

4. Windows (DOS) Commands – 1:

(Date, Time, Prompt, md, cd, rd, path)

Open Command Line / Command Prompt in any Windows OS
(Run as Administrator)

- Date /t –

```
C:\>date /t  
06-04-2024
```

- Date (you get option to change date after this command) –

```
C:\>date  
The current date is: 06-04-2024  
Enter the new date: (dd-mm-yy)
```

- Time /t –

```
C:\>time /t  
14:51
```

- Time –

```
C:\>time  
The current time is: 14:51:10.01  
Enter the new time:
```

- Cls (Use to clear the terminal)–

```
C:\>cls
```

- prompt \$p\$g=Time\$t –

```
C:\>$p$g=Time$t  
'$p$g' is not recognized as an internal or external command,  
operable program or batch file.
```

- cd (Use to move through file) –

```
C:\>cd new
```

- md (Use to create directory) –

```
C:\new>md a b c
```

- rd (Use to delete file or directory) –

```
C:\new>rd a b c

C:\new>cd \

C:\>rd new
```

- path –

```
C:\>path
PATH=C:\app\SUMIT\virtual\product\12.2.0\dbhome_1\bin;C:\Program Files (x86)\VMware\VMware Workstation\bin;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\Win
owsPowerShell\cmd;C:\windows\System32\OpenSSH\;C:\Program Files (x86)\Microsoft SQL Server\160\Tools\Binn\;C:\Program Files\Microsoft SQL Server\160\Tools\Binn\;C:\Program Files\Microso
ft SQL Server\Client SDK\ODBC\170\Tools\Binn\;C:\Program Files\Microsoft SQL Server\160\DTs\Binn\;C:\Program Files (x86)\Microsoft SQL Server\160\DTs\Binn\;C:\Program Files\Azure Data Stud
io\bin;C:\Program Files\Git\cmd;C:\Program Files\nodejs\;C:\Program Files\MySQL\MySQL Server 8.0\bin;C:\Program Files\MySQL\MySQL Shell 8.0\bin;C:\Users\SUMIT\AppData\Local\Programs\Python
\Python312\Scripts\;C:\Users\SUMIT\AppData\Local\Programs\Python\Python312\;C:\Users\SUMIT\AppData\Local\Microsoft\WindowsApps\;C:\Program Files\Azure Data Studio\bin;C:\Users\SUMIT\AppData
\Local\Programs\Microsoft VS Code\bin;C:\WinGW\bin;C:\Program Files\JetBrains\PyCharm Community Edition 2023.3.2\bin;;C:\Users\SUMIT\AppData\Local\GitHubDesktop\bin;C:\Users\SUMIT\AppData
Roaming\npm
```

- Chkdsk –

```
C:\Windows\System32>chkdsk
The type of the file system is NTFS.
Volume label is Windows.

WARNING! /F parameter not specified.
Running CHKDSK in read-only mode.

Stage 1: Examining basic file system structure ...
 537088 file records processed.
File verification completed.
Phase duration (File record verification): 9.25 seconds.
 20766 large file records processed.
Phase duration (Orphan file record recovery): 18.27 milliseconds.
 0 bad file records processed.
Phase duration (Bad file record checking): 0.62 milliseconds.

Stage 2: Examining file name linkage ...
 903 reparse records processed.
 755828 index entries processed.
Index verification completed.
Phase duration (Index verification): 19.44 seconds.
 0 unindexed files scanned.
Phase duration (Orphan reconnection): 3.09 seconds.
 0 unindexed files recovered to lost and found.
Phase duration (Orphan recovery to lost and found): 0.74 milliseconds.
 903 reparse records processed.
Phase duration (Reparse point and Object ID verification): 9.14 milliseconds.

Stage 3: Examining security descriptors ...
Security descriptor verification completed.
Phase duration (Security descriptor verification): 71.57 milliseconds.
 109371 data files processed.
Phase duration (Data attribute verification): 1.48 milliseconds.
CHKDSK is verifying Usn Journal...
 40654888 USN bytes processed.
Usn Journal verification completed.
Phase duration (USN journal verification): 256.64 milliseconds.

Windows has scanned the file system and found no problems.
No further action is required.

396787711 KB total disk space.
162141116 KB in 375452 files.
 258636 KB in 109372 indexes.
 0 KB in bad sectors.
 664667 KB in use by the system.
 65536 KB occupied by the log file.
233723292 KB available on disk.
```

