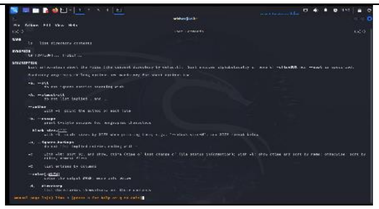
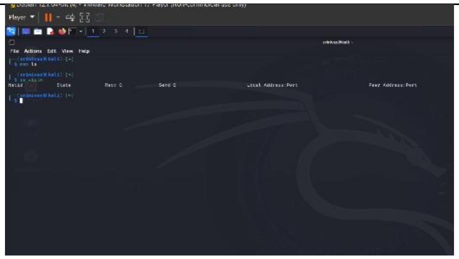
Topic Name:

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

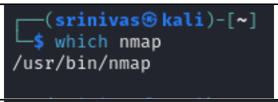
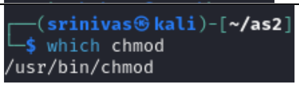
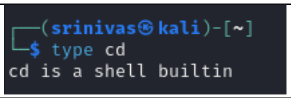
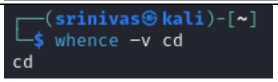
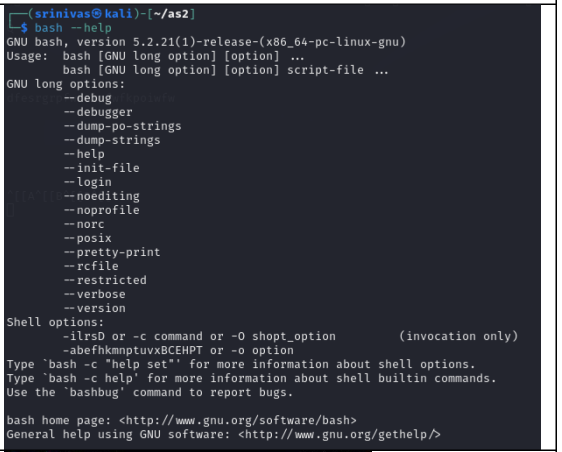
- Getting Help
- Basic Commands
- Navigation
- File System
- simple shell script

1. Getting Help

Task	Command Name	Syntax	Example	Screenshots
To get manual page for the known command	Command name is '**man**'	man[command name]	man ls	
To get manual page for the unknown command	Command name is '**which**'	which command_name	ss -tuln	

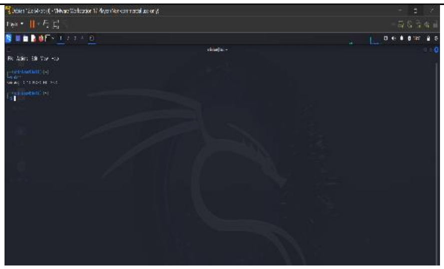
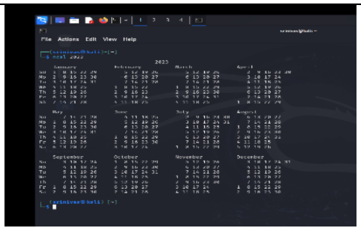
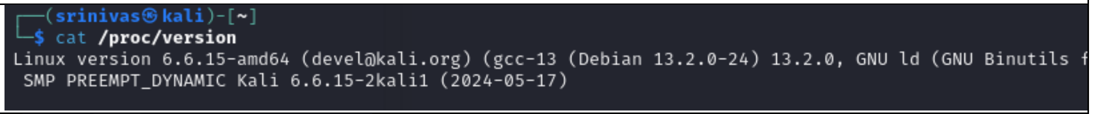
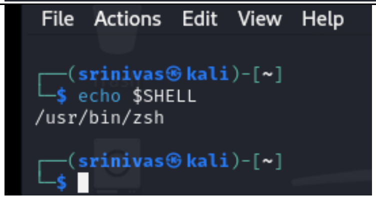
```
NAME
       ls - list directory contents
SYNOPSIS
       ls [OPTION]... [FILE]...
DESCRIPTION
[illegible]
```

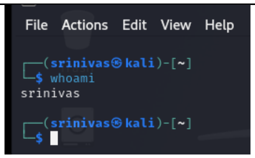
```
File  Actions  Edit  View  Help
┌──(srinivas㉿kali)-[~]
└─$ man ls

┌──(srinivas㉿kali)-[~]
└─$ ss -tuln
Netid        State        Recv-Q        Send-Q        Local Address:Port        Peer Address:Port

┌──(srinivas㉿kali)-[~]
└─$
```

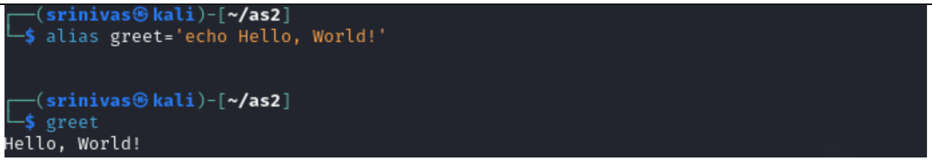
To know the source file binary	Command is ‘**where**’	where command_name	which nmap	 ┌──(srinivas㉿kali)-[~] └─$ which nmap /usr/bin/nmap
To know the path of the command	Command is ‘**which**’	which command _name	which chmod	 ┌──(srinivas㉿kali)-[~/as2] └─$ which chmod /usr/bin/chmod
To know the command is external or internal	Command is ‘**type**’	type command_type	type cd	 ┌──(srinivas㉿kali)-[~] └─$ type cd cd is a shell builtin
To get help for the internal command	Command is **‘whence’**	whence command_name	whence cd	 ┌──(srinivas㉿kali)-[~] └─$ whence -v cd cd
To list out bash commands	Command is **‘help’**	bash --help	Bash --help	
To know the usage of the command	Command is apropos	man command_name	man ls	

