



Linux For User

Day3_linuxUSER.md



What was our last class about?

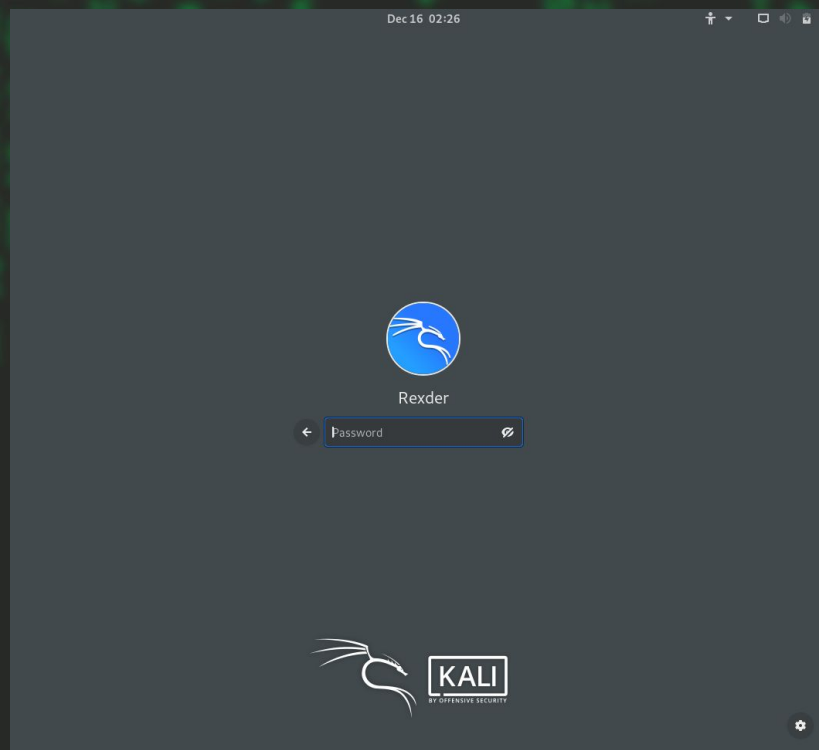
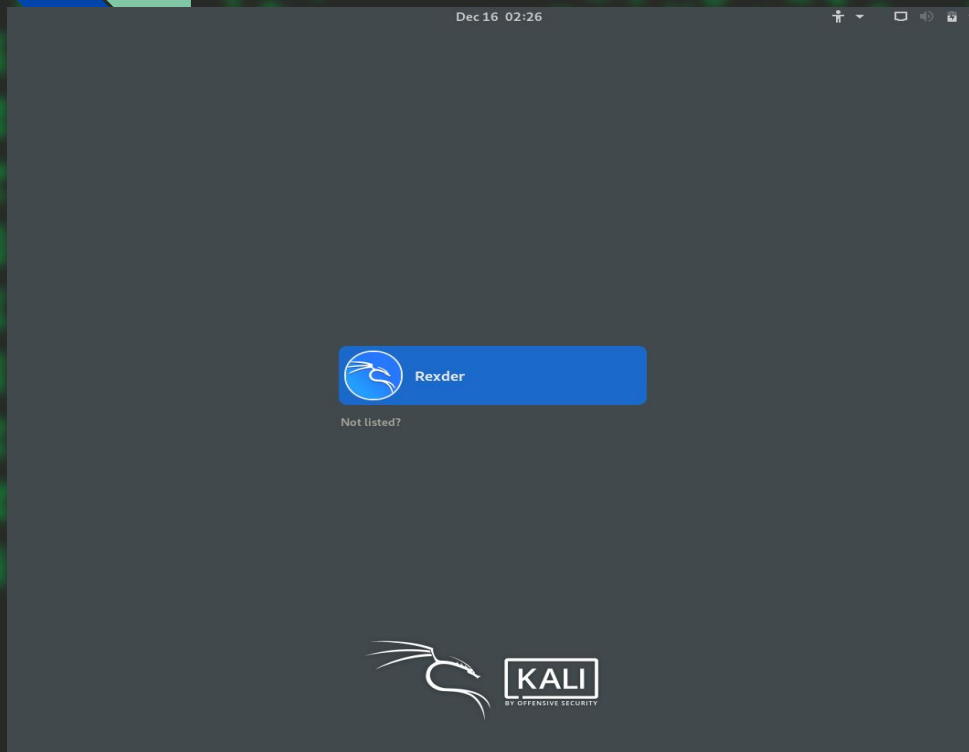
- Topics
- Concepts
- Assignments



Overview of kali linux

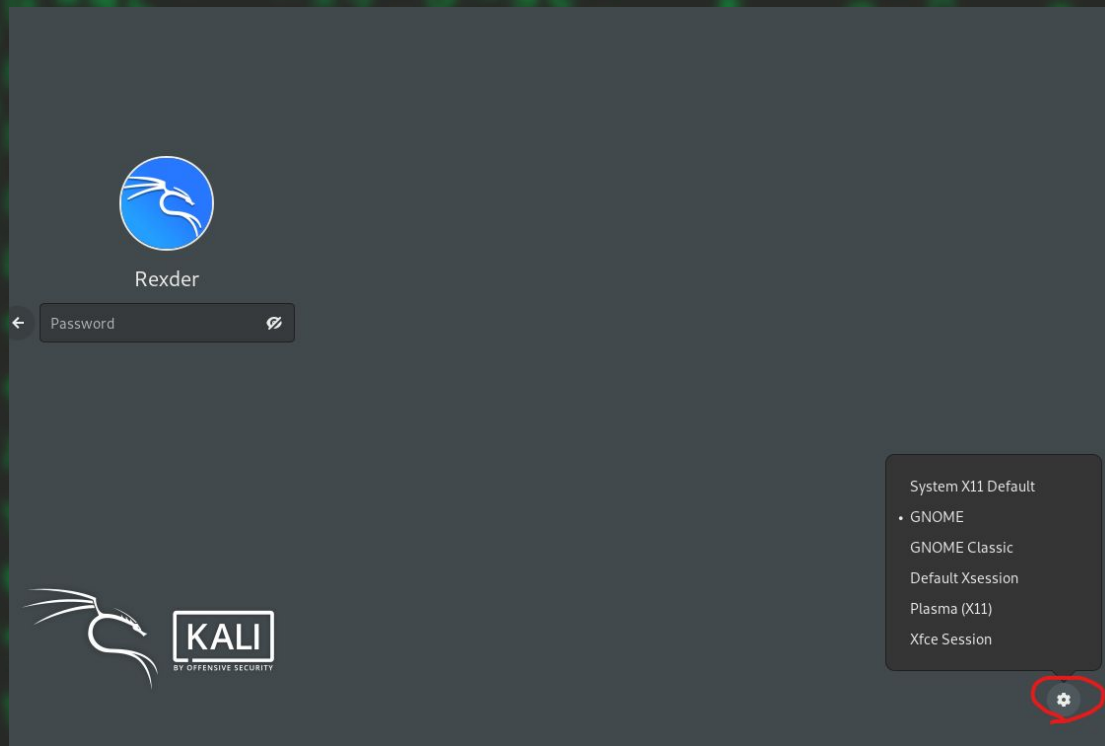
- Specifically, we are going to show see with “kali linux” with gnome Desktop Environment but you can Use any kind, BUT Debian is recommended.
- Previously known as Backtrack

Login

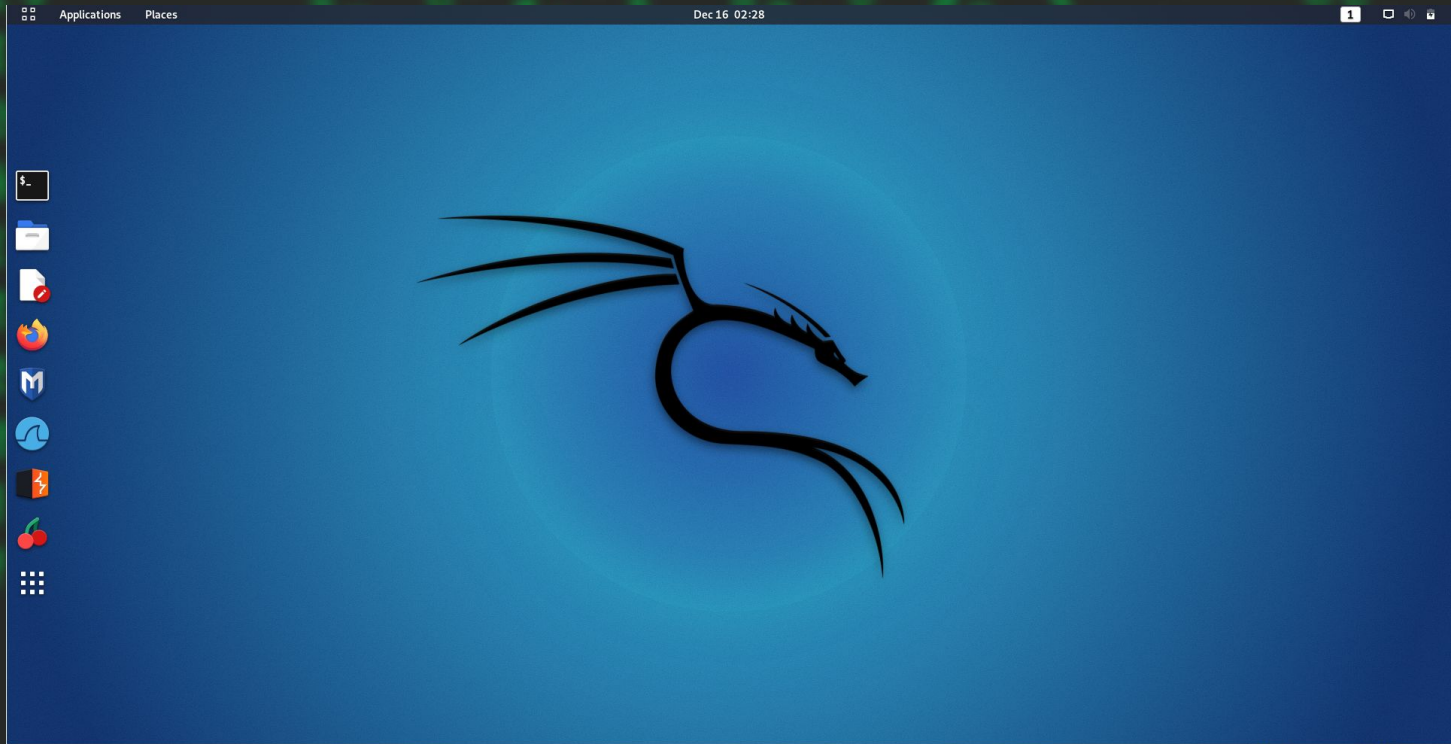


Cont ...