5. Windows (DOS) Commands – 1.1: (copy, xcopy, format, fidsk, cls, defrag, del, move)

In new folder on desktop create a new text file (Containing any simple statement like: test file)

- Using the cd command go into that folder –

```
C:\>cd test
```

- copy (Use to copy file) –

```
C:\test>copy new.txt new2.txt  
1 file(s) copied.
```

- xcopy (use to copy file or directory) –

```
C:\test>xcopy new.txt new3.txt  
Does new3.txt specify a file name  
or directory name on the target  
(F = file, D = directory)? f  
C:\new.txt  
1 File(s) copied
```

- DEFRAG c: -a –

```
C:\test>DEFRAG c: -a  
Invoking analysis on Windows (C:)...  
  
Pre-Optimization Report:  
  
Volume Information:  
Volume size           = 378.40 GB  
Free space             = 223.41 GB  
Total fragmented space = 13%  
Largest free space size = 201.70 GB  
  
Note: File fragments larger than 64MB are not included in the fragmentation statistics.  
  
It is recommended that you defragment this volume.  
  
The operation completed successfully.  
  
Post Defragmentation Report:  
  
Volume Information:  
Volume size           = 378.40 GB  
Free space             = 223.41 GB  
Total fragmented space = 13%  
Largest free space size = 201.70 GB  
  
Note: File fragments larger than 64MB are not included in the fragmentation statistics.  
  
It is recommended that you defragment this volume.
```

- Del (Use to delete file or directory –

```
C:\test>del new.txt new2.txt new3.txt  
C:\test>cd /  
C:\>del test
```

Practical 6

- Is command in Linux:

The ls command is commonly used to identify the files and directories in the working directory.

```
(kali@kali)-[~]  
$ ls  
Desktop Documents Downloads Music Pictures Public Templates Videos
```

- pwd command in Linux:

The pwd command is mostly used to print the current working directory on your terminal.

```
(kali@kali)-[~]  
$ pwd  
/home/kali
```

- mkdir command in Linux:

This mkdir command allows you to create fresh directories in the terminal itself. The default syntax is mkdir and the new directory will be created.

```
(kali@kali)-[~]  
$ mkdir os_Practical
```

- cd command in Linux:

The cd command is used to navigate between directories.

```
(kali@kali)-[/home]  
$ cd kali
```

- rmdir command in Linux:

The rmdir command is used to delete permanently an empty directory.

```
(kali@kali)-[~]  
$ rmdir os_Practical
```

- cp command in Linux:

The cp command of Linux is equivalent to copy-paste and cut-paste in Windows.

```
(kali㉿kali)-[~/os_Practical]  
$ cp simple simple2
```

- mv command in Linux:

The mv command is generally used for renaming the files in Linux.

```
(kali㉿kali)-[~/os_Practical]  
$ mv simple2 new_simple
```

- rm command in Linux:

rm command in Linux is generally used to delete the files created in the directory.

```
(kali㉿kali)-[~/os_Practical]  
$ rm simple
```

Practical 7

- Uname command in Linux:

The uname command is used to check the complete OS information of the system.

```
(kali@kali)-[~]  
$ uname  
Linux
```

- locate command in Linux:

The locate command is generally used to locate the files in the database. Use an asterisk (*) to search for content that contains two or more words. As an example: locate first*file. This command will search the database for the files that contain these two names first and file.

```
(kali@kali)-[~]  
$ locate new_file  
/var/lib/plocate/plocate.db: No such file or directory
```

- touch command in Linux:

The touch command creates an empty file when put in the terminal in this format as touch <file name>.

```
(kali@kali)-[~/new_file]  
$ touch hello
```

- ln command in Linux:

