**Topic Name:**

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

* Explore file structure
* File management commands
* Absolute path and Relative path
* Globbing
* Scripting

File Structure

1. Under the root directory there are many files like

/bin , /boot , /dev , /etc , ….

Find out the importance of those files

Example : /etc is for user account details

|  |  |  |
| --- | --- | --- |
| **S.No** | **Directory** | **Usage** |
| 1 | / | Root directory |
| 2 | /bin | Binary files |
| 3 | /boot | System’s boot process |
| 4 | /dev | Directory in Linux is a special filesystem that provides access to device files |
| 5 | /etc | user account details |
| 6 | /home | Directory that contains standard location where user-specific files and directories are stored |
| 7 | /lib | Directory that contains shared library files that are essential for system and application operations |
| 8 | /proc | Virtual filesystem that provides a mechanism for the kernel to expose information about the system and running processes |
| 9 | /sbin | Contains essential system binaries and executable files that are generally used for system administration and maintenance tasks |
| 10 | /tmp | Temporary storage location used by the operating system and applications to store temporary files |
| 11 | /var | Directory that contains filesystem hierarchy that contains variable data files that are expected to change over time |
| 12 |  |  |
| 13 |  |  |

1. In Linux, there are three different files

Regular file

Directory

Special file

Block file

Character file

Socket file

Pipe file

Fill the below table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| File Type | Represented by (Hint ls) | Role | How to create | How to check | Location | Screen shot |
| Regular file | - | To store data |  | NA |  |  |
| * Text file | - | To store texted data |  | NA |  |  |
| * Compressed file | - | Compressed data will be stored |  | NA |  |  |
| * Image | - | Stores image file | NA | NA |  |  |
| Directory | d | Stores different type of files |  | NA |  |  |
| Block file | b |  | NA | NA |  |  |
| Character file |  |  | NA | NA |  |  |
| Socket file |  |  | NA | NA |  |  |
| pipe file |  |  | NA | NA |  |  |

1. Globbing
2. Go back to CYS
3. Create multiple subdirectories using single command

LS

Unit1

command

glob

Unit2

command

grep

Unit3

Constructs

**mkdir -p unit1/Command unit1/glob unit2/Command unit2/glob unit3/Constructs**

1. Navigate to unit1/glob
2. Create the following files :

Commands.txt

Commands1.txt

Commands2.txt

page1.html

page2.html

page3.html

file1

file10

file11

file2

File2

File3

file33

fileAB

filea

fileA

fileAAA

file(

file 2

* + 1. List all files starting with file
    2. List all files starting with File
    3. List all files starting with file and ending in a number.
    4. List all files starting with file and ending with a letter
    5. List all files starting with File and having a digit as fifth character.
    6. List all files starting with File and having a digit as fifth character and nothing else.
    7. List (with ls) all files starting with a letter and ending in a number.
    8. List (with ls) all files that have exactly five characters.

== Ls ?????? or ls [a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9][a-zA-Z0-9]

* + 1. List (with ls) all files that start with f or F and end with 3 or A.l
    2. List (with ls) all files that start with f have i or R as second character and end in a number.
    3. List all files that do not start with the letter F.
    4. Remove all the \*.html
    5. Rename \*.txt to \*.json

1. Absolute path and relative path

Use rm, mv, cp, ls with absolute path and relative path as per your choice.

=>

rm ~/Shashank/file1.txt

rm file1.txt

mv ~/Shashank/file2.txt file3.txt

cp file3.txt file4.txt

cp file4.txt ~/Shashank/file5.txt

ls -l ~/Shashank

1. Wildcards

|  |  |  |  |
| --- | --- | --- | --- |
| Notation | Use | Example | Screenshot |
| \* | It matches zero or more characters in filenames or directory names | ls file\*.txt |  |
| ? | It matches only one characters in filenames or directory names | ls file?.txt |  |
| [ ] | (Character class) Matches specific sets or ranges of characters | ls file[1-9].txt |  |
| [! ] | Matches any character that is not a member of the set characters | ls file[!a].txt |  |
| { } | Allows to create multiple strings or paths from a single pattern | ls file{1..3}.txt |  |

More on Character class

|  |  |  |  |
| --- | --- | --- | --- |
| Notation | Use | Example | Screenshot |
| [:alnum:] |  |  |  |
| [:alpha:] |  |  |  |
| [:digit:] |  |  |  |
| [:lower:] |  |  |  |
| [:upper:] |  |  |  |

4. change permission

1. Change the permission set of /work/readme.txt so that only the user (owner) can read,write, and execute it. Use absolute mode.

* chmod 700 /work/readme.txt

1. Change the permission set of /work/readme.txt so that any user can read it, the group can read/write to it and the user (owner) can read/write/execute it. Use absolute mode.

* chmod 764 /work/readme.txt

1. Change the permission set of /bin/bash so that only the user (owner) can read/write/ execute, group, and any user can execute it. However, whenever anyone executes it, it should run with the privileges of the owner user. Use absolute mode.

* sudo chmod 4755 /bin/bash

1. Change the permission set of /work/readme.txt so that only the user (owner) can read, write, and execute it. Use relative mode.

* chmod u+rwx /work/readme.txt

1. Change the permission set of /work/readme.txt so that any user can read it, the group can read/write to it and the user (owner) can read/write/execute it. Use relative mode.

* chmod u+rwx,g+rw,o+r /work/readme.txt

1. Change the permission set of /work/readme.txt so that only the user (owner) can read/write/ execute, group, and any user can execute it. However, whenever anyone executes it, it should run with the privileges of the group. Use absolute mode.

* sudo chmod 2755 /work/readme.txt

1. Change the permission set of /work/readme.txt so that only the owner can rename or delete this file while maintaining the existing permissions. Use absolute mode.

* chmod 644 /work/readme.txt

1. What are the default permissions for the new file?

* read and write for the owner, read-only for group and others

1. What was the command to view the file permissions?

* ls- l filename

1. Change chmod.exercises permissions to -r--r--r—

* chmod 644 chmod.exercises

1. Change the file permissions to Read only for the owner, group and all other users.

* chmod 444 filename

1. What was the command for changing the file permissions to -r--r--r--?

* chmod 444 filename

1. Change chmod.exercises permissions to -rw-r-----

* chmod 640 chmod.exercises

1. Change the file permissions to match the following:
   1. owner: Read and Write
   2. group: Read
   3. other: no permissions (None)

* chmod 640 filename

1. What was the command for changing the file permissions to -rw-r-----?

=> chmod 640 chmod.exercises

1. Change chmod.exercises permissions to -rwxr-x—x

* chmod 751 chmod.exercises

1. Change the file permissions to match the following:
   1. owner: Read, Write and Execute
   2. group: Read and Execute
   3. other: Execute

* chmod 751 filename

1. What was the command for changing the file permissions to -rwxr-x--x?

* chmod 751 filename