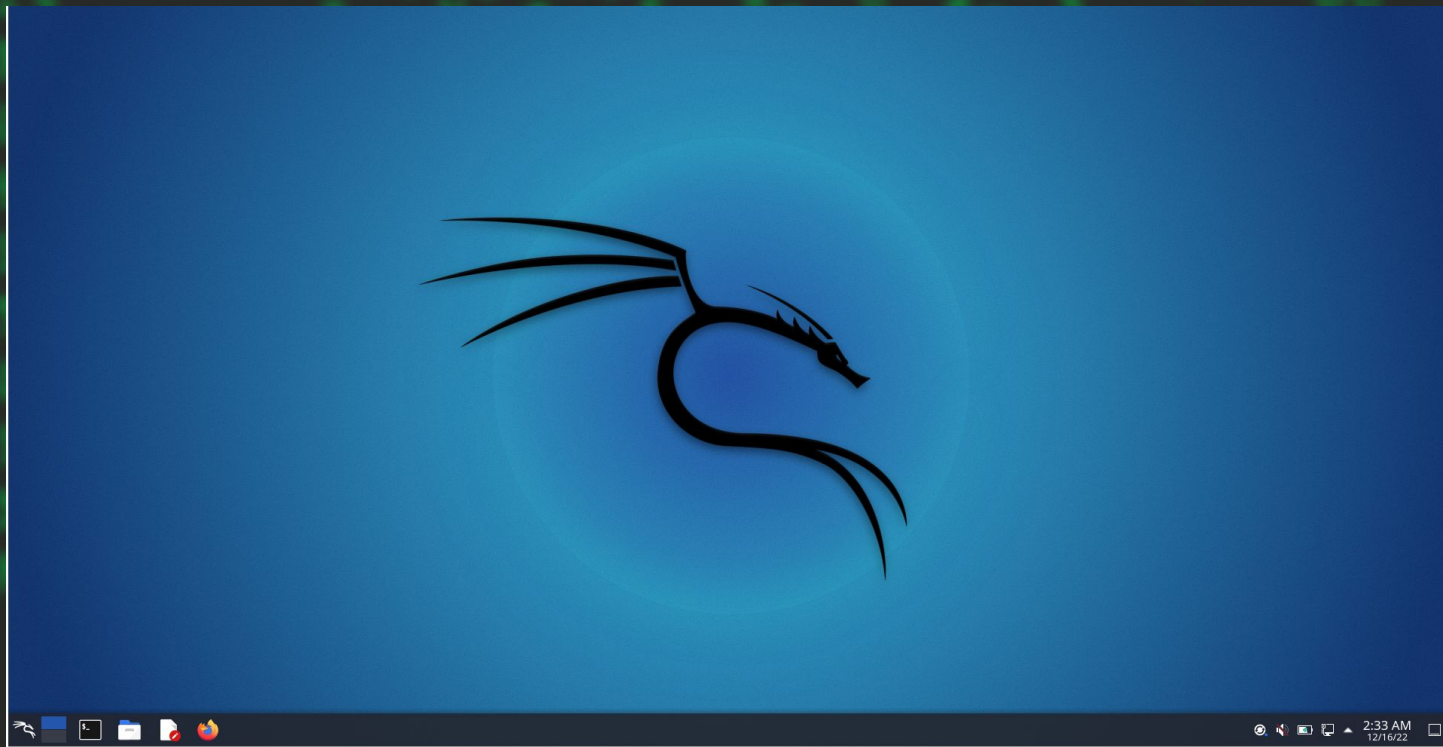
- You can switch between your Desktop environment



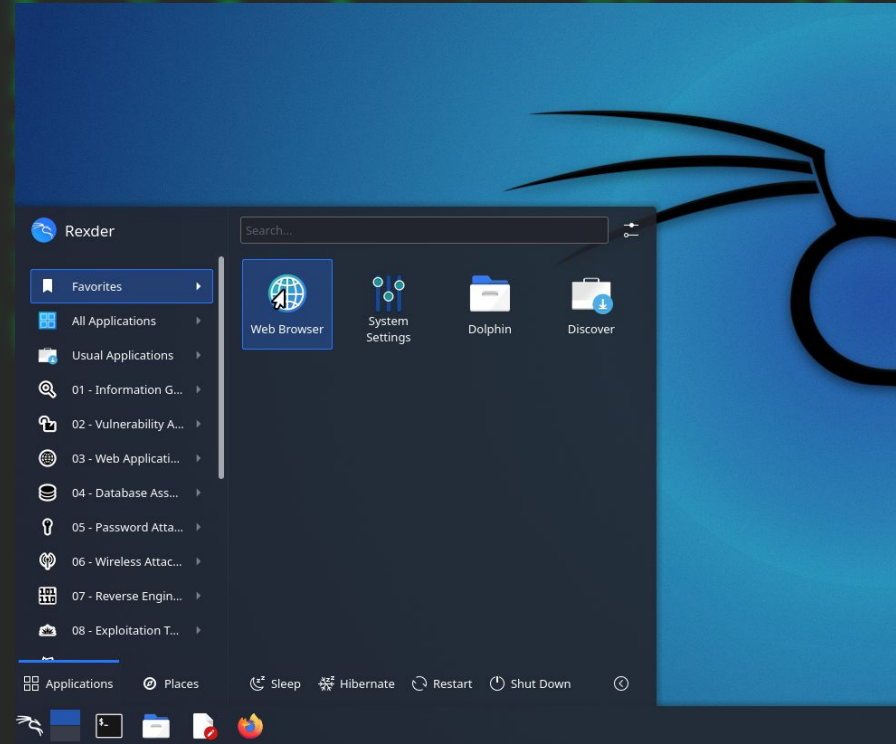
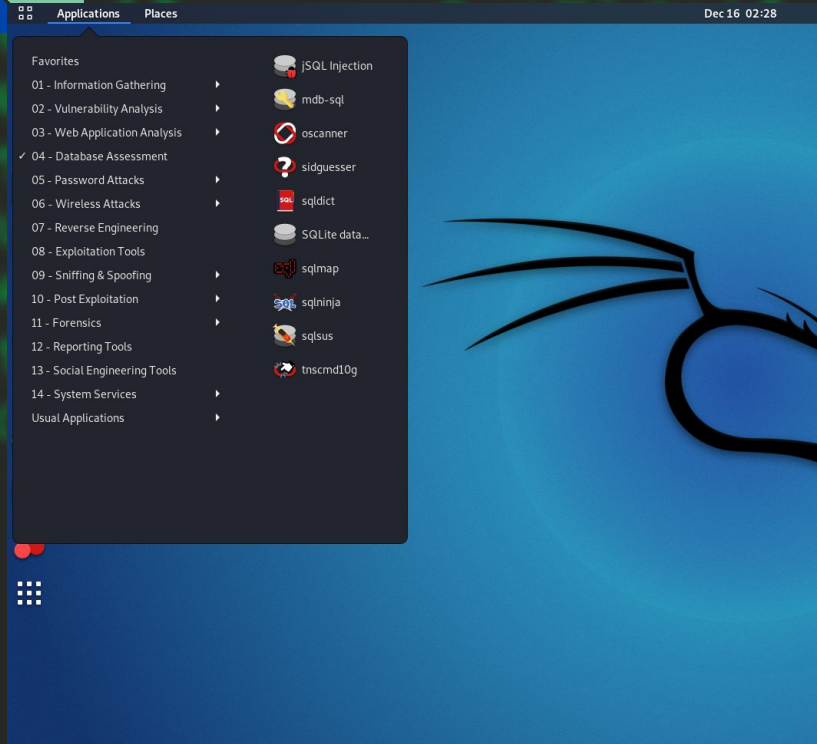
Main screen - Gnome



Cont... - KDE plasma

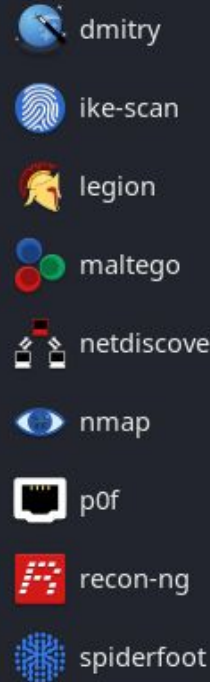


Menu



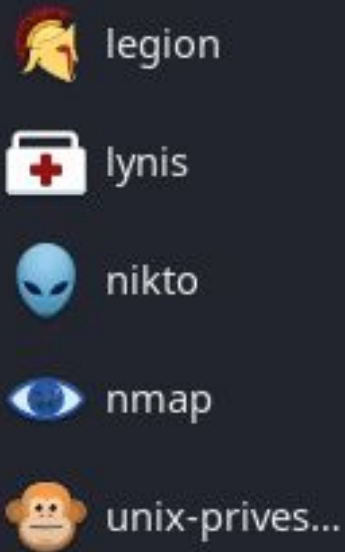
1) Information gathering

- Tools for information gathering, in system, network, host




2) Vulnerability Analysis

- **Tools for Finding Vulnerabilities**



3) Web Application Analysis

- **Tools for Finding Vulnerabilities and exploits on websites.**

 burpsuite

 commix


 httrack

 paros

 skipfish

 sqlmap

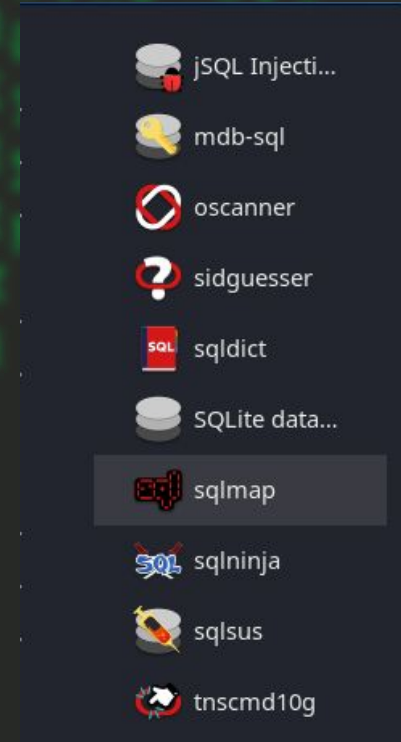
 webScarab

 wpScan

 ZAP

4) Database Assessment

- Tools for Finding Vulnerabilities and exploits on Databases.



