

# The CLI

ccc

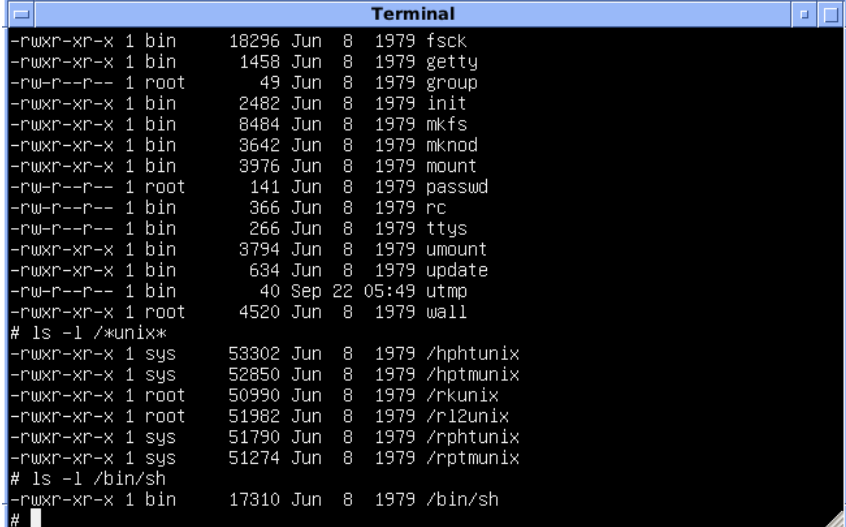
# What is a CLI?

- CLI stands for the “Command Line Interface”
- It’s the “Green Terminal Screen” you’ve seen in movies used by “Hackers”.
- It used to be the only way to interact with your computer. (Until the Beginning in 1979 with Apple Lisa from Steve Jobs and his team.
- CLIs began with UNIX in the early 1960’s/1970’s



# Unix CLI

- One of the earliest and most influential command-line interfaces was the one used by Unix, developed in the late 1960s and early 1970s at AT&T Bell Labs. Unix introduced many of the concepts and commands that are still used in modern CLIs today. It featured a text-based shell that allowed users to interact with the operating system by entering commands and receiving text-based output.

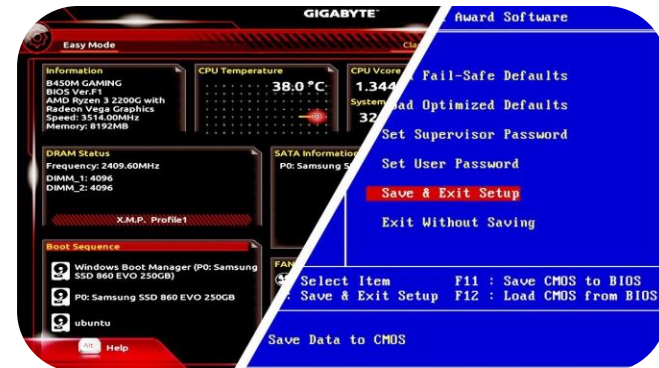
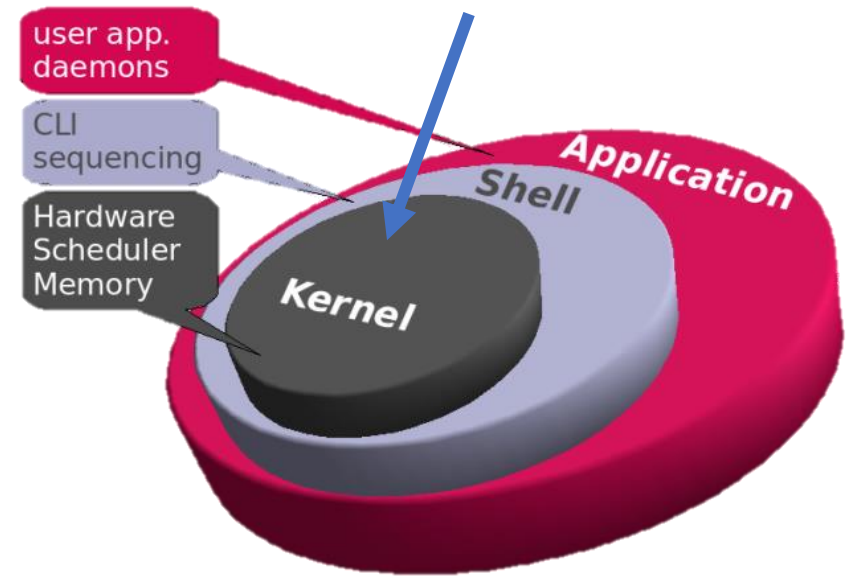


```
Terminal
-rwxr-xr-x 1 bin      18296 Jun  8  1979 fsck
-rwxr-xr-x 1 bin      1458 Jun  8  1979 getty
-rw-r--r-- 1 root       49 Jun  8  1979 group
-rwxr-xr-x 1 bin      2482 Jun  8  1979 init
-rwxr-xr-x 1 bin      8484 Jun  8  1979 mkfs
-rwxr-xr-x 1 bin      3642 Jun  8  1979 mknod
-rwxr-xr-x 1 bin      3976 Jun  8  1979 mount
-rw-r--r-- 1 root       141 Jun  8  1979 passwd
-rw-r--r-- 1 bin       366 Jun  8  1979 rc
-rw-r--r-- 1 bin       266 Jun  8  1979 ttys
-rwxr-xr-x 1 bin      3794 Jun  8  1979 umount
-rwxr-xr-x 1 bin       634 Jun  8  1979 update
-rw-r--r-- 1 bin        40 Sep 22 05:49 utmp
-rwxr-xr-x 1 root      4520 Jun  8  1979 wall
# ls -l /*unix*
-rwxr-xr-x 1 sys      53302 Jun  8  1979 /hptunix
-rwxr-xr-x 1 sys      52850 Jun  8  1979 /hptmunix
-rwxr-xr-x 1 root     50990 Jun  8  1979 /rkunix
-rwxr-xr-x 1 root     51982 Jun  8  1979 /rl2unix
-rwxr-xr-x 1 sys      51790 Jun  8  1979 /rphtunix
-rwxr-xr-x 1 sys      51274 Jun  8  1979 /rptmunix
# ls -l /bin/sh
-rwxr-xr-x 1 bin      17310 Jun  8  1979 /bin/sh
#
```

