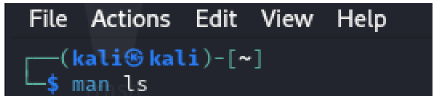
Topic Name:

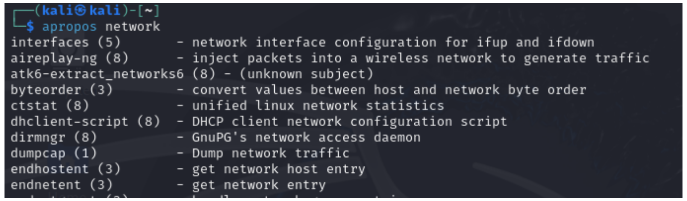
The main aim of this lab session is to provide hands-on experience on

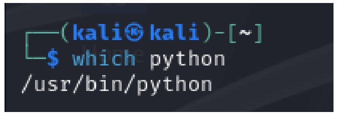
- Getting Help
- Basic Commands
- Navigation
- File System
- simple shell script

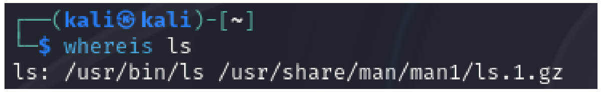
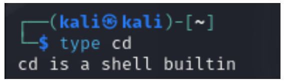
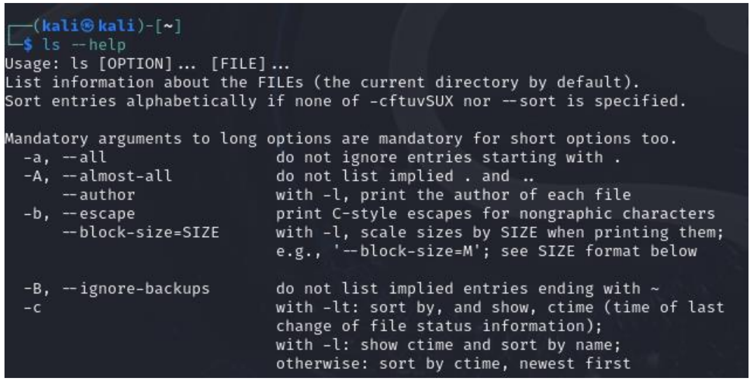
1. Getting Help

Task	Command Name	Syntax	Example	Screenshots

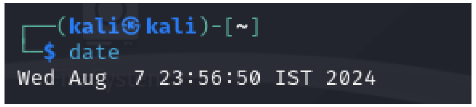
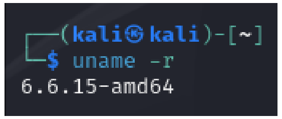
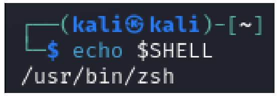
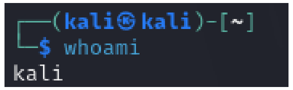
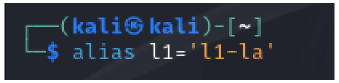
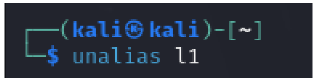
To get manual page for the known command	'man'	man [command_name]	man ls	
				`┌──(kali㉿kali)-[~]`
				`└─$ man ls`

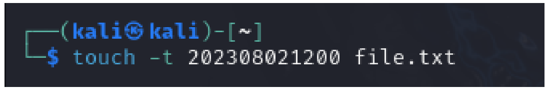
To get manual page for the unknown command	'apropos'	apropos [keyword]	apropos network	`┌──(kali㉿kali)-[~]`
				`└─$ apropos network`
				`interfaces (5)       - network interface configuration for ifup and ifdown`
				`aireplay-ng (8)      - inject packets into a wireless network to generate traffic`
				`atk6-extract_networks6 (8) - (unknown subject)`
				`byteorder (3)        - convert values between host and network byte order`
				`ctstat (8)           - unified linux network statistics`
				`dhclient-script (8)  - DHCP client network configuration script`
				`dirmngr (8)          - GnuPG's network access daemon`
				`dumpcap (1)          - Dump network traffic`
				`endhostent (3)       - get network host entry`
				`endnetent (3)        - get network entry`

To know the source file binary	'which'	which [command_name]	which python	`┌──(kali㉿kali)-[~]`
				`└─$ which python`
				`/usr/bin/python`

