

19CS416-CS-Ex-3-Linux-Commands

Linux is an open-source operating system, and its kernel is the heart of the OS, facilitating communication between hardware and software. One of the key advantages of Linux is its customizability; developers can modify the Linux kernel to create their own tailored operating systems.

Linux Commands

Linux commands are executed in the terminal, which is case-sensitive. This guide covers some basic and advanced commands used in Linux.

1. ls Command

The `ls` command is used to display a list of contents in a directory.

Syntax:

```
ls
```



Output:

```
untitled5.ipynb
(base) %n@m %1~ %# pwd
/Users/apple
(base) %n@m %1~ %# █
```

2. pwd Command

The `pwd` command displays the location of the current working directory.

Syntax:

```
pwd
```



Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ rename 's/\.txt$/\.pdf/' *.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a                Desktop          examples.desktop  Music            Python-3.8.0
Akash            Directory        hello.c           Newfolder       sample
a.out            Documents        hello.i           pico            snap
composer.phar    Downloads        hello.o           Pictures         Templates
demo1.pdf        eclipse          hello.s           project         Test.pdf
Demo.sh          eclipse-installer index.html        Public          Videos
Demo.txt~        eclipse-workspace mail              Python
```

3. mkdir Command

The `mkdir` command is used to create a new directory.

Syntax:

```
mkdir <directory_name>
```



Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ head Demo.txt
1
2
3
4
5
6
7
8
9
10
```

4. rmdir Command

The `rmdir` command is used to delete a directory.

Syntax:

```
rmdir <directory_name>
```



Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ tail Demo.txt
2
3
4
5
6
7
8
9
10
11
```

5. cd Command

The `cd` command is used to change the current directory.

Syntax:

```
cd <directory_name>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ cd /usr/local/bin
$ ls
ls: write error: 'htetee': Invalid argument
apple@apples-MacBook-Pro ~ (main)
$ cd /usr/local/bin
$ ls
ls: write error: 'htetee': Invalid argument
```

6. cat Command

The `cat` command is used to create, display, and concatenate files.

Syntax:

```
cat [OPTION]... [FILE]...
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ grep -o hello sample.txt
hello
```

7. cp Command

The `cp` command is used to copy files or directories.

Syntax:

```
cp <source_file> <destination_file>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ echo "hello world" | tr "a-z" "A-Z"
HELLO WORLD
```

8. gedit Command

`gedit` is a general-purpose text editor used to create and edit text files.

Syntax:

```
gedit <file_name>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ chmod 755 script.sh
```

9. su Command

The `su` command provides administrative access to another user.

Syntax:

```
su <username>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ tar -xvf archive.tar
```

10. mv Command

The `mv` command is used to move a file or directory from one location to another.

Syntax:

```
mv <file_name> <directory_path>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ chown apple sample.txt
apple@apples-MacBook-Pro ~ (main)
[$ ls -l sample.txt
-rw-r--r-- 1 apple  staff  30 Apr 26 13:41 sample.txt
apple@apples-MacBook-Pro ~ (main)
```

11. rename Command

The `rename` command is used to rename files.

Syntax:

```
rename 's/old-name/new-name/' <files>
```



Output:

```
javatpoint@javatpoint-Inspiron-3542:~/project$ make
echo "Hello World!"
Hello World!
```

12. head Command

The `head` command displays the first 10 lines of a file.

Syntax:

```
head <file_name>
```



Output:

```
7Users/apple
(base) %n%m %1~ %# mkdir file1
(base) %n%m %1~ %# ls
AndroidStudioProjects %n%m %1~ %#
```

13. tail Command

The `tail` command displays the last 10 lines of a file.

Syntax:

```
tail <file_name>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
    inet 127.0.0.1 netmask 0xff000000
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
    nd6 options=201<PERFORMNUD,DAD>
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=0<> mtu 1280
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST>
```

14. id Command

The `id` command displays the user ID (UID) and group ID (GID).

Syntax:

```
id
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ host www.tutorpoint.com
www.tutorpoint.com has address 185.53.177.53
www.tutorpoint.com has IPv6 address 64:ff9b::b935:b135
www.tutorpoint.com mail is handled by 5 mail.h-email.net.
apple@apples-MacBook-Pro ~ (main)
```

15. grep Command

The `grep` command is used to search for a pattern within files.

Syntax:

```
command | grep <search_word>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ gzip sample.txt sample1.txt
sample.txt.gz already exists -- do you wish to overwrite (y or n)? y
sample1.txt.gz already exists -- do you wish to overwrite (y or n)? y
apple@apples-MacBook-Pro ~ (main)
```

16. tr Command

The `tr` command is used to translate or delete characters.

Syntax:

```
command | tr <old> <new>
```



Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sort marks.txt
alen-70
alex-50
carry-85
celena-90
jon-75
justin-80
```

17. chmod Command

The `chmod` command is used to change the access mode (permissions) of a file.

