

	<p align="center"> Bansilal Ramnath Agarwal Charitable Trust's Vishwakarma Institute of Information Technology Department of Artificial Intelligence and Data Science </p>	
<p>Student Name: Siddhesh Dilip Khairnar</p>		
<p>Class: T.Y.</p>	<p>Division: B</p>	<p>Roll No: 372028</p>
<p>Semester: V</p>		<p>Academic Year: 2023 - 24</p>
<p>Subject Name & Code: Cloud Computing and DevOps: ADUA31203</p>		
<p>Title of Assignment: Linux commands practice</p>		
<p>Date of Performance: 16/08/2023</p>	<p>Date of Submission: 20/08/2023</p>	

Assignment: 1

1. **mkdir** (Make Directory): This command is used to create new directories. You provide the name of the directory you want to create as an argument.

```
siddhesh@LAPTOP-USTR35KT:~$ mkdir folder{1..5}
siddhesh@LAPTOP-USTR35KT:~$ ls
folder1 folder2 folder3 folder4 folder5
siddhesh@LAPTOP-USTR35KT:~$ |
```

2. **cd** (Change Directory): This command is used to change the current working directory. You can use it to navigate through the file system and move into different directories.

```
siddhesh@LAPTOP-USTR35KT:~$ cd folder1
siddhesh@LAPTOP-USTR35KT:~/folder1$ |
```

3. **echo**: The ``echo`` command is used to display a line of text or output to the terminal. It takes the text or string provided as an argument and prints it to the standard output. This command is often used for displaying messages, variables, or other textual information on the screen. It's a simple way to provide information or feedback to the user when running scripts or interacting with the command-line interface.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ echo "Geeks for Geeks"
Geeks for Geeks
```

4. **ls** (List): The ``ls`` command is used to list the contents of a directory. It shows you the names of files and subdirectories within the specified directory.

```
siddhesh@LAPTOP-USTR35KT:~$ ls
folder1 folder2 folder3 folder4 folder5
```

5. **more**: The ``more`` command is used to display the contents of a text file in a paginated manner. It allows you to view a file one screenful at a time.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ more -d sample.txt
I Name is Siddhesh Khairnar
```

6. **cp** (Copy): The ``cp`` command is used to copy files or directories from one location to another. It creates a duplicate of the specified file or directory.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ more -d sample2.txt
I Name is Siddhesh Khairnar
```

7. mv (Move): The `mv` command is used to move files or directories from one location to another. It's also used to rename files and directories.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ mv sample.txt sample1.txt
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
sample1.txt  sample2.txt
```

8. grep: The `grep` command is used to search for specific patterns (text strings or regular expressions) within files. It's often used to find occurrences of a particular word or phrase.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ grep -i "UNix" sample1.txt
```

9. mv (Move): The `mv` command is used to move files or directories from one location to another. It's also used to rename files and directories.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ mv sample1.txt folder2
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
folder2  sample2.txt
```

- 10.rm (Remove): The `rm` command is used to delete files and directories. Be cautious when using this command, as deleted data is typically not recoverable.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ rm folder2
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
sample2.txt
```

- 11.find: The `find` command is used to search for files and directories in a specified location and perform actions on them based on conditions you specify.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ find ./sample2.txt
./sample2.txt
```

- 12.pwd (Print Working Directory): This command displays the full path of the current working directory, showing you where you are in the file system.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ pwd
/home/siddhesh/folder1
```

- 13.tar: The `tar` command is used to create and manipulate archive files. It can compress multiple files and directories into a single archive file, making it easier to transport or share them.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
file.tar  sample2.txt
```

- 14.rmdir: The rmdir command is used to remove empty directories from a system. The directory must be empty, and you must have written permission in its parent directory. The syntax for the rmdir command is rmdir <dirname>.

```
siddhesh@LAPTOP-USTR35KT:~$ rmdir folder5
siddhesh@LAPTOP-USTR35KT:~$ ls
folder1  folder2  folder3  folder4
siddhesh@LAPTOP-USTR35KT:~$ |
```

