

# Unit - V

**Managing Files and Directories** : Create and edit text files, Search for files, Perform operations on files and directories, Process text files, Manipulate file output

**Managing Kernel Modules** : Explore the linux kernel, Install and configure kernel modules, Monitor kernel modules,

**Managing the Linux Boot Process** : Configure linux boot components, Configure GRUB2

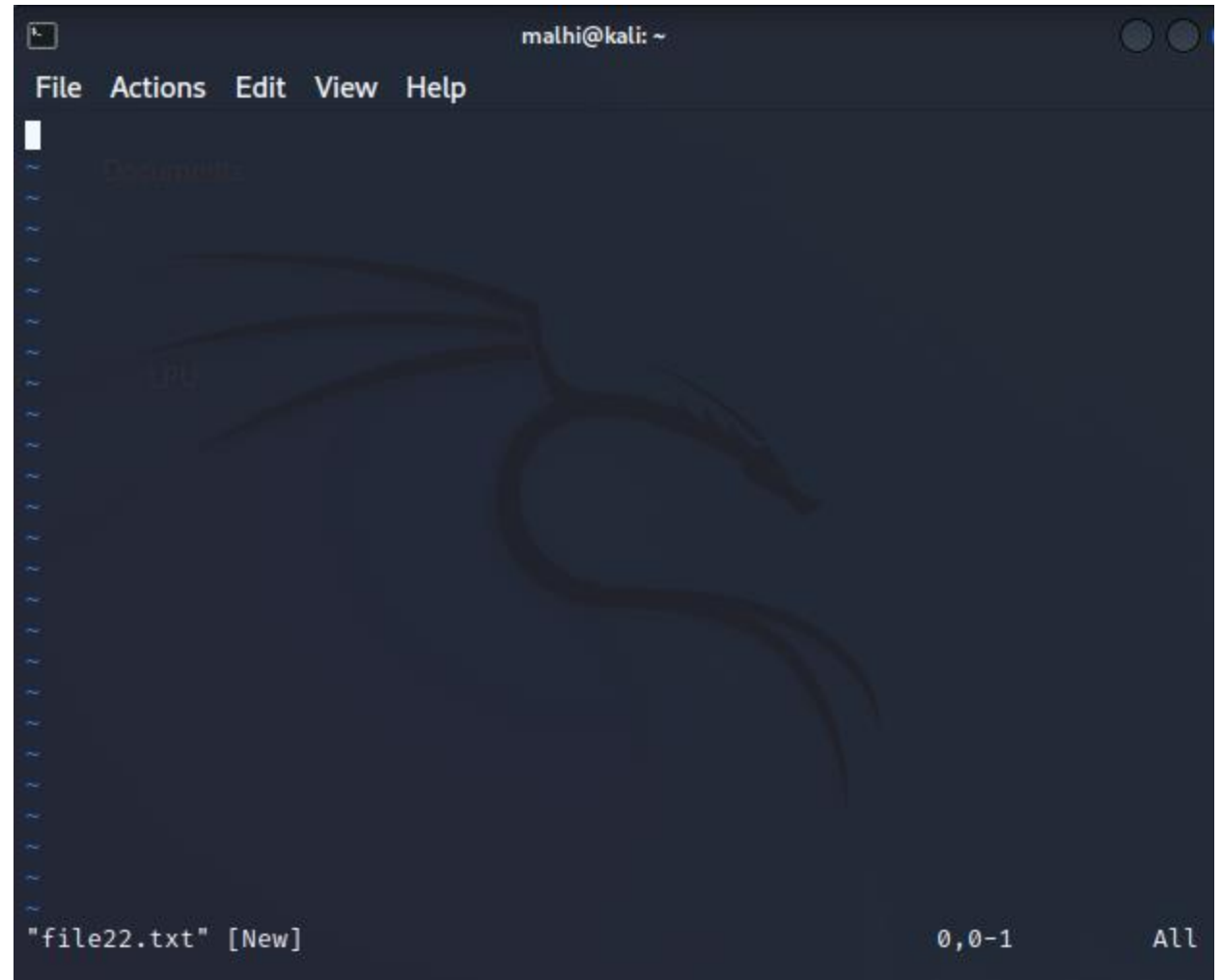
# Create and Edit Text Files

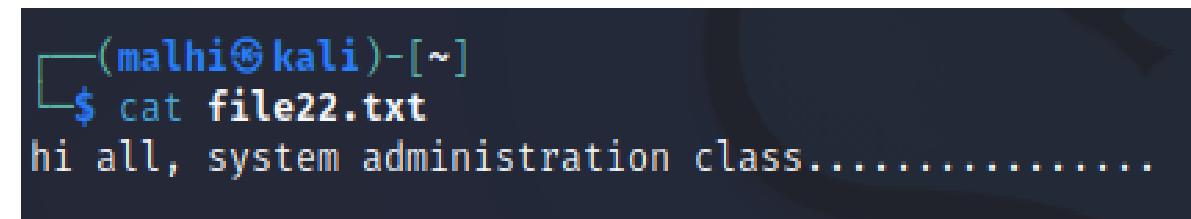
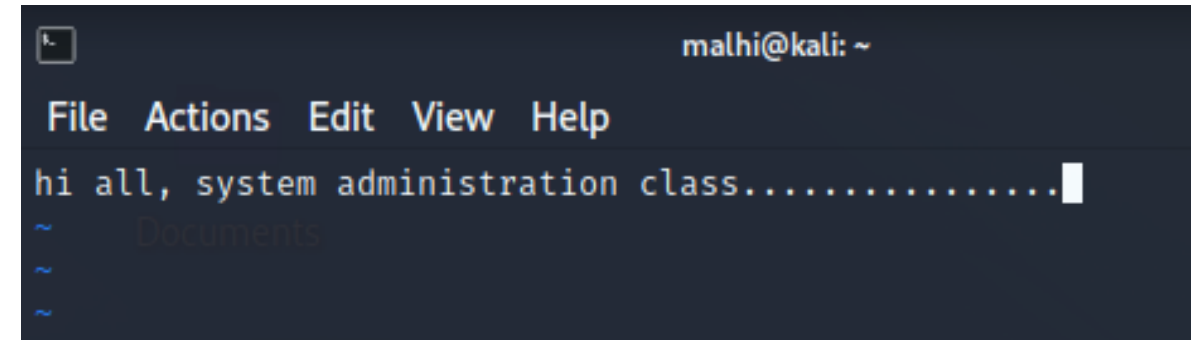
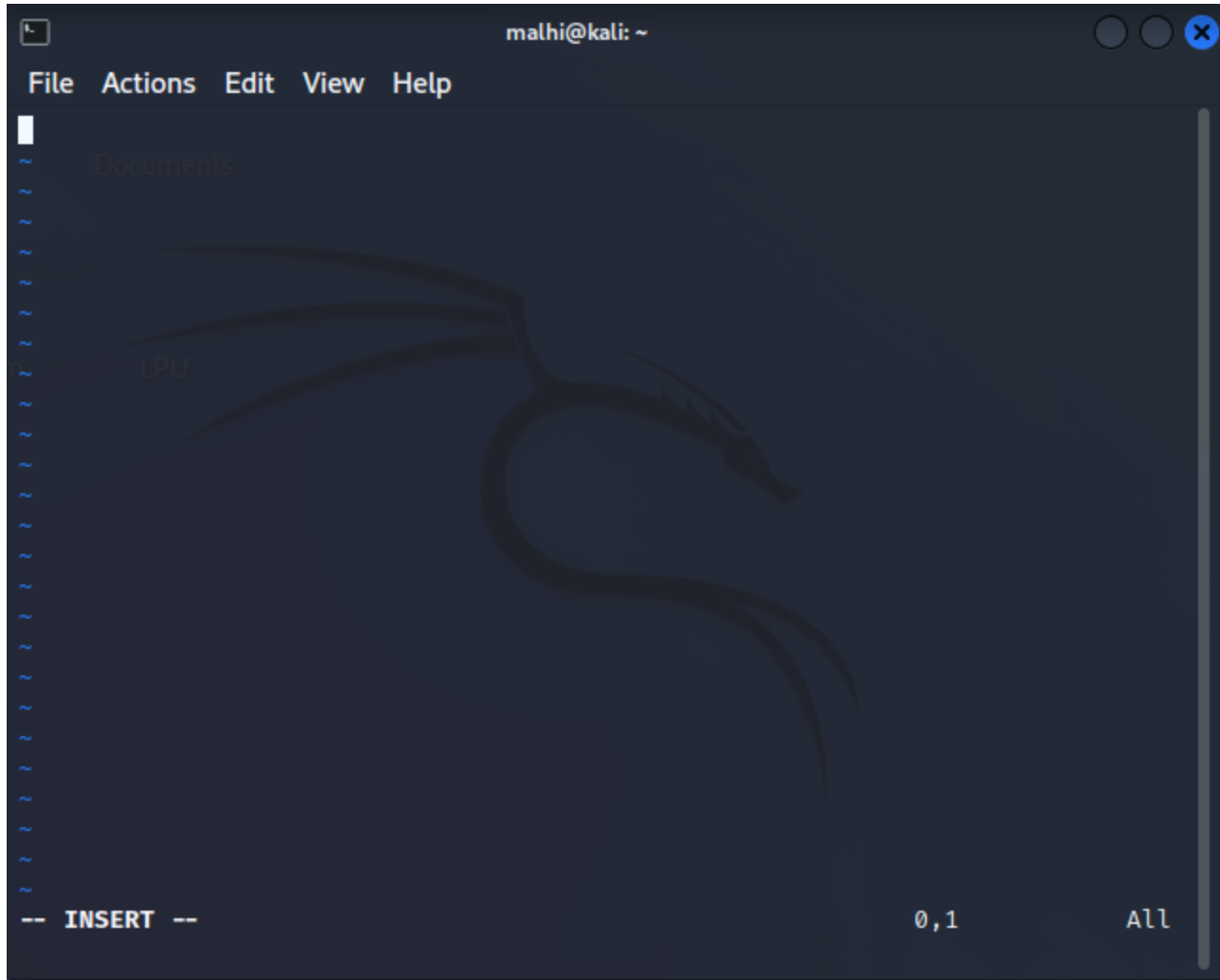
- TEXT EDITORS: A text editor is an application that enables you to view, create, or modify the contents of text files.
- Vim
- Nano