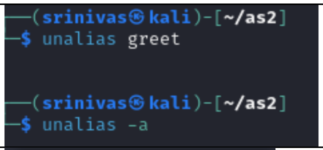
```
┌──(srinivas㉿kali)-[~/as2]
└─$ bash --help
GNU bash, version 5.2.21(1)-release-(x86_64-pc-linux-gnu)
Usage:  bash [GNU long option] [option] ...
        bash [GNU long option] [option] script-file ...
GNU long options:
        --debug
        --debugger
        --dump-po-strings
        --dump-strings
        --help
        --init-file
        --login
        --noediting
        --noprofile
        --norc
        --posix
        --pretty-print
        --rcfile
        --restricted
        --verbose
        --version
Shell options:
        -ilrsD or -c command or -O shopt_option         (invocation only)
        -abefhkmnptuvxBCEHPT or -o option
Type `bash -c "help set"' for more information about shell options.
Type `bash -c help' for more information about shell builtin commands.
Use the `bashbug' command to report bugs.

bash home page: <http://www.gnu.org/software/bash>
General help using GNU software: <http://www.gnu.org/gethelp/>
```

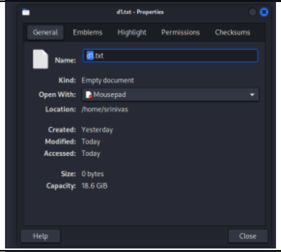
2. Basic Commands

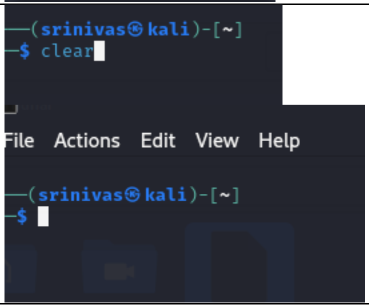
Task	Command Name	Syntax	Example	Screenshots
To know today's date	date	date	date	
To print calendar	cal	ncal	ncal 2023	
To print kernel version	cat/proc/version	cat/proc/version	cat /proc/version	 ┌──(srinivas㉿kali)-[~] └─$ cat /proc/version Linux version 6.6.15-amd64 (devel@kali.org) (gcc-13 (Debian 13.2.0-24) 13.2.0, GNU ld (GNU Binutils f SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-05-17)
To print default shell	echo $SHELL	echo $SHELL	echo $SHELL	 File Actions Edit View Help ┌──(srinivas㉿kali)-[~] └─$ echo $SHELL /usr/bin/zsh ┌──(srinivas㉿kali)-[~] └─$

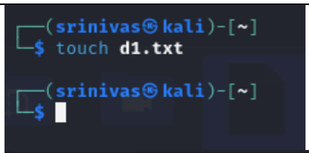
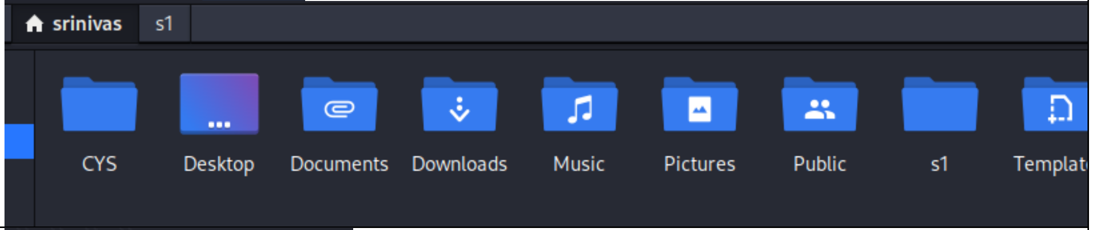
To print currently logged in user	whoami	whoami	whoami	 File Actions Edit View Help
				`┌──(srinivas㉿kali)-[~]`
				`└─$ whoami`
				`srinivas`
				`┌──(srinivas㉿kali)-[~]`
				`└─$`

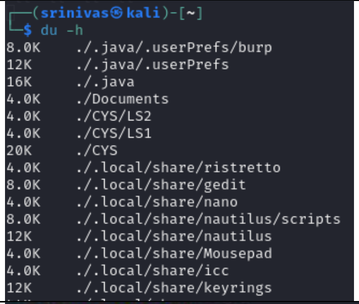
To create shortcut for command	alias	alias shortcut_name=command	alias greet='echo Hello, World!'	 `┌──(srinivas㉿kali)-[~/as2]`
				`└─$ alias greet='echo Hello, World!'`
				`┌──(srinivas㉿kali)-[~/as2]`
				`└─$ greet`
				`Hello, World!`

