**AMRUTHESH**

**241059041**

**LINUX OS & SCRIPTING LAB**

**M.E – CYBER SECURITY**

**File Structure**

1. **Root directory (/)**: The top-level directory in the Linux filesystem hierarchy.
2. **Binary files (/bin)**: Contains essential command binaries that are required for booting and repairing the system.
3. **Boot (/boot)**: Contains the Linux kernel and files needed for booting.
4. **Device files (/dev)**: Contains device files that represent hardware devices.
5. **Configuration files (/etc)**: Contains configuration files for the system and applications.
6. **Home directories (/home)**: Contains the home directories of all users.
7. **Libraries (/lib)**: Contains shared library files used by binaries in /bin and /sbin.
8. **Process information (/proc)**: Virtual filesystem providing information about system processes.
9. **System binaries (/sbin)**: Contains essential system binaries used by the system administrator.
10. **Temporary files (/tmp)**: Temporary files created by applications.
11. **Variable data files (/var)**: Contains files that are expected to grow, like logs and mail spools.

**File Types**

| **File Type** | **Represented by (Hint: ls)** | **Role** | **How to create** | **How to check** | **Location** |
| --- | --- | --- | --- | --- | --- |
| Regular file | - | Store data, text, or binary information. | touch file | ls -l | /home |
| Text file | - | Store plain text. | touch file.txt | ls -l | /home |
| Compressed file |  | Store compressed data. | gzip file | file file.gz | /var |
| Image |  | Store image data. | convert | file image | /images |
| Directory | d | Organize files. | mkdir dir | ls -ld | / |
| Block file | b | Represent block devices. | mknod | ls -l | /dev |
| Character file | c | Represent character devices. | mknod | ls -l | /dev |
| Socket file | s | Facilitate communication between processes. | socket | ls -l | /run |
| Pipe file | p | Facilitate inter-process communication. | mkfifo | ls -l | /tmp |

**Globbing**

Commands for globbing operations:

1. **Create subdirectories using a single command:**

mkdir -p Unit1/command/glob Unit2/command/grep Unit3/constructs

1. **Navigate to unit1/glob and create the specified files:**

cd Unit1/glob

touch 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. **Globbing operations:**
   * List all files starting with file:

ls file\*

* + List all files starting with File:

ls File\*

* + List all files starting with file and ending in a number:

ls file\*[0-9]

* + List all files starting with file and ending with a letter:

ls file\*[a-zA-Z]

* + List all files starting with File and having a digit as the fifth character:

ls File???[0-9]\*

* + List all files starting with File and having a digit as the fifth character and nothing else:

ls File???[0-9]

* + List all files starting with a letter and ending in a number:

ls [a-zA-Z]\*[0-9]

* + List all files that have exactly five characters:

ls ?????

* + List all files that start with f or F and end with 3 or A:

ls [fF]\*[3A]

* + List all files that start with f, have i or R as the second character, and end in a number:

ls f[iR]\*[0-9]

* + List all files that do not start with the letter F:

ls [^F]\*

* + Remove all \*.html files:

rm \*.html

* + Rename \*.txt to \*.json:

for file in \*.txt; do mv "$file" "${file%.txt}.json"; done

**Absolute Path and Relative Path**

* **Use rm, mv, cp, ls with absolute path:**

rm /home/user/Unit1/glob/Commands.json

mv /home/user/Unit1/glob/file1 /home/user/Unit1/glob/file01

cp /home/user/Unit1/glob/file10 /home/user/Unit1/glob/file20

ls /home/user/Unit1/glob

* **Use rm, mv, cp, ls with relative path:**

rm ./Commands.json

mv ./file1 ./file01

cp ./file10 ./file20

ls ./

**Wildcards**

* **\*:** Matches any number of characters.
  + Example: ls \* lists all files.
* **?:** Matches a single character.
  + Example: ls ?ile\* matches file with one character prefix.
* **[ ]:** Matches any single character within brackets.
  + Example: ls file[0-9]\* lists files starting with file followed by a number.
* **[! ]:** Matches any character not in the brackets.
  + Example: ls [!F]\* lists files not starting with F.
* **{ }:** Matches a comma-separated list of values.
  + Example: ls {file1,file2} lists file1 and file2.

**More on Character Classes**

* **[:alnum:]:** Matches any alphanumeric character.
* **[:alpha:]:** Matches any alphabetic character.
* **[:digit:]:** Matches any digit.
* **[:lower:]:** Matches any lowercase letter.
* **[:upper:]:** Matches any uppercase letter.

**Change Permissions**

1. **/work/readme.txt - User can read/write/execute (absolute mode):**

chmod 700 /work/readme.txt

1. **/work/readme.txt - Any user can read, group can read/write, user can read/write/execute (absolute mode):**

chmod 764 /work/readme.txt

1. **/bin/ - User can read/write/execute, group and others can execute, runs with owner privileges (absolute mode):**

chmod 4751 /bin/

1. **/work/readme.txt - User can read/write/execute (relative mode):**

chmod u+rwx,go-rwx /work/readme.txt

1. **/work/readme.txt - Any user can read, group can read/write, user can read/write/execute (relative mode):**

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

1. **/work/readme.txt - User can read/write/execute, group and others can execute, runs with group privileges (absolute mode):**

chmod 2755 /work/readme.txt

1. **/work/readme.txt - Only the owner can rename or delete, maintaining existing permissions (absolute mode):**

chmod 700 /work/readme.txt

1. **Default permissions for a new file:**

Typically -rw-r--r-- (umask 022).

1. **Command to view file permissions:**

ls -l

1. **Change chmod.exercises permissions to -r--r--r--:**

chmod 444 chmod.exercises

1. **Change file permissions to Read-only for owner, group, and others:**

chmod 444 filename

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

chmod 640 chmod.exercises

1. **Command for changing file permissions to -rw-r-----:**

chmod 640 filename

1. **Change chmod.exercises permissions to -rwxr-x--x:**

chmod 751 chmod.exercises

1. **Command for changing file permissions to -rwxr-x--x:**

chmod 751 filename