- 15.touch: The touch command in Linux is used to create, change and modify timestamps of a file. It is a standard command used in UNIX/Linux operating system.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ touch file{0..9}.txt
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
file.tar  file1.txt  file3.txt  file5.txt  file7.txt  file9.txt
file0.txt  file2.txt  file4.txt  file6.txt  file8.txt  sample2.txt
```

- 16.cat: The cat command is a command-line utility for displaying, concatenating, writing to, or appending files. It is a very versatile command with many uses.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ cat> sample1.txt
Roll no:372028
siddhesh@LAPTOP-USTR35KT:~/folder1$ ls
file0.txt  file2.txt  file4.txt  file6.txt  file8.txt  sample1.txt
file1.txt  file3.txt  file5.txt  file7.txt  file9.txt  sample2.txt
siddhesh@LAPTOP-USTR35KT:~/folder1$ |
```

- 17.head: The head command in Linux prints the first lines of a file or input stream. It is often used to troubleshoot problems by quickly viewing the beginning of a file.

```

line1
line2
line3
line4
line5
vedant@DESKTOP-JSP30IF:~/linux$ cat file1.txt
line1
line2
line3
line4
line5
line6
line7
line8
line9
line104
line11

```

- 18.tail: The tail command, as the name implies, prints the last N no. of data of the given input. By default, it prints the last 10 lines of the specified files. If more than one file name is provided, then data from each file is precedes by its file name.

```

line9
line104
line11
line12vedant@DESKTOP-JSP30IF:~/linux$ |

```

- 19.tac: The tac command in Linux is used to concatenate and print files in reverse. It is the reverse of the cat command. The tac command will write each FILE to standard output, the last line first. When no file is specified then this command will read the standard input.

```

line12line11
line104
line9
line8
line7
line6
line5
line4
line3
line2
line1

```

- 20.df: The df command in Linux is used to display information about file system disk space usage. It retrieves the information from /proc/mounts or /etc/mtab. By default, df command shows disk space in Kilobytes (KB) and uses the SI unit suffixes for clarity.

```
siddhesh@LAPTOP-USTR35KT:~$ df
Filesystem      1K-blocks      Used    Available Use% Mounted on
none            4026336         4      4026332   1% /mnt/wsl
none           424273220 154330284 269942936 37% /usr/lib/wsl/drivers
none            4026336         0      4026336   0% /usr/lib/wsl/lib
/dev/sdc        1055762868 1564208 1000495188 1% /
none            4026336         80     4026256   1% /mnt/wslg
rootfs          4023088        1936     4021152   1% /init
none            4023120         0      4023120   0% /dev
none            4026336        876     4025460   1% /run
none            4026336         0      4026336   0% /run/lock
none            4026336         0      4026336   0% /run/shm
none            4026336         0      4026336   0% /run/user
tmpfs           4026336         0      4026336   0% /sys/fs/cgroup
none            4026336         76     4026260   1% /mnt/wslg/versions.txt
none            4026336         76     4026260   1% /mnt/wslg/doc
drvfs           424273220 154330284 269942936 37% /mnt/c
drvfs           51199996 41696788 9503208 82% /mnt/d
snapfuse        65024        65024         0 100% /snap/core20/1891
snapfuse        94080        94080         0 100% /snap/lxd/24061
snapfuse        54528        54528         0 100% /snap/snapd/19122
siddhesh@LAPTOP-USTR35KT:~$ |
```

- 21.type: The type command tells you whether a Linux command is built-in shell command, where is its executable located and whether it is aliased to some other command.

```
siddhesh@LAPTOP-USTR35KT:~$ type rmdir
rmdir is hashed (/usr/bin/rmdir)
siddhesh@LAPTOP-USTR35KT:~$ type cd
cd is a shell builtin
siddhesh@LAPTOP-USTR35KT:~$ type ls
ls is aliased to 'ls --color=auto'
```

- 22.clear: The clear command in Linux is used to clear the terminal screen. To use it, simply type clear and press Enter. You can also use the keyboard shortcut Ctrl+L to clear the terminal screen.