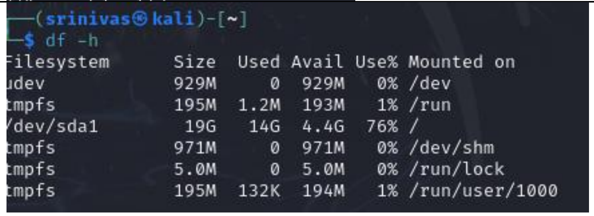
To delete shortcut	unalias	unalias shortcut_name	unalias ll	 `──(srinivas㉿kali)-[~/as2]`
				`─$ unalias greet`
				`──(srinivas㉿kali)-[~/as2]`
				`─$ unalias -a`

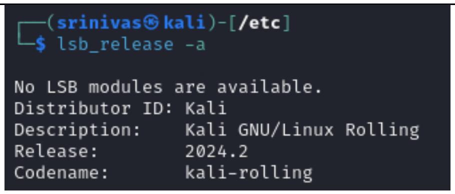
To change the timestamp of the file	touch	touch –t <yearmonthdaytime>	touch -t 202308052223 s1	 d1.txt - Properties General Emblems Highlight Permissions Checksums Name: d1.txt Kind: Empty document Open With: Mousepad Location: /home/srinivas Created: Yesterday Modified: Today Accessed: Today Size: 0 bytes Capacity: 18.6 GiB Help Close

To clear the screen	clear	clear	clear	 `──(srinivas㉿kali)-[~]`
				`─$ clear`
				File Actions Edit View Help
				`──(srinivas㉿kali)-[~]`
				`─$`

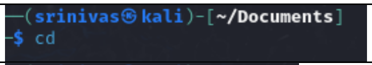
To create empty files	touch	touch.filename	touch d1.txt	 `┌──(srinivas㉿kali)-[~]`
				`└─$ touch d1.txt`
				
				`┌──(srinivas㉿kali)-[~]`
				`└─$ ▮`
				 srinivas
				s1
				CYS
				Desktop
				Documents
				Downloads
				Music
				Pictures
				Public
				s1
				Templat

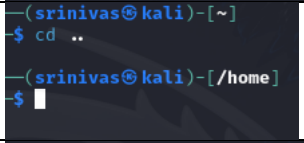
To know disk usage	du	du [options][path]	du –h	 `┌──(srinivas㉿kali)-[~]`
				`└─$ du -h`
				`8.0K    ./.java/.userPrefs/burp`
				`12K     ./.java/.userPrefs`
				`16K     ./.java`
				`4.0K    ./Documents`
				`4.0K    ./CYS/LS2`
				`4.0K    ./CYS/LS1`
				`20K     ./CYS`
				`4.0K    ./.local/share/ristretto`
				`8.0K    ./.local/share/gedit`
				`4.0K    ./.local/share/nano`
				`8.0K    ./.local/share/nautilus/scripts`
				`12K     ./.local/share/nautilus`
				`4.0K    ./.local/share/Mousepad`
				`4.0K    ./.local/share/icc`
				`12K     ./.local/share/keyrings`

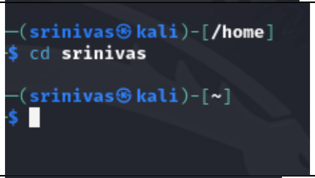
To know free space in the system	df	df[options]	df -h	 `┌──(srinivas㉿kali)-[~]`
				`└─$ df -h`
				`Filesystem      Size  Used Avail Use% Mounted on`
				`udev            929M     0  929M   0% /dev`
				`tmpfs           195M  1.2M  193M   1% /run`
				`/dev/sda1        19G   14G  4.4G  76% /`
				`tmpfs           971M     0  971M   0% /dev/shm`
				`tmpfs           5.0M     0  5.0M   0% /run/lock`
				`tmpfs           195M  132K  194M   1% /run/user/1000`

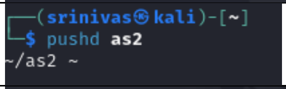
To know about the Linux release	lsb_release	lsb_release -a	lsb_release -a	 `┌──(srinivas㉿kali)-[/etc]`
				`└─$ lsb_release -a`
				
				`No LSB modules are available.`
				`Distributor ID: Kali`
				`Description:    Kali GNU/Linux Rolling`
				`Release:        2024.2`
				`Codename:       kali-rolling`

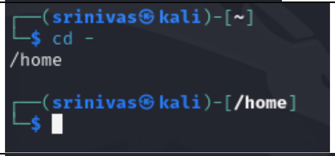
3. Navigation

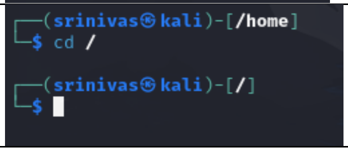
Task	Command	Syntax	Screenshots
To navigate home directory	cd	cd	 `─(srinivas㉿kali)-[~/Documents]` `$ cd`

To navigate to the parent directory	cd ..	cd ..	 `─(srinivas㉿kali)-[~]`
			`$ cd ..`
			`─(srinivas㉿kali)-[/home]`
			`$`

To navigate to the child directory	cd <directory name>	cd <directory name>	 `─(srinivas㉿kali)-[/home]`
			`$ cd srinivas`
			`─(srinivas㉿kali)-[~]`
			`$`

