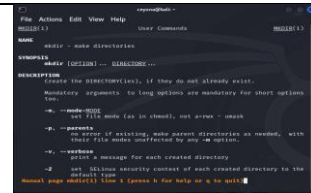
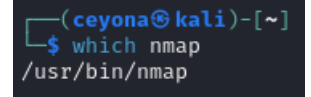
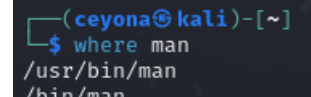
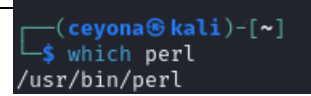
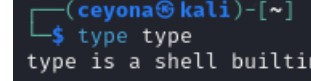


## Topic Name:

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

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

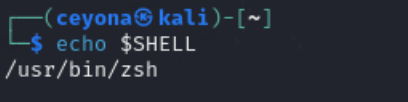
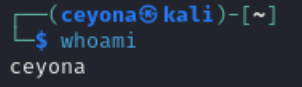
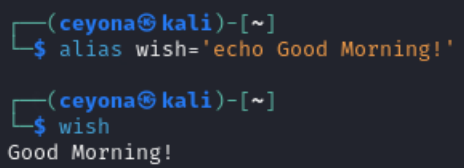
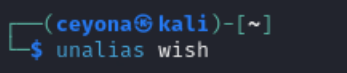
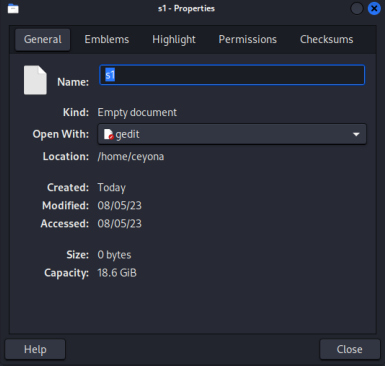
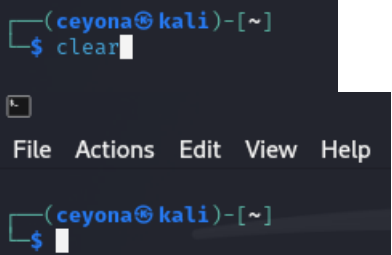
### 1. Getting Help


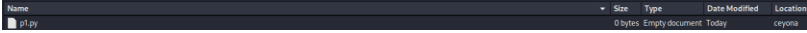
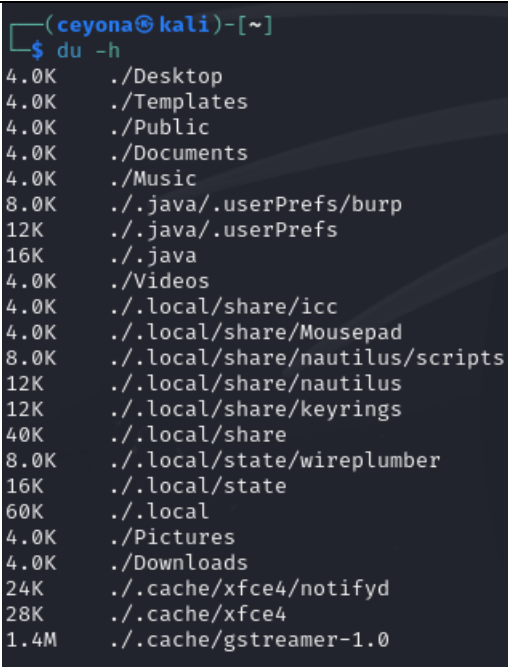
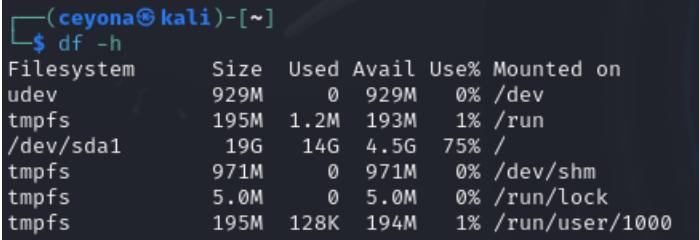
Task	Command Name	Syntax	Example	Screenshots
To get manual page for the known command	man	man command-name	man mkdir	
To get manual page for the unknown command	which	Which command-name	Which nmap	
To know the source file binary	where	where command-name	where man	
To know the path of the command	which	Which command-name	which perl	
To know the command is external or internal	type	type command-name	type type	