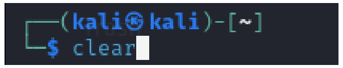
To know the path of the command	‘whereis’	whereis [command_name]	whereis ls	 ┌──(kali㉿kali)-[~] └─$ whereis ls ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
To know the command is external or internal	‘type'	type [command_name]	type cd	 ┌──(kali㉿kali)-[~] └─$ type cd cd is a shell builtin
To get help for the internal command	‘help’	help [internal_command_name]	help cd	Error Message : command_not_found
To list out bash commands	‘compgen -c'	compgen -c	compgen -c	Error Message : command_not_found
To know the usage of the command	‘[command_name] --help’	[command_name] --help	ls --help	 ┌──(kali㉿kali)-[~] └─$ ls --help Usage: ls [OPTION]... [FILE]... List information about the FILEs (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified. Mandatory arguments to long options are mandatory for short options too. -a, --all do not ignore entries starting with . -A, --almost-all do not list implied . and .. --author with -l, print the author of each file -b, --escape print C-style escapes for nongraphic characters --block-size=SIZE with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below -B, --ignore-backups do not list implied entries ending with ~ -c with -lt: sort by, and show, ctime (time of last change of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first

2. Basic Commands

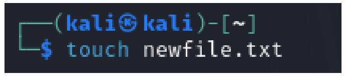
Task	Command Name	Syntax	Example	Screenshots
To know today's date	'date'	date	date	 ┌──(kali㉿kali)-[~] └─$ date Wed Aug 7 23:56:50 IST 2024
To print calendar	'cal'	cal [month] [year]	cal 8 2024	Error Message : command_not_found
To print kernel version	'uname'	uname -r	uname -r	 ┌──(kali㉿kali)-[~] └─$ uname -r 6.6.15-amd64
To print default shell	'echo'	echo $SHELL	echo $SHELL	 ┌──(kali㉿kali)-[~] └─$ echo $SHELL /usr/bin/zsh
To print currently logged in user	'whoami'	whoami	whoami	 ┌──(kali㉿kali)-[~] └─$ whoami kali
To create shortcut for command	'alias'	alias [shortcut_name]='[command]'	alias l1='ls –la'	 ┌──(kali㉿kali)-[~] └─$ alias l1='l1-la'
To delete shortcut	'unalias'	unalias [shortcut_name]	unalias l1	 ┌──(kali㉿kali)-[~] └─$ unalias l1

<table>
<tr><td>To change the timestamp of the file</td><td>‘touch’</td><td>touch [options] [file]</td><td>touch –t 202308021200 file.txt</td><td>


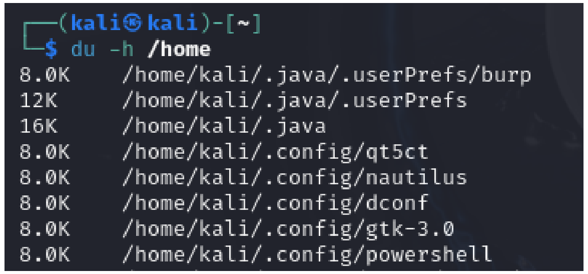
```
┌──(kali㉿kali)-[~]
└─$ touch -t 202308021200 file.txt
```

</td></tr>
<tr><td>To clear the screen</td><td>‘clear’</td><td>clear</td><td>clear</td><td>


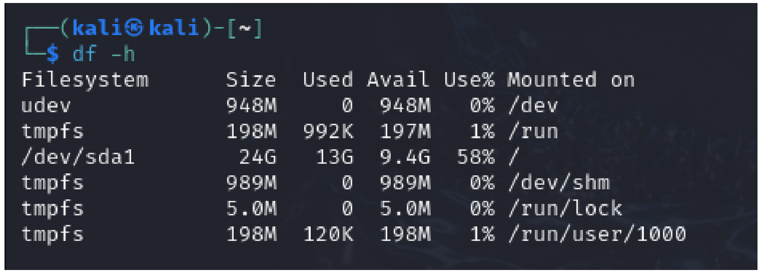
```
┌──(kali㉿kali)-[~]
└─$ clear
```

</td></tr>
<tr><td>To create empty files</td><td>‘touch’</td><td>touch [file_name]</td><td>touch newfile.txt</td><td>


```
┌──(kali㉿kali)-[~]
└─$ touch newfile.txt
```