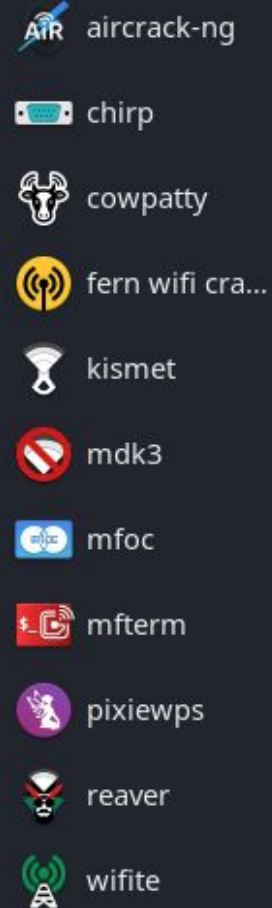
5) Password Attacks

- Tools for exploiting Passwords for login, websites, application, windows..



6) Wireless Attacks

- Tools for exploiting Wireless Systems like wifi, bluetooth..



7) Reverse Engineering

- Tools for exploiting Softwares, Mobile Applications and any binary files



apktool



bytecode-vi...



clang



clang++



dex2jar



edb-debug...



ghidra



jadx-gui



jasnoob



NASM shell



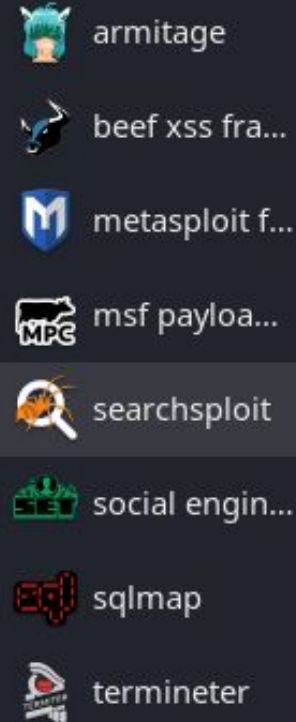
ollydbg



radare2

8) Exploitation Tools

- Tools for exploiting Softwares, Mobile ,Computers ,websites and any things



9) Sniffing & Spoofing

- Tools for Listening or hijacking networks



driftnet



ettercap-gr...



hamster



macchanger



mitmproxy



netsniff-ng




responder




wireshark


10) POST exploitation


- **Tools for Maintaining our access. Used after exploiting a system**


 backdoor-f...

 exe2hex

 mimikatz

 nishang

 powersploit

 proxychains4

 weeveily

11) Forensics

- Tools for Doing researches and investigate a Cyber Attacks.



autopsy



binwalk



bulk_extrac...



chkrootkit



foremost



galleta



hashdeep

12) Reporting tools

- **Tools for Report preparation. After some forensic you will get data and you will write report and these tools will help you.**



13) Social Engineering tools

- Tools Used for Social Engineering attacks



14) System Services

- Buttons used to start some services.



beef start



beef stop



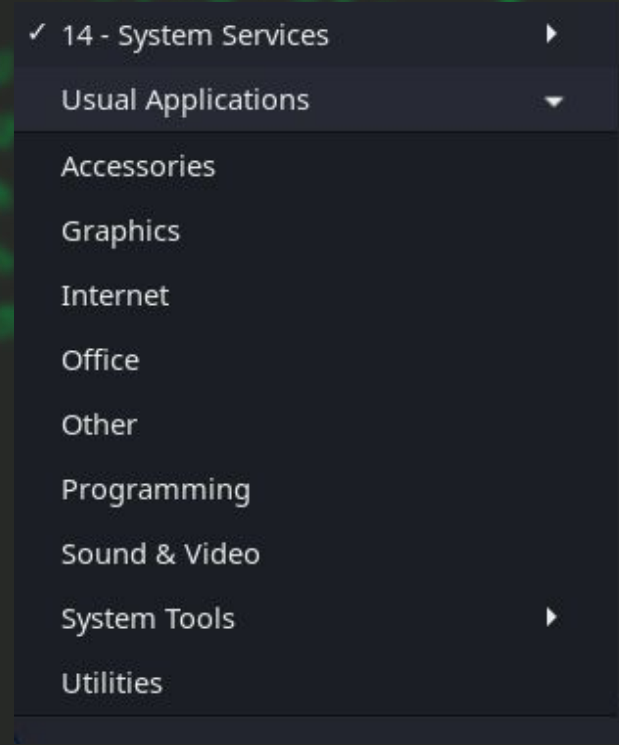
dradis start



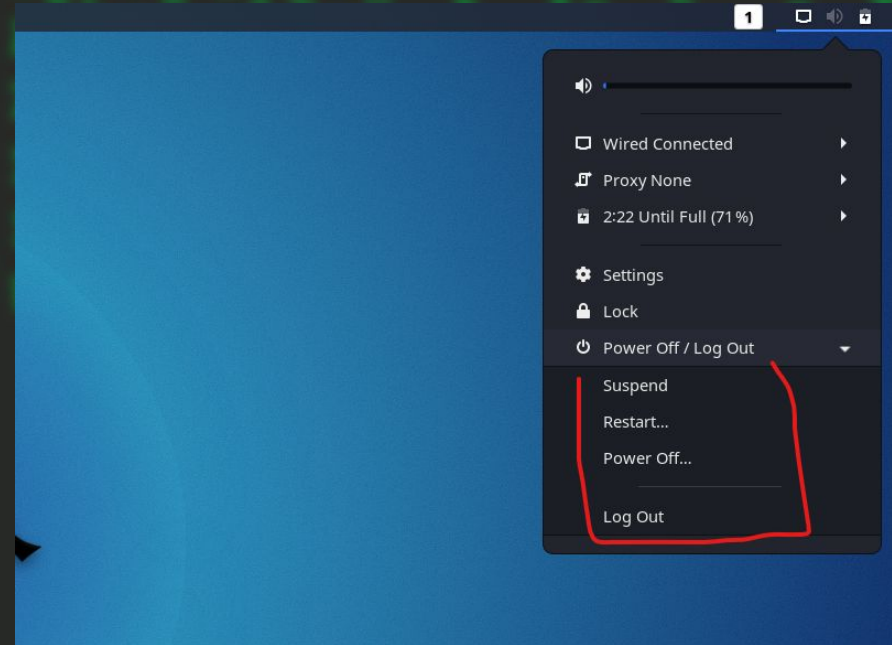
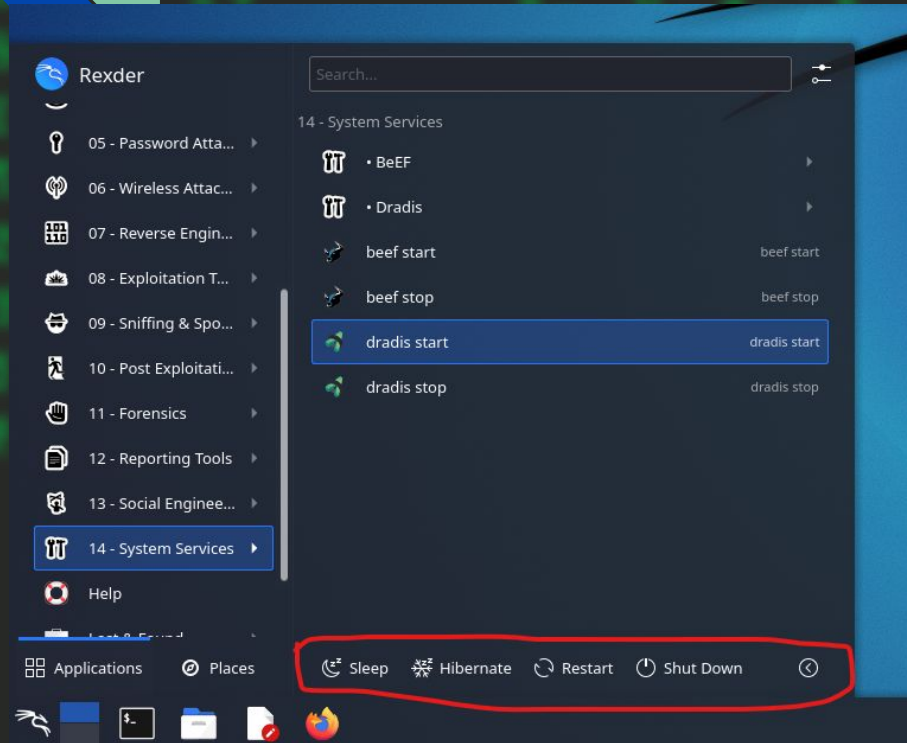
dradis stop