Alternate command to cd	pushd	pushd <directory name>	 `┌──(srinivas㉿kali)-[~]`
			`└─$ pushd as2`
			`~/as2 ~`

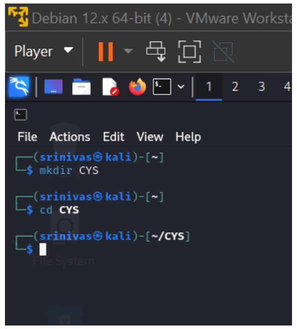
To go back to the previous directory	cd -	cd -	 `┌──(srinivas㉿kali)-[~]`
			`└─$ cd -`
			`/home`
			`┌──(srinivas㉿kali)-[/home]`
			`└─$`

To go to the root directory	cd /	cd /	 `┌──(srinivas㉿kali)-[/home]`
			`└─$ cd /`
			`┌──(srinivas㉿kali)-[/]`
			`└─$`

4. File System

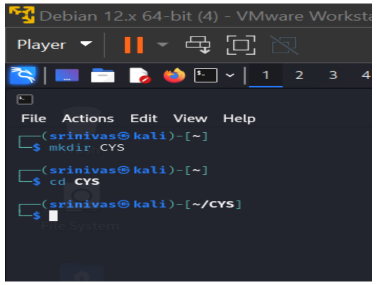
Task	**Syntax**	**Command**
How to identify the file system	**lsblk -f**	**lsblk –f**

a. Create Folder “CYS”


Debian 12.x 64-bit (4) - VMware Worksta
Player
File Actions Edit View Help

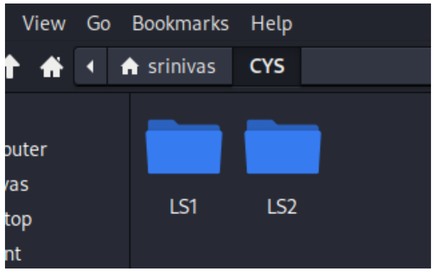
```
┌──(srinivas㉿kali)-[~]
└─$ mkdir CYS

┌──(srinivas㉿kali)-[~]
└─$ cd CYS

┌──(srinivas㉿kali)-[~/CYS]
└─$ 
```

b. Navigate to CYS

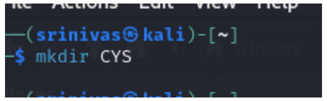

Debian 12.x 64-bit (4) - VMware Worksta
Player
File Actions Edit View Help

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

┌──(srinivas㉿kali)-[~]
└─$ cd CYS

┌──(srinivas㉿kali)-[~/CYS]
└─$ 
```

c. Create folder LS1 and LS2 under CYS



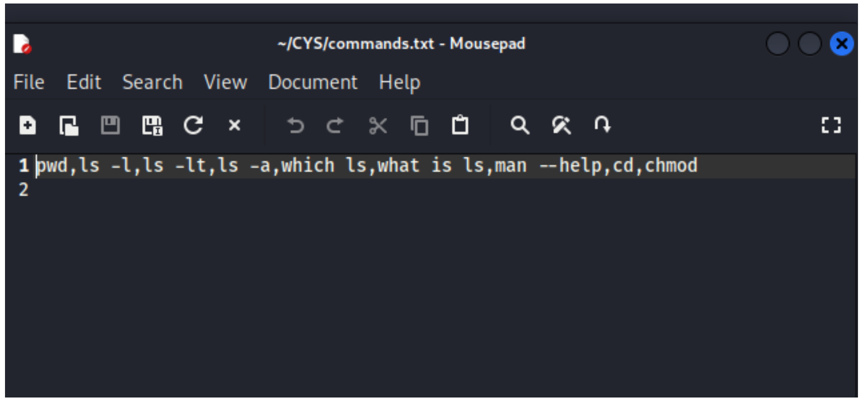
d. Go back to CYS



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

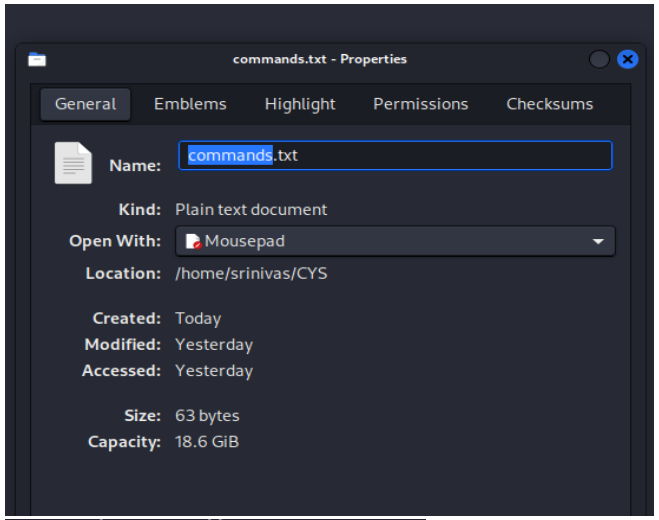
e. Working with Files

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

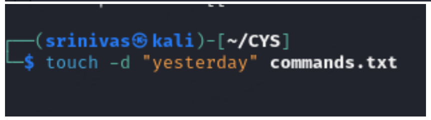

~/CYS/commands.txt - Mousepad
File Edit Search View Document Help

