PW SKILLS | DevOps and Cloud Computing

Linux Fundamentals for System Administration

Objective

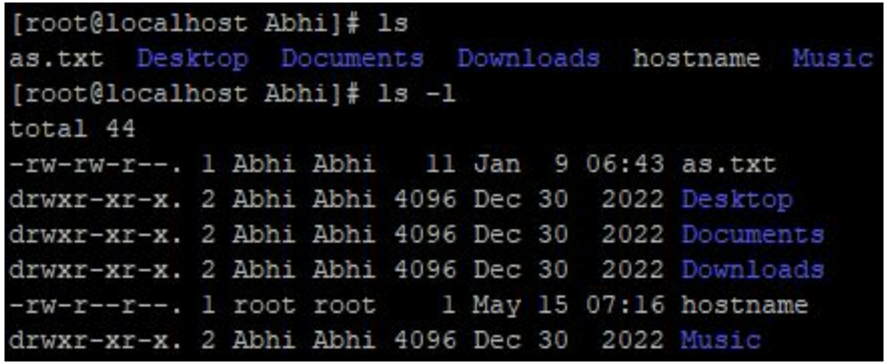
- Navigate the Linux file system and execute basic commands.
- Manage users, groups, and permissions effectively.
- Perform package management and basic system administration tasks.
- Understand and apply file permissions and ownership commands.
- Monitor and manage disk space usage with essential Linu tools.

PW SKILLS

Linux Terminal & Its Importance

Linux Terminal

- The Linux terminal is a crucial component of the Linux operating system, providing a command-line interface (CLI) that allows users to interact with the system through text-based commands.



```
[root@localhost Abhi]# ls
as.txt  Desktop  Documents  Downloads  hostname  Music
[root@localhost Abhi]# ls -l
total 44
-rw-rw-r--. 1 Abhi Abhi   11 Jan  9 06:43 as.txt
drwxr-xr-x. 2 Abhi Abhi 4096 Dec 30  2022 Desktop
drwxr-xr-x. 2 Abhi Abhi 4096 Dec 30  2022 Documents
drwxr-xr-x. 2 Abhi Abhi 4096 Dec 30  2022 Downloads
-rw-r--r--. 1 root root    1 May 15 07:16 hostname
drwxr-xr-x. 2 Abhi Abhi 4096 Dec 30  2022 Music
```

Linux Terminal

Importance of Linux Terminal in system administration:

- Command Execution
- Scripting & Automation
- Remote Management
- Resource Efficiency
- Learning & Control

Pop Quiz

Q. What is the primary function of the Linux terminal?

A

To provide a graphical user interface

B

To execute commands and interact with the operating system

Pop Quiz

Q. What is the primary function of the Linux terminal?

A

To provide a graphical user interface

B

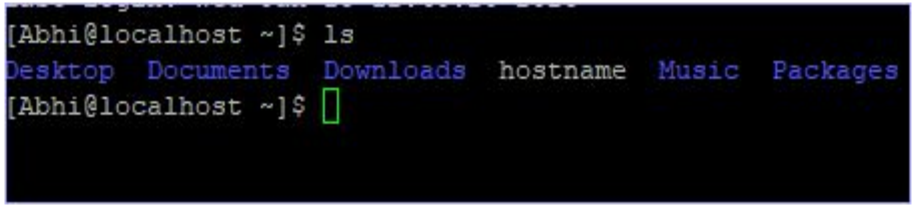
To execute commands and interact with the operating system

Demonstrating Basic Commands

Commands: ls, cd, pwd, mkdir, rm & cp

1. ls - List Directory Contents:

The ls command is used to display the files and directories in the current directory.

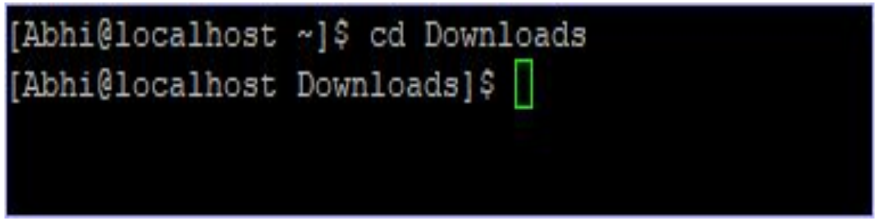


```
[Abhi@localhost ~]$ ls
Desktop  Documents  Downloads  hostname  Music  Packages
[Abhi@localhost ~]$
```

Commands: ls, cd, pwd, mkdir, rm & cp

2. cd - Change Directory:

The cd command allows you to navigate between directories.

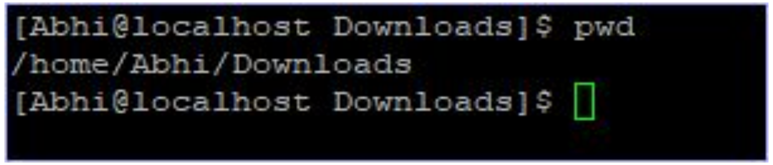


```
[Abhi@localhost ~]$ cd Downloads
[Abhi@localhost Downloads]$ 
```

Commands: ls, cd, pwd, mkdir, rm & cp

3. pwd - Print Working Directory:

The pwd command displays the full path of the current working directory.