The ln command is used to create a shortcut link to another file. This is among the most important Linux commands to know if you want to operate as a Linux administrator.

```
(kali@kali)-[~]  
$ ln -s new new_file
```

- **cat command in Linux:**

The cat command is the simplest command to use when you want to see the contents of a particular file. The only issue is that it simply unloads the entire file to your terminal. If you want to navigate around a huge file, should use less command alternatively.

```
(kali㉿kali)-[~/new]
$ cat wne
hello
it's me
and who
might you be
```

- **clear command in Linux:**

The clear command is a standard command to clear the terminal screen.

```
(kali㉿kali)-[~/new]
$ clear
```

- **ps command in Linux:**

ps command in Linux is used to check the active processes in the terminal.

```
(kali㉿kali)-[~/new]
$ ps
  PID TTY          TIME CMD
 8864 pts/1        00:00:03 zsh
16899 pts/1        00:00:00 ps
```

- **man command in Linux:**

The man command displays a user manual for any commands or utilities available in the Terminal, including their name, description, and options. Command to view the full manual: man <command name>

```
(kali㉿kali)-[~/new]
$ man -f ls
ls: nothing appropriate.
```


Practical 8

- **grep command in Linux:**

The grep command is used to find a specific string in a series of outputs. For example, if you want to find a string in a file, you can use the syntax:
| grep "<string to find>".

- **echo command in Linux:**

echo command in Linux is specially used to print something in the terminal.

```
(kali㉿kali)-[~]  
$ echo "Hello welcome to pactical no 8"  
Hello welcome to pactical no 8  
  
(kali㉿kali)-[~]  
$ touch new.txt  
  
(kali㉿kali)-[~]  
$ echo "Hello welcome to pactical no 8" > new.txt
```

- **whoami command in Linux:**

The whoami command provides basic information that is extremely useful when working on multiple systems. In general, if you are working with a single computer, you will not require it as frequently as a network administrator.

```
(kali㉿kali)-[~]  
$ whoami  
kali
```

- **sort command in Linux:**

The sort command is used generally to sort the output of the file. Let's use the command and see the output.

```
(kali㉿kali)-[~]  
$ sort new.txt  
Hello welcome to pactical no 8  
  
(kali㉿kali)-[~]  
$ echo "This is me sumit" > new.txt  
  
(kali㉿kali)-[~]  
$ sort new.txt  
This is me sumit  
  
(kali㉿kali)-[~]  
$ cat new.txt  
This is me sumit
```

- **cal command in Linux:**

The cal command is not the most famous command in the terminal but it functions to view the calendar for a particular month in the terminal.

Let's see how this works.

```
[root@linux_machine ~]# cal
      July 2020
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 31

[root@linux_machine ~]#
```

- **whereis command in Linux:**

whereis command in Linux is generally used to see the exact location of any command typed after this. Let's see how this performs.

```
(kali@kali)-[~]
$ whereis kali
kali:
```

- **df command in Linux:**

df command in Linux gets the details of the file system.

```
(kali@kali)-[~]
$ df
Filesystem      1K-blocks    Used Available Use% Mounted on
udev             959868         0   959868    0% /dev
tmpfs            200308      1268   199040    1% /run
/dev/sda1      82083148 14937492 62930108   20% /
tmpfs           1001524         0   1001524    0% /dev/shm
tmpfs             5120         0     5120    0% /run/lock
tmpfs           200304      128   200176    1% /run/user/1000
```

- **wc command in Linux:**

wc command in Linux indicates the number of words, characters, lines, etc using a set of options.

- wc -w shows the number of words
- wc -l shows the number of lines
- wc -m shows the number of characters present in a file

```
(kali㉿kali)-[~]  
$ wc -w new.txt  
6 new.txt  
  
(kali㉿kali)-[~]  
$ wc -l new.txt  
1 new.txt  
  
(kali㉿kali)-[~]  
$ wc -m new.txt  
31 new.txt  
  
(kali㉿kali)-[~]  
$ cat new.txt | grep "Hello"  
Hello welcome to pactical no 8  
  
(kali㉿kali)-[~]  
$ echo "This is me sumit" > new.tx
```