15) Usually used applications

- **Softwares for some basic purposes**

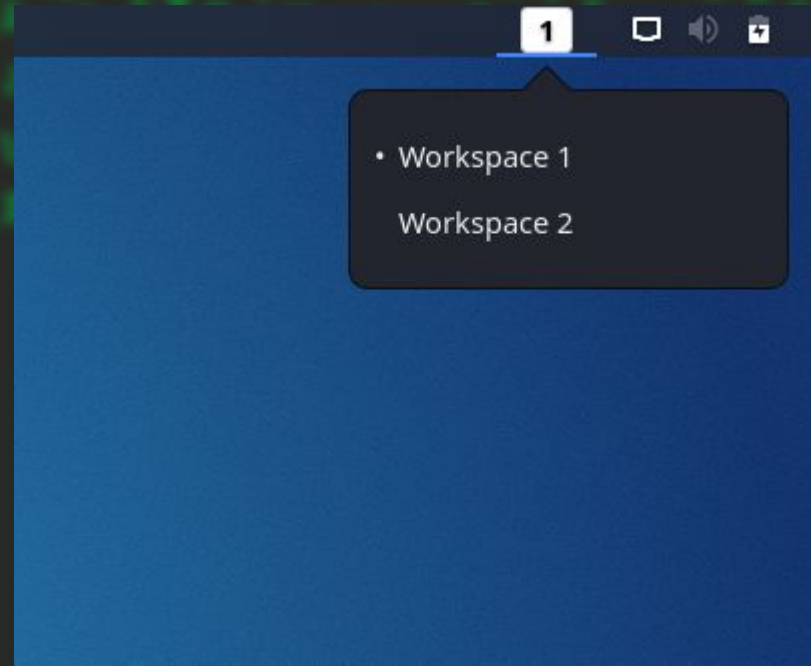
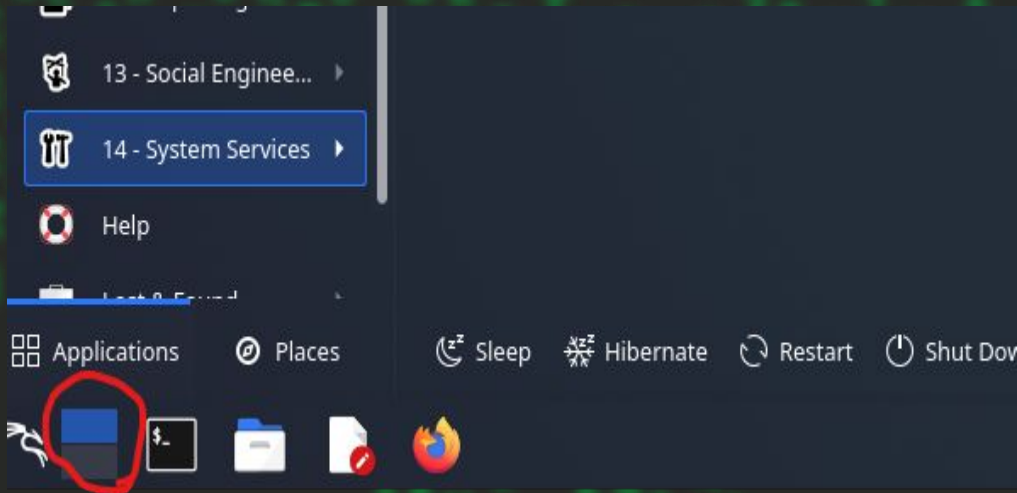


Shutdown,lock,restart

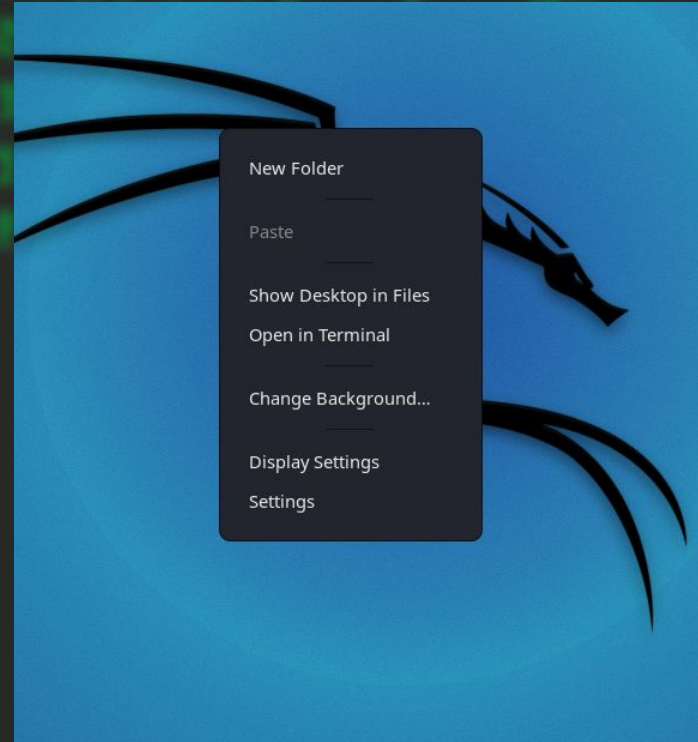
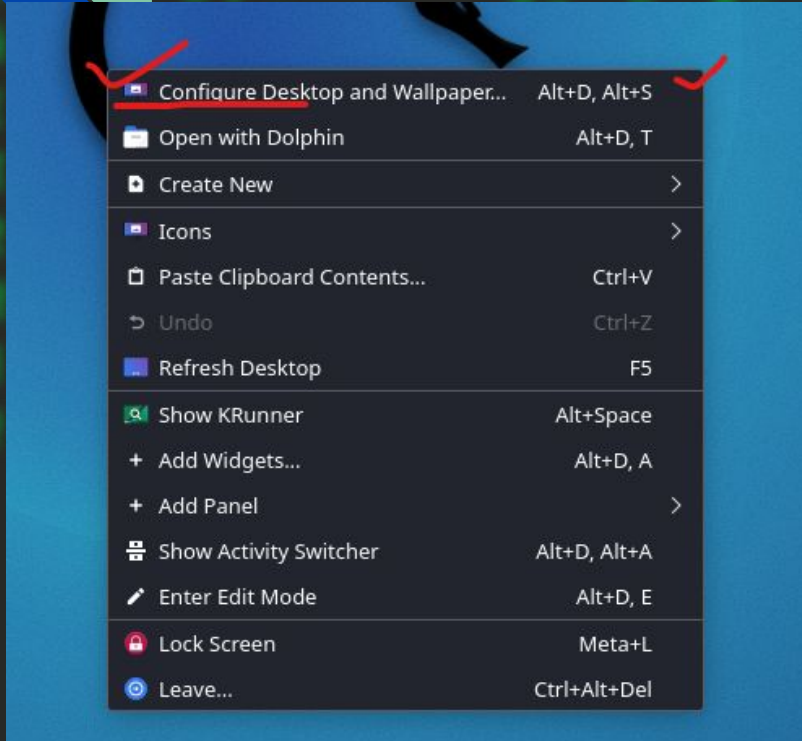


Workspace manager

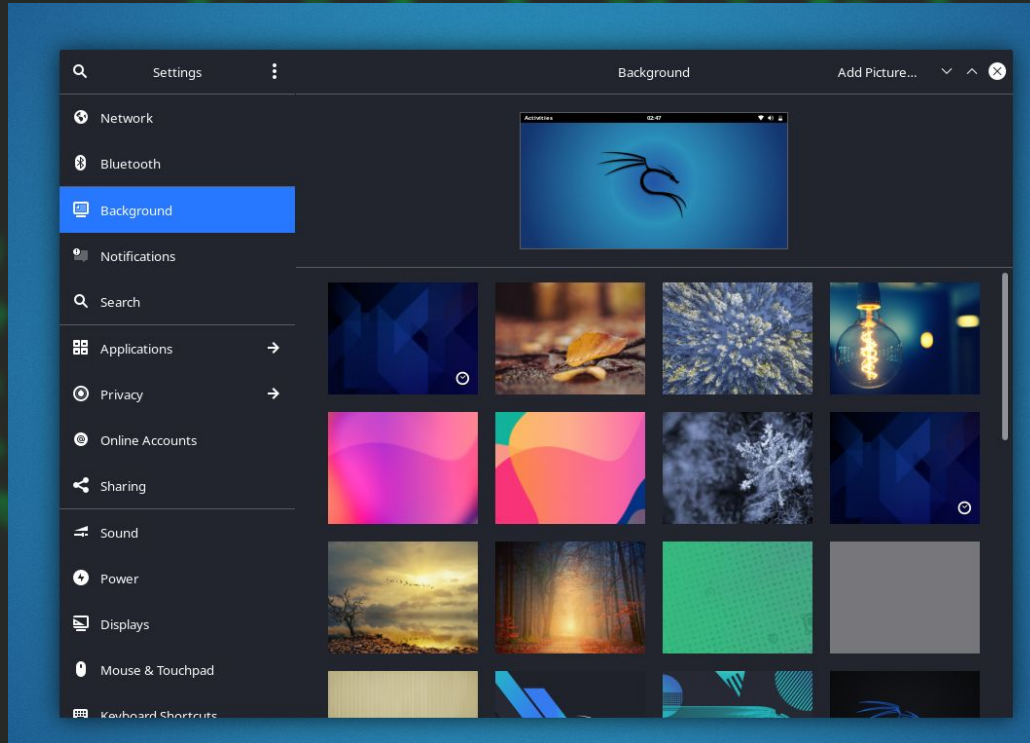
Used to Classify our works on different windows