Here are some examples of how to use the clear command:

- * clear: Clears the entire terminal screen.
- * clear -l: Clears the current line of the terminal screen.
- * clear -n: Clears the next n lines of the terminal screen.

The clear command can be used in any Linux environment, including Ubuntu, Fedora, and CentOS.

```
siddhesh@LAPTOP-USTR35KT:~$ type rmdir
rmdir is hashed (/usr/bin/rmdir)
siddhesh@LAPTOP-USTR35KT:~$ type cd
cd is a shell builtin
siddhesh@LAPTOP-USTR35KT:~$ type ls
ls is aliased to `ls --color=auto'
siddhesh@LAPTOP-USTR35KT:~$ clear

siddhesh@LAPTOP-USTR35KT:~$ |
```

- 23.id: The id command is a basic Linux command used to confirm the identity of a specified Linux user. It is also used to find user and group names, along with the UID and GID of any user in Linux. By default, the id command is available in all the Linux operating systems.

```
siddhesh@LAPTOP-USTR35KT:~$ id
uid=1000(siddhesh) gid=1000(siddhesh) groups=1000(siddhesh),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(video),46(plugdev),118(netdev)
siddhesh@LAPTOP-USTR35KT:~$ |
```

- 24.passwd: The passwd command changes passwords for user accounts.
Syntax: passwd [options] [LOGIN]

```
prajwal@DESKTOP-MBE740D:~/linux$ passwd
Changing password for prajwal.
Current password:
passwd: Authentication token manipulation error
passwd: password unchanged
prajwal@DESKTOP-MBE740D:~/linux$ |
```

- 25.groupadd: The groupadd command creates a new group account using the values specified on the command line plus the default values from the system. The new group will be entered into the system files as needed.

```
siddhesh@LAPTOP-USTR35KT:~$ groupadd test_group
groupadd: Permission denied.
groupadd: cannot lock /etc/group; try again later.
siddhesh@LAPTOP-USTR35KT:~$ sudo groupadd test_group
[sudo] password for siddhesh:
siddhesh@LAPTOP-USTR35KT:~$ sudo tail /etc/group
tcpdump:x:113:
ssh:x:114:
landscape:x:115:
fwupd-refresh:x:116:
admin:x:117:
netdev:x:118:siddhesh
lxd:x:119:
siddhesh:x:1000:
systemd-coredump:x:999:
test_group:x:1001:
siddhesh@LAPTOP-USTR35KT:~$
```

- 26.cut: The cut command in UNIX is a command for cutting out the sections from each line of files and writing the result to standard output. It can be used to cut parts of a line by byte position, character, and field. Basically, the cut command slices a line and extracts the text. It is necessary to specify option with command otherwise it gives error. If more than one file name is provided, then data from each file is not precedes by its file name.

```
Lie1
Lie2
Lie3
Lie4
Lie5
Lie6
Lie7
Lie8
Lie9
Lie1
Lie1
Lie1
```

- 27.tr: The tr command is a UNIX command-line utility for translating or deleting characters. It supports a range of transformations including uppercase to lowercase, squeezing repeating characters, deleting specific characters, and basic find and replace.


```
vedant@DESKTOP-JSP30IF:~/linux$ cat abc.txt | tr "aBcE" "ZXCv"
Z
X
C
DVFghIJK
```

- 28.wc: The wc command in Linux is a command-line tool that counts the number of lines, words, and bytes in a file. It can also be used to print the length of the longest line in a file. The wc command is a very versatile tool and can be used for a variety of tasks, such as checking the size of a file, counting the number of words in a document, and finding the longest line in a file.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ wc sample2.txt
 1  5 28 sample2.txt
siddhesh@LAPTOP-USTR35KT:~/folder1$ |
```

- 29.od: The od command in Linux is a utility that can be used to display the contents of a file in different formats. The default format is octal, but it can also be displayed in hexadecimal, decimal, and ASCII. The od command is useful for debugging Linux scripts and for visualizing data that is not in a human-readable format.

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ od -b sample2.txt
00000000 111 040 116 141 155 145 040 151 163 040 123 151 144 144 150 145
00000020 163 150 040 113 150 141 151 162 156 141 162 012
00000034

siddhesh@LAPTOP-USTR35KT:~/folder1$ od -c sample2.txt
00000000 I      N   a   m   e       i   s       S   i   d   d   h   e
00000020 s   h      K   h   a   i   r   n   a   r  \n
00000034
siddhesh@LAPTOP-USTR35KT:~/folder1$ |
```

