**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 | Cmd name-man | man(command name) | man ls |  |
| To get manual page for the unknown command | Cmd name -which | which(command name) | ss -tuln |  |
| To know the source file binary | Cmd name-where | where (command name) | which nmap |  |
| To know the path of the command | Cmd -which | which (command name) | which ls |  |
| To know the command is external or internal | Cmd - type | type(command name) | type chmod |  |
| To get help for the internal command | Cmd -whence | whence (command name) | whence ls |  |
| To list out bash commands | Cmd -help | bash –help(command name) | bash --help |  |
| To know the usage of the command | Cmd - apropos | man (command name) | man chmod |  |

1. **Basic Commands**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Task | Command Name | Syntax | Example | Screenshots |
| To know today’s date | date | date | date |  |
| To print calendar | cal | ncal | ncal 2022 |  |
| To print kernel version | cat /proc/version | cat /proc/version | cat /proc/version |  |
| 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 -cmd | alias greet=’echo hello, world!’ |  |
| To delete shortcut | unalias | unalias | Unalias |  |
| To change the timestamp of the file | touch | touch -t | touch –t 202406052223 s1 |  |
| To clear the screen | clear | clear | clear |  |
| To create empty files | touch | touch .filename | touch m1.txt |  |
| To know disk usage | du | du | du -D |  |
| To know free space in the system | df | df | df -a |  |
| To know about the Linux release | lsb\_release | lsb\_release -a | lsb\_release -a |  |

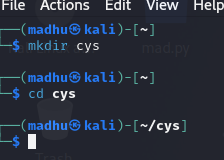
1. **Navigation**

|  |  |  |  |
| --- | --- | --- | --- |
| Task | Command | Syntax | Screenshots |
| To navigate home directory | cd | cd |  |
| To navigate to the parent directory | cd .. | cd .. |  |
| To navigate to the child directory | cd(directory name) | cd(directory name) |  |
| Alternate command to cd | pushd | pushd(directory name) |  |
| To go back to the previous directory | cd - | cd - |  |
| To go to the root directory | cd / | cd / |  |

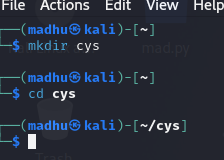
1. **File System**

|  |  |  |
| --- | --- | --- |
| **Task** | **Syntax** | **Command** |
| **How to identify the file system** | **df -T** | **df -T** |

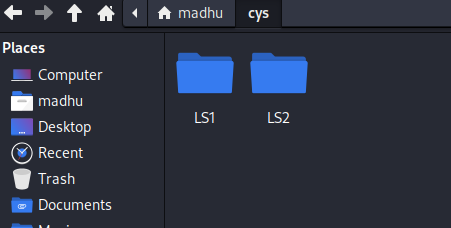
1. Create Folder “CYS”