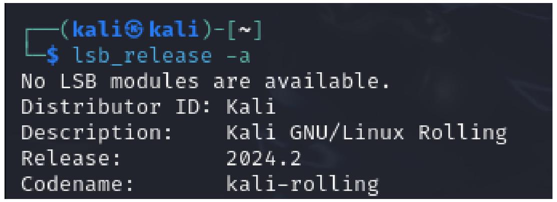
</td></tr>
<tr><td>To know disk usage</td><td>‘du’</td><td>du [options] [directory]</td><td>du -h /home</td><td>


```
┌──(kali㉿kali)-[~]
└─$ du -h /home
8.0K    /home/kali/.java/.userPrefs/burp
12K     /home/kali/.java/.userPrefs
16K     /home/kali/.java
8.0K    /home/kali/.config/qt5ct
8.0K    /home/kali/.config/nautilus
8.0K    /home/kali/.config/dconf
8.0K    /home/kali/.config/gtk-3.0
8.0K    /home/kali/.config/powershell
```

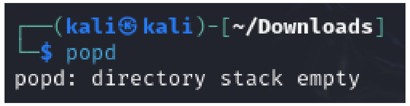
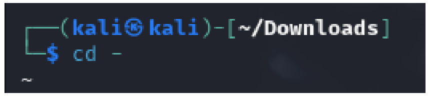
</td></tr>
<tr><td>To know free space in the system</td><td>‘df’</td><td>df [options]</td><td>df -h</td><td>


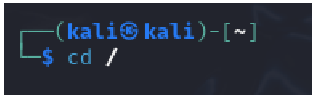
```
┌──(kali㉿kali)-[~]
└─$ df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            948M     0  948M   0% /dev
tmpfs           198M  992K  197M   1% /run
/dev/sda1        24G   13G  9.4G  58% /
tmpfs           989M     0  989M   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           198M  120K  198M   1% /run/user/1000
```

</td></tr>
</table>

To know about the Linux release	‘lsb_release’	lsb_release –a	lsb_release -a	 ┌──(kali㉿kali)-[~] └─$ lsb_release -a No LSB modules are available. Distributor ID: Kali Description: Kali GNU/Linux Rolling Release: 2024.2 Codename: kali-rolling

3. Navigation

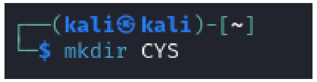
Task	Command	Syntax	Screenshots
To navigate home directory	‘cd’	cd	 ┌──(kali㉿kali)-[~] └─$ cd
To navigate to the parent directory	‘cd’	cd ..	 ┌──(kali㉿kali)-[~] └─$ cd ..
To navigate to the child directory	‘cd’	cd [child_directory_name]	 ┌──(kali㉿kali)-[~] └─$ cd Downloads
Alternate command to cd	‘pushd’/’popd’	pushd [directory], popd	 ┌──(kali㉿kali)-[~/Downloads] └─$ popd popd: directory stack empty
To go back to the previous directory	‘cd’	cd -	 ┌──(kali㉿kali)-[~/Downloads] └─$ cd - ~

To go to the root directory	‘cd’	cd /	 ┌──(kali㉿kali)-[~] └─$ cd /

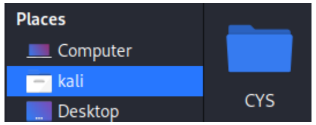
4. File System

Task	Syntax	Command
How to identify the file system	file –s [device]	‘file’

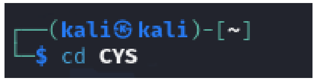
a. Create Folder “CYS”



```
┌──(kali㉿kali)-[~]
└─$ mkdir CYS
```

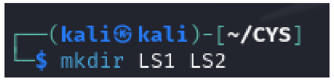


b. Navigate to CYS

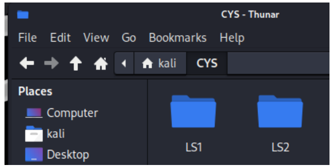


```
┌──(kali㉿kali)-[~]
└─$ cd CYS
```

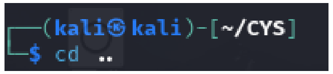
c. Create folder LS1 and LS2 under CYS



```
┌──(kali㉿kali)-[~/CYS]
└─$ mkdir LS1 LS2
```