# vim

```
(malhi@kali)-[~]  
$ vim file22.txt
```

- Press i to enter into insert mode
- Then write your content
- After writing, exit from insert mode by pressing escape key
- To save write **:wq**

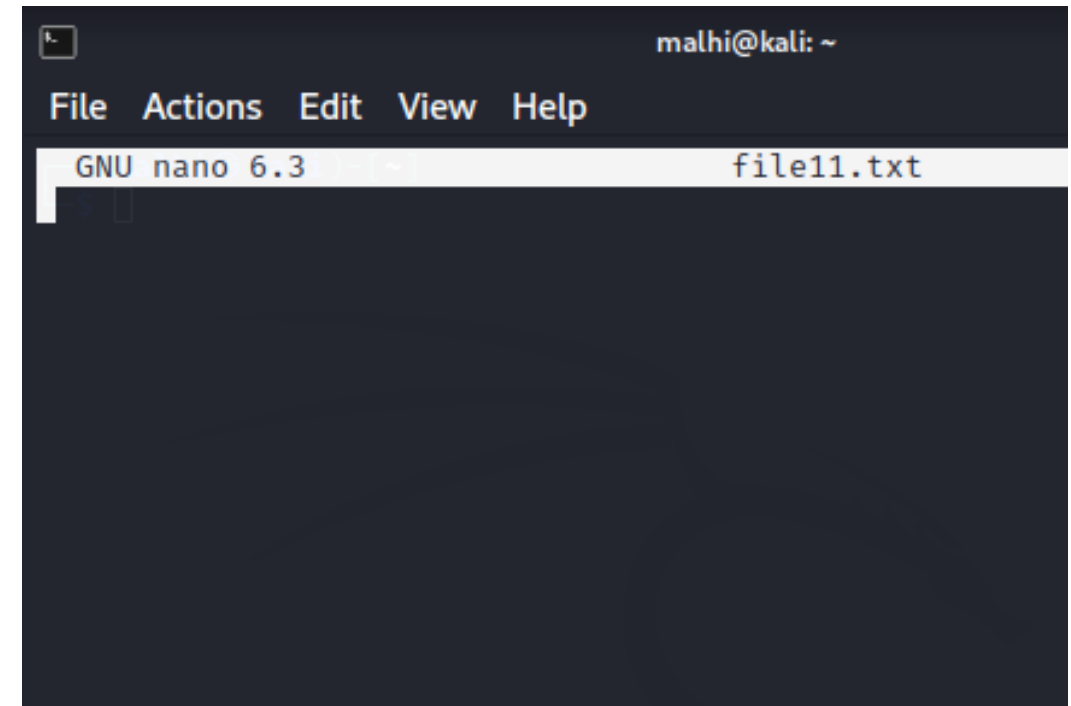




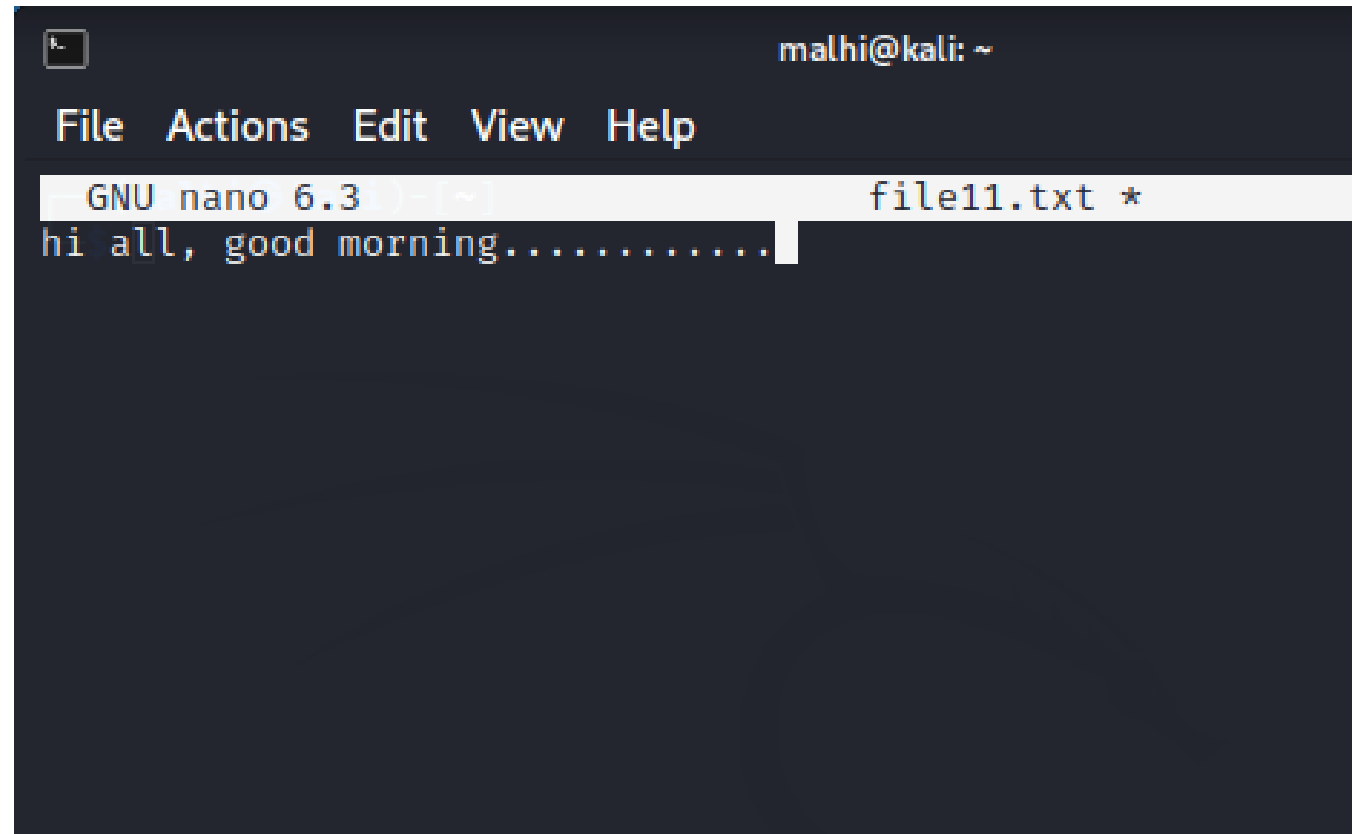
# nano

- nano

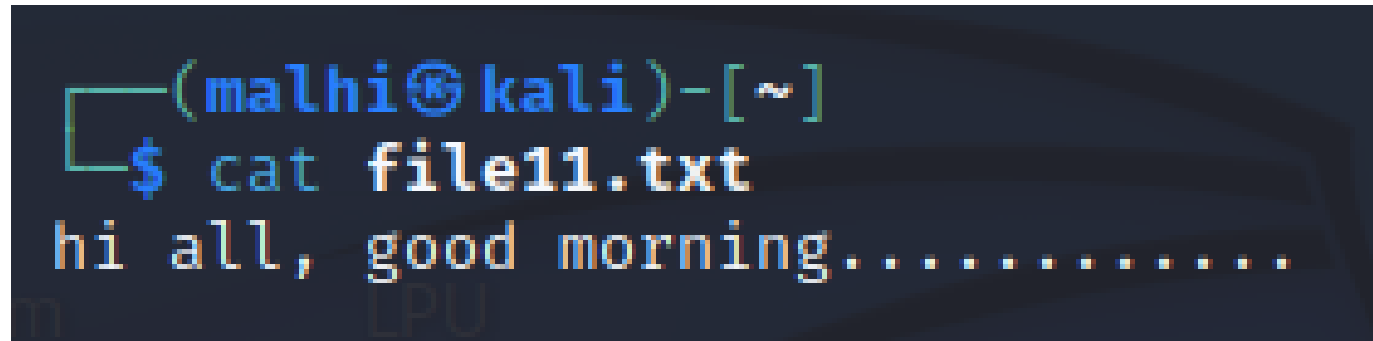
```
(malhi@kali)-[~]  
$ nano file11.txt
```



- Ctrl + O to save
- Ctrl + C to cancel the current work
- Ctrl + X to exit
- Then press Y



```
malhi@kali: ~
File Actions Edit View Help
GNU nano 6.3 file11.txt *
hi all, good morning.....
```



```
(malhi@kali)-[~]
$ cat file11.txt
hi all, good morning.....
```

# Search for Files

- There will be times that you create a file and forget where in the directory structure you put it.
- There will also be times when you don't know the exact location of files created by the system, applications, or other users.