Desktop Properties

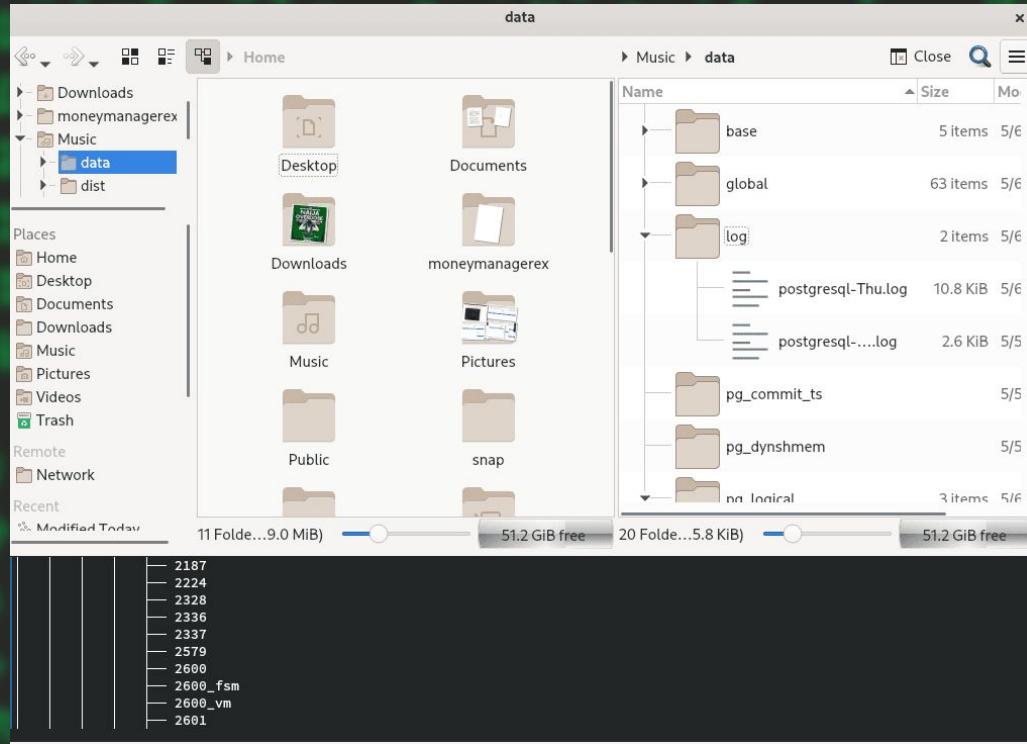


Background Changes



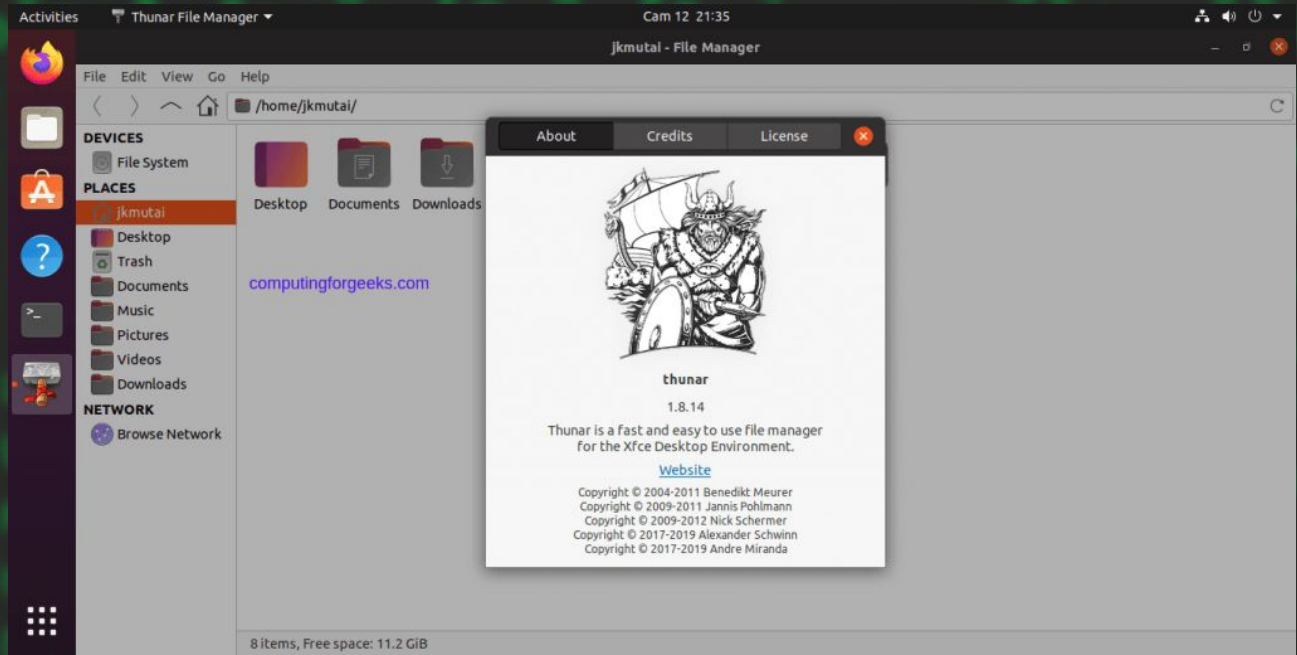
Folder managers

1) Dolphin



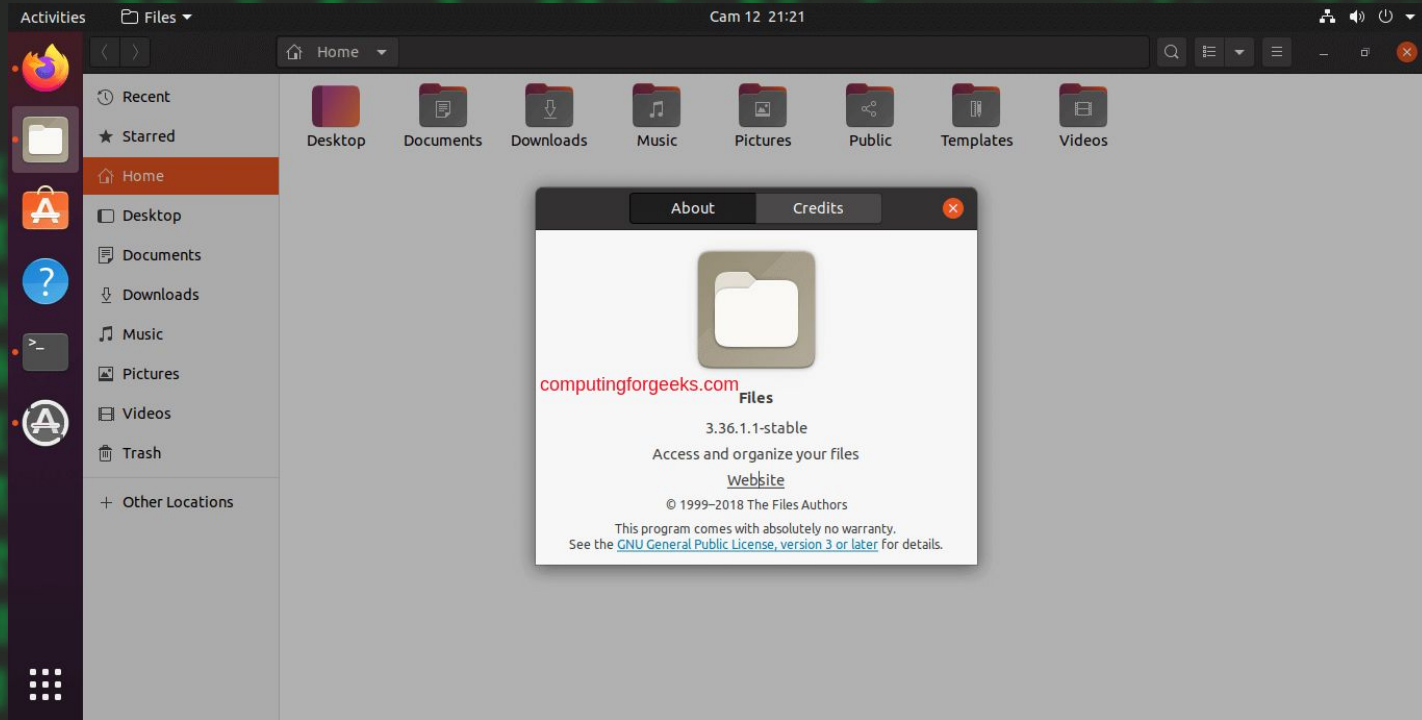
Folder managers

2) Thunar



Folder managers

3) Nautilus



On window it is NOT FREE

[Home](#) / [Utilities & tools](#) / Shrestha Files Pro X - File Explorer for Windows



Shrestha Files Pro X - File Explorer for Windows

JPT Games and Apps

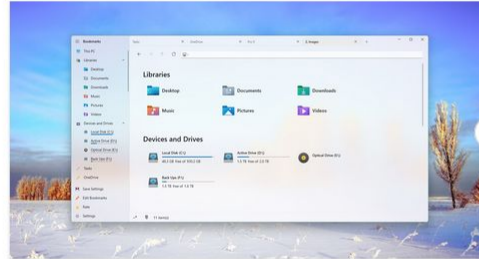
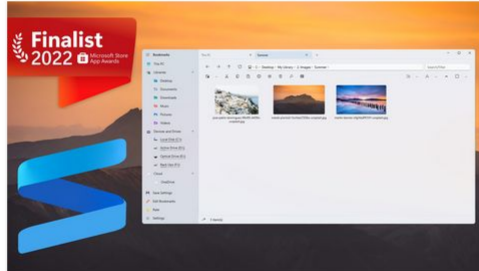
☆☆☆☆☆ 0 | [Utilities & tools](#)

[Get in Store app](#)

~~USD\$9.99~~ USD\$7.99

0

SAVE USD\$2.00



Details

Release Year: 2020

Available in 43 languages

Published by

JPT Games and Apps

[Contact information](#)

Terms

[Privacy policy](#)

Developer and IT

[App badge](#)

[Endpoint Manager](#)

Description


[Note: The app uses .NET Runtime 6. This updated app does not support Windows 10 S.]

Shrestha Files is a finalist in Microsoft Store App Award 2022 in the File Management Category!

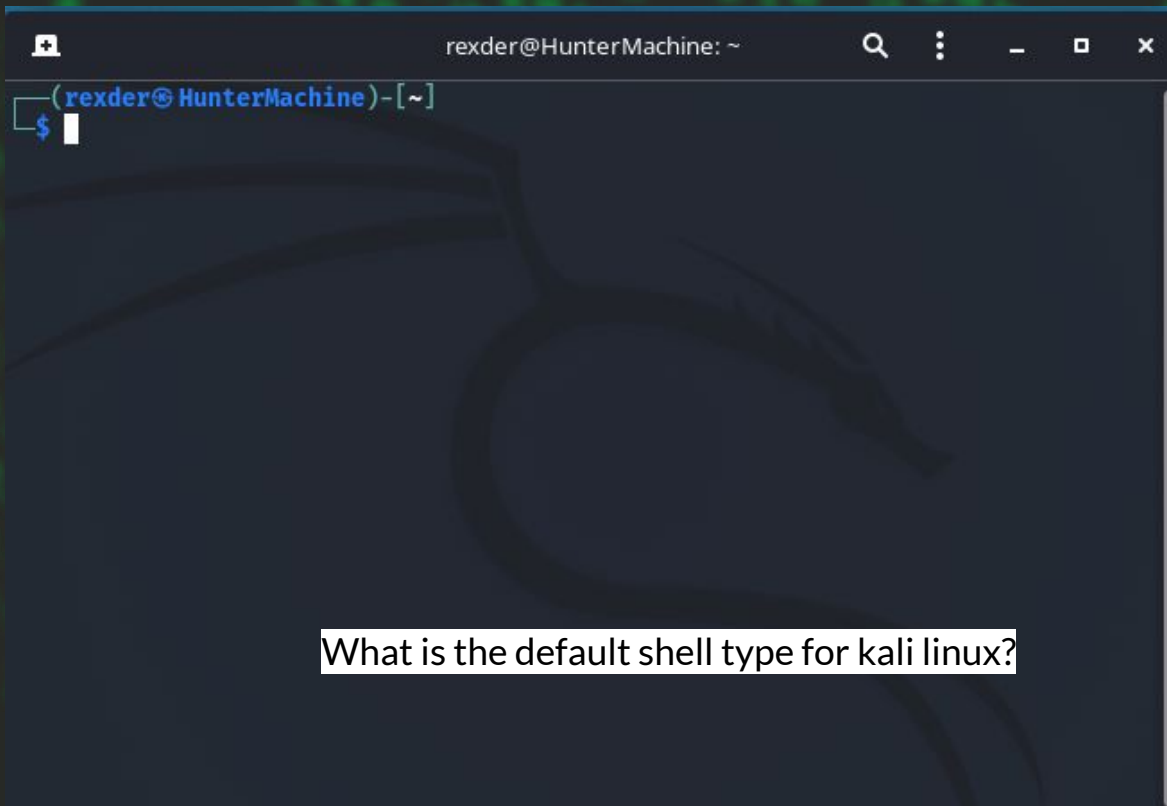
Welcome to Shrestha Files Pro X - the neXt version of Shrestha Files Pro! Shrestha Files Pro X offers the convenience of tabs and dual pane with a