To get help for the internal command	whence	Whence command-name	Whence	<pre>(ceyona@kali)-[~] \$ whence cd cd</pre>
To list out bash commands	help	Bash --help	Bash --help	<pre>(ceyona@kali)-[~] \$ 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-ps-strings   --dump-strings   --help   --init-file   --login   --noediting   --noprofile   --norc   --posix   --pretty-print   --rcfile   --restricted   --verbose   --version  Shell options:   -l[ist] or -c command or -O shopt_option      (invocation only)   -sh[ell]kshoptuvxv8CHPT 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: &lt;http://www.gnu.org/software/bash&gt; General help using GNU software: &lt;http://www.gnu.org/gethelp/&gt;</pre>
To know the usage of the command	apropos	apropos command-name	Apropos calculator	<pre>(ceyona@kali)-[~] \$ apropos calculator mate-calc (1) - (mate-calculator) - The MATE Desktop Environment Calcu mate-calc-cmd (1) - A console calculator for the MATE Desktop Environment. mate-calculator (1) - (mate-calculator) - The MATE Desktop Environment Calcu mcalc (1) - scientific calculator for X</pre>

## 2. Basic Commands

Task	Command Name	Syntax	Example	Screenshots
To know today's date	date	date	date	<pre>(ceyona@kali)-[~] \$ date Tue Aug 6 20:53:38 IST 2024</pre>
To print calendar	cal	cal	cal	<pre>(ceyona@kali)-[~] \$ cal       August 2024 Su Mo Tu We Th Fr Sa                 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</pre>
To print kernel version	cat /proc/version	cat /proc/version	cat /proc/version	<pre>(ceyona@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 for Debian) 2.42) #1 SMP PREEMPT_DYNAMIC Kali 6.6.15-2kali1 (2024-05-17)</pre>

To print default shell	echo \$SHELL	echo \$SHELL	echo \$SHELL		
To print currently logged in user	whoami	whoami	whoami		
To create shortcut for command	alias	alias shortcut-name=command-name	alias wish='echo Good Morning!'		
To delete shortcut	unalias	unalias shortcut-name	unalias wish		
To change the timestamp of the file	touch	touch -t<yearmonthdaytime>	touch -t 202308281840		
To clear the screen	clear	clear	clear		

To create empty files	touch	touch.filename	touch p1.py	 
To know disk usage	du	du[options][path]	du -h	
To know free space in the system	df	df[options]	df -h	

To know about the Linux release	lsb_release	lsb_release -a	lsb_release -a	<pre>(ceyona@kali)-[~] \$ lsb_release -a No LSB modules are available. Distributor ID: Kali Description:    Kali GNU/Linux Rolling Release:        2024.2 Codename:       kali-rolling</pre>	
---------------------------------	-------------	----------------	----------------	--	--

### 3. Navigation

Task	Command	Syntax	Screenshots
To navigate home directory	cd	cd	<pre>(ceyona@kali)-[~] \$ cd</pre>
To navigate to the parent directory	cd ..	cd ..	<pre>(ceyona@kali)-[~/Documents] \$ cd ..  (ceyona@kali)-[~] \$</pre>
To navigate to the child directory	cd <directory name>	cd <directory name>	<pre>(ceyona@kali)-[~] \$ cd Documents  (ceyona@kali)-[~/Documents] \$</pre>
Alternate command to cd	pushd	pushd<directory name>	<pre>(ceyona@kali)-[~] \$ pushd Desktop ~/Desktop ~</pre>
To go back to the previous directory	cd -	cd -	<pre>(ceyona@kali)-[~/Desktop] \$ cd - ~  (ceyona@kali)-[~] \$</pre>
To go to the root directory	cd /	cd /	<pre>(ceyona@kali)-[~/Desktop] \$ cd /  (ceyona@kali)-[/] \$</pre>