Kernal vs Shell vs Application

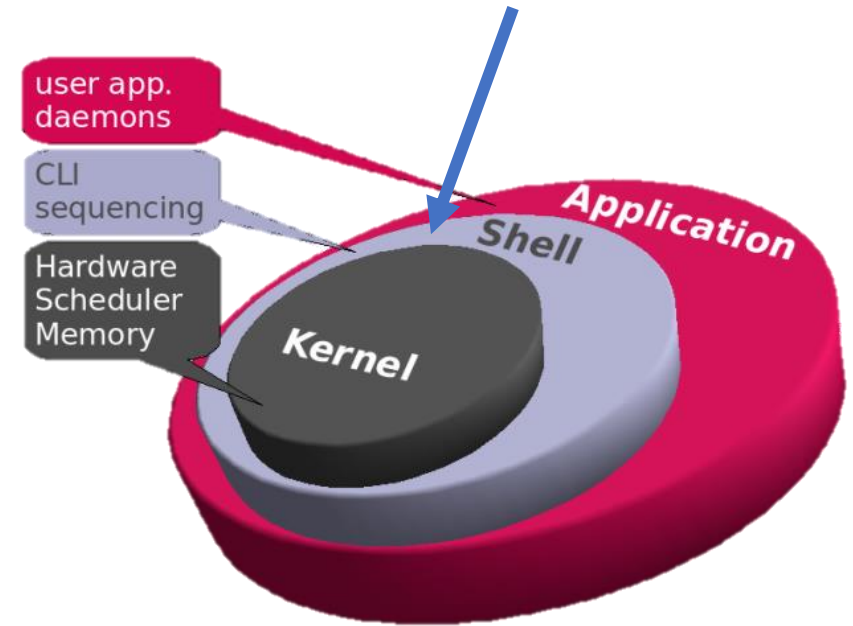
# Kernal vs Shell vs Application

- **Kernal:** The core component of an operating system. It is responsible for managing the hardware resources of the computer, such as CPU, memory, input/output devices, and system calls.
- *Intermediary* between ***the hardware and the software*** running on a computer.
- The Kernel enforces security and access control policies, ensuring that different processes and users can't interfere with or harm each other.