# THE locate COMMAND

- The `locate` command performs a quick search for any specified string in file names and paths stored in the `mlocate` database.

```
(root@kali)-[~]  
# locate Desktop  
/home/malhi/Desktop  
/usr/lib/python3/dist-packages/xdg/DesktopEntry.py  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front-Speaker-Headset.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear-Line.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear.conf  
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop-HiFi.conf  
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop.conf  
/usr/share/dbus-1/services/org.freedesktop.portal.Desktop.service  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.bitwarden.BitwardenDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.elsevier.MendeleyDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.onlyoffice.DesktopEditors.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.wire.WireDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/io.github.shiftey.Desktop.svg  
/usr/share/man/man3/File::DesktopEntry.3pm.gz  
/usr/share/perl5/File/DesktopEntry.pm  
/usr/share/powershell-empire/empire/server/csharp/Covenant/Data/ReferenceSourceLibraries/Seatbelt  
/Seatbelt/Commands/Products/RemoteDesktopConnectionManagerCommand.cs
```



- If locate command is not working, then needs to update the db used by locate command.
  - `sudo updatedb`

# locate COMMAND OPTIONS

| Option | Used To  |
|--------|--|
| -r     | Search for file names using regular expressions.                               |
| -c     | Display only the number of matching entries found, rather than the file names. |

```
(root@kali)-[~]
# locate -r Desktop
/home/malhi/Desktop
/usr/lib/python3/dist-packages/xdg/DesktopEntry.py
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front-Speaker-Headset.conf
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front.conf
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear-Line.conf
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear.conf
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop-HiFi.conf
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop.conf
/usr/share/dbus-1/services/org.freedesktop.portal.Desktop.service
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.bitwarden.BitwardenDesktop.svg
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.elsevier.MendeleyDesktop.svg
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.onlyoffice.DesktopEditors.svg
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.wire.WireDesktop.svg
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/io.github.shiftey.Desktop.svg
/usr/share/man/man3/File::DesktopEntry.3pm.gz
/usr/share/perl5/File/DesktopEntry.pm
/usr/share/powershell-empire/empire/server/csharp/Covenant/Data/ReferenceSourceLibraries/Seatbelt
/Seatbelt/Commands/Products/RemoteDesktopConnectionManagerCommand.cs
```

```
(root@kali)-[~]
# locate -c Desktop
17
```

| Option                 | Used To   |
|------------------------|---|
| -e                     | Return only files that exist at the time of search.           |
| -i                     | Ignore the casing in file names or paths.                     |
| -n {number of entries} | Return only the first few matches up to the specified number. |

```
(root@kali)-[~]  
# locate -e Desktop  
/home/malhi/Desktop  
/usr/lib/python3/dist-packages/xdg/DesktopEntry.py  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front-Speaker-Headset.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear-Line.conf  
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear.conf  
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop-HiFi.conf  
/usr/share/alsa/ucm2/USB-Audio/Realtek/ALC1220-VB-Desktop.conf  
/usr/share/dbus-1/services/org.freedesktop.portal.Desktop.service  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.bitwarden.BitwardenDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.elsevier.MendeleyDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.onlyoffice.DesktopEditors.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/com.wire.WireDesktop.svg  
/usr/share/icons/Flat-Remix-Blue-Dark/apps/scalable/io.github.shiftey.Desktop.svg  
/usr/share/man/man3/File::DesktopEntry.3pm.gz  
/usr/share/perl5/File/DesktopEntry.pm  
/usr/share/powershell-empire/empire/server/csharp/Covenant/Data/ReferenceSourceLibraries/Seatbelt  
/Seatbelt/Commands/Products/RemoteDesktopConnectionManagerCommand.cs
```

**(root@kali)-[~]**

**#** locate -i Desktop

/etc/apparmor.d/abstractions/freedesktop.org

/etc/apparmor.d/abstractions/xdg-desktop

/etc/dbus-1/system.d/org.freedesktop.DisplayManager.conf

/etc/dbus-1/system.d/org.freedesktop.GeoClue2.Agent.conf

/etc/dbus-1/system.d/org.freedesktop.GeoClue2.conf

/etc/dbus-1/system.d/org.freedesktop.ModemManager1.conf

/etc/skel/.config/autostart/fix-duplicated-xfce-panel-launcher.desktop

```
(root@kali)-[~]
```

```
# locate -n 5 Desktop
```

```
/home/malhi/Desktop
```

```
/usr/lib/python3/dist-packages/xdg/DesktopEntry.py
```

```
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front-Speaker-Headset.conf
```

```
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Front.conf
```

```
/usr/share/alsa/ucm2/USB-Audio/Dell/Desktop-Rear-Line.conf
```

# THE find COMMAND

- The `find` command enables you to search a specific location for files and directories that adhere to some search criteria.
- The `-type` option enables you to specify the type of object you're looking for, such as `d` for directory or `f` for file. The `-name` option is where you specify the name of the object you're looking for.
- `find /home/user -type f -name 2019 report`

```
(root@kali)-[~]  
# find /root/LPU -type f -name F33.txt  
/root/LPU/F33.txt
```

```
(malhi@kali)-[~]  
$ find ./Desktop -name abc.txt  
./Desktop/Documents/Desktop/abc.txt  
./Desktop/abc.txt
```

```
DOCUMENTS  
└─(malhi@kali)-[~]  
└─$ ls Desktop  
abc.txt Documents LPU
```

```
└─(malhi@kali)-[~]  
└─$ find ./Desktop/abc.txt -delete
```

```
└─(malhi@kali)-[~]  
└─$ ls Desktop  
Documents LPU
```



```
(malhi@kali)-[~]
```

```
$ ls Desktop/Documents/Desktop
```

```
abc.txt '[Documents]' '[LPU]' qwerty SA '[XYZZ]'
```

```
def.txt ghi.txt qwe rty '[WWW]'
```

```
(malhi@kali)-[~]  
$ find ./Desktop/Documents/Desktop -print  
./Desktop/Documents/Desktop  
./Desktop/Documents/Desktop/rty  
./Desktop/Documents/Desktop/[Documents]  
./Desktop/Documents/Desktop/[XYZZ]  
./Desktop/Documents/Desktop/ghi.txt  
./Desktop/Documents/Desktop/qwe  
./Desktop/Documents/Desktop/SA  
./Desktop/Documents/Desktop/[WWW]  
./Desktop/Documents/Desktop/abc.txt  
./Desktop/Documents/Desktop/[LPU]  
./Desktop/Documents/Desktop/qwerty  
./Desktop/Documents/Desktop/def.txt
```

# find VS. locate COMMANDS

- The `locate` command searches a database and retrieves information on files present on your system.
- However, *failure to keep this database updated* may produce outdated results.
- The `find` command, on the other hand, performs a live search of the file system and may concentrate on a specific location.
- The `find` command may take more time to complete a search than the `locate` command.

`locate` searches  
the `mlocate`  
database



`find` searches  
the drive



*The find vs. locate commands.*

- What are some advantages of using the `find` command over using the `locate` command?
- The `locate` command requires that a database be updated in order to perform accurate searches, whereas `find` does not. Also, `locate` cannot filter its search by specific directories, whereas `find` can. However, `locate` may be able to perform searches more quickly in certain cases.

# THE which COMMAND

- The `which` command displays the complete path of a specified command by searching the directories assigned to the `PATH` variable.

```
(malhi@kali)-[~]  
$ which cat  
/usr/bin/cat
```

# THE whereis COMMAND

- The `whereis` command is used to display various details associated with a command.

```
(root@kali)-[~]  
# whereis ls  
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
```

- Where `/bin/ls` indicates the location of the `ls` command and `/usr/share/man/man1/ls.1.gz` indicates the location of the man pages for the `ls` command.

Displaying location information for a command.

```
(malhi@kali)-[~]
```

```
$ whereis cat
```

```
cat: /usr/bin/cat /usr/share/man/man1/cat.1.gz
```



# Perform Operations on Files and Directories

- There are many ways you can manage a file once you've created and edited it to your liking.
- **THE cat COMMAND:** The `cat` command, short for concatenate, can display, combine, and create text files. It is most frequently used to display the contents of small text files, as it does not have a screen scrolling capability.

```
(malhi@kali)-[~]
```

```
$ cat -n f111.txt
```

```
1 Hi all,,,this is system administration class.
```

```
2
```

```
3
```

```
4 you are in LPU.
```

```
5 section is KM003.
```

```
(malhi@kali)-[~]
```

```
$ cat -b f111.txt
```

```
1  Hi all,,,this is system administration class.
```

```
2  you are in LPU.
```

```
3  section is KM003.
```

```
(malhi@kali)-[~]
```

```
$ cat -s f111.txt
```

```
Hi all,,,this is system administration class.
```

```
you are in LPU.
```

```
section is KM003.
```

```
(malhi@kali)-[~]
```

```
$ cat -e f111.txt
```

```
Hi all,,,this is system administration class.$
```

```
$
```

```
$
```

```
you are in LPU.$
```

```
section is KM003.$
```

# THE head AND tail COMMANDS

- The head command displays the first 10 lines of each file. The tail command displays the last 10 lines of each file.
- These commands are useful when you only need to see the beginning or the end of a file.