Linux Commands

- Linux Systems uses shell. The shell help us to Communicate with the kernel and helps to execute codes.
- This is the shells icon --> The icon is a blue square with a white border. Inside the square, there is a white dollar sign (\$) followed by a white underscore (_).
- Shell also called “Terminal”

The shell 🥰



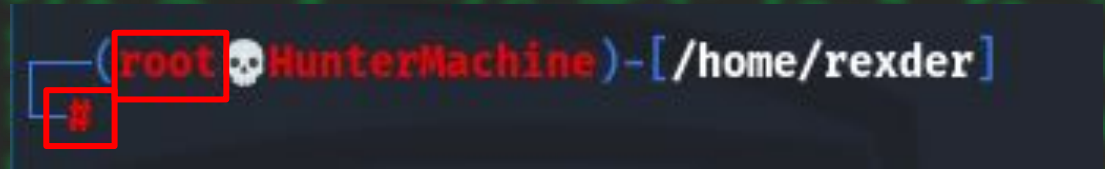
What is the default shell type for kali linux?

Cont...

- The terminal have 5 parts.
 - **Username** = rexder
 - **Hostname** = HunterMachine
 - **Current Directory** = PATH
 - **Priviledge** = \$-(user) , #-(root)
 - **Command place** = _
- Home directory is Denoted by '~'
- Local directory is Denoted by '.'
- All directories is Denoted by '*'
 - We will see them in future use.



```
(rexder@HunterMachine) - [~]  
$
```



```
(root@HunterMachine) - [/home/rexder]  
#
```

Directory = folder

Linux Command Basics

- On linux there are over 1000 commands. But we are Expected to know the main and useful only.
- Also those commands have their own **options** and **arguments**.

```
(rexder@HunterMachine)-[~]  
$ command --option argument
```

- **Options**: Additional Settings they take
- **Arguments**: different inputs to have output.
- Example:

```
(rexder@HunterMachine) - [~]  
$ maths --add 1 1
```

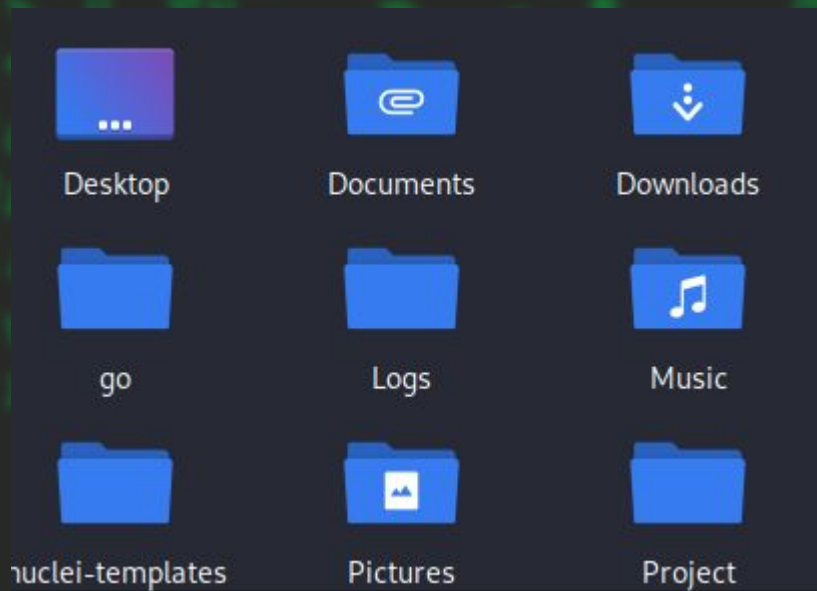


What is **command**

“Small programs that
do one task well”

ls / List Directory

- SYNOPSIS
 - **ls [OPTION] [FILE]**
- DESCRIPTION
 - List information about the Files (the current directory by default).



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



tree

- SYNOPSIS
 - **tree [FOLDER]**
- DESCRIPTION
 - List information about the Files and Folders with tree structure(the current directory by default).

```
~ > $ tree GTSTv1/
GTSTv1/
├── Day1 - Introduction.md
├── Notes
│   └── Season1
└── README.md

3 directories, 2 files
```

cont...

- ls -l -l: listed
- ls -a -a: Hidden
- ls filename
- ls -R -R: recursive

You can combine them,

ls -Rla

Linux hidden files start with dot.

```
(rexder@HunterMachine)-[~]  
$ ls -l  
total 32  
drwxr-xr-x 2 rexder rexder 4096 Dec 16 02:32 Desktop  
drwxr-xr-x 2 rexder rexder 4096 Dec 6 03:03 Documents  
drwxr-xr-x 4 rexder rexder 4096 Dec 9 08:08 Downloads  
drwxr-xr-x 2 rexder rexder 4096 Dec 6 03:03 Music  
drwxr-xr-x 2 rexder rexder 4096 Dec 15 02:21 Pictures  
drwxr-xr-x 2 rexder rexder 4096 Dec 6 03:03 Public  
drwxr-xr-x 2 rexder rexder 4096 Dec 6 03:03 Templates  
drwxr-xr-x 2 rexder rexder 4096 Dec 6 03:03 Videos
```

```
(rexder@HunterMachine)-[~]  
$ ls -la  
.  
..  
.bash_logout  
.bashrc  
.bashrc.original  
.cache  
.config  
Desktop  
Documents  
Downloads  
.face  
.face.icon  
.gtkr-2.0  
.kde  
.local  
.mozilla  
Music  
Pictures  
.profile  
Public  
Templates  
.vboxclient-clipboard.pid  
.vboxclient-display-svga-x11.pid  
.vboxclient-draganddrop.pid  
.vboxclient-seamless.pid  
Videos  
.zsh_history  
.zshrc
```




cd / Change Directory

SYNOPSIS

cd [directory]

DESCRIPTION

It is used to change current working directory.

```
(rexder@HunterMachine) ~  
$ cd Downloads
```

```
(rexder@HunterMachine) ~  
$
```




cont...

- `cd /` => root
- `cd` => Back to User's home dir
- `cd ..` => 1x Back
- `cd ../..` => 2x Back
- `cd foldername`

If folder name have space you have to add the name inside " folder name "

`cd "folder name"`

```
(rexder@HunterMachine)-[~]  
$ cd /  
  
(rexder@HunterMachine)-[/]  
$ cd ..  
  
(rexder@HunterMachine)-[/]  
$ cd home  
  
(rexder@HunterMachine)-[/home]  
$
```

Pwd / print working directory

SYNOPSIS

`pwd [options]`

DESCRIPTION

It prints the path of the working directory, starting from the root.

Example after typing pwd:

```
(rexder@HunterMachine)-[~/Documents]  
$ pwd  
/home/rexder/Documents
```

PWD(1)	User Commands	PWD(1)
NAME pwd - print name of current/working directory		
SYNOPSIS pwd [OPTION]...		
DESCRIPTION Print the full filename of the current working directory.		
-L, --logical use PWD from environment, even if it contains symlinks		
-P, --physical avoid all symlinks		
--help display this help and exit		
--version output version information and exit		



echo

SYNOPSIS

echo [string]

DESCRIPTION

echo command in linux is used to display line of text/string that are passed as an argument .

```
(rexder@HunterMachine)-[/home]  
$ echo "Hello Geeztech Security Testers."  
Hello Geeztech Security Testers.
```


Output Redirecting

- You can write Output of any commands into files, This is Called Redirecting.
- Todo this we will use the `>` Sign.
 - `echo text > file.txt`
- You can add texts(**append**)
 - `echo text >> file.txt`

```
(rexder@HunterMachine)-[~]  
$ echo "Hello Geeztech Security Testers." > geez.txt  
  
(rexder@HunterMachine)-[~]  
$ ls  
Desktop Downloads Music Public Videos  
Documents geez.txt Pictures Templates
```




cat / head / tail / less

```
(rexder@HunterMachine)-[~]  
$ cat geez.txt  
Hello Geeztech Security Testers.
```

- SYNOPSIS

- cat [FILE]...
- head [FILE]...
- tail [FILE]...
- less [FILE]...