# Kernal vs Shell vs Application

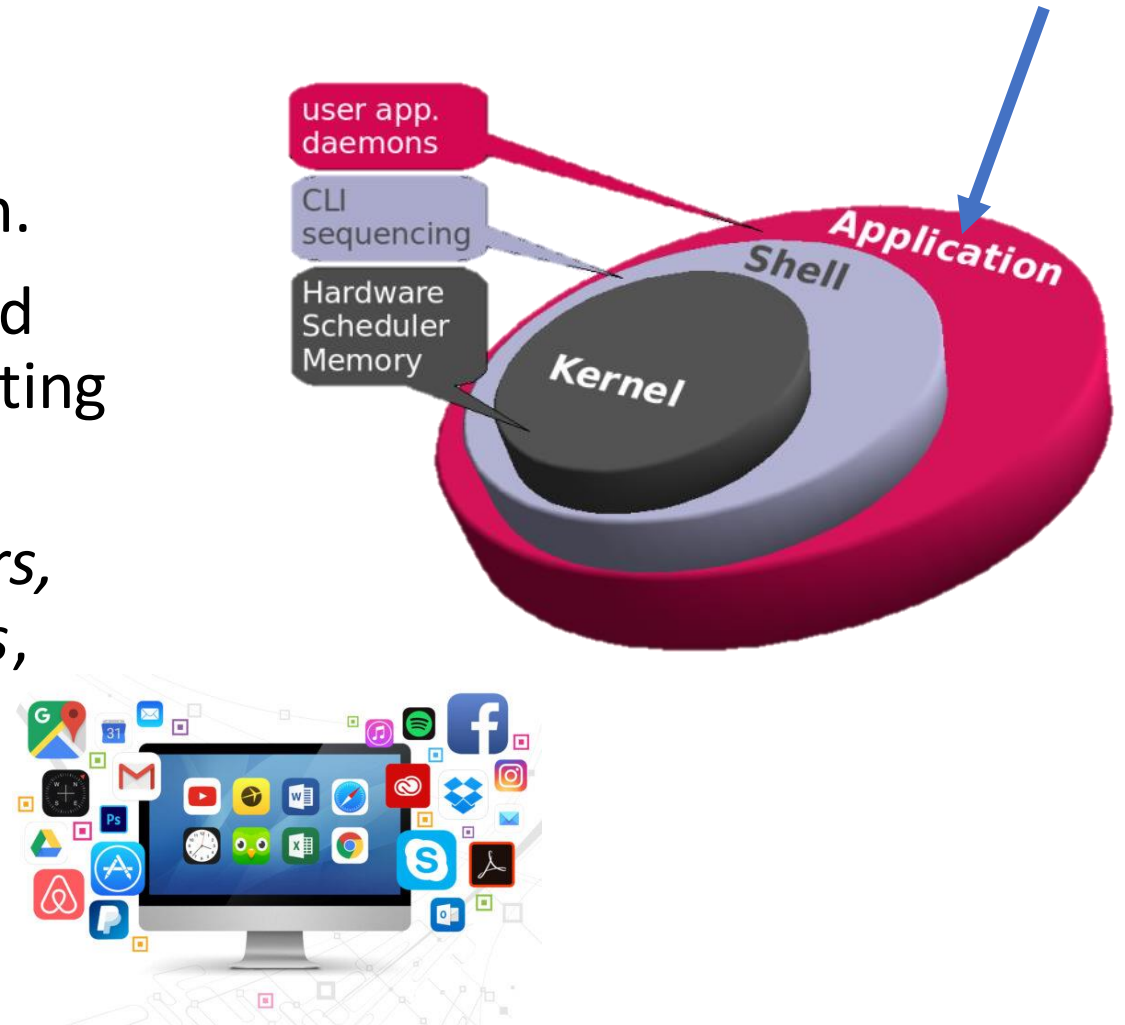
- **Shell:** User interface that allows users to interact with the operating system and **run commands**. It serves as an intermediary between the user and the kernel.
- Interprets user commands, whether *typed in manually or executed from scripts*, and then communicates with the kernel to perform the requested operations.