#### 4. File System

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

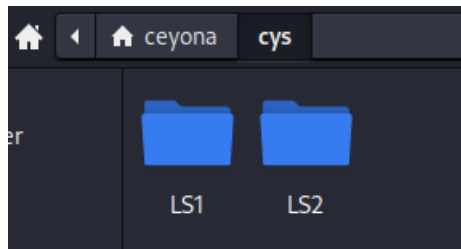
- a. Create Folder "CYS"

```
(ceyona@kali)-[~]  
$ mkdir cys  
  
(ceyona@kali)-[~]  
$ cd cys  
  
(ceyona@kali)-[~/cys]  
$
```

- b. Navigate to CYS

```
(ceyona@kali)-[~]  
$ mkdir cys  
  
(ceyona@kali)-[~]  
$ cd cys  
  
(ceyona@kali)-[~/cys]  
$
```

- c. Create folder LS1 and LS2 under CYS

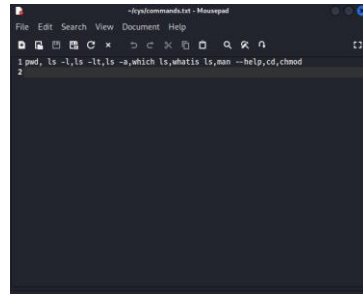


- d. Go back to CYS

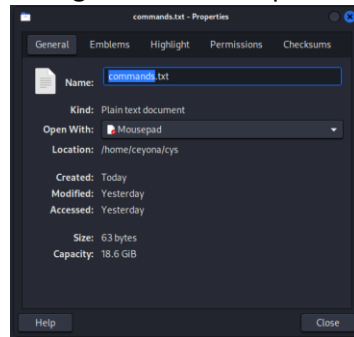
```
(ceyona@kali)-[~/cys]  
$
```

e. Working with Files

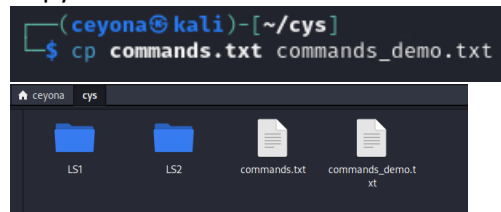
- i. Add commands which you learnt during lab session in the file commands.txt



- ii. Change the timestamp of the file to yesterday

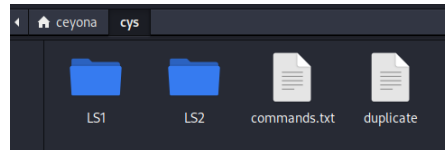


- iii. Copy the contents from the file commands.txt to commands\_demo.txt



- iv. Rename the file commands\_demo.txt to duplicate

```
(ceyona@kali)-[~/cys]  
$ mv commands_demo.txt duplicate
```



- v. Delete the file duplicate

```
(ceyona@kali)-[~/cys]
$ rm duplicate
(ceyona@kali)-[~/cys]
$
```

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

```
(ceyona@kali)-[~/cys]
$ cp commands.txt ../LS2/unit4.txt
(ceyona@kali)-[~/cys]
$ cp commands.txt ../LS2/unit5.txt
```

- vii. Delete the contents from unit5 (using absolute path)

```
(ceyona@kali)-[~/cys]
$ cp commands.txt ../LS2/unit5.txt
```