```
pwd,ls -l,ls -lt,ls -a,which ls,what is ls,man --help,cd,chmod
```

ii. Change the timestamp of the file to yesterday

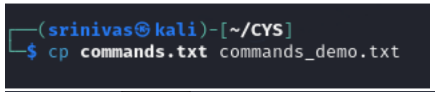

commands.txt - Properties
General Emblems Highlight Permissions Checksums

Name: commands.txt
Kind: Plain text document
Open With: Mousepad
Location: /home/srinivas/CYS
Created: Today
Modified: Yesterday
Accessed: Yesterday
Size: 63 bytes
Capacity: 18.6 GiB

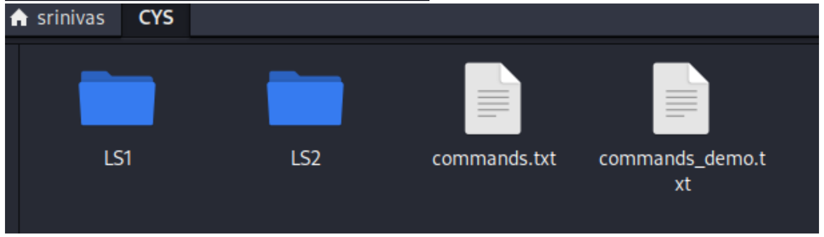


```
┌──(srinivas㉿kali)-[~/CYS]
└─$ touch -d "yesterday" commands.txt
```

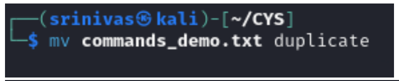
iii. Copy the contents from the file commands.txt to commands_demo.txt



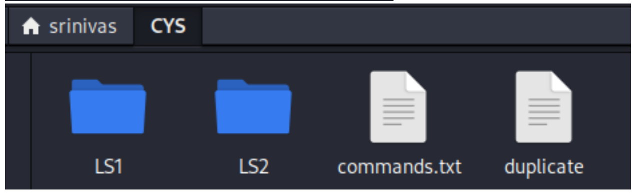
```
┌──(srinivas㉿kali)-[~/CYS]
└─$ cp commands.txt commands_demo.txt
```



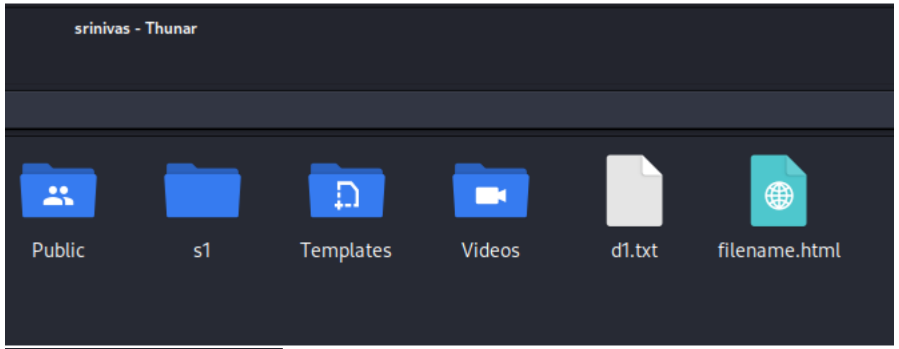
iv. Rename the file commands_demo.txt to duplicate

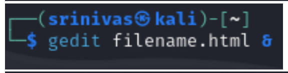


```
┌──(srinivas㉿kali)-[~/CYS]
└─$ mv commands_demo.txt duplicate
```

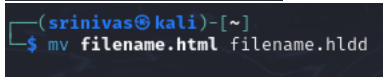


v. Rename all .html to .hldd

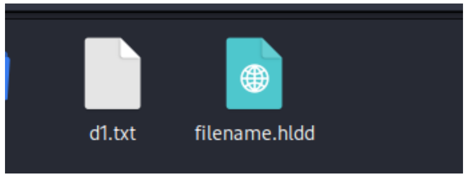




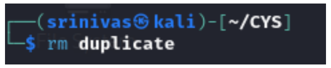
```
┌──(srinivas㉿kali)-[~]
└─$ gedit filename.html &
```