# head

- displays the first 10 lines

```
(malhi@kali)-[~/Desktop]
$ head test.txt
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
Hi all.....
```

# tail

- displays the last 10 lines

```
(malhi@kali)-[~/Desktop]
$ tail test.txt
You are in Lovely Professional University.
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
Hi all.....
This is System Administration Class.
You are in Lovely Professional University.
```

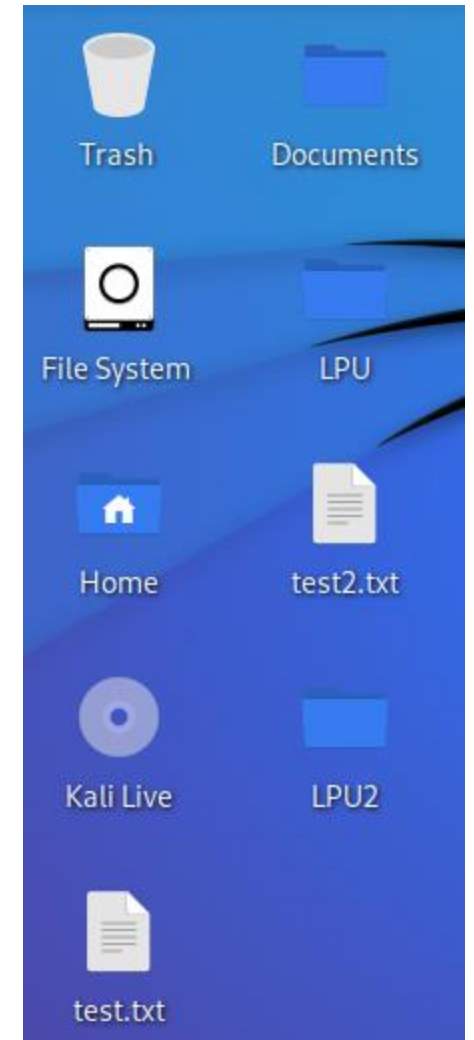


# THE `cp` COMMAND

- The `cp` command enables you to copy and then paste a file or directory.
- The initial object is left where it is, but an exact duplicate of that object is created at the destination you specify.
- When you copy directories, you must specify the `-R` option to copy the specified directory recursively.

```
You(malhi@kali)-[~/Desktop]al Unive
└─$ cp test.txt test2.txt
This is System Administration Class
You(malhi@kali)-[~/Desktop]al Unive
└─$ ls
Documents@ LPU test2.txt test.txt
```

```
Hi all.....  
This (malhi@kali)-[~/Desktop]n Class.  
$ cpe-Rn LPU LPU2 Professional University.  
Hi all.....  
This (malhi@kali)-[~/Desktop]n Class.  
$ lse in Lovely Professional University.  
Documents.. LPU.. LPU2 test2.txt test.txt
```



# THE `mv` COMMAND

- The `mv` command moves files and directories to other locations.
- It is similar to the `cp` command, but does not leave the initial object in place.
- Therefore, `mv` is more like a cut and paste operation.

```
(malhi@kali) - [~/Desktop]  
$ mv test2.txt LPU2
```

```
(malhi@kali) - [~/Desktop]  
$ cd LPU2
```

```
(malhi@kali) - [~/Desktop/LPU2]  
$ ls  
test2.txt
```

# ls COLORS

- In the Bash shell, when you execute the `ls` command, you may have noticed that the results sometimes appear in different colors.
- These colors distinguish different types of files.
- By default, some of the colors are:
  - Default color: Normal/text file
  - Blue: Directory
  - Sky blue: Symbolic link or audio file
  - Green: Executable file

- Yellow with black background: Device
- Pink: Image file
- Red: Archive file
- Red with black background: Broken link

# THE mkdir AND rmdir COMMANDS

- The mkdir command is used to create (or make) a directory. You supply the name of the directory as an argument.
- The rmdir command is used to remove directories, but only those that are empty (i.e., contain no files or subdirectories).

```
(malhi@kali)-[~/Desktop]  
$ rmdir LPU2  
rmdir: failed to remove 'LPU2': Directory not empty
```



- In order to delete a with actual contents, you must use the `rm -R` command.

```
(malhi@kali) - [~/Desktop]  
$ rm -R LPU2
```

```
(malhi@kali) - [~/Desktop]  
$ ls  
Documents  LPU  test.txt
```

# Process Text Files

- **THE echo COMMAND**

The `echo` command is used to display a line of text on the terminal.

- You can also use the `echo` command to write text to a file by providing the string after the `echo` command and redirecting to the file.

```
(malhi@kali)-[~]  
$ echo hi all.....  
hi all.....
```

```
(malhi@kali)-[~]  
$ echo "hi all....." > test.txt  
  
(malhi@kali)-[~]  
$ cat test.txt  
hi all.....
```

- The echo command can **overwrite** the content in the file by redirecting and **not appending**.

# THE printf COMMAND

- The `printf` command is similar to `echo`, but provides the user with much more control over how the output is formatted.
- You can supply various format characters within the text you want to output, using a backslash (`\`) to indicate when they are being used.

```
(malhi@kali)-[~/Desktop]
$ printf "Hi all ... \n This is System Administration Class ..."
Hi all ...
This is System Administration Class ...
```

# THE `tr` COMMAND

- The `tr` command is used to translate a string of characters.
- It is predominantly used to change the case of letters in a file.
- This command acts only on a stream of characters and does not accept file names as arguments.
- You must use redirection to actually change a file.

```
(malhi@kali)-[~/Desktop]
$ cat test.txt | tr [a-l] [A-L]
HI ALL.....
THIS Is SyStEm ADmInIstrAtIon CLAss.
You ArE In LovELy ProFEssIonAL UnIvErSity.
HI ALL.....
THIS Is SyStEm ADmInIstrAtIon CLAss.
You ArE In LovELy ProFEssIonAL UnIvErSity.
HI ALL.....
THIS Is SyStEm ADmInIstrAtIon CLAss.
You ArE In LovELy ProFEssIonAL UnIvErSity.
HI ALL.....
THIS Is SyStEm ADmInIstrAtIon CLAss.
You ArE In LovELy ProFEssIonAL UnIvErSity.
HI ALL.....
THIS Is SyStEm ADmInIstrAtIon CLAss.
You ArE In LovELy ProFEssIonAL UnIvErSity.
```

```
(malhi@kali)-[~/Desktop]
$ tr [s-y] [S-Y] < test.txt
Hi all.....
ThiS iS SYSTem AdminiSTraTion ClaSS.
YoU are in LoVeLY ProfeSSional UniVerSiTY.
Hi all.....
ThiS iS SYSTem AdminiSTraTion ClaSS.
YoU are in LoVeLY ProfeSSional UniVerSiTY.
Hi all.....
ThiS iS SYSTem AdminiSTraTion ClaSS.
YoU are in LoVeLY ProfeSSional UniVerSiTY.
Hi all.....
ThiS iS SYSTem AdminiSTraTion ClaSS.
YoU are in LoVeLY ProfeSSional UniVerSiTY.
Hi all.....
ThiS iS SYSTem AdminiSTraTion ClaSS.
YoU are in LoVeLY ProfeSSional UniVerSiTY.
```

# THE `wc` COMMAND

- The word count (`wc`) command is used to count the number of lines, words, and characters in a text file.

```
(malhi@kali)-[~/Desktop]  
$ wc test.txt  
18  78 594 test.txt
```



- The `wc` command provides various options that enable you to specify the nature of the output.
- `-c` Display the byte count.

```
(malhi@kali)-[~]  
$ cat test.txt  
hi all.....
```

```
(malhi@kali)-[~]  
$ wc -c test.txt  
13 test.txt
```

```
(malhi@kali)-[~]  
$ wc test.txt  
1  2 13 test.txt
```

- -m Display the character count.

```
(malhi@kali)-[~]  
$ wc -m test.txt  
13 test.txt
```

- -l Display the newline count.

```
(malhi@kali)-[~]  
$ cat test.txt  
  
SA  
  
LPU  
  
(malhi@kali)-[~]  
$ wc -l test.txt  
4 test.txt
```

- -w Display the word count.

```
(malhi@kali)-[~]  
$ wc -w test.txt  
2 test.txt
```

- The sort command arranges the lines in a file.

```
(malhi@kali)-[~]  
$ sort test.txt
```

LPU

SA

- -k{column numbers} Specify field values. For example, -k2 indicates the second field.

```
(malhi@kali)-[~/Desktop]
$ cat tab_data.txt
```

|      |        |     |
|------|--------|-----|
| sem1 | ajay   | 7.8 |
| sem2 | riya   | 9.4 |
| sem4 | alok   | 6.8 |
| sem3 | abhay  | 9.1 |
| sem1 | lohit  | 8.3 |
| sem6 | gagan  | 8.9 |
| sem7 | amisha | 7.8 |

```
(malhi@kali)-[~/Desktop]
$ sort -k2 tab_data.txt
```

|      |        |     |
|------|--------|-----|
| sem3 | abhay  | 9.1 |
| sem1 | ajay   | 7.8 |
| sem4 | alok   | 6.8 |
| sem7 | amisha | 7.8 |
| sem6 | gagan  | 8.9 |
| sem1 | lohit  | 8.3 |
| sem2 | riya   | 9.4 |

- `-r` Sort fields in descending order.
- By default, the fields are sorted in ascending order.

```
(malhi@kali)-[~/Desktop]
$ sort tab_data.txt
sem1      ajay      7.8
sem1      lohith    8.3
sem2      riya      9.4
sem3      abhay     9.1
sem4      alok      6.8
sem6      gagan     8.9
sem7      amisha    7.8
```

```
(malhi@kali)-[~/Desktop]
$ sort -r tab_data.txt
sem7      amisha    7.8
sem6      gagan     8.9
sem4      alok      6.8
sem3      abhay     9.1
sem2      riya      9.4
sem1      lohith    8.3
sem1      ajay      7.8
```