- DESCRIPTION

- All are Used to show the content of a file.
- Head and Tail will display the 10 lines of the file.

```
$ cat VlanNote -n  
1 # Commands  
2 - enable  
3 - conf t  
4  
5 > To Create A VLAN  
6 - vlan <VLAN_NUM>  
7 - name <VLAN_NAME>  
8 - exit  
9  
10 > Assigning Single Ports to a VLAN  
11 - interface  
12 - int fa0/2  
13 - switchport mode access  
14 - switchport access vlan <VLAN_NUM>  
15 # We Use access because we are using accessport for the end users.  
16
```

-n will add number lines

```
~ > $ head VlanNote  
# Commands  
- enable  
- conf t
```



touch

SYNOPSIS

```
touch [FILE1] [FILE2] [FILE3]
```

DESCRIPTION

Creates any kind of Files with the name you gave it. With empty inside.

```
(rexder@HunterMachine)-[~]
$ ls
Desktop  Downloads  Music  Public  Videos
Documents  geez.txt  Pictures  Templates

(rexder@HunterMachine)-[~]
$ touch Day3.md

(rexder@HunterMachine)-[~]
$ ls
Day3.md  Documents  geez.txt  Pictures  Templates
Desktop  Downloads  Music  Public  Videos
```

Mkdir / make directory

- SYNOPSIS

- mkdir [FOLDER-NAME1]
[FOLDER-NAME2]
[FOLDER-NAME3]
- Mkdir -p folder1/folder2/folder3

- DESCRIPTION

- Creates Folder with the name u gave it.
- DON'T forget to add the " " when you are using folders with space between them.

```
(rexder@HunterMachine)-[~]  
$ ls  
Desktop  Downloads  Music  Public  Videos  
Documents  geez.txt  Pictures  Templates  
  
(rexder@HunterMachine)-[~]  
$ mkdir GeezTech  
  
(rexder@HunterMachine)-[~]  
$ ls  
Desktop  Downloads  geez.txt  Pictures  Templates  
Documents  GeezTech  Music  Public  Videos
```

```
(rexder@HunterMachine)-[~]  
$ ls  
Desktop  Downloads  Music  Public  Videos  
Documents  geez.txt  Pictures  Templates  
  
(rexder@HunterMachine)-[~]  
$ mkdir "Geez Tech"  
  
(rexder@HunterMachine)-[~]  
$ ls  
Desktop  Downloads  geez.txt  Pictures  Templates  
Documents  'Geez Tech'  Music  Public  Videos
```




Cont ...

```
~/GTSTv1 > $ ls
'Day1 - Introduction.md'  README.md
~/GTSTv1 > $ mkdir Notes/Season1
mkdir: cannot create directory 'Notes/Season1': No such file or directory
~/GTSTv1 > $ mkdir -p Notes/Season1
~/GTSTv1 > $ ls -la Notes/
total 12
drwxrwxr-x 3 rexder rexder 4096 Oct 25 17:03 ./
drwxrwxr-x 5 rexder rexder 4096 Oct 25 17:03 ../
drwxrwxr-x 2 rexder rexder 4096 Oct 25 17:03 Season1/
```




clear

SYNOPSIS

clear

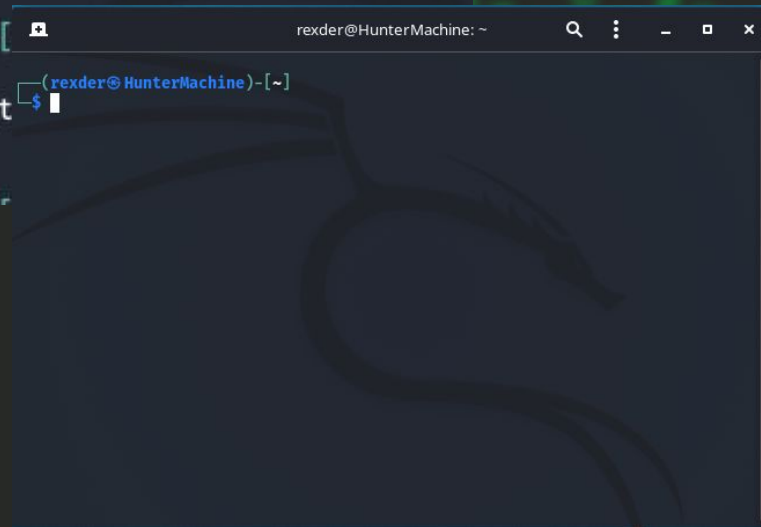
DESCRIPTION

Clears your screen.

```
(rexder@HunterMachine)-[~]  
$ ls  
Desktop  Downloads  Music      Public      Videos  
Documents geez.txt   Pictures   Templates
```

```
(rexder@HunterMachine)-[~]  
$ touch Day3.md
```

```
(rexder@HunterMachine)-[~]  
$ ls  
Day3.md  Documents  geez.txt  
Desktop  Downloads  Music
```





rm / remove

SYNOPSIS

rm [FILE1][FILE2][FILE3]

DESCRIPTION

Remove file.

```
(rexder@HunterMachine)-[~]
$ ls
Day3.md  Documents  geez.txt  Pictures  Templates
Desktop  Downloads  Music     Public    Videos

(rexder@HunterMachine)-[~]
$ rm Day3.md

(rexder@HunterMachine)-[~]
$ ls
Desktop  Downloads  Music  Public  Videos
Documents  geez.txt  Pictures  Templates
```



cont...

- `rm -r` => recursive(4 folders)
- `rm -i` => for prompt(ask)
- `rm -f` => force delete

```
(rexder@HunterMachine)-[~]  
$ ls  
Desktop      Downloads    geez.txt     Pictures     Templates  
Documents    'Geez Tech' Music        Public       Videos  
  
(rexder@HunterMachine)-[~]  
$ rm -rf 'Geez Tech'  
  
(rexder@HunterMachine)-[~]  
$ rm -i geez.txt  
rm: remove regular file 'geez.txt'? yes
```

You can use them in combination
too like, `rm -rf 'filename'`

cp | mv / copy,move

SYNOPSIS

cp [oldFILEplace] [newfilePlace]

Mv [oldFILEplace] [newfilePlace]

DESCRIPTION

Copy/move files & folders.

- If you want to Copy Folder which contains Files, You have to use **-r** option

```
~ > $ cp -r Desktop/ /root/
```

```
(rexder@HunterMachine)-[~]  
$ ls  
Desktop    Downloads  Music      Public     Videos  
Documents  geez.txt   Pictures   Templates  
  
(rexder@HunterMachine)-[~]  
$ cp geez.txt Documents  
  
(rexder@HunterMachine)-[~]  
$ cd Documents  
  
(rexder@HunterMachine)-[~/Documents]  
$ ls  
geez.txt
```




...Cont

You can Use the Notations
too to Move and Copy file/
Folders.

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

```
(rexder@HunterMachine) - [~]  
$ cp 'Cyber sec note.pdf' ../../tmp
```

```
(rexder@HunterMachine) - [/tmp]  
$ mv adb.1000.log ~/Downloads/
```

```
(rexder@HunterMachine) - [/tmp]  
$ mv * /root
```



Breaktime

TASKS:

- 1) Create 2 Folders called **gtst** & **linux**
 - 2) Open **gtst**, create file called **linux.txt** with text “Hello this is my first text”
 - 3) create python file with empty inside called **code.py**
 - 4) create markdown file with “ # Test is Going. ” called **'hacker'**
 - 5) Move the python file you created to the **linux** folder.
- 

grep - global search for regular expression

- **grep [options] pattern [files]**

- The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for global search for regular expression and print out).

```
(rexder@HunterMachine)-[~]  
$ ls  
Day3.md Desktop Downloads Music Public Videos  
Day4.md Documents grep_test.txt Pictures Templates  
  
(rexder@HunterMachine)-[~]  
$ grep "my" grep_test.txt  
hello my friend my name is nathan , my friend is good boy.
```

cont...

- **grep -i "search" file**
 - - case insensitive
- **grep -c "search" file**
 - - count numbers line
- **grep -l "search" ***
 - - displays filename
- **grep -r "search" foldername**
 - - search text in folders
- **grep -v 'term' filename**
 - To display without this term
- **grep -n "term" file**
 - To display the term find number line
- **grep -o 'pattern' filename**
 - To Display that specific word only

```
(rexder@ HunterMachine)-[~]
$ ls
Day3.md  debdabe.txt  Documents  grep_test.txt  Pictures  Templates  Videos
Day4.md  Desktop      Downloads  Music          Public    topsecret.txt

(rexder@ HunterMachine)-[~]
$ grep -i "hello" *
debdabe.txt:HELLO Ethiopia
grep: Desktop: Is a directory
grep: Documents: Is a directory
grep: Downloads: Is a directory
grep_test.txt:hello my friend my name is nathan , my friend is good boy.
grep: Music: Is a directory
grep: Pictures: Is a directory
grep: Public: Is a directory
grep: Templates: Is a directory
topsecret.txt:hello there
grep: Videos: Is a directory
```

```
(rexder@ HunterMachine)-[~]
$ ls
Day3.md  debdabe.txt  Documents  grep_test.txt  Pictures  Templates  Videos
Day4.md  Desktop      Downloads  Music          Public    topsecret.txt

(rexder@ HunterMachine)-[~]
$ grep -R "hello" *
grep_test.txt:hello my friend my name is nathan , my friend is good boy.
topsecret.txt:hello there
```



```
(rexder@HunterMachine) - [~/CTF/HTB/Machines]
$ grep -ran HTB
trickster/shop/admin634ewutrx1jgitlooaj/themes/default/package-lock.json:3799:      "integrity": "sha512-5nqDSx
I6WT+QqR547RwxQapgjQBmktkdP+HTBb/a/zLsbzERTONyUB5pefh5TtjQ==",
trickster/shop/admin634ewutrx1jgitlooaj/themes/new-theme/package-lock.json:4616:      "integrity": "sha512-5nqD
oLI6WT+QqR547RwxQapgjQBmktkdP+HTBb/a/zLsbzERTONyUB5pefh5TtjQ==",
trickster/rex/Metadata/Slic3r_PE_model.config:4:  <metadata type="object" key="name" value="TRICKSTER.HTB"/>
trickster/rex/Metadata/Slic3r_PE_model.config:6:  <metadata type="volume" key="name" value="TRICKSTER.HTB"/>
trickster/rex/Metadata/Slic3r_PE_model.config:10:  <slic3rpe:text text="TRICKSTER.HTB" style_name="NORMAL" fon
0;0;0;700;0;0;0;1;0;0;0;0;Segoe UI" font_descriptor_type="wxFontDescriptor_Windows" line_height="20" depth="5.6
```

```
(rexder@HunterMachine) - [~/CTF/HTB/Machines]
$ grep -rno HTB
trickster/shop/admin634ewutrx1jgitlooaj/themes/default/package-lock.json:3799:HTB
trickster/shop/admin634ewutrx1jgitlooaj/themes/new-theme/package-lock.json:4616:HTB
trickster/rex/Metadata/Slic3r_PE_model.config:4:HTB
trickster/rex/Metadata/Slic3r_PE_model.config:6:HTB
trickster/rex/Metadata/Slic3r_PE_model.config:10:HTB
```

```
(rexder@HunterMachine) - [~/CTF/HTB/Machines]
$ echo "Cyber Security is Very Fun Field" > hope.txt
```

```
(rexder@HunterMachine) - [~/CTF/HTB/Machines]
$ grep -c 'Fun' hope.txt
```

1



Wc - word count

SYNOPSIS

wc [OPTION]... [FILE]...

DESCRIPTION

It is used to find out **number of lines**, **word count**, **byte** and **characters count** in the files specified in the file arguments.