- viii. Navigate to root

```
(ceyona@kali)-[~/cys]
$ cd /
(ceyona@kali)-[/]
$
```

- ix. List all the files under root

```
(ceyona@kali)-[/]
$ ls -la
total 60
drwxr-xr-x 19 root root 4096 Aug  5 20:48 .
drwxr-xr-x 19 root root 4096 Aug  5 20:48 ..
drwxr-xr-x  2 root root 4096 Aug  5 20:48 cache
lrwxrwxrwx  1 root root    7 Aug  5 20:34 bin -> usr/bin
drwxr-xr-x  3 root root 4096 Aug  5 20:39 boot
drwxr-xr-x 17 root root 3488 Aug  7 20:20 dev
drwxr-xr-x 185 root root 12288 Aug  7 21:11 etc
drwxr-xr-x  3 root root 4096 Aug  5 20:58 home
lrwxrwxrwx  1 root root    7 Aug  5 20:35 initrd.img -> boot/initrd.img-6.6
-rw-r--r--  1 root root    7 Aug  5 20:35 initrd.img.old -> boot/initrd.img-6.6
lrwxrwxrwx  1 root root    7 Aug  5 20:34 lib -> usr/lib
lrwxrwxrwx  1 root root    7 Aug  5 20:34 lib64 -> usr/lib64
lrwxrwxrwx  1 root root    9 Aug  5 20:45 lib32 -> usr/lib32
drwxr-xr-x  2 root root 16384 Aug  5 20:34 lost+found
drwxr-xr-x  3 root root 4096 Aug  5 20:34 media
drwxr-xr-x  2 root root 4096 Aug  5 20:34 mnt
drwxr-xr-x  3 root root 4096 Aug  5 20:45 opt
drwxr-xr-x 242 root root    0 Aug  7 19:57 proc
drwxr-xr-x  4 root root 4096 Aug  5 21:02 root
drwxr-xr-x 34 root root 8448 Aug  7 20:10 run
lrwxrwxrwx  1 root root    6 Aug  5 20:34 sbin -> usr/sbin
drwxr-xr-x  3 root root 4096 Aug  5 20:47 srv
drwxr-xr-x 13 root root    0 Aug  7 19:57 sys
drwxrwxrwt 16 root root 4096 Aug  7 21:12 tmp
drwxr-xr-x 16 root root 4096 Aug  5 20:45 usr
drwxr-xr-x 12 root root 4096 Aug  5 21:02 var
lrwxrwxrwx  1 root root    7 Aug  5 20:35 vmlinuz -> boot/vmlinuz-6.6.15-amd64
lrwxrwxrwx  1 root root    7 Aug  5 20:35 vmlinuz.old -> boot/vmlinuz-6.6.15-amd64
```