# THE cut COMMAND

The `cut` command extracts the specified lines of text from a file.

```
(malhi@kali)-[~/Desktop]
$ cut -b 1-3 test.txt
Hi
Thi
You
Hi
Thi
You
Hi
Thi
You
Hi
Thi
You
Hi
Thi
You
Hi
Thi
You
Hi
Thi
You
```

```
(malhi@kali)-[~/Desktop]
└─$ cut -b -5 test.txt
Hi al
This
You a
Hi al
This
You a
Hi al
This
You a
Hi al
This
You a
Hi al
This
You a
Hi al
This
You a
```

```
(malhi@kali)-[~/Desktop]
└─$ cut -b 3- test.txt
all.....
is is System Administration Class.
u are in Lovely Professional University.
all.....
is is System Administration Class.
u are in Lovely Professional University.
all.....
is is System Administration Class.
u are in Lovely Professional University.
all.....
is is System Administration Class.
u are in Lovely Professional University.
all.....
is is System Administration Class.
u are in Lovely Professional University.
```



- `-c` Specify the number of the character to cut from each line.

```
(malhi@kali)-[~/Desktop]
$ cut -c 2,5,9 test.txt
il.
h S
oai
il.
h S
oai
il.
h S
oai
il.
h S
oai
il.
h S
oai
il.
h S
oai
```

```
(malhi@kali)-[~/Desktop]
$ cut -c 2-6 test.txt
i all
his i
ou ar
i all
his i
ou ar
i all
his i
ou ar
i all
his i
ou ar
i all
his i
ou ar
```



# THE paste COMMAND

- The `paste` command is used to merge lines from text files horizontally.

```
(malhi@kali)-[~/Desktop]
$ paste -d , cn.txt cc.txt
System Administration,INT239
```

```
(malhi@kali)-[~/Desktop]
$ paste -d "|" cn.txt cc.txt
System Administration|INT239
```

# THE diff COMMAND

- The `diff` command is used to compare text files. The command displays the two files and the differences between them.

```
(malhi@kali)-[~/Desktop]
$ diff cc.txt cn.txt
1c1
< INT239
_
> System Administration
```

- `1c1` means that line 1 in the first file needs to be changed to match line number 1 in the second file.

# THE awk COMMAND

- The `awk` command performs pattern matching on files.

```
(malhi@kali)-[~/Desktop]
$ awk '{print}' tab_data.txt
```

|      |        |     |
|------|--------|-----|
| sem1 | ajay   | 7.8 |
| sem2 | riya   | 9.4 |
| sem4 | alok   | 6.8 |
| sem3 | abhay  | 9.1 |
| sem1 | lohit  | 8.3 |
| sem6 | gagan  | 8.9 |
| sem7 | amisha | 7.8 |

```
(malhi@kali)-[~/Desktop]
$ awk '/sem1/ {print}' tab_data.txt
```

|      |       |     |
|------|-------|-----|
| sem1 | ajay  | 7.8 |
| sem1 | lohit | 8.3 |

# THE sed COMMAND

- The `sed` or stream editor command is a program that you can use to modify text files according to various parameters. The `sed` command can also be used for global search and replace actions.
- `s` Substitute the first occurrence of the string in the file.

```
(malhi@kali)-[~/Desktop]
$ sed 's/sem1/SEM1/' tab_data.txt
SEM1      ajay      7.8
sem2      riya      9.4
sem4      alok      6.8
sem3      abhay     9.1
SEM1      lohith    8.3
sem6      gagan     8.9
sem7      amisha    7.8
```

- `$ sed 's/sem1/SEM1/2' tab_data.txt`

- `$ sed 's/sem1/SEM1/g' tab_data.txt`

# THE `ln` COMMAND

- The `ln` command is used to create a link to a file.
- Linking enables a file name in one directory (the link) to point to a file in another directory (the target).
- A link does not contain data of its own, only a reference to the target file.
- Any changes to the link will reflect in the target file.
- If you don't specify the link name, the `ln` command will create the link in your current working directory.



- TYPES OF LINKS

- Using the `ln` command, you can create two types of links: hard and symbolic (soft).
- Hard and symbolic links are a feature of the file system and are common in most file systems supported by Linux. The ext2, ext3, ext4, and XFS file systems all support hard and symbolic links.
  - A hard link is a reference to another file; it enables the file's data to have more than one name in different locations in the same file system. Applications treat a hard link as a real file. If the original file is deleted after a hard link is created, all its contents will still be available in the linked file. This is because the inode of a hard link is the same as its target; in other words, it points to the same object on the file system. Hard links cannot be created between two directories, nor can they be created between two files in different file systems.

- A symbolic link is a reference to a file or directory that can span multiple file systems.  
If the original file or directory is deleted after a symbolic link is created, then the original content is lost.
- This is because the inode of a symbolic link is different than its target; in other words, it points to a different object on the file system.
- A symbolic link is also known as a soft link.

- A link in UNIX is a pointer to a file.
- Like pointers in any programming languages, links in UNIX are pointers pointing to a file or a directory.
- Creating links is a kind of shortcuts to access a file.
- Links allow more than one file name to refer to the same file, elsewhere.

- There are two types of links :

- 1.Soft Link or Symbolic links

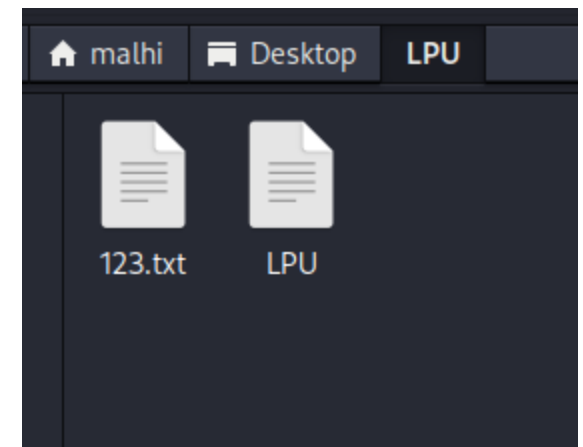
- 2.Hard Links

- These links behave differently when the source of the link (what is being linked to) is moved or removed.
  - Symbolic links are not updated (they merely contain a string which is the path name of its target);
  - hard links always refer to the source, even if moved or removed.

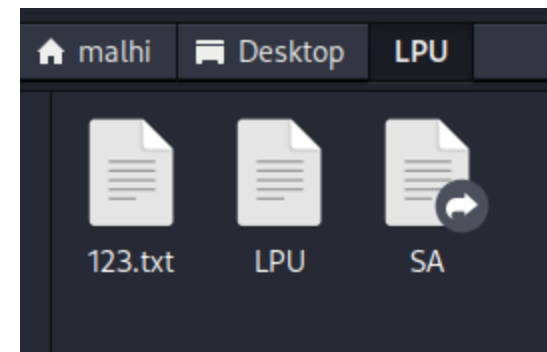
- For example, if we have a file a.txt.
- If we create a hard link to the file and then delete the file, we can still access the file using hard link.
- But if we create a soft link of the file and then delete the file, we can't access the file through soft link and soft link becomes dangling.

```
(malhi@kali)-[~/Desktop/LPU]
$ ln 123.txt LPU

(malhi@kali)-[~/Desktop/LPU]
$ ls
123.txt  LPU
```



```
ads  
└─(malhi@kali)-[~/Desktop/LPU]  
└─$ ln -s 123.txt SA  
  
stem└─(malhi@kali)-[~/Desktop/LPU]  
└─$ ls  
123.txt LPU SA
```





# Manipulate File Output

- TEXT STREAMS: A text stream is a sequence of one or more lines of text that applications can leverage to read from or write to a particular device or system component.

- Standard input, or stdin, is a text stream that acts as the source for command input.

Standard input for the Linux command-line is usually generated from the keyboard. In the case of the GUI, the standard input can also come from the mouse.

- Standard output, or stdout, is a text stream that acts as the destination for command output. By default, standard output from a Linux command is directed to the CLI.
- Standard error, or stderr, is a text stream that is used as the destination for error messages. By default, the standard error stream prints error messages at the CLI.

- **INPUT/OUTPUT REDIRECTION**

Redirection is the process of accepting input data from a source other than the keyboard and sending output data to a destination other than the display device.

- **REDIRECTION OPERATORS**

There are several operators that are used to redirect input or output.