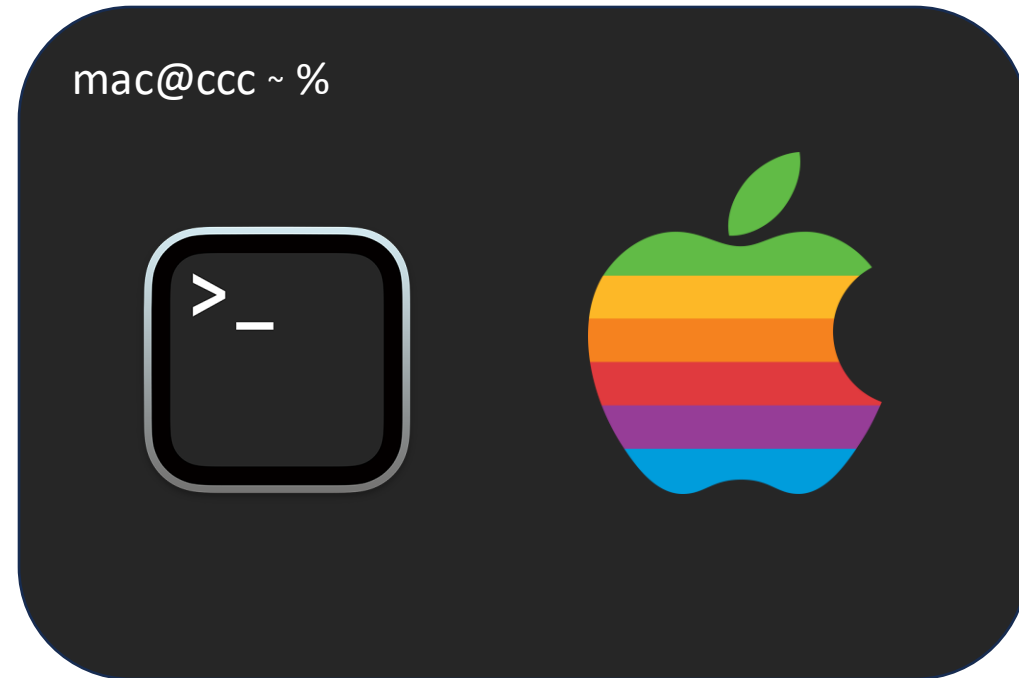
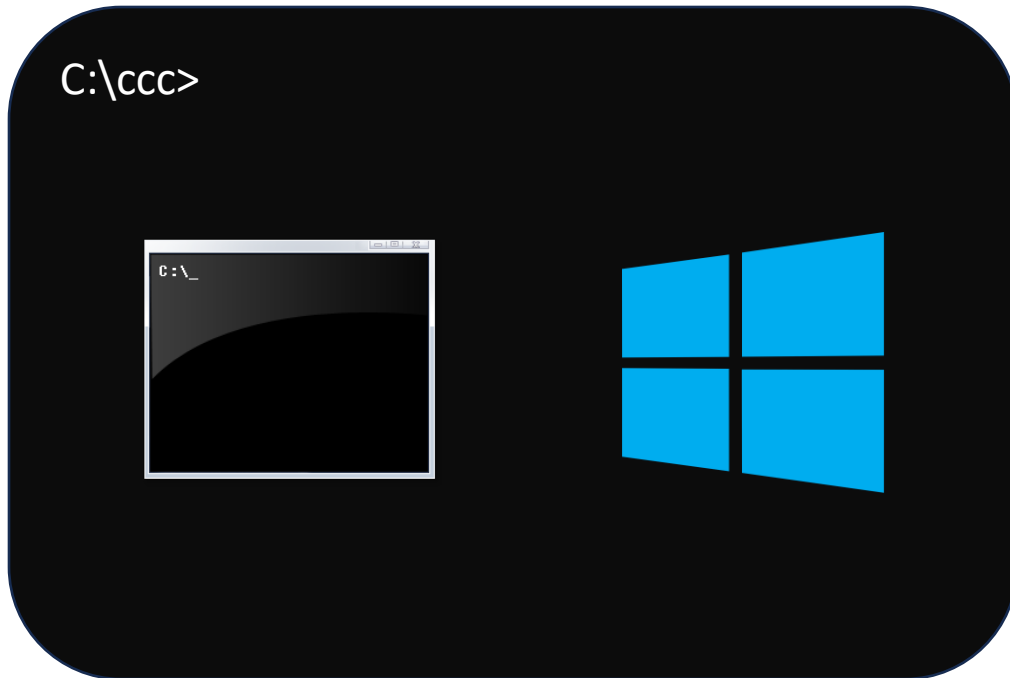
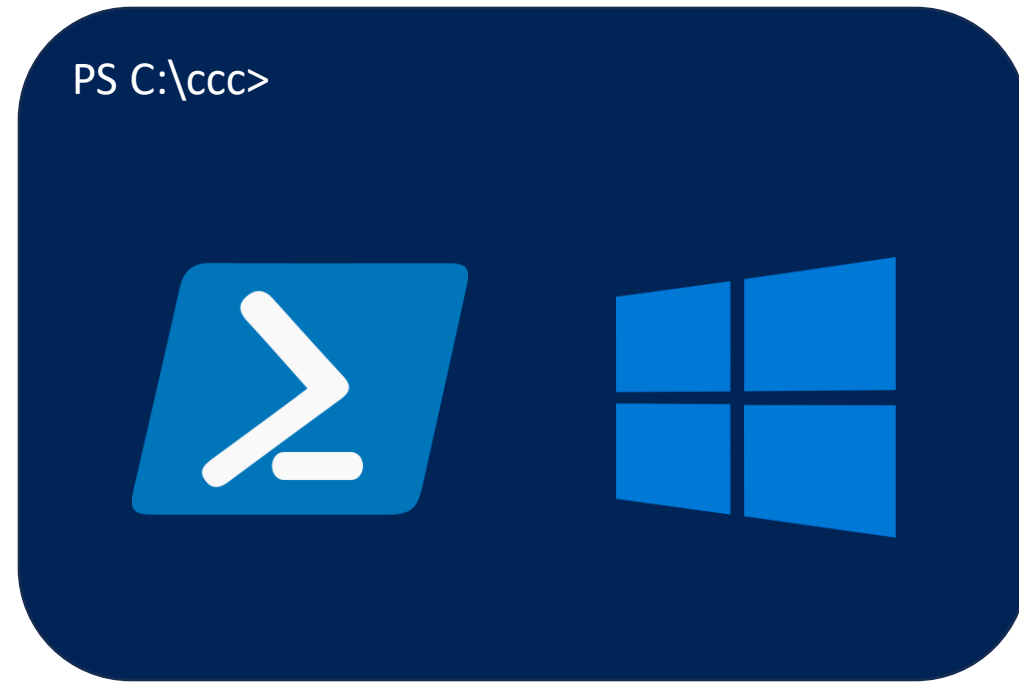
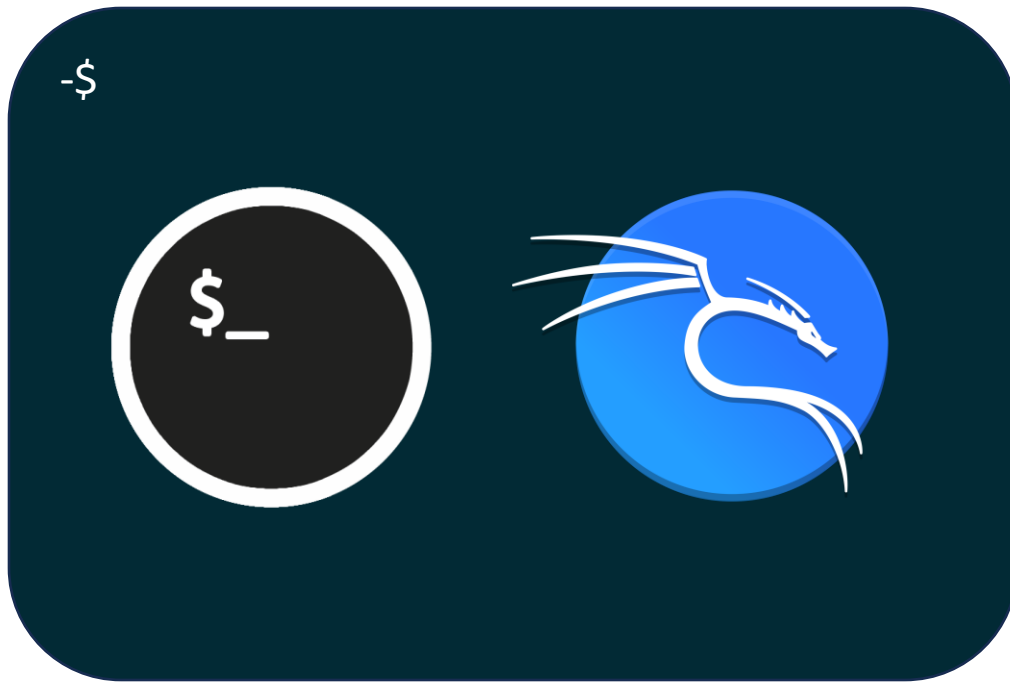
```
kali@kali:~$ cd Desktop
kali@kali:~/Desktop$ cd Files
kali@kali:~/Desktop/Files$ ls
image1.png  java.png  pics.png  picture.png  pp.png  screen.png
kali@kali:~/Desktop/Files$ rm pics.png
kali@kali:~/Desktop/Files$ ls
image1.png  java.png  picture.png  pp.png  screen.png
```