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

```
(ceyona@kali: /)
ls -la
total 60
drwxr-xr-x 19 root root 4096 Aug  5 20:48 .
drwxr-xr-x 19 root root 4096 Aug  5 20:48 ..
drwxr-xr-x  2 root root 4096 Aug  5 20:48 -lacho
lrwxrwxrwx  1 root root    7 Aug  5 20:34 bin -> usr/bin
drwxr-xr-x  2 root root 4096 Aug  5 20:59 boot
drwxr-xr-x 17 root root 3680 Aug  7 20:20 dev
drwxr-xr-x 185 root root 12288 Aug  7 21:11 etc
drwxr-xr-x  3 root root 4096 Aug  5 20:58 home
lrwxrwxrwx  1 root root    7 Aug  5 20:35 initrd.img -> boot/initrd.img-6.6
-rw-r--r--  1 root root    7 Aug  5 20:35 initrd.img.eld -> boot/initrd.img
-rw-r--r--  1 root root    7 Aug  5 20:35 initrd.img.eld -> boot/initrd.img
-rw-r--r--  1 root root    7 Aug  5 20:34 lib -> usr/lib
lrwxrwxrwx  1 root root    7 Aug  5 20:34 lib -> usr/lib
lrwxrwxrwx  1 root root    9 Aug  5 20:45 lib32 -> usr/lib32
lrwxrwxrwx  1 root root    9 Aug  5 20:34 lib64 -> usr/lib64
drwxr-xr-x  2 root root 16384 Aug  5 20:36 lostfound
drwxr-xr-x  2 root root 4096 Aug  5 20:34 media
drwxr-xr-x  2 root root 4096 Aug  5 20:24 mnt
drwxr-xr-x  3 root root 4096 Aug  5 20:45 opt
drwxr-xr-x 242 root root    0 Aug  7 10:57 proc
drwxr-xr-x  4 root root 4096 Aug  5 21:02 root
drwxr-xr-x 34 root root 8440 Aug  7 20:10 run
lrwxrwxrwx  1 root root    8 Aug  5 20:36 sbin -> usr/sbin
drwxr-xr-x  3 root root 4096 Aug  5 20:47 srv
drwxr-xr-x 13 root root    0 Aug  7 10:57 sys
drwxrwxrwx 16 root root 4096 Aug  7 21:11 tmp
drwxr-xr-x 16 root root 4096 Aug  5 20:45 usr
drwxr-xr-x 32 root root 4096 Aug  5 21:02 var
lrwxrwxrwx  1 root root    7 Aug  5 20:35 vmlinuz -> boot/vmlinuz-6.6.15-amd64
lrwxrwxrwx  1 root root    7 Aug  5 20:35 vmlinuz.eld -> boot/vmlinuz-6.6.15-amd64
```

- xi. Navigate to /etc/passwd

```
(ceyona@kali)-[/]
$ cd /etc
```

- xii. Open the file passwd

```
(ceyona@kali)-[/etc]
$ cat passwd
```

- xiii. Explore the file passwd

```
(ceyona@kali)-[~/cys]
$ less passwd
```

- xiv. Navigate to /etc/group and explore

```

(ceyona@kali)-[/etc]
$ cat group
root:x:0:/:usr/lib/tpm/bin/false
daemon:x:1:/:usr/sbin/nologin
bin:x:2:/:usr/sbin/nologin
sys:x:3:/:usr/sbin/nologin
adm:x:4:ceyona:/:usr/sbin/nologin
tty:x:5:/:usr/sbin/nologin
disk:x:6:/:usr/sbin/nologin
lp:x:7:/:usr/sbin/nologin
mail:x:8:/:usr/sbin/nologin
news:x:9:/:usr/sbin/nologin
uucp:x:10:/:usr/sbin/nologin
man:x:12:/:usr/sbin/nologin
proxy:x:13:/:usr/sbin/nologin
kmem:x:15:/:usr/sbin/nologin
dialout:x:20:ceyona:/:usr/sbin/nologin
fax:x:21:/:usr/sbin/nologin
voice:x:22:/:usr/sbin/nologin
cdrom:x:24:ceyona:/:usr/sbin/nologin
floppy:x:25:ceyona:/:usr/sbin/nologin

```

## f. Difference between

### i. GUI vs. CLI

GUI (Graphical User Interface):

- **Visual Interaction:** GUI relies on visual elements like windows, icons, buttons, and menus. Users interact with these elements using a mouse, touchpad, or touchscreen.
- **User-Friendly:** Generally considered more intuitive and easier for beginners because it provides a visual representation of options and tasks.
- **Learning Curve:** Easier to learn for most people due to its visual nature. Users can often discover functions by exploring the interface.
- **Multitasking:** Typically allows users to multitask more effectively with multiple windows and applications visible simultaneously.
- **Performance:** May use more system resources (like memory and processing power) due to the graphical elements.
- **Examples:** Windows operating system, macOS, most modern web applications, and graphical desktop applications.