# PIPING

- Piping is the process of combining the standard I/O streams of commands. It uses the standard output of one command as the standard input for another command. The output format of the first command should be compatible with the format that the second command works with. The pipe operator (|) can be used with most commands in Linux.

```
(malhi@kali)-[~]
```

```
$ ls documents
```

|           |            |                 |           |        |
|-----------|------------|-----------------|-----------|--------|
| Desktop   | file11.txt | file33.txt      | Pictures  | Videos |
| Downloads | file1.save | Music           | Public    |        |
| f111.txt  | file22.txt | nano.10154.save | Templates |        |

```
(malhi@kali)~  
$ ls > f111.txt
```

```
(malhi@kali)-[~]
```

```
$ cat f111.txt
```

```
Desktop LPU
```

```
Downloads
```

```
f111.txt
```

```
file11.txt
```

```
file1.save
```

```
file22.txt
```

```
file33.txt
```

```
Music
```

```
nano.10154.save
```

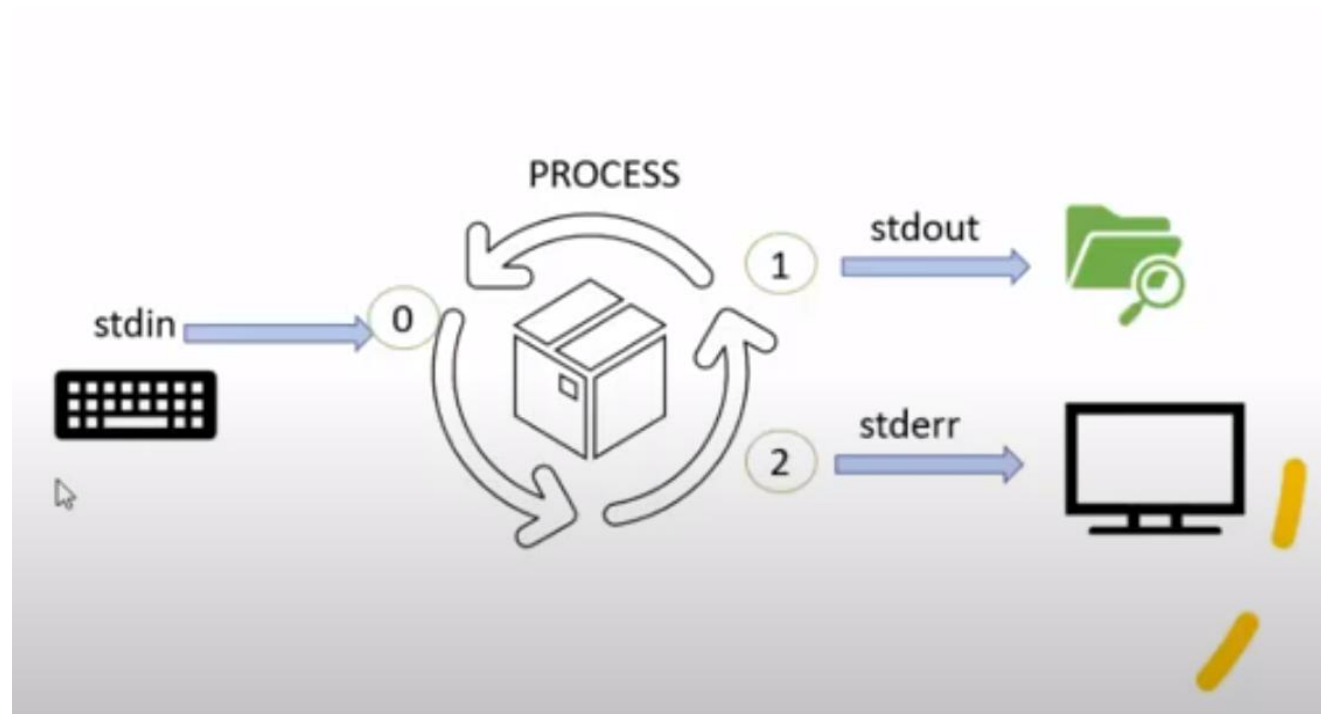
```
Pictures
```

```
Public
```

```
Templates
```

```
Videos
```





```
(malhi@kali)-[~]
```

```
$ ls documents
```

```
Desktop    file11.txt  file33.txt    Pictures  Videos
Downloads  file1.save  Music         Public
f111.txt   file22.txt  nano.10154.save  Templates
```

```
(malhi@kali)-[~]
```

```
$ cat sw
```

```
cat: sw: No such file or directory
```

```
(malhi@kali)-[~]
```

```
$ cat sw > f111.txt
```

```
cat: sw: No such file or directory
```

```
(malhi@kali)-[~]
```

```
$ cat sw 2> f111.txt
```

```
(malhi@kali)-[~]
```

```
$ cat f111.txt
```

```
cat: sw: No such file or directory
```

```
(malhi@kali)-[~]  
$ cat sw &> f111.txt
```

```
(malhi@kali)-[~]  
$ ls &>> f111.txt
```

```
(malhi@kali)-[~]  
$ cat f111.txt  
cat: sw: No such file or directory  
Desktop  
Downloads  
f111.txt  
file11.txt  
file1.save  
file22.txt  
file33.txt  
Music  
nano.10154.save  
Pictures  
Public  
Templates  
Videos
```

# Display the top 5 files/directories according to their size in the current working directory

```
(malhi@kali)-[~]  
$ ls -ls  
total 52  
drwxr-xr-x 4 malhi malhi 4096 Apr 19 00:23 Desktop  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Downloads  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Music  
drwxr-xr-x 2 malhi malhi 4096 Nov 15 23:23 Pictures  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Public  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Templates  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Videos  
-rw-r--r-- 1 malhi malhi 161 Apr 19 04:11 f111.txt  
-rw-r--r-- 1 malhi malhi 52 Mar 29 13:00 file22.txt  
-rw----- 1 malhi malhi 45 Mar 29 00:50 nano.10154.save  
-rw-r--r-- 1 malhi malhi 33 Mar 29 12:52 file11.txt  
-rw-r--r-- 1 malhi malhi 28 Mar 29 12:48 file33.txt  
-rw----- 1 malhi malhi 7 Mar 29 00:51 file1.save  
  
(malhi@kali)-[~]  
$ ls -ls | head -n 5  
total 52  
drwxr-xr-x 4 malhi malhi 4096 Apr 19 00:23 Desktop  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Downloads  
drwxr-xr-x 2 malhi malhi 4096 Oct 30 13:50 Music  
drwxr-xr-x 2 malhi malhi 4096 Nov 15 23:23 Pictures
```

# THE xargs COMMAND (extended arguments)

- The `xargs` command reads from standard input and executes a command for each argument provided.

```
(malhi@kali)-[~/Desktop]  
$ xargs -a tab_data.txt  
sem1 ajay 7.8 sem2 riya 9.4 sem4 alok 6.8 sem3 abhay 9.1 sem1 lohit 8.3 sem6  
gagan 8.9 sem7 amisha 7.8
```

- It shows the content of file.(read from file instead of standard input)

# THE tee COMMAND

- The `tee` command reads the standard input, sends the output to the default output device (the CLI), and also copies the output to each specified file.

```
(malhi@kali)-[~/Desktop]
$ ls -l | tee test.txt
total 24
-rw-r--r-- 1 malhi malhi    7 Apr 19 13:30 cc.txt
-rw-r--r-- 1 malhi malhi   22 Apr 19 13:31 cn.txt
drwxr-xr-x 3 malhi malhi 4096 Mar 22 13:19 Documents
drwxr-xr-x 2 malhi malhi 4096 Apr 25 13:58 LPU
-rw-r--r-- 1 malhi malhi  117 Apr 19 07:18 tab_data.txt
-rw-r--r-- 1 malhi malhi   594 Apr 17 23:43 test.txt

(malhi@kali)-[~/Desktop]
$ cat test.txt
total 24
-rw-r--r-- 1 malhi malhi    7 Apr 19 13:30 cc.txt
-rw-r--r-- 1 malhi malhi   22 Apr 19 13:31 cn.txt
drwxr-xr-x 3 malhi malhi 4096 Mar 22 13:19 Documents
drwxr-xr-x 2 malhi malhi 4096 Apr 25 13:58 LPU
-rw-r--r-- 1 malhi malhi  117 Apr 19 07:18 tab_data.txt
-rw-r--r-- 1 malhi malhi   594 Apr 17 23:43 test.txt
```

# Managing Kernel Modules

- The kernel is the core of an operating system. All other components rely on it.
- The kernel manages file system access, memory, processes, devices, and resource allocation on a system.
- The kernel also controls all the hardware devices plugged into the system.
- It is one of the first elements to be loaded on startup and remains in the main memory during the computer's operation.
- The kernel also contains system-level commands and other functions that are normally hidden from users.

- The kernel space is simply where the kernel executes the services that it provides.
- The user space is the area of memory that includes everything outside of kernel space.



# TYPES OF KERNELS

- Kernels can be classified as monolithic or microkernel. In a monolithic kernel, all system modules, such as device drivers or file systems, run in kernel space. As a result, a monolithic kernel can interact quickly with devices.
- In a microkernel architecture, the kernel itself runs the minimum amount of resources necessary to actually implement a fully functional operating system. Compared to monolithic kernels, microkernels have smaller kernel spaces and instead have larger user spaces.

- A **device driver** is a software program that enables a computer's operating system to identify the characteristics and functions of a hardware device, communicate with it, and control its operations. It acts as an interface between the operating system and hardware devices such as storage drives, printers, scanners, monitors, and keyboards. Device drivers can be included in the operating system or installed on demand.

# THE `uname` COMMAND

- By default, `uname` prints the name of the kernel—Linux. You can view the kernel version number of your current system by using the `uname -r` command. You can also enter `uname -i` to view the hardware platform. To print all information, enter the `uname -a` command.

```
(malhi@kali)-[~]  
$ uname  
Linux  
  
(malhi@kali)-[~]  
$ uname -a  
Linux kali 5.18.0-kali5-686-pae #1 SMP PREEMPT_DYNAMIC Debian 5.18.5-1kali6 (2022-07-07) i686 GNU/Linux  
  
(malhi@kali)-[~]  
$ uname -r  
5.18.0-kali5-686-pae  
  
(malhi@kali)-[~]  
$ uname -i  
unknown
```

# KERNEL LAYERS

- The kernel performs various functions to control and manage the operations of a system. It is composed of several layers that operate in kernel space.
- System Call Interface (SCI): Handles system calls sent from user applications to the kernel.

- Process management: Handles different processes by allocating separate execution space on the processor and ensuring that the running of one process does not interfere with other processes.

# Memory Management

- Manages the computer's memory, which is one of the complex tasks performed by the kernel. Like processor sharing, the system's memory also needs to be shared among different user space resources.