- 30.du -ah:

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ du -ah
4.0K    ./sample1.txt
4.0K    ./sample2.txt
0       ./file2.txt
0       ./file1.txt
0       ./file0.txt
0       ./file7.txt
0       ./file5.txt
0       ./file4.txt
0       ./file3.txt
0       ./file8.txt
0       ./file6.txt
0       ./file9.txt
```

31.du -sh:

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ du -sh
12K .
siddhesh@LAPTOP-USTR35KT:~/folder1$ |
```

32.ps:

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ ps
  PID TTY          TIME CMD
  579 pts/0        00:00:00 bash
 1724 pts/0        00:00:00 ps
```

33.ps -ef:

```
siddhesh@LAPTOP-USTR35KT:~/folder1$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1      0  0 17:25 ?           00:00:00 /sbin/init
root           2      1  0 17:25 ?           00:00:00 /init
root           5      2  0 17:25 ?           00:00:00 plan9 --control-socket 6
root          54      1  0 17:25 ?           00:00:00 /lib/systemd/systemd-jou
root          74      1  0 17:25 ?           00:00:00 /lib/systemd/systemd-ude
systemd+      82      1  0 17:25 ?           00:00:00 /lib/systemd/systemd-net
root         242      1  0 17:25 ?           00:00:00 snapfuse /var/lib/snapd/
root         248      1  0 17:25 ?           00:00:00 snapfuse /var/lib/snapd/
root         250      1  0 17:25 ?           00:00:03 snapfuse /var/lib/snapd/
systemd+      257      1  0 17:25 ?           00:00:00 /lib/systemd/systemd-res
root         260      1  0 17:25 ?           00:00:00 /usr/lib/accountsservice
message+      261      1  0 17:25 ?           00:00:00 /usr/bin/dbus-daemon --s
root         267      1  0 17:25 ?           00:00:00 /usr/bin/python3 /usr/bi
root         268      1  0 17:25 ?           00:00:00 /usr/lib/policykit-1/pol
syslog        269      1  0 17:25 ?           00:00:00 /usr/sbin/rsyslogd -n -i
root         270      1  0 17:25 ?           00:00:00 /usr/lib/snapd/snapd
root         276      1  0 17:25 ?           00:00:00 /lib/systemd/systemd-log
root         277      1  0 17:25 ?           00:00:00 /usr/lib/udisks2/udisksd
root         293      1  0 17:25 ?           00:00:00 /usr/sbin/ModemManager
root         316      1  0 17:25 ?           00:00:00 /usr/sbin/cron -f
daemon       320      1  0 17:25 ?           00:00:00 /usr/sbin/atd -f
root         332      1  0 17:25 hvco      00:00:00 /sbin/agetty -o -p -- \u
root         335      1  0 17:25 ?           00:00:00 /usr/bin/python3 /usr/sh
root         338      1  0 17:25 tty1      00:00:00 /sbin/agetty -o -p -- \u
root          577      2  0 17:25 ?           00:00:00 /init
root          578     577  0 17:25 ?           00:00:00 /init
siddhesh      579     578  0 17:25 pts/0      00:00:00 -bash
root          580      2  0 17:25 pts/1      00:00:00 /bin/login -f
siddhesh      754      1  0 17:25 ?           00:00:00 /lib/systemd/systemd --u
siddhesh      755     754  0 17:25 ?           00:00:00 (sd-pam)
siddhesh      762     580  0 17:25 pts/1      00:00:00 -bash
siddhesh     1729     579  0 19:38 pts/0      00:00:00 ps -ef
```