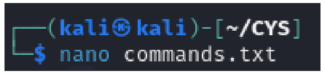
d. Go back to CYS



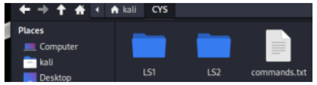
```
┌──(kali㉿kali)-[~/CYS]
└─$ cd ..
```

e. Working with Files

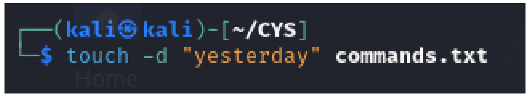
i. Add commands which you learnt during lab session in the file commands.txt



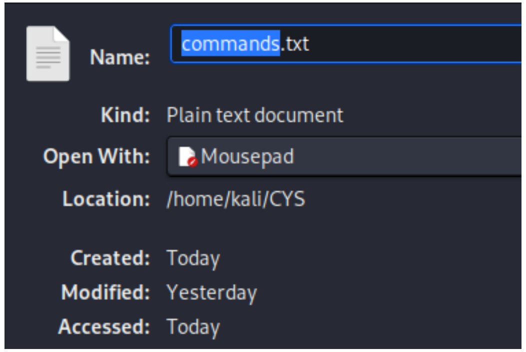
```
┌──(kali㉿kali)-[~/CYS]
└─$ nano commands.txt
```



ii. Change the timestamp of the file to yesterday



```
┌──(kali㉿kali)-[~/CYS]
└─$ touch -d "yesterday" commands.txt
```



Name: commands.txt

Kind: Plain text document

Open With: Mousepad

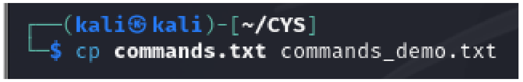
Location: /home/kali/CYS

Created: Today

Modified: Yesterday

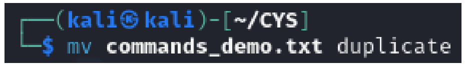
Accessed: Today

iii. Copy the contents from the file commands.txt to commands_demo.txt



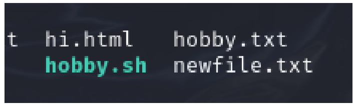
```
┌──(kali㉿kali)-[~/CYS]
└─$ cp commands.txt commands_demo.txt
```

iv. Rename the file commands_demo.txt to duplicate



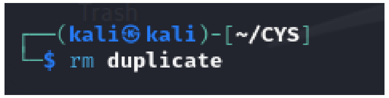
```
┌──(kali㉿kali)-[~/CYS]
└─$ mv commands_demo.txt duplicate
```

v. Rename all .html to .hldd



```
t  hi.html    hobby.txt
   hobby.sh   newfile.txt
```

vi. Delete the file duplicate

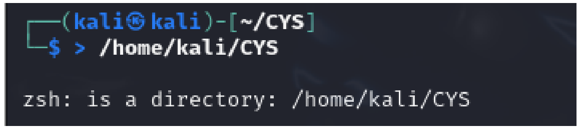


```
┌──(kali㉿kali)-[~/CYS]
└─$ rm duplicate
```

vii. Copy the contents commands.txt to unit4 and unit5 (using relative path)

Error Message – “cannot stat ‘unit4/’ : Not a directory”

viii. Delete the contents from unit5 (using absolute path)

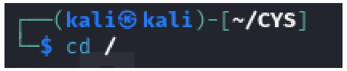


```
┌──(kali㉿kali)-[~/CYS]
└─$ > /home/kali/CYS

zsh: is a directory: /home/kali/CYS
```

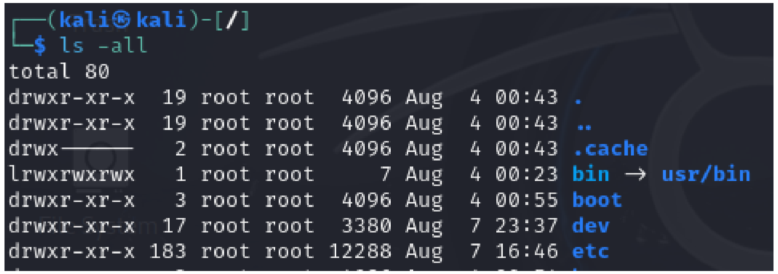
ix.

