timer mount target multiseat inhibit
snapshot path socket swap session pam cgroup dbus

systemd Libraries

dbus-1 libpam libcap libcryptsetup tcpwrapper libaudit libnotify

Linux Kernel

cgroups autofs kdbus

kill

kill command in Linux (located in `/bin/kill`), is a built-in command which is used to terminate processes manually. *kill* command sends a signal to a process which terminates the process. If the user doesn't specify any signal which is to be sent along with kill command then default *TERM* signal is sent that terminates the process.

kill -l :To display all the available signals you can use below command option:

`ps`: Is used to list all processes and their pids.

```
$kill pid
```

```
kill {-signal | -s signal} pid
```

cp

cp stands for **copy**. This command is used to copy files or group of files or directory. It creates an exact image of a file on a disk with different file name. *cp* command require at least two filenames in its arguments.

Syntax: cp Src_file Dest_file

```
$ ls
```

```
a.txt
```

```
$ cp a.txt b.txt
```

```
$ ls
```

```
a.txt  b.txt
```


mv

mv [Option] source destination

mv stands for **move**. mv is used to move one or more files or directories from one place to another in a file system like UNIX. It has two distinct functions:

It has two distinct functions:

(i) It renames a file or folder.

(ii) It moves a group of files to a different directory.

```
$ ls  
a.txt  b.txt  c.txt  d.txt
```

```
$ mv a.txt geek.txt
```

```
$ ls  
b.txt  c.txt  d.txt  
geek.txt
```

```
mv example.txt ~/Documents
```

Manage access to the root account

Sudo, the one command to rule them all. It stands for “super user do!” and is pronounced like “sue dough”.

If you prefix “sudo” with any Linux command, it will run that command with elevated privileges. Elevated privileges are required to perform certain administrative tasks.

The sudoers file

This file is the seedy underbelly of sudo. It controls who can use the sudo command to gain elevated privileges. It is usually located at **/etc/sudoers**.

```
username  ALL=(ALL)    ALL //gives user "username" sudo access%wheel
ALL=(ALL)    ALL //Gives all users that belong to the wheel group sudo access
```

GREP

Grep command can be used to find or search a regular expression or a string in a text file. To demonstrate this, let's create a text file *welcome.txt* and add some content as shown.

1. Search any line that contains the word in filename on Linux:
grep 'word' filename
2. Perform a case-insensitive search for the word 'bar' in Linux and Unix:
grep -i 'bar' file1
3. Look for all files in the current directory and in all of its subdirectories in Linux for the word 'httpd':
grep -R 'httpd' .
4. Search and display the total number of times that the string 'nixcraft' appears in a file named frontpage.md:
grep -c 'nixcraft' frontpage.md

search for files

find command options *starting/path* expression

1. *find* . -name thisfile.txt

If you need to know how to find a file in Linux called thisfile.txt, it will look for it in current and sub-directories.

2. *find* /home -name *.jpg

Look for all .jpg files in the /home and directories below it.

3. *find* . -type f -empty

Look for an empty file inside the current directory.

4. *find* /home -user randomperson -mtime 6 -iname ".db"

Look for all .db files (ignoring text case) that have been changed in the preceding 6 days by a user called *randomperson*.

Users and Groups

A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

A user- group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file.

Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else.

Linux Permissions

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

- **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
- **Write:** The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
- **Execute:** In Windows, an executable program usually has an extension “.exe” and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

rw- only read and write no execute USER

rw- only read and write no execute GROUP

r- only read no execute no write OTHERS

Provide all permission to user. RW to group and R to others

To check file permissions

`ls -l`

The characters are pretty easy to remember.

r = read permission

w = write permission

x = execute permission

- = no permission

chmod

Absolute(Numeric) Mode in Linux

In this mode, file **permissions are not represented as characters but a three-digit octal number**.
The table below gives numbers for all for permissions types.

Number	Permission Type	Symbol
0	No Permission	—
1	Execute	—x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r—
5	Read + Execute	r-x
6	Read +Write	rw-
7	Read + Write +Execute	rwX

chmod

chmod 400 filename

chmod 674 filename



Command Line Redirection

On a command line, *redirection* is the process of using the input/output of a file or command to use it as an input for another file. It is similar but different from pipes, as it allows reading/writing from files instead of only commands.

Redirecting Output

The > symbol is used to redirect output by taking the output from the command on the left and passing as input to the file on the right.

Pipe shell command

The | command is called a *pipe*. It is used to *pipe*, or transfer, the standard output from the command on its left into the standard input of the command on its right.

Append Redirect shell command

The >> shell command is used to redirect the standard output of the command on the left and append (add) it to the end of the file on the right.

Archive, backup, compress, unpack, and uncompress files

The `tar` command creates tar files by converting a group of files into an archive. It also can extract tar archives, display a list of the files included in the archive, add additional files to an existing archive, and various other kinds of operations.

```
tar [OPERATION_AND_OPTIONS] [ARCHIVE_NAME] [FILE_NAME(s)]
```

```
tar -cf archive.tar file1 file2 file3
```

```
tar -czf archive.tar.gz file1 file2
```

```
tar -cjf archive.tar.bz2 file1 file2
```

Listing Archive

```
tar -tf archive.tar
```

```
tar -tvf archive.tar
```

Extracting Archive

```
tar -xf archive.tar
```

```
tar -xvf archive.tar
```

```
tar -xf archive.tar -C /opt/files
```

List, set, and change standard file permissions

Create a file named file1, another file named file2 create a tar archive from files 1 and file 2. Change the permission of the file so that user can (read,write,execute) and groups can (read) while everyone else has no permissions. Rename the file1 to sample and copy the contents of sample to another file called sample2 using cat. List and check the file permissions given to sample, sample2 and the archive.