# Kernal vs Shell vs Application

- **Application:** the highest level in the software stack of an operating system.
- Consists of **user-level applications** and software that run on top of the operating system.
- Applications can include *web browsers, word processors, email clients, games,* and many other types of software.



# Modern CLIs





# Terminal Apps



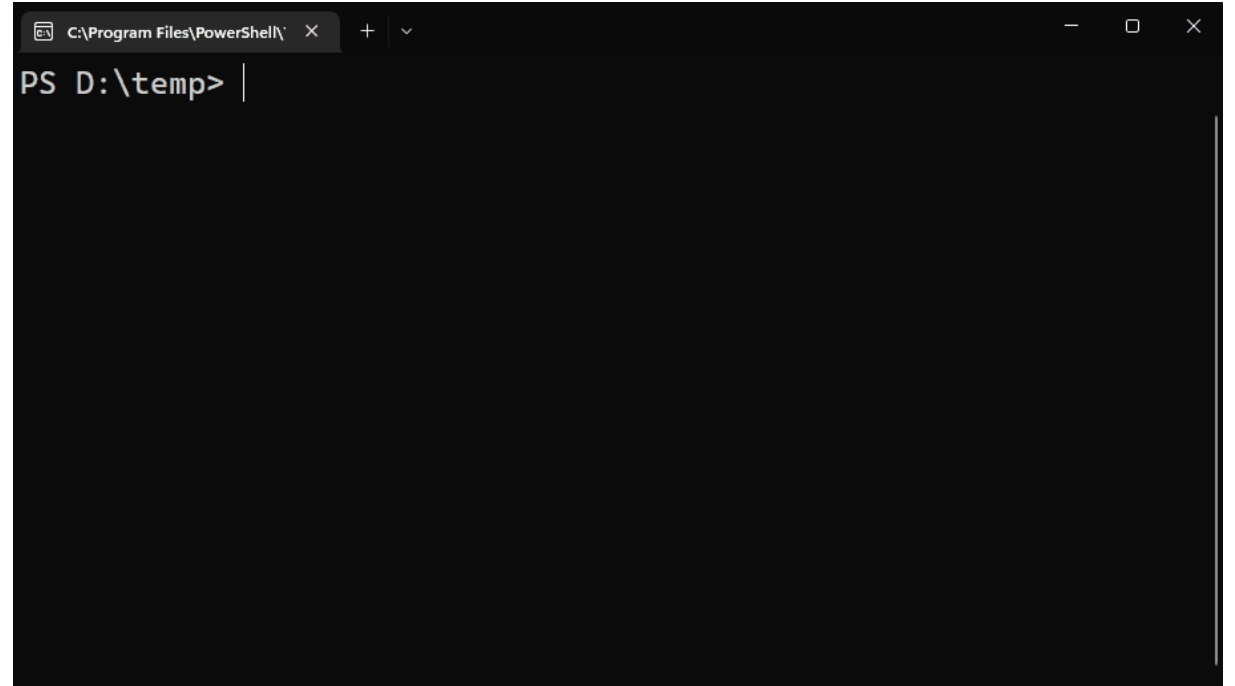
Windows Terminal



Mac iTerm2

# Command Line Tips

- Use **Tab** key to auto complete commands
- Use **Up arrow** key to sort through previously used commands
- Use **CTRL-C** to end a command early
- **Right Click** to copy and paste



# List File Command

```
-$ ls
```

```
Desktop Documents Downloads Music  
Pictures Public Templates Videos
```

```
PS C:\ccc> ls
```

```
Directory of C:\
```

```
12/07/2019 02:14 AM <DIR> PerfLogs  
09/19/2023 09:05 PM <DIR> Program Files  
11/19/2020 12:33 AM <DIR> Program Files (x86)  
09/19/2023 09:07 PM <DIR> Users  
09/19/2023 08:57 PM <DIR> Windows  
0 File(s) 0 bytes  
5 Dir(s) 41,916,641,280 bytes free
```

```
C:\ccc> dir
```

```
Directory of C:\
```

```
12/07/2019 02:14 AM <DIR> PerfLogs  
09/19/2023 09:05 PM <DIR> Program Files  
11/19/2020 12:33 AM <DIR> Program Files (x86)  
09/19/2023 09:07 PM <DIR> Users  
09/19/2023 08:57 PM <DIR> Windows  
0 File(s) 0 bytes  
5 Dir(s) 41,916,641,280 bytes free
```

```
mac@ccc ~ % ls
```

```
Desktop Downloads Movies Pictures  
Documents Library Music Public
```

Current  
Working  
Path  
Command

```
-$ pwd  
  
/home/kali
```

```
PS C:\ccc> pwd  
  
Path  
----  
C:\Users\Public
```

```
C:\ccc> cd  
  
C:\Users\Public
```

```
mac@ccc ~ % pwd  
  
/Users/TimApple
```

# Switching Directory Command

```
-$ cd Downloads
```

**\*In Downloads Directory**

```
-$ cd ..
```

**\*Back to Parent Directory**

```
-$ cd /
```

**\*Back to Root Directory**

```
-$ cd /home/kali/Downloads
```

**\*Back to Downloads Directory**

```
PS C:\ccc> cd .\Downloads\
```

**\*In Downloads Directory**

```
PS C:\ccc\...> cd ..
```

**\*Back to Parent Directory**

```
PS C:\ccc\...> cd C://
```

**\*Back to Root Directory**

```
PS C:\ccc\...> cd Users\Public\Downloads
```

**\*Back to Downloads Directory**

```
C:\ccc> cd Downloads
```

**\*In Downloads Directory**

```
C:\ccc\...> cd ..
```

**\*Back to Parent Directory**

```
C:\ccc\...> cd C://
```

**\*Back to Root Directory**

```
C:\ccc\...> cd Users\Public\Downloads
```

**\*Back to Downloads Directory**

```
mac@ccc ~ % cd Downloads
```

**\*In Downloads Directory**

```
mac@ccc ~ % cd ..
```

**\*Back to Parent Directory**

```
mac@ccc ~ % cd /
```

**\*Back to Root Directory**

```
mac@ccc ~ % /Users/TimApple/Downloads
```

**\*Back to Downloads Directory**

## New File Command

```
-$ touch demo.txt
```

```
-$ ls
```

```
-$ demo.txt
```

```
PS C:\ccc> ni demo.txt
```

```
PS C:\ccc\...> dir
```