```
┌──(srinivas㉿kali)-[~]
└─$ mv filename.html filename.hldd
```

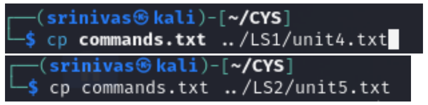


vi. Delete the file duplicate



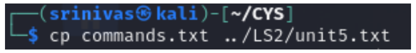
```
┌──(srinivas㉿kali)-[~/CYS]
└─$ rm duplicate
```

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



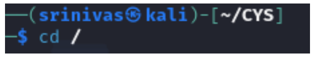
```
┌──(srinivas㉿kali)-[~/CYS]
└─$ cp commands.txt ../LS1/unit4.txt
┌──(srinivas㉿kali)-[~/CYS]
└─$ cp commands.txt ../LS2/unit5.txt
```

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



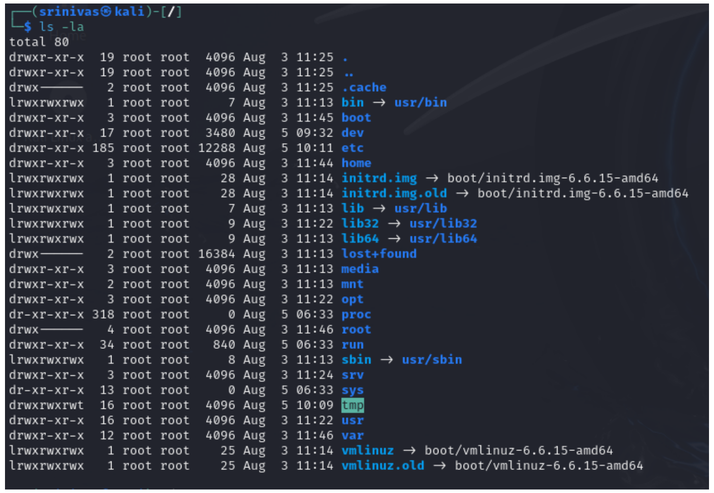
```
┌──(srinivas㉿kali)-[~/CYS]
└─$ cp commands.txt ../LS2/unit5.txt
```

ix. Navigate to root



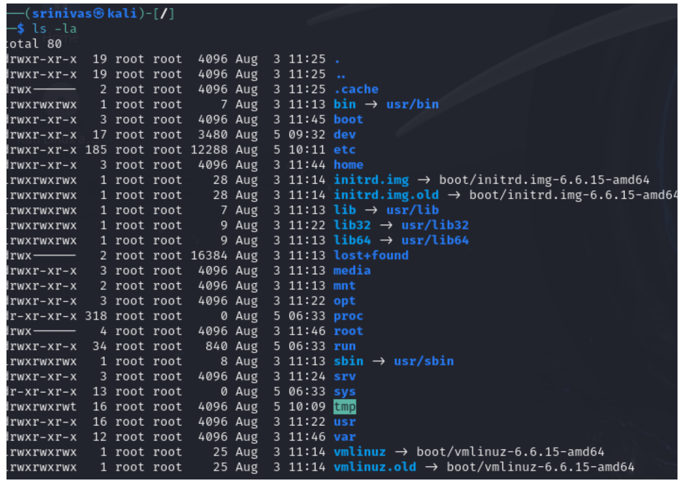
```
──(srinivas㉿kali)-[~/CYS]
─$ cd /
```

x. List all the files under root



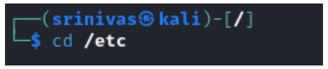
```
┌──(srinivas㉿kali)-[/]
└─$ ls -la
total 80
drwxr-xr-x  19 root root  4096 Aug  3 11:25 .
drwxr-xr-x  19 root root  4096 Aug  3 11:25 ..
drwx------   2 root root  4096 Aug  3 11:25 .cache
lrwxrwxrwx   1 root root     7 Aug  3 11:13 bin → usr/bin
drwxr-xr-x   3 root root  4096 Aug  3 11:45 boot
drwxr-xr-x  17 root root  3480 Aug  5 09:32 dev
drwxr-xr-x 185 root root 12288 Aug  5 10:11 etc
drwxr-xr-x   3 root root  4096 Aug  3 11:44 home
lrwxrwxrwx   1 root root    28 Aug  3 11:14 initrd.img → boot/initrd.img-6.6.15-amd64
lrwxrwxrwx   1 root root    28 Aug  3 11:14 initrd.img.old → boot/initrd.img-6.6.15-amd64
lrwxrwxrwx   1 root root     7 Aug  3 11:13 lib → usr/lib
lrwxrwxrwx   1 root root     9 Aug  3 11:22 lib32 → usr/lib32
lrwxrwxrwx   1 root root     9 Aug  3 11:13 lib64 → usr/lib64
drwx------   2 root root 16384 Aug  3 11:13 lost+found
drwxr-xr-x   3 root root  4096 Aug  3 11:13 media
drwxr-xr-x   2 root root  4096 Aug  3 11:13 mnt
drwxr-xr-x   3 root root  4096 Aug  3 11:22 opt
dr-xr-xr-x 318 root root     0 Aug  5 06:33 proc
drwx------   4 root root  4096 Aug  3 11:46 root
drwxr-xr-x  34 root root   840 Aug  5 06:33 run
lrwxrwxrwx   1 root root     8 Aug  3 11:13 sbin → usr/sbin
drwxr-xr-x   3 root root  4096 Aug  3 11:24 srv
dr-xr-xr-x  13 root root     0 Aug  5 06:33 sys
drwxrwxrwt  16 root root  4096 Aug  5 10:09 tmp