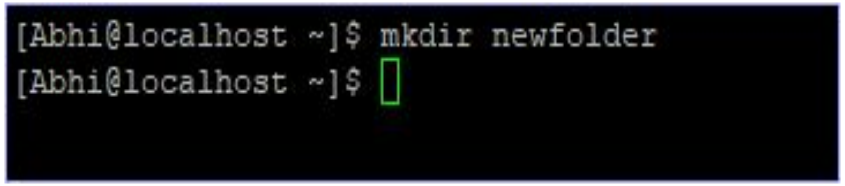


```
[Abhi@localhost Downloads]$ pwd
/home/Abhi/Downloads
[Abhi@localhost Downloads]$ 
```

Commands: ls, cd, pwd, mkdir, rm &   cp

4. mkdir - Make Directory:

The mkdir command is used to create a new directory.

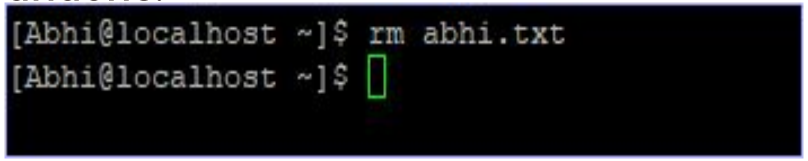


```
[Abhi@localhost ~]$ mkdir newfolder
[Abhi@localhost ~]$ 
```

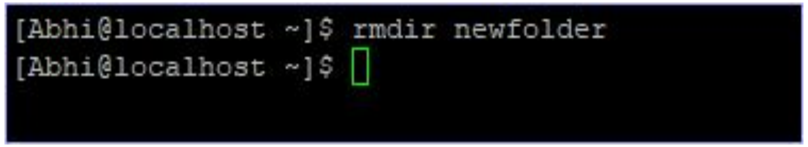
Commands: ls, cd, pwd, mkdir, rm & cp

5. rm - Remove Files or Directories:

The rm command is used to delete files or directories. Be cautious as this action cannot be undone.



```
[Abhi@localhost ~]$ rm abhi.txt
[Abhi@localhost ~]$ 
```



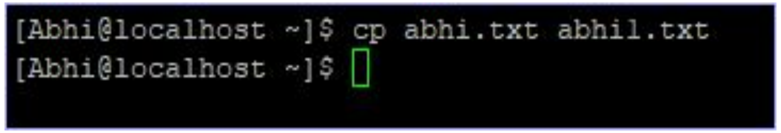
```
[Abhi@localhost ~]$ rmdir newfolder
[Abhi@localhost ~]$ 
```

Commands: ls, cd, pwd, mkdir, rm & cp

6. cp - Copy Files or Directories:

The cp command allows you to copy files or directories from one location to another.

```
cp <source_file> <destination>
```



```
[Abhi@localhost ~]$ cp abhi.txt abhil.txt
[Abhi@localhost ~]$ 
```

Pop Quiz

Q. Which command is used to change the current directory?

A

cd

B

pwd

PW SKILLS

Pop Quiz

Q. Which command is used to change the current directory?

A

cd

B

pwd

PW SKILLS

How to navigate the file system, including absolute & Relative paths

Understanding File System Paths

Absolute Paths

Definition: An absolute path is a complete path from the root directory ‘/’ to the desired file or directory. It always starts with a ‘/’.

Example: ‘/home/user/documents/report.txt’ refers to the file ‘report.txt’ located in the documents directory, which is under the ‘user’ directory in the ‘home’ directory.

Understanding File System Paths

Relative Paths

Definition: A relative path refers to a location in relation to the current working directory. It does not start with a ‘/’.

Example: If your current directory is ‘/home/user’, then ‘documents/report.txt’ refers to the same file as above.

Basic Navigation Commands

1. Change Directory (cd):

- Absolute path example:

```
cd /home/user/documents
```

- Relative Path Example (if currently in /home/user):

```
cd documents
```

Basic Navigation Commands

2. List Directory Contents (ls):

- Example:

```
ls
```

- To list contents of another directory using an absolute path:

```
ls /home/user/documents
```

Basic Navigation Commands

Practical Examples:

1. Navigating to a subdirectory:

- If you are currently in /home/user and want to go to projects:

```
cd projects
```

2. Navigating back to Home Directory:

```
cd ~
```

Basic Navigation Commands

Practical Examples:

3. Accessing a File with an Absolute Path:

```
cat /home/user/documents/notes.txt
```

4. Using Relative Paths:

```
cat notes.txt
```

Pop Quiz

Q. If your current directory is '/home/user', what does 'cd documents' do?

A

Changes to /documents

B

Changes to /home/user/documents

Pop Quiz

Q. If your current directory is '/home/user', what does 'cd documents' do?

A

Changes to /documents

B

Changes to /home/user/documents

cat, less & grep for viewing & searching files

cat, less & grep

‘Cat’

- **Purpose:** Displays the contents of a file or concatenates multiple files.

```
cat filename.txt
```

Common Options:

- -n: Number all output lines.
- -b: Number non-empty output lines.

cat, less & grep

‘Less’

Purpose: A pager program that allows viewing file contents one screen at a time, suitable for large files.

```
less filename.txt
```

Navigation: Use spacebar to scroll down, ‘b’ to scroll up, and ‘/search term’ to search within the file.

cat, less & grep

‘Grep’

- **Purpose:** Searches for specific patterns in files and outputs