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	9/30/2023 12:33 PM		0 demo.txt

```
C:\ccc> copy nul demo.txt
```

```
1 file(s) copied.
```

```
C:\ccc\...> dir
```

09/30/2023 12:29 PM	<DIR>	.
09/30/2023 12:29 PM	<DIR>	..
09/30/2023 12:29 PM		0 demo.txt
1 File(s)	0 bytes	
2 Dir(s)	34,306,015,232 bytes free	

```
mac@ccc ~ % touch demo.txt
```

```
mac@ccc ~ % ls  
demo.txt
```

# Edit a File Command

```
-$ vim demo.txt
```

~

~

"demo.txt" 0L, 0B 0,0-1 All

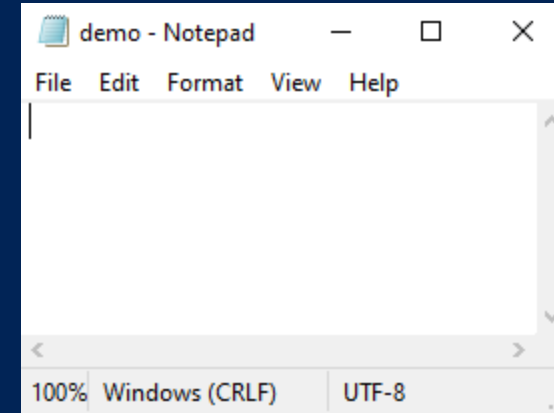
**\*Inside of the VIM editor**

**(\*Shiftkey + :) + wq**

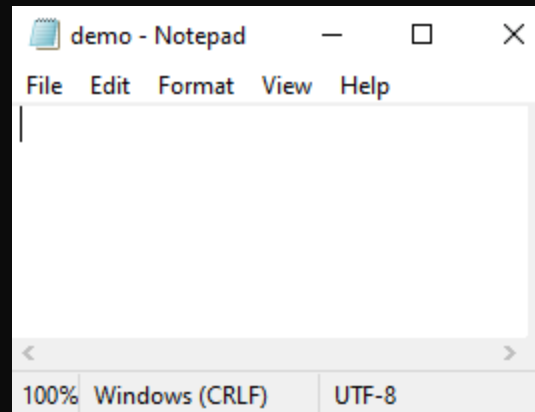
:wq

**\*Exited vim**

```
PS C:\ccc> notepad demo.txt
```



```
C:\ccc> notepad demo.txt
```



```
mac@ccc ~ % vim demo.txt
```

~

~

"demo.txt<200b>" [New]

**\*Inside of the VIM editor**

**(\*Shiftkey + :) + wq**

:wq

**\*Exited vim**

# File Quick View Command

```
-$ cat demo.txt
```

```
Hello World!
```

```
PS C:\ccc> gc demo.txt
```

```
Hello World!
```

```
C:\ccc> type demo.txt
```

```
Hello World!
```

```
mac@ccc ~ % cat demo.txt
```

```
Hello World!
```



# New Directory/ Folder Command

```
-$ mkdir DemoFolder
```

```
-$ ls  
DemoFolder demo.txt
```

```
-$ cd DemoFolder
```

```
-$ pwd  
/home/kali/Documents/DemoFolder
```

```
PS C:\ccc> mkdir DemoFolder
```

```
PS C:\ccc\...> dir
```

Mode	LastWriteTime	Length	Name
d----	9/30/2023 6:50 PM		DemoFolder

```
C:\ccc> mkdir DemoFolder
```

```
C:\ccc\...> dir  
09/30/2023 06:47 PM <DIR> .  
09/30/2023 06:47 PM <DIR> ..  
09/30/2023 01:02 PM      12 demo.txt  
09/30/2023 06:47 PM <DIR> DemoFolder  
1 File(s)      12 bytes  
3 Dir(s) 34,302,484,480 bytes free
```

```
mac@ccc ~ % mkdir DemoFolder
```

```
mac@ccc ~ % ls  
DemoFolder demo.txt
```

```
mac@ccc ~ % cd DemoFolder
```

```
mac@ccc ~ % pwd  
/home/kali/Documents/DemoFolder
```