drwxr-xr-x  16 root root  4096 Aug  3 11:22 usr
drwxr-xr-x  12 root root  4096 Aug  3 11:46 var
lrwxrwxrwx   1 root root    25 Aug  3 11:14 vmlinuz → boot/vmlinuz-6.6.15-amd64
lrwxrwxrwx   1 root root    25 Aug  3 11:14 vmlinuz.old → boot/vmlinuz-6.6.15-amd64
```

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



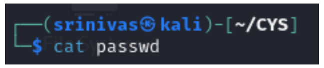
```
──(srinivas㉿kali)-[/]
─$ ls -la
otal 80
rwxr-xr-x  19 root root  4096 Aug  3 11:25 .
rwxr-xr-x  19 root root  4096 Aug  3 11:25 ..
rwx------   2 root root  4096 Aug  3 11:25 .cache
rwxrwxrwx   1 root root     7 Aug  3 11:13 bin → usr/bin
rwxr-xr-x   3 root root  4096 Aug  3 11:45 boot
rwxr-xr-x  17 root root  3480 Aug  5 09:32 dev
rwxr-xr-x 185 root root 12288 Aug  5 10:11 etc
rwxr-xr-x   3 root root  4096 Aug  3 11:44 home
rwxrwxrwx   1 root root    28 Aug  3 11:14 initrd.img → boot/initrd.img-6.6.15-amd64
rwxrwxrwx   1 root root    28 Aug  3 11:14 initrd.img.old → boot/initrd.img-6.6.15-amd64
rwxrwxrwx   1 root root     7 Aug  3 11:13 lib → usr/lib
rwxrwxrwx   1 root root     9 Aug  3 11:22 lib32 → usr/lib32
rwxrwxrwx   1 root root     9 Aug  3 11:13 lib64 → usr/lib64
rwx------   2 root root 16384 Aug  3 11:13 lost+found
rwxr-xr-x   3 root root  4096 Aug  3 11:13 media
rwxr-xr-x   2 root root  4096 Aug  3 11:13 mnt
rwxr-xr-x   3 root root  4096 Aug  3 11:22 opt
r-xr-xr-x 318 root root     0 Aug  5 06:33 proc
rwx------   4 root root  4096 Aug  3 11:46 root
rwxr-xr-x  34 root root   840 Aug  5 06:33 run
rwxrwxrwx   1 root root     8 Aug  3 11:13 sbin → usr/sbin
rwxr-xr-x   3 root root  4096 Aug  3 11:24 srv
r-xr-xr-x  13 root root     0 Aug  5 06:33 sys
rwxrwxrwt  16 root root  4096 Aug  5 10:09 tmp
rwxr-xr-x  16 root root  4096 Aug  3 11:22 usr
rwxr-xr-x  12 root root  4096 Aug  3 11:46 var
rwxrwxrwx   1 root root    25 Aug  3 11:14 vmlinuz → boot/vmlinuz-6.6.15-amd64
rwxrwxrwx   1 root root    25 Aug  3 11:14 vmlinuz.old → boot/vmlinuz-6.6.15-amd64
```

xii. Navigate to /etc/passwd



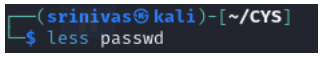
```
┌──(srinivas㉿kali)-[/]
└─$ cd /etc
```

xiii. Open the file passwd



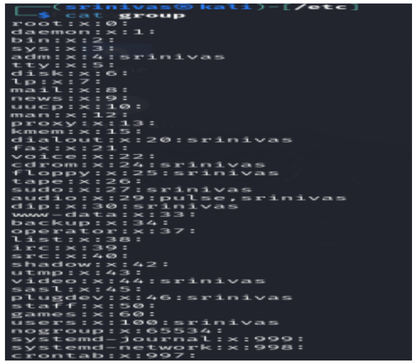
```
┌──(srinivas㉿kali)-[~/CYS]
└─$ cat passwd
```

xiv. Explore the file passwd