Line(-l) word(-w) byte(-c)

```
(rexder@HunterMachine)-[~]  
$ wc grep_test.txt  
1 13 59 grep_test.txt
```

```
(rexder@HunterMachine)-[~]  
$ cat grep_test.txt  
hello my friend my name is nathan , my friend is good boy.
```

```
(rexder@HunterMachine)-[~]  
$ wc -l grep_test.txt  
1 grep_test.txt
```



Multiple Command Executions

- You can run/ execute multiple commands in 1 line.
- using 3 methods:
 - And (&&)
 - Or (||)
 - Pipeing(|)



AND (&&)

On AND operation All commands you entered will be executed. If both are working without error

```
(rexder@HunterMachine)-[~]  
$ touch Day3.md && echo "Multi-lineing"  
Multi-lineing  
  
(rexder@HunterMachine)-[~]  
$ ls  
Day3.md  Documents  Music      Public      Videos  
Desktop  Downloads  Pictures   Templates
```

```
(rexder@HunterMachine)-[~]  
$ touh test && echo "not working"  
Command 'touh' not found, did you mean:  
  command 'touch' from deb coreutils  
Try: sudo apt install <deb name>
```




OR (||)

On OR operation the command will be executed. If it have error or not

```
(rexder@HunterMachine)-[~]  
$ touch Day4.md || echo "Multi-lineing"  
  
(rexder@HunterMachine)-[~]  
$ ls  
Day3.md  Desktop  Downloads  Pictures  Templates  
Day4.md  Documents Music      Public    Videos
```

```
(rexder@HunterMachine)-[~]  
$ touh test || echo "not working"  
Command 'touh' not found, did you mean:  
  command 'touch' from deb coreutils  
Try: sudo apt install <deb name>  
not working
```



Piping (|)

On pipe, will help you run commands by using the output of the 1st command as the input for the next one.

```
(rexder@HunterMachine)-[~]  
$ cat grep_test.txt | grep "hello"  
hello my friend my name is nathan , my friend is good boy.
```

```
(rexder@HunterMachine)-[~]  
$ cat grep_test.txt  
hello my friend my name is nathan , my friend is good boy.
```

```
(rexder@HunterMachine)-[~]  
$ cat grep_test.txt | wc -l  
1
```



EXERCISE 2

1) Create A text file with name paragraph.txt

“ Hello my name is <Yourname>, i got this course of linux and said hello, and am trying to learn linux. I love linux, so i said HELLO world “

2) How many times did the word hello written ?

3) how many byte is the size of this text

4) how many words are there?

Sed / Stream Editor

- A powerful command-line tool for parsing and transforming text in Linux.
- Processes files line-by-line, making it efficient for large text processing tasks.
- Common Uses:
 - Text substitution and replacement.
 - Deleting or selecting specific lines.
 - Efficient text manipulation for tasks like network information gathering or penetration testing logs.
- Syntax: `sed [options] 'command' file`
 - Substitute (Replace): `sed 's/old/new/' file`
 - You can use the `|` sign instead of `/` too. `'S|old|new|'`
 - Use `'g'` on the end to replace if it finds the word more than 1 time in one line. `'s/old/new/g'`
 - Delete: `sed '/pattern/d' file`

```
~/GTSTv1 > $ cat temp.txt
Hello my name is Nathan, i got this course of linux
x, so i said HELLO world
~/GTSTv1 > $ cat temp.txt | sed 's/Nathan/Minte/'
Hello my name is Minte, i got this course of linux a
, so i said HELLO world
```

```
~/GTSTv1 > $ cat temp.txt
Hello my name is Nathan, i got this course of linux and said Hello, and
nux. I love linux, so i said HELLO world
~/GTSTv1 > $ cat temp.txt | sed 's/Hello/Hi/'
Hi my name is Nathan, i got this course of linux and said Hello, and am
. I love linux, so i said HELLO world
~/GTSTv1 > $ cat temp.txt | sed 's/Hello/Hi/g'
Hi my name is Nathan, i got this course of linux and said Hi, and am try
love linux, so i said HELLO world
```


Cont...

```
~/GTSTv1 > $ cat temp.txt
Hello my name is Nathan, i got this course of linux a
to learn linux. I love linux, so i said HELLO world
Hello my name is Nati, i got this course of linux and
o learn linux. I love linux, so i said HELLO world
Hello my name is Nathan, i got this course of linux a
to learn linux. I love linux, so i said HELLO world
Hello my name is Minte i got this course of linux ar
to learn linux. I love linux, so i said HELLO world
~/GTSTv1 > $ cat temp.txt | sed '/Nathan/d'
Hello my name is Nati, i got this course of linux and
o learn linux. I love linux, so i said HELLO world
Hello my name is Minte i got this course of linux ar
to learn linux. I love linux, so i said HELLO world
```

awk

- A versatile command-line text-processing tool.
- Ideal for pattern scanning and data extraction in structured text.
- Named after its creators: Aho, Weinberger, and Kernighan.
- Features:
 - Processes text line-by-line.
 - Supports complex pattern matching.
 - Allows field-based text manipulation, making it great for column-based data like CSV files.
- Basic Syntax:
 - `awk 'pattern { action }' file.txt`
 - `awk '{print $1, $3}' file.txt`
 - # Prints columns 1 and 3
 - `awk '/pattern/ {print $0}' file.txt`
 - # Prints lines matching "pattern"

```
~/GTSTv1 > $ cat log.txt
ID,Name,Department,Salary,Join Date
101,John Doe,Engineering,65000,2022-05-01
102,Jane Smith,Marketing,55000,2021-08-15
103,Emily Johnson,Sales,70000,2020-09-22
104,Michael Brown,Engineering,72000,2019-12-01
105,Jessica White,Sales,68000,2021-01-10
106,Chris Green,Marketing,53000,2023-02-25
107,Sarah Black,Engineering,69000,2022-07-20
108,Tom Blue,Sales,71000,2018-06-18
109,Amy Gold,Engineering,65000,2023-03-30
110,David Silver,Marketing,56000,2020-11-11
```

```
~/GTSTv1 > $ cat log1.txt
Name Salary
John Doe 65000
Jane Smith 55000
Emily Johnson 70000
```

Cont...

- Awk by Default Determine, Columns if they are separated by space(), Here Space is known As **Delimiter**.
 - To Change Delimiter ' -F "Sign" '
- **Built-in Variables:**
 - **\$0**: Entire line of text.
 - **\$1, \$2,...**: Represents each column (field) in a line.
 - **NR**: Line (record) number.
 - **NF**: Number of fields in the current record.
- You can Use it with file or Pipped.
 - Awk 'options' file.txt
 - Cat 'file.txt' | awk 'options'

```
~/GTSTv1 > $ cat log1.txt | awk '{print $1}'
Name
John
Jane
Emily
~/GTSTv1 > $ cat log1.txt | awk '{print $2}'
Salary
Doe
Smith
Johnson
```

```
107,Sarah Black,Engineering,69000,2022-07-20
108,Tom Blue,Sales,71000,2018-06-18
109,Amy Gold,Engineering,65000,2023-03-30
110,David Silver,Marketing,56000,2020-11-11
~/GTSTv1 > $ cat log.txt | awk -F "," '{print $2}'
Name
John Doe
Jane Smith
Emily Johnson
Michael Brown
```


...

```
~/GTSTv1 > $ cat log.txt
ID,Name,Department,Salary,Join_Date
101,John Doe,Engineering,65000,2022-05-01
102,Jane Smith,Marketing,55000,2021-08-15
103,Emily Johnson,Sales,70000,2020-09-22
104,Michael Brown,Engineering,72000,2019-12-01
105,Jessica White,Sales,68000,2021-01-10
106,Chris Green,Marketing,53000,2023-02-25
107,Sarah Black,Engineering,69000,2022-07-20
108,Tom Blue,Sales,71000,2018-06-18
109,Amy Gold,Engineering,65000,2023-03-30
110,David Silver,Marketing,56000,2020-11-11
```

```
~/GTSTv1 > $ cat log.txt | awk -F "," '{/6/ {print $2}}'
John Doe
Jessica White
Chris Green
Sarah Black
Tom Blue
Amy Gold
David Silver
```

awk '\$3 > 50 {print \$1, \$3}' file.txt # Prints rows where column 3 is > 50

```
~/GTSTv1 > $ cat log.txt | awk -F "," '$4 > 68000 {print $2}'
Name
Emily Johnson
Michael Brown
Sarah Black
Tom Blue
```

```
~/GTSTv1 > $ cat log.txt | awk -F "," '{print $NF}'
Join_Date
2022-05-01
2021-08-15
2020-09-22
2019-12-01
2021-01-10
```

```
~/GTSTv1 > $ cat log.txt | awk -F "," '{print $0}'
ID,Name,Department,Salary,Join_Date
101,John Doe,Engineering,65000,2022-05-01
102,Jane Smith,Marketing,55000,2021-08-15
103,Emily Johnson,Sales,70000,2020-09-22
104,Michael Brown,Engineering,72000,2019-12-01
```

Sum of a Column:

```
~/GTSTv1 > $ cat log.txt | awk -F "," '{sum += $4} END {print "Total: ",sum}'
Total: 644000
```




Exercise #1 - sed

- Use this data and Do the Following Questions
- Data:
 - 1, John Doe, Sales, \$3500
 - 2, Jane Smith, IT, \$4000
 - 3, Sam Brown, Marketing, \$3800
 - 4, Lisa Ray, IT, \$4200
 - 5, Tom Scott, Sales, \$3300
 - 6, Anna White, Marketing, \$4100
 - 7, Bill Blue, Sales, \$3600
 - 8, Emma Gray, IT, \$3900
- Questions
 - Replace all occurrences of "IT" with "Information Technology".
 - Delete lines that contain "Sales".



Exercise #2 - awk

- Using the Data Below, Do the questions
- Data
 - Student,Subject,Grade
 - Alice,Math,89
 - Bob,Science,78
 - Charlie,Math,95
 - Alice,Science,92
 - Bob,Math,85
 - Charlie,Science,87

Challenge:

1. **Total Grade:** Calculate the Total grade for each student.
2. **Student Names:** Display the student Name Only.
3. **Specific Subject:** Print all grades for **Science**.



Class is Over!

Assignment

- 1) Write notes on github
- 2) Do the exercises again those are the most basic and essential once.