The kernel maps or allocates the available memory to applications or programs on request and frees the memory automatically when the execution of the programs is complete, so that it can be allocated to other programs.

- File system management:  
Manages the filesystem, which involves storing, organizing, and tracking files and data on a computer.
- Device management: Manages devices by controlling device access and interfacing between user applications and hardware devices of the computer.

# Install and Configure Kernel Modules

- A **kernel module** is a system-level object that extends the functionality of the kernel. It can be dynamically loaded into the kernel or unloaded from the kernel when required. It enables the kernel to update or recompile itself without requiring the system to reboot.
- The advantages of kernel modules are:
  - They reduce the burden on the kernel because otherwise all of the modules' functionality would have to be added directly to the kernel.
  - Dynamic loading of kernel modules facilitates lower memory consumption.
  - They avoid having to rebuild and reboot the system when new functionality is required.Kernel module file consists of a .ko extension. Modules built for a specific kernel version may not be compatible with another version of the kernel.



# KERNEL MODULE MANAGEMENT COMMANDS

- Kernel module management commands enable you to view, load, unload, or modify kernel modules.
- `lsmod`: Display the currently loaded kernel modules, their sizes, usage details, and their dependent modules.

```
(malhi@kali)-[~]  
$ lsmod  
Module                Size  Used by  
vsock_loopback        16384  0  
vmw_vsock_virtio_transport_common 28672  1 vsock_loopback  
rfkill                24576  2  
vmw_vsock_vmci_transport 32768  2  
vsock                 32768  7 vsock_loopback,vmw_vsock_vmci_transport,vmw_v  
sock_virtio_transport_common  
qrtr                  32768  4  
sunrpc               376832  1  
intel_rapl_msr        20480  0  
vmw_balloon           24576  0
```

# THE modprobe COMMAND

- The `modprobe` command is used to add or remove modules from a kernel.

# THE depmod COMMAND

- In order for `modprobe` to accurately install dependent modules, it reads the `modules.dep` file to identify how modules are linked to one another. The `depmod` command is used to update this database of dependencies so that `modprobe` can function properly.

# KERNEL MODULE CONFIGURATION

- The `/etc/modprobe.conf` file is a configuration file that contains settings that apply persistently to all the modules loaded on the system.
- It is used to configure modules and their dependencies and also specify module aliases.
- An alias is just an alternative name to use for a module.

# KERNEL PARAMETERS

- In addition to loading modules into the kernel at runtime, you can also change some of the kernel's parameters while it is running.
- You can use these parameters to improve system performance, harden security, configure networking limitations, change virtual memory settings, and more.

# THE sysctl COMMAND

- The `sysctl` command is used to view or set kernel parameters at runtime. It has various options.
- `-a` Display all parameters and their current values.

```
(malhi@kali)-[~]  
$ sysctl -a  
crypto.fips_enabled = 0  
debug.exception-trace = 1  
debug.kprobes-optimization = 1  
dev.cdrom.autoclose = 1  
dev.cdrom.autoeject = 0  
dev.cdrom.check_media = 0  
dev.cdrom.debug = 0  
dev.cdrom.info = CD-ROM information, Id: cdrom.c 3.20 2003/12/17  
dev.cdrom.info =  
dev.cdrom.info = drive name:          sr0  
dev.cdrom.info = drive speed:         1  
dev.cdrom.info = drive # of slots:    1  
dev.cdrom.info = Can close tray:      1  
dev.cdrom.info = Can open tray:       1  
dev.cdrom.info = Can lock tray:       1  
dev.cdrom.info = Can change speed:    1  
dev.cdrom.info = Can select disk:     0  
dev.cdrom.info = Can read multisession: 1  
dev.cdrom.info = Can read MCN:        1  
dev.cdrom.info = Reports media changed: 1  
dev.cdrom.info = Can play audio:      1
```

# Monitor Kernel Modules

- After you install and configure kernel modules, it's a good idea to monitor those modules.
- In this, you'll verify that the modules you installed were actually loaded into the kernel, and that any configurations you made were implemented properly.

# THE /proc/ DIRECTORY

- The /proc/ directory is a virtual file system (VFS) that provides significant information about the kernel's running process.

```
(malhi@kali)-[~]  
$ cat /proc/cpuinfo  
processor       : 0  
vendor_id      : GenuineIntel  
cpu family     : 6  
model          : 42  
model name     : Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz  
stepping       : 7  
microcode      : 0x2f  
cpu MHz        : 2195.012  
cache size     : 3072 KB  
physical id    : 0  
siblings       : 1
```



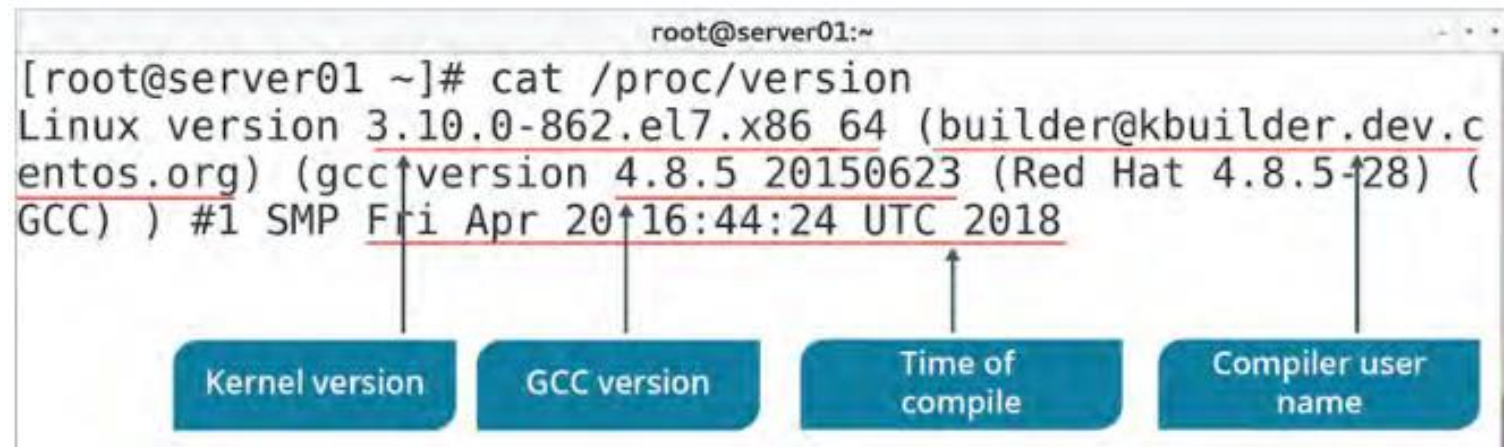
# THE `/proc/version` FILE

- The `/proc/version` file specifies several points of information about the Linux kernel:
- The version of the Linux kernel currently running.
- The version of the GNU Compiler Collection (GCC) used to compile the kernel.
- The user name of the kernel compiler.
- The time the kernel was compiled.

```
(malhi@kali)-[~]
```

```
$ cat /proc/version
```

```
Linux version 5.18.0-kali5-686-pae (devel@kali.org) (gcc-11 (Debian 11.3.0-3)
11.3.0, GNU ld (GNU Binutils for Debian) 2.38) #1 SMP PREEMPT_DYNAMIC Debian
5.18.5-1kali6 (2022-07-07)
```



A terminal window titled 'root@server01:~' displays the output of the command 'cat /proc/version'. The output text is: 'Linux version 3.10.0-862.el7.x86\_64 (builder@kbuilder.dev.centos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-28) (GCC) ) #1 SMP Fri Apr 20 16:44:24 UTC 2018'. Below the text, four blue boxes with white text are connected to the output by vertical arrows. The boxes are labeled 'Kernel version', 'GCC version', 'Time of compile', and 'Compiler user name'. The arrows point from the boxes to the corresponding parts of the output: 'Kernel version' points to '3.10.0-862.el7.x86\_64', 'GCC version' points to 'gcc version 4.8.5 20150623', 'Time of compile' points to '16:44:24', and 'Compiler user name' points to 'builder@kbuilder.dev.centos.org'.

```
root@server01:~  
[root@server01 ~]# cat /proc/version  
Linux version 3.10.0-862.el7.x86 64 (builder@kbuilder.dev.c  
entos.org) (gcc version 4.8.5 20150623 (Red Hat 4.8.5-28) (  
GCC) ) #1 SMP Fri Apr 20 16:44:24 UTC 2018
```

Kernel version      GCC version      Time of compile      Compiler user name

# THE dmesg COMMAND

- The `dmesg` ("display message" or "driver message") command is used to print any messages that have been sent to the kernel's message buffer during and after system boot. Device drivers send messages to the kernel indicating the status of modules and parameters that the drivers interface with. These drivers can also send diagnostic messages to the kernel in case they encounter errors.

```
(root@kali)-[~]
# dmesg -H
[Apr25 23:49] Linux version 5.18.0-kali5-686-pae (devel@kali.org) (gcc-11 (Debian 11.3.0-3) 11.>
[ +0.000000] Disabled fast string operations
[ +0.000000] x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'
[ +0.000000] x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'
[ +0.000000] x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'
[ +0.000000] x86/fpu: xstate_offset[2]: 576, xstate_sizes[2]: 256
[ +0.000000] x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using 'standard'>
[ +0.000000] signal: max sigframe size: 1760
[ +0.000000] BIOS-provided physical RAM map:
[ +0.000000] BIOS-e820: [mem 0x0000000000000000-0x0000000000009ebff] usable
```

# Managing the Linux Boot Process

- **Configure Linux Boot Components:** Booting is the process of starting or restarting a computer and loading an operating system for the user to access. In the boot process, a booting environment reads a small program that is stored in read-only memory (ROM). This program then executes various operations in RAM that bootstrap the operating system and make it available for use.