CLI (Command-Line Interface):

- **Text-Based Interaction:** CLI relies on text commands entered via a keyboard. Users type specific commands to perform tasks or execute programs.
- **Efficient for Experts:** Preferred by more experienced users and system administrators for its speed and precision.
- **Learning Curve:** Can be more difficult to learn due to the need to memorize commands and syntax.
- **Scripting and Automation:** Highly effective for scripting and automating repetitive tasks. Users can write scripts to perform complex sequences of actions.
- **Performance:** Typically uses fewer system resources as it doesn't require graphical rendering.
- **Examples:** Terminal in Unix/Linux, Command Prompt and PowerShell in Windows, and the shell interfaces in various operating systems.

## ii. man vs info

man (Manual Pages):

- **Purpose:** Provides access to the traditional Unix manual pages, which contain documentation about commands, system calls, library functions, and other aspects of the system.
- The documentation is divided into sections, with each section containing information about a specific type of content (e.g., commands, library functions). The layout is text-based and often includes sections like NAME, SYNOPSIS, DESCRIPTION, OPTIONS, and SEE ALSO.
- **Example:** To read the manual page for the `ls` command, you would use `man ls`.

Info:

- **Purpose:** Provides access to the GNU info system, which offers a more extensive and hierarchical documentation format, often used for GNU software and projects.
- **Format:** The documentation is organized in a tree-like structure with nodes and hyperlinks, allowing for more detailed and structured information. It supports rich formatting, such as tables of contents, cross-references, and embedded hyperlinks.
- **Example:** To read the info documentation for the `ls` command, you would use `info coreutils 'ls invocation'` (assuming you are looking for the GNU coreutils version of `ls`).

### iii. which vs. whereis

Which:

- The `which` command is used to locate the executable file associated with a command by searching through directories listed in the `PATH` environment variable.
- It provides the full path to the executable file that will be run when the command is executed.
- The output is limited to the location of the command's executable and does not include other related files or documentation.
- If the command is not found in the `PATH`, `which` returns no result or an indication that the command is not available.

Whereis:

- The `whereis` command searches for the binary, source code, and manual page files related to a command in standard directories.
- It provides paths to multiple types of files, including the executable, source code, and documentation, if available.
- The search scope of `whereis` extends beyond the `PATH` environment variable to include system-wide directories and locations.
- The output is more comprehensive than `which`, showing various related files associated with the command, not just the executable.

### iv. Terminal vs shell

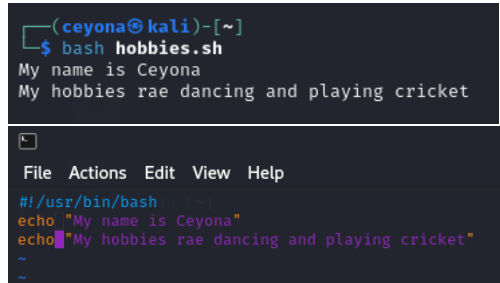
Terminal:

- A terminal is a software application or physical device that provides a text-based interface for user interaction with the operating system.
- It acts as a window or environment where users can input commands and view the output from the system.
- Terminal emulators like `gnome-terminal`, `xterm`, and `Terminal.app` facilitate access to the command-line interface in graphical environments.
- Essentially, the terminal is the medium through which users interact with the shell and the broader command-line environment.

Shell:

- A shell is a command-line interpreter that processes and executes user commands typed into a terminal.
- It provides features like scripting, variable management, and control flow to automate and manage tasks.
- Common shells include `bash`, `zsh`, and `fish`, each offering different features and functionalities.
- The shell operates within the terminal, interpreting commands and returning results to the user through the terminal interface.

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



```
(ceyona@kali)-[~]  
$ bash hobbies.sh  
My name is Ceyona  
My hobbies rae dancing and playing cricket  
  
File Actions Edit View Help  
#!/usr/bin/bash  
echo "My name is Ceyona"  
echo "My hobbies rae dancing and playing cricket"  
~  
~
```

## 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