## Delete Commands

```
-$ ls
```

```
DemoFolder demo.txt
```

```
-$ rm demo.txt
```

```
-$ rmdir DemoFolder
```

```
-$ ls
```

```
*Empty
```

```
PS C:\ccc> dir
```

```
d----- 9/30/2023 6:50 PM DemoFolder
```

```
-a----- 9/30/2023 1:02 PM 12 demo.txt
```

```
PS C:\ccc\...> rm demo.txt
```

```
PS C:\ccc\...> rmdir DemoFolder
```

```
PS C:\ccc\...> dir
```

```
*Empty
```

```
C:\ccc> dir
```

```
09/30/2023 07:00 PM 0 demo.txt
```

```
09/30/2023 06:59 PM <DIR> DemoFolder
```

```
C:\ccc\...> del demo.txt
```

```
C:\ccc\...> rmdir DemoFolder
```

```
C:\ccc\...> dir
```

```
*Empty
```

```
mac@ccc ~ % ls
```

```
DemoFolder demo.txt
```

```
mac@ccc ~ % rm demo.txt
```

```
mac@ccc ~ % rmdir DemoFolder
```

```
mac@ccc ~ % ls
```

```
*Empty
```

# Chmod

The "**chmod**" command, short for "change mode," is a command used in Unix-like operating systems

Used change the permissions (mode) of files and directories. It allows users to specify who can read, write, and execute a file or directory.

User | Groups | Others

```
-$ chmod 764 demo.txt
```

**\*This changes the permissions on the file using a binary number system**

**1 = execute**

**2 = write**

**4 = read**

**the first number in the set represents the user**

**the second number represent the group**

**the third number represents others\***

# Chmod Chart

Octal	Binary	File Mode
0	000	---
1	001	--X
2	010	-W-
3	011	-WX
4	100	r--
5	101	r-X
6	110	rw-
7	111	rwx

# Linux file permissions cheatsheet

@sysxplore

```
sysxplore@zorinos:~$ ls -l
total 93792
```

```
-rwx-rw-r-- 1 1 sysxplore sysxplore 20 Jul 18 4096 backup.sh
-rw-rw-r-- 1 1 sysxplore sysxplore 0 Jun 23 15 12:59 demo
drwxr-xr-x 2 2 sysxplore sysxplore 4096 May 15 10:42 Desktop
```

Number of linked hard-links  
Owner of the file  
Group that own the file  
File size in bytes  
File modification date and time  
File name

File type  
User permissions  
Group permissions  
Other (everyone) permissions

User Owner Permissions Group Owner Permissions Other Permissions

**-rwxrw-r--**

r	Read	4
w	Write	2
x	Execute	1
7		

r	Read	4
w	Write	2
-	No permission	0
6		

r	Read	4
-	No permission	0
-	No permission	0
4		

SUID Permission

Group Owner Permissions

User Owner Permissions Other Permissions

**-rwsrw-r--**

\$ chmod u+s file

SGID Permission

Group Owner Permissions

User Owner Permissions Other Permissions

**drwxrwsr--**

\$ chmod g+s directory\_name

Sticky Bit Permission

Group Owner Permissions

User Owner Permissions Other Permissions

**drwxrwxrwt**

\$ chmod +t directory\_name

Binary	Octal	Permissions	Representation
000	0 (0+0+0)	No Permission	---
001	1 (0+0+1)	Execute	--x
010	2 (0+2+0)	Write	-w-
011	3 (0+2+1)	Write + Execute	-wx
100	4 (4+0+0)	Read	r--
101	5 (4+0+1)	Read + Execute	r-x
110	6 (4+2+0)	Read + Write	rw-
111	7 (4+2+1)	Read + Write + Execute	rwX

Owner	Group	Other
r w x	r w x	r w x
S s	S s	T t

Capital S is an error it occurs if you set SUID bit or SGID bit to a file without execute (x) bit set

Capital T is an error it occurs if you set Sticky bit to a file without execute (x) bit set

@sysxplore

```
-$ chmod 764 demo.txt
```

```
-$ ls -l
```

```
total 0
```

```
-rwxrw-r-- 1 kali kali 0 Oct  2 23:20 demo.txt
```

```
-$ chmod 432 demo.txt
```

```
-$ ls -l
```

```
total 0
```

```
-r---wx-w- 1 kali kali 0 Oct  2 23:20 demo.txt
```

# SSH

```
-$ ssh StarID@199.17.28.80
```

```
StarID@199.17.28.80's password: ****
```

```
PS C:\ccc> ssh StarID@199.17.28.80
```

```
StarID@199.17.28.80's password: ****
```

```
C:\ccc> ssh StarID@199.17.28.80
```

```
StarID@199.17.28.80's password: ****
```

```
mac@ccc ~ % ssh StarID@199.17.28.80
```

```
StarID@199.17.28.80's password: ****
```

Practice



# Live Activity

- Make two folders inside of the Downloads directory
  - CCC
  - CCDC
- Make a new txt file in both
  - Ccc.txt
  - Ccdc.txt
- Use notepad or VIM to add text to the files
- Cat the files data to the screen
- Give custom permissions to ccc.txt
  - User = rwx
  - Group = rw
  - Other = r
- Show updated permissions in terminal