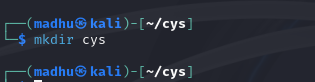
1. Navigate to CYS



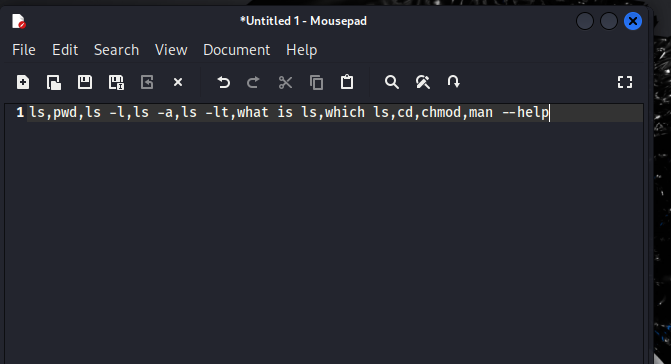
1. Create folder LS1 and LS2 under CYS



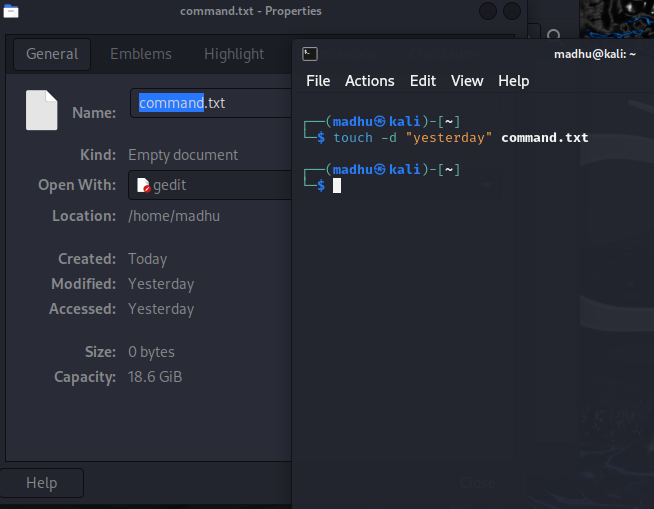
1. Go back to CYS



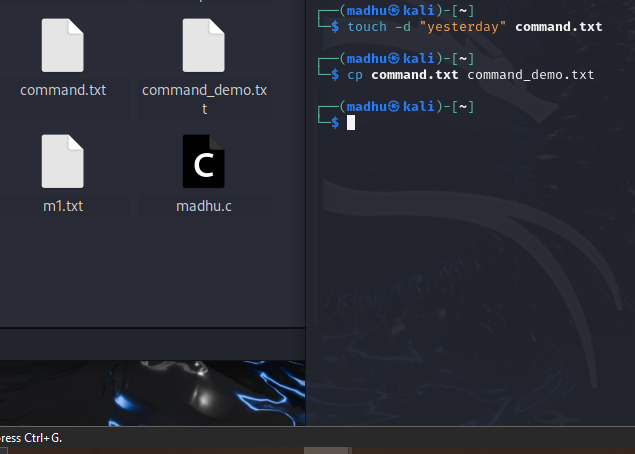
1. Working with Files
2. Add commands which you learnt during lab session in the file commands.txt



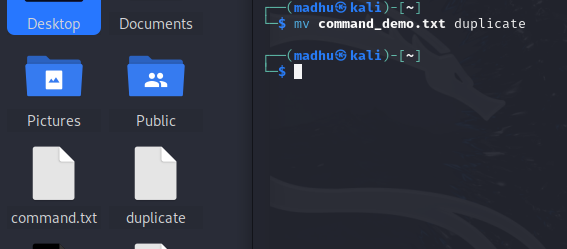
1. Change the timestamp of the file to yesterday



1. Copy the contents from the file commands.txt to commands\_demo.txt



1. Rename the file commands\_demo.txt to duplicate



1. Rename all .html to .hldd

1. Delete the file duplicate



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





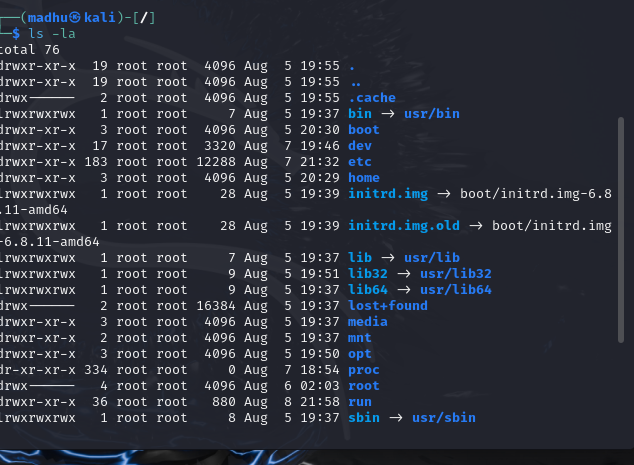
1. Delete the contents from unit5 (using absolute path)



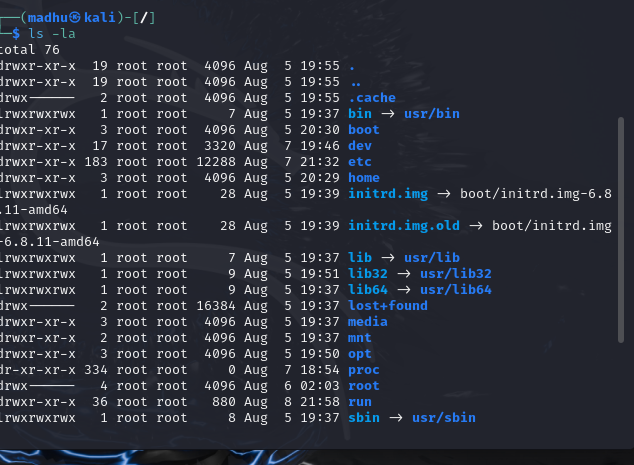
1. Navigate to root



1. List all the files under root



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



1. Navigate to /etc/passwd



1. Open the file passwd



1. Explore the file passwd



1. Navigate to /etc/group and explore

1. **Difference between** 
   * + 1. GUI vs. CLI

### Graphical User Interface (GUI):

* **Visual Interaction:** GUIs provide a visual way to interact with the computer. Users interact through graphical elements like windows, icons, buttons, and menus.
* **Ease of Use:** Typically more intuitive and easier for beginners because it visually represents tasks and processes. You can click on icons and navigate through menus rather than remembering commands.
* **Complexity Management:** Helps manage complexity by abstracting detailed commands and options behind graphical controls.
* **Learning Curve:** Generally, there's less of a learning curve because users don't need to memorize commands.
* **Performance:** GUIs can be more resource-intensive, as they require more system resources to render graphical elements.