x. Navigate to root



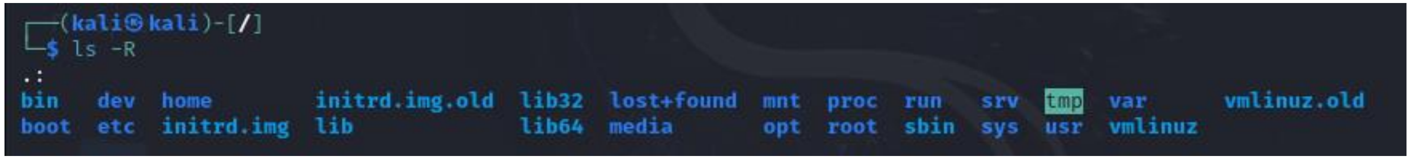
```
┌──(kali㉿kali)-[~/CYS]
└─$ cd /
```

xi. List all the files under root



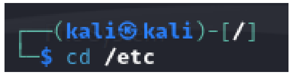
```
┌──(kali㉿kali)-[/]
└─$ ls -all
total 80
drwxr-xr-x  19 root root  4096 Aug  4 00:43 .
drwxr-xr-x  19 root root  4096 Aug  4 00:43 ..
drwx------   2 root root  4096 Aug  4 00:43 .cache
lrwxrwxrwx   1 root root     7 Aug  4 00:23 bin → usr/bin
drwxr-xr-x   3 root root  4096 Aug  4 00:55 boot
drwxr-xr-x  17 root root  3380 Aug  7 23:37 dev
drwxr-xr-x 183 root root 12288 Aug  7 16:46 etc
```

xii. Explore all the folders (Do not delete any folder)



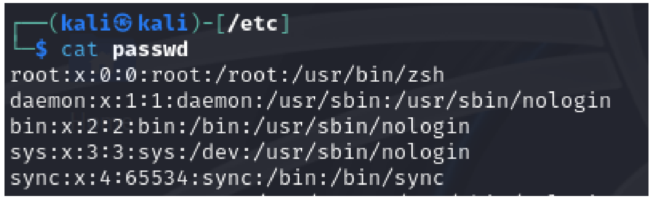
```
┌──(kali㉿kali)-[/]
└─$ ls -R
.:
bin   dev  home        initrd.img.old  lib32  lost+found  mnt  proc  run   srv  tmp  var      vmlinuz.old
boot  etc  initrd.img  lib             lib64  media       opt  root  sbin  sys  usr  vmlinuz
```

xiii. Navigate to /etc/passwd



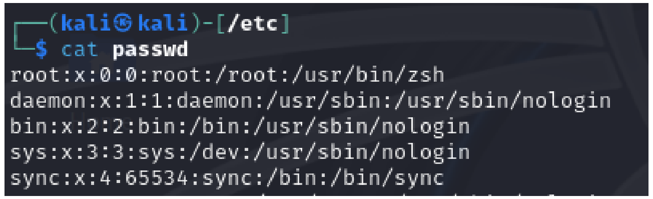
```
┌──(kali㉿kali)-[/]
└─$ cd /etc
```

xiv. Open the file passwd



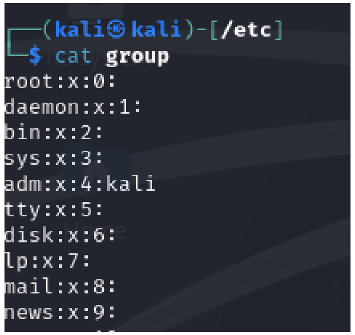
```
┌──(kali㉿kali)-[/etc]
└─$ cat passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

xv. Explore the file passwd



```
┌──(kali㉿kali)-[/etc]
└─$ cat passwd
root:x:0:0:root:/root:/usr/bin/zsh
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
```

xvi. Navigate to /etc/group and explore