- A **boot loader** is the small program stored in ROM that loads the kernel from a storage device, and then starts the operating system. A boot environment like BIOS reads the boot loader from ROM so that the boot loader can execute the necessary operations to start the process.

- The **Basic Input/Output System (BIOS)** is a standard for firmware interfaces and is stored on a computer motherboard's ROM chip. When a computer with BIOS is powered on, the BIOS firmware is the first to run; this enables it to test the various hardware components in a computer, as well as run a boot loader so that an operating system can start.

# UEFI

- **Unified Extensible Firmware Interface (UEFI)** is newer firmware technology that has largely replaced BIOS by bringing with it several key advantages. UEFI runs faster than BIOS, can operate within a greater amount of memory, can access storage drives of currently unattainable sizes, can access more hardware types, and has improved security protections. Most modern motherboards, as well as the pre-assembled PCs that use them, ship with UEFI.

- A **sector** is the smallest unit of storage read from or written to a drive. A sector stores 512 bytes of data by default. On hard disk drives, a collection of sectors is called a track. The number of sectors in a track may vary, and so does their capacity to hold data.



- The **master boot record (MBR)** is the first physical sector on a storage drive and a type of partition structure. The MBR boot sector contains the boot loader that loads the operating system into memory. It also contains the partition table of the storage drive. MBR determines what sectors are available to each partition, as well as which partition is considered bootable and which partitions are not.

- The **GUID Partition Table (GPT)** is a successor to MBR that makes up for the latter's shortcomings. Like MBR, it is a partition structure, but it employs a more modern design and is part of the UEFI standard. Every partition on a drive is assigned a globally unique identifier—a GUID—to distinguish it from every other partition on (theoretically) every drive.

# initrd

- The initial ramdisk (initrd) refers to the root file system that is temporarily loaded into memory upon system boot. The initrd loads along with the kernel, which controls its functionality. The initrd enables the system to be started in two phases. In the first phase, the system is booted with the minimal set of modules required to load the main or the permanent root file system. In the second phase, when the main root file system is mounted, the previously mounted initrd file system is removed and the user space boot process continues.

# THE initrd IMAGE

- The Linux initrd image is an archive file containing all the essential files that are required for booting the operating system.

# THE `mkinitrd` COMMAND

- The `mkinitrd` command is used to create the `initrd` image for preloading the kernel modules.

# THE `/boot/` DIRECTORY

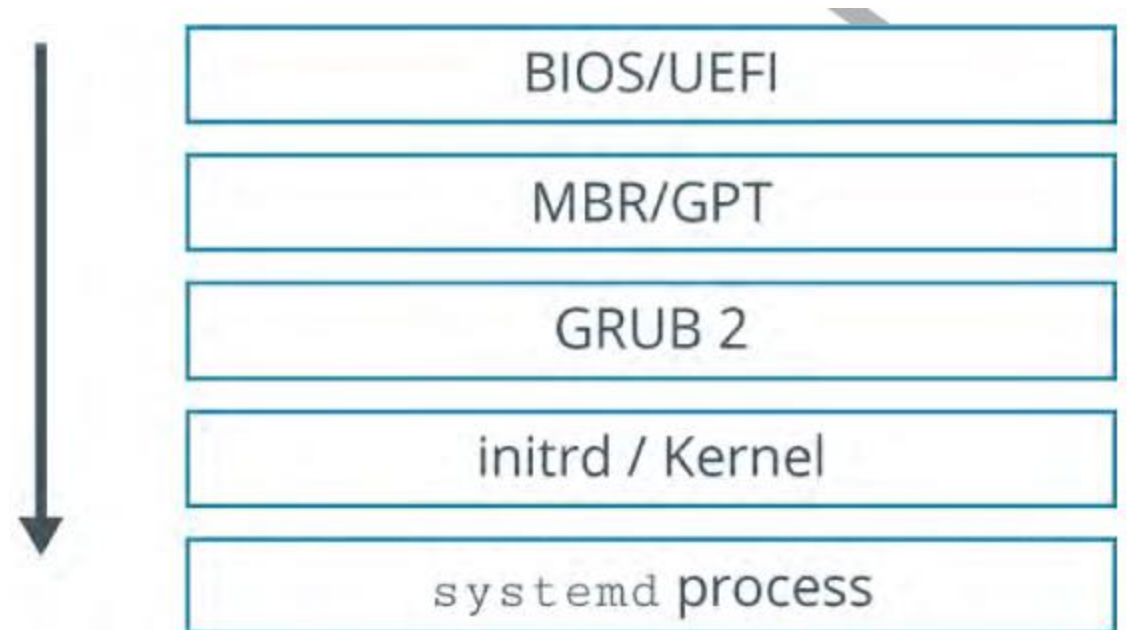
- As defined by the Filesystem Hierarchy Standard (FHS), the `/boot/` directory contains files that are used to facilitate the Linux boot process.

# THE dracut COMMAND

- The `dracut` command is used to generate an `initramfs` image, similar to how `mkinitrd` is used to generate an `initrd` image.

# THE BOOT PROCESS

- The boot process is repeated each time your computer is started by loading the operating system from a storage device. It involves a series of sequential steps that can be divided into BIOS/UEFI initialization, boot loader, kernel and initrd/initramfs initialization, and boot scripts.





- 1. The processor checks for the BIOS/UEFI firmware and executes it. This is also where the power-on self-test (POST) occurs.
- 
- 2. BIOS/UEFI checks for bootable media from internal storage devices or peripherals like USB thumb drives and DVD-ROMs. It locates a valid device to boot the system.
- 
- 3. BIOS/UEFI loads the primary boot loader from the MBR/GPT partition into memory. It also loads the partition table along with it.
- 
- 4. The user is prompted by GRUB 2 to select the operating system they want to boot. If the user does not respond, then the default operating system will be booted.

- 5. The boot loader determines the kernel and locates the corresponding kernel binary. It then uploads the respective initrd image into memory and transfers control of the boot process to the kernel.
- 6. The kernel configures the available hardware drivers, including processors, I/O subsystems, and storage devices. It decompresses the initrd image and mounts it to load the necessary drivers. If the system implemented any virtual devices, such as LVM or software RAID, then they are initialized.
- 7. The kernel mounts the main root partition and releases unused memory. To set up the user environment, the systemd program is run. It becomes process ID 1.
- 8. The systemd program searches for the default.target file, which contains details about the services to be started. It mounts the file system based on the /etc/fstab file and begins the process of starting services. On most systems, the target will either be multi-user.target or graphical.target.

9. If graphical mode is selected, then a display manager like XDM or KDM is started and the login window is displayed on the screen.

10. The user enters a user name and password to log in to the system.

- 11. The system authenticates the user. If the user is valid, then various profile files are executed.
- 12. The shell is started and the system is ready for the user to work on.

# KERNEL PANIC

Kernel panic is a mechanism by which the system detects there has been a fatal error and responds to it. A fatal error typically results in the system becoming unstable or totally unusable. Software that handles kernel panics will display an error message to the user and dump the current state of kernel memory to a storage device for later debugging.

Depending on how the system is configured, the panic handler will either reboot the system automatically, or wait for the user to do so.

# Configure GRUB 2

- GNU GRUB:  
The GNU GRand Unified Bootloader (GNU GRUB) is a boot loader developed by the GNU Project that became popular on Unix-like systems. It enables users to choose which operating system or kernel version to boot in a multi-platform environment.

# GRUB 2 IMPROVEMENTS

- GRUB 2 is more than simply a newer version of GRUB; it is a complete redesign and rewrite of the GRUB system. GRUB 2 offers administrators more control over the boot process, boot devices, and boot behavior.

# GRUB 2 INSTALLATION

- The `grub2-install` command is used to install the GRUB 2 boot loader on a storage device. It copies GRUB 2 files into the `/boot/grub2` directory and, on some platforms, installs GRUB 2 into the boot sector. However, `grub2-install` applies to BIOS systems, not UEFI. To install GRUB 2 on a UEFI system, use a package manager to install the `grub2-efi` package. Installing this package will copy GRUB 2 files onto the EFI system partition (ESP) in the `/boot/efi` directory.

# THE grub.cfg FILE

- The `grub.cfg` file is the main configuration file for the GRUB 2 boot loader. On BIOS systems, it is located in the `/boot/grub2/` directory. On UEFI systems, it is located in the `/boot/efi/EFI/<distro>/` directory.



# THE `/etc/grub.d/` DIRECTORY

- The `/etc/grub.d/` directory contains scripts that are used to build the main `grub.cfg` file.

# THE /etc/default/grub FILE

- The `/etc/default/grub` file contains GRUB 2 display menu settings that are read by the `/etc/grub.d/` scripts and built into the `grub.cfg` file. It enables you to change options such as how many seconds GRUB 2 will wait before automatically selecting the default boot option.

```
(root@kali)-[~]
# cat /etc/default/grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet"
GRUB_CMDLINE_LINUX="initrd=/install/gtk/initrd.gz"

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"
```

# THE grub2-mkconfig COMMAND

- The `grub2-mkconfig` command generates a new `grub.cfg` configuration file,  
and is used to update an existing `grub.cfg` file.

- ***ADDITIONAL READING IS  
STRONGLY RECOMMENDED***