**Command-Line Interface (CLI):**

* **Text-Based Interaction:** CLIs use text commands to perform tasks. Users type commands into a terminal or command prompt to execute operations.
* **Efficiency:** Often faster and more efficient for experienced users because it can perform complex tasks with concise commands. Automation and scripting are also easier with CLI.
* **Learning Curve:** Has a steeper learning curve as users need to memorize and understand various commands and their options.
* **Resource Usage:** Generally requires fewer system resources because it doesn’t need to render graphical elements.

* + - 1. man vs info

**man Command:**

* **Purpose:** man stands for "manual." It is used to display the manual pages for commands, system calls, library functions, and other topics. It provides detailed documentation on how to use various commands and programs.
* **Usage:** To get help on a command or topic, you use man followed by the command or topic name.
* **Content:** Manual pages (man pages) are organized into sections, including name, synopsis, description, options, and examples. They provide comprehensive information about the command's usage and options.

**who Command:**

* **Purpose:** The who command is used to display information about users currently logged into the system. It provides details such as the usernames, terminal lines, login times, and originating IP addresses or hostnames.
* **Usage:** Simply type who at the command prompt
* **Content:** The output typically includes columns for the username, terminal line (TTY), login time, and sometimes the originating IP address or hostname.

**Summary:**

* **man:** Used to view documentation for commands and system features. It helps you understand how to use different commands and utilities.
* **who:** Provides information about users currently logged into the system. It’s useful for system monitoring and administration.

So, man is about getting help and documentation, while who is about system user activity.

* + - 1. which vs. whereis

The which and whereis commands are both used in Unix-like operating systems to locate executables and related files, but they serve slightly different purposes and provide different levels of detail.

**which Command:**

* **Purpose:** The which command is used to find the location of executables that are in your PATH. It helps you determine the path of the executable that will be run when you type a command in the terminal.
* **Usage:** To use which, simply type
* **Output:** It provides a single path to the executable that will be executed when you run the command, based on the current PATH environment variable

**whereis Command:**

* **Purpose:** The whereis command is used to locate the binary, source, and manual page files for a command. It searches a broader set of directories and provides more comprehensive information about the locations of related files.
* **Usage:** To use whereis, type
* **Output:** It provides multiple paths for the command’s binary, source code, and man pages, depending on what it finds.

**Summary:**

* **which:** Locates the path of the executable file that would be executed for a given command, based on the PATH environment variable.
* **whereis:** Locates the binary, source, and manual page files for a given command, providing a broader set of file locations.

.

* + - 1. Terminal vs shell

**Terminal:**

* **Definition:** A terminal is a software application or hardware device that provides a text-based interface to interact with the operating system. It allows users to input commands and see output from those commands.
* **Types:** There are both physical and virtual terminals. Physical terminals were hardware devices used in earlier computing eras, while virtual terminals are software-based windows or applications in modern systems (e.g., GNOME Terminal, xterm, iTerm2).
* **Function:** The terminal emulates the physical terminals of the past, enabling users to interact with the system by providing an interface where they can type commands and view results.

**Shell:**

* **Definition:** A shell is a command-line interpreter that provides the user interface for accessing the operating system's services. It processes the commands entered by the user and executes them.
* **Types:** There are various types of shells, including:
  + **Bash (Bourne Again Shell):** The default shell on many Linux distributions.
  + **Zsh (Z Shell):** Known for its powerful features and customization.
  + **Fish (Friendly Interactive Shell):** Known for user-friendly features and modern design.
  + **Sh (Bourne Shell):** One of the earliest Unix shells.
  + **Tcsh:** An enhanced version of the C shell (csh).
* **Function:** The shell interprets the commands, executes them, and returns the output to the terminal. It also provides scripting capabilities, environmental variable management, and other advanced features.

**Summary:**

* **Terminal:** The application or interface that allows you to interact with the operating system using text input and output. It provides the environment where you can run a shell.
* **Shell:** The command-line interpreter that processes and executes the commands you type into the terminal.

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

**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