```
┌──(kali㉿kali)-[/etc]
└─$ cat group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:kali
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
```

f. Difference between

i. GUI vs. CLI

Feature	GUI (Graphical User Interface)	CLI (Command Line Interface)
Interface	Visual with windows, icons, and menus	Text-based with commands and output
Interaction	Mouse and keyboard for clicking and selecting	Keyboard for typing commands
Ease of Use	Generally more user-friendly and intuitive	Requires knowledge of commands and syntax

Feature	GUI (Graphical User Interface)	CLI (Command Line Interface)
Learning Curve	Generally easier for beginners	Steeper learning curve, especially for advanced tasks
Speed	Can be slower due to graphical rendering	Often faster as it avoids graphical overhead
Resource Usage	Higher resource usage due to graphical elements	Lower resource usage
Automation	Limited automation capabilities	Strong automation with scripting and batch files
Accessibility	Useful for users who prefer visual interaction	Preferred by advanced users and administrators
Customization	Limited compared to CLI	Highly customizable with scripts and commands
Feedback	Immediate visual feedback	Feedback through text output
Multitasking	Easier to switch between tasks visually	Task switching done via command execution
Examples	Windows Explorer, GNOME, macOS Finder	Bash, PowerShell, Command Prompt

ii. man vs info

Feature	`man` (Manual Pages)	`info`
Purpose	Provides documentation for commands and system functions	Provides detailed and structured documentation for commands and software
Format	Text-based, often less navigable	Hypertext format with sections and links
Navigation	Linear, with section headings and plain text	Non-linear, with hyperlinks and navigation options
Search Capability	Search within the manual page using `/`	Search with `s` or use the menu system
Sections	Divided into sections (e.g., NAME, SYNOPSIS, DESCRIPTION)	Divided into nodes and chapters
Usage	man command	info command
Display Style	Usually one large page with scrolling or page navigation	Structured with menus, sections, and links

Feature	man (Manual Pages)	info
Additional Info	Provides concise information	Often more comprehensive with examples and detailed explanations
Examples	man ls	info coreutils 'ls invocation'
Integration	Generally integrated with the system’s help tools	Often used for GNU and other open-source documentation

iii. which vs. whereis

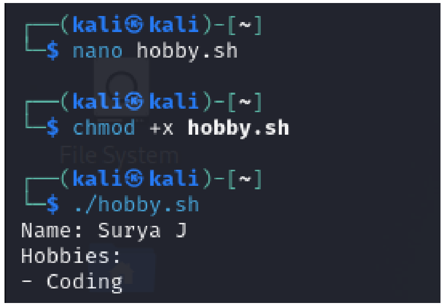
Feature	which	whereis
Purpose	Locates the path of an executable in the PATH environment variable	Locates the binary, source, and manual page files for a command
Command Syntax	which command	whereis command
Output	Returns the path of the executable	Returns paths to the binary, source, and man pages
Scope	Searches only the directories in the PATH	Searches predefined system directories for binaries, sources, and man pages
Use Case	To find the exact location of executables	To get comprehensive information about the command’s files
Example	which ls might return /bin/ls	whereis ls might return ls: /bin/ls /usr/share/man/man1/ls.1.gz
Specificity	More focused on executables	Broader search including source code and documentation
Customization	Limited to PATH variable	Can search specific paths using options like -b, -m, and -s
Options	Few options, mainly --all for all matches	Several options for more refined searches (e.g., -b for binaries only, -m for man pages)

iv. Terminal vs shell

Feature	Terminal	Shell
Definition	An interface for accessing the shell	A command interpreter that executes commands

Feature	Terminal	Shell
Purpose	Provides a window or environment to run the shell	Interprets and executes user commands
Functionality	Acts as a wrapper around the shell	Processes and runs commands, scripts, and programs
Interaction	Users type commands into the terminal	The shell reads and executes these commands
Examples	GNOME Terminal, xterm, Konsole, Terminal.app	Bash, Zsh, Fish, PowerShell
User Interface	Typically includes features like tabs, text settings, and scrollback	Primarily text-based with command-line interface
Persistence	Maintains a session for the shell	Executes commands in the current session or subshell
Usage	To access the shell and interact with the system	To execute commands, run scripts, and manage processes
Customization	Limited to terminal emulator settings	Highly customizable through shell configuration files (e.g., .bashrc, .zshrc)
Multitasking	Allows multiple shell sessions through tabs or windows	Handles command execution and job control within a single session
Visuals	Can display colors, fonts, and styles	Outputs text and command results

g. Write a simple shell script to print your name and your hobbies!



```
┌──(kali㉿kali)-[~]
└─$ nano hobby.sh

┌──(kali㉿kali)-[~]
└─$ chmod +x hobby.sh

┌──(kali㉿kali)-[~]
└─$ ./hobby.sh
Name: Surya J
Hobbies:
- Coding
```

Interesting commands to Explore

Banner

History

Note: Include your screenshots

Evaluation :

Marks : 10 (Deadline : 4 – Originality :3 – Completeness :3)

Deadline: 06.08.2024

“All our dreams can come true if we have the courage to pursue them.”

- Walt Disney