```
┌──(srinivas㉿kali)-[~/CYS]
└─$ less passwd
```

xv. Navigate to /etc/group and explore



```
┌──(srinivas㉿kali)-[/etc]
└─$ cat group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:srinivas
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:srinivas
fax:x:21:
voice:x:22:
cdrom:x:24:srinivas
floppy:x:25:srinivas
tape:x:26:
sudo:x:27:srinivas
audio:x:29:pulse,srinivas
dip:x:30:srinivas
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
shadow:x:42:
utmp:x:43:
video:x:44:srinivas
sasl:x:45:
plugdev:x:46:srinivas
staff:x:50:
games:x:60:
users:x:100:srinivas
nogroup:x:65534:
systemd-journal:x:999:
systemd-network:x:998:
crontab:x:997:
```

f. Difference between

i. GUI vs. CLI

Graphical User Interface (GUI)

1. User Interaction:

- **GUI:** Interaction is through graphical elements like windows, icons, buttons, and menus using a mouse or touchscreen.
- **CLI:** Interaction is through text commands typed on a keyboard.

2. Ease of Use:

- **GUI:** Generally easier for beginners as it provides a visual interface that is more intuitive and user-friendly.
- **CLI:** Can be more challenging for beginners due to the need to memorize and type commands, but very powerful for experienced users.

3. Speed and Efficiency:

- **GUI:** May be slower and less efficient for experienced users, especially for repetitive tasks.
- **CLI:** Often faster and more efficient for experienced users, especially for complex and repetitive tasks, as commands can be scripted and automated.

4. Resource Usage:

- **GUI:** Consumes more system resources (RAM and CPU) because it requires graphical processing.
- **CLI:** Consumes fewer system resources, making it ideal for low-power systems or remote server management.

5. Remote Access:

- **GUI:** Requires more bandwidth and may need additional setup (e.g., VNC, Remote Desktop Protocol) to access remotely.
- **CLI:** Can be accessed easily over low-bandwidth connections using tools like SSH.

6. Customization:

- **GUI:** Customization is usually limited to the themes, icons, and window behavior provided by the desktop environment.
- **CLI:** Highly customizable through various shell configurations, scripts, and aliases.

7. Error Handling:

- **GUI:** Provides visual feedback and often more user-friendly error messages.
- **CLI:** Error messages are text-based and may be less intuitive, requiring the user to understand and troubleshoot based on the output.

Command Line Interface (CLI)

1. Flexibility:

- **GUI:** Limited to the functionalities provided by the graphical applications.

- **CLI:** Highly flexible; users can combine commands and create scripts to perform a wide range of tasks.

2. Learning Curve:

- **GUI:** Easier learning curve due to visual interaction.
- **CLI:** Steeper learning curve as it requires learning and remembering various commands and their syntax.

3. Task Complexity:

- **GUI:** Suitable for simple to moderately complex tasks; complex tasks may be cumbersome.
- **CLI:** Suitable for both simple and highly complex tasks; can handle complex tasks more efficiently.

4. Automation:

- **GUI:** Limited automation capabilities; some automation is possible with tools like macros.
- **CLI:** Excellent for automation; scripts and cron jobs can automate virtually any task.

5. Multi-tasking:

- **GUI:** Supports multitasking with multiple windows and tabs.
- **CLI:** Supports multitasking through terminal multiplexers like `tmux` or `screen`, and background processes.

Examples

GUI Example:

- Using a file manager (like Nautilus or Dolphin) to copy files by dragging and dropping.

ii. **man vs info**

man (Manual Pages):

- **Purpose:** Provides concise reference documentation for commands, system calls, library functions, file formats, games, and more.
- **Structure:** Each man page is a single, self-contained document, often divided into sections like NAME, SYNOPSIS, DESCRIPTION, OPTIONS, EXAMPLES, and SEE ALSO.
- **Usage:** Primarily used for quick look-up of command syntax, options, and basic descriptions.

info (Info Pages):

- **Purpose:** Provides more comprehensive and detailed documentation, often with more explanation, background information, and examples.
- **Structure:** Organized as a hypertext system with nodes, allowing for navigation through different sections and chapters using links.
- **Usage:** Used for more in-depth study and understanding of a topic, often including tutorials and extensive examples.

iii. which vs. whereis

The which command is used to locate the executable file associated with the given command by searching the directories listed in the PATH environment variable.

Key Points:

- It only searches for executable files.
- It looks through the directories specified in the PATH environment variable.
- It returns the path of the first match it finds.
- Useful for determining the actual location of the executable that will be run when a command is entered.

The whereis command is used to locate the binary, source, and manual page files for a command.

iv. Terminal vs shell

A terminal is an interface that allows users to interact with the system by entering text-based commands. It emulates a traditional text terminal within a graphical environment.

Key Points:

- **Hardware/Software Interface:** Originally, terminals were hardware devices, but now they are usually software applications that provide a text input/output interface.
- **Examples:** GNOME Terminal, Konsole, xterm, Terminal (macOS).
- **Function:** The terminal serves as a container for the shell. It captures user input (commands) and displays output from the shell and other command-line programs.
- **Graphical Environment:** In a graphical desktop environment, the terminal is a window that provides access to the command line.
- **Features:** Terminals can support multiple tabs, color schemes, font customization, and other graphical features.

Example: Opening a terminal in GNOME might involve searching for "Terminal" in the applications menu and launching it. This opens a window where you can type commands.

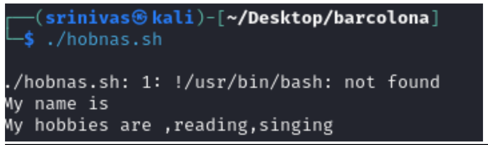
A shell is a command-line interpreter that provides a user interface for accessing the services of the operating system. It interprets and executes the commands entered by the user.

Key Points:

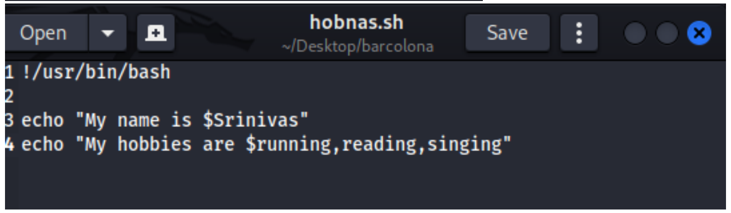
- **Command-Line Interpreter:** The shell processes and executes user commands, scripts, and programs.
- **Examples:** Bash (Bourne Again SHell), Zsh (Z Shell), Ksh (Korn Shell), Fish (Friendly Interactive SHell).
- **Function:** The shell provides features like scripting, command history, job control, aliases, and environment variables.
- **Text-Based Interface:** The shell operates entirely in a text-based interface, taking input from the user and providing text output.
- **Scripting:** Shells allow users to write and execute shell scripts for automating tasks.

Example: When you open a terminal and see a prompt like $, you are interacting with a shell (e.g., Bash). Typing echo "Hello, World!" and pressing Enter will execute the command in the shell.

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



```
┌──(srinivas㉿kali)-[~/Desktop/barcolona]
└─$ ./hobnas.sh

./hobnas.sh: 1: !/usr/bin/bash: not found
My name is
My hobbies are ,reading,singing
```



```
!/usr/bin/bash

echo "My name is $Srinivas"
echo "My hobbies are $running,reading,singing"
```

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