Syntax:

```
chmod <options> <permissions> <file_name>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ cal
      April 2025
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```

18. tar Command

The `tar` command is used to create or extract archive files.

Syntax:

```
tar [options] [archive-file] [files_to_archive]
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ clear
```

19. chown Command

The `chown` command is used to change the ownership of a file.

Syntax:

```
chown <owner_name> <file_name>
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ echo "This is the body of the email." | mail -s "Subject of the email" recipient@example.com
apple@apples-MacBook-Pro ~ (main)
$
```

20. make Command

The `make` command is used to build and maintain groups of programs.

Syntax:

```
make [-f makefile] [options] [targets]
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
$ df
Filesystem      512-blocks      Used Available Capacity  iused      ifree %used    Mounted on
/dev/disk1s5s1  489562928    30115992 242630288      12%    502388 1213151440      0%      /
devfs              385          385          0    100%       667          0    100%      /dev
/dev/disk1s4     489562928    2099240 242630288        1%         2 1213151440      0%    /System/Volumes/VM
/dev/disk1s2     489562928     647864 242630288        1%      1469 1213151440      0%    /System/Volumes/Preboot
/dev/disk1s6     489562928      12024 242630288        1%        95 1213151440      0%    /System/Volumes/Update
/dev/disk1s1     489562928   211594336 242630288      47%   1178709 1213151440      0%    /System/Volumes/Data
map auto_home          0          0          0    100%         0          0    100%    /System/Volumes/Data/home
You have mail in /var/mail/apple
```

21. ifconfig Command

The `ifconfig` command is used to configure network interfaces.

Syntax:

```
ifconfig [options] [interface]
```



Output:

```
to update your account to use zsn, please run  cnsn -s /bin/zsn .
For more details, please visit https://support.apple.com/kb/HT208050.
[(base) %n@%m %1~ %# find sample.txt /Users/apple/sample
sample.txt
/Users/apple/sample
/Users/apple/sample/sample.txt
(base) %n@%m %1~ %# █
```

22. host Command

The `host` command is used to display the IP address for a given domain name.

Syntax:

```
host <domain_name> or <ip_address>
```



Output:

```
[(base) %n@%m %1~ %# rmdir file1
(base) %n@%m %1~ %# █
```

23. gzip Command

The `gzip` command is used to compress files, replacing the original file with a compressed one with a `.gz` extension.

Syntax:

```
gzip <file1> <file2> <file3>...
```



Output:

```
untitled5.ipynb
(base) %n@m %1~ %# cd desktop
(base) %n@m %1~ %# pwd
/Users/apple/desktop
(base) %n@m %1~ %#
```

24. sort Command

The `sort` command is used to sort the contents of a file alphabetically.

Syntax:

```
sort <file_name>
```



Output:

```
untitled5.ipynb      exp1-prg.py      number-random.py
(base) %n@m %1~ %# cat < hello.txt
Paragraphs are the building blocks of papers. Many students define paragraphs in terms of length:
unity and coherence of ideas among sentences is what constitutes a paragraph.(base) %n@m %1~ %#
(base) %n@m %1~ %# cat hello.txt
Paragraphs are the building blocks of papers. Many students define paragraphs in terms of length:
unity and coherence of ideas among sentences is what constitutes a paragraph.(base) %n@m %1~ %#
```

25. cal Command

The `cal` command displays the current month's calendar with the current date highlighted.

Syntax:

```
cal
```



Output:

```
(base) %n@m %1~ %# cp hello.txt hello1.txt
(base) %n@m %1~ %#
```

26. clear Command

The `clear` command clears the terminal screen.

Syntax:

```
clear
```



Output:

```
dave@howtogeek:~/work$ gedit ana.c
```

27. mail Command

The `mail` command is used to send emails from the command line.

Syntax:

```
mail
```



Output:

```
(base) apple@apples-MacBook-Pro ~ % su apple
Password:
```

28. df Command

The `df` command displays the disk space usage of file systems.

Syntax:

```
df
```



Output:

```
apple@apples-MacBook-Pro ~ (main)
[$ mv hello.txt apple
apple@apples-MacBook-Pro ~ (main)
$ █
```

29. find Command

The `find` command is used to search for files in a directory hierarchy.

Syntax:

```
find <directory> -name <file_name>
```



Output:

```
(base) %n@%m %l~ %# ls
AndroidStudioProjects
Applications
Cisco Packet Tracer 8.2.2
Desktop
Downloads
IdeaProjects
Library
Movies
Music
PCA-EXP-1-SUM-ARRAY-GPU-AY-23-24
Pictures
Public
PycharmProjects
Untitled.ipynb
Untitled1.ipynb
Untitled2.ipynb
Untitled3.ipynb
Untitled4.ipynb
Untitled5.ipynb
Untitled6.ipynb
Untitled7.ipynb
Untitled8.ipynb
Untitled9.ipynb
(base) %n@%m %l~ %#

VirtualBox VMs
Virtualbox vms1
a.l
a.out
a.tab.c
a.y
arp.py
arp1.py
arp_client.py
arp_server.py
at.l
at.y
auth.l
auth.y
b.l
b.y
class.html
client_stop.py
compiler.c
compiler.txt
data-engineer-handbook
e.l
e.y
exp-gen
expl-pro.py

exp3.py
exp31.py
exp9.ipynb
file.l
file1.l
file1.tab.c
file1.tab.h
file1.y
forage-cognizant.ipynb
hello
hello.c
hello.dSYM
hello.f08
hello.html
hello.txt
intro.c
k.txt
lex.yy.c
lex_program
m.l
n.y
miniconda3
mod.l
mod.y
number-random.py

pap.html
qns1.java
rarp.py
rarp1.py
received_hello
s.l
s.tab.c
s.y
sa.l
sa.y
sam
sam.c
sam.dSYM
sam.py
sample.asm
sample.py
sample2.py
server.py
server_stop.py
sliding_client.py
sliding_server.py
sql
student_form.html
style.css
tab.c

tcp-client.py
tcp-sever.py
test.c
test.f08
test.html
test.java
test.l
test.py
test.y
text.c
today.html
today1.html
today2.html
traversy-js-challenges
try.html
try1.l
try2.l
try3.l
try4.l
var.c
var1.c
website.html
y.tab.c
y.tab.c.save
y.tab.h
```

Result

All basic and advanced operations were successfully performed through appropriate Linux commands, with the system responding accurately to each, confirming correct execution and expected behavior, thereby demonstrating the effectiveness and reliability of the Linux command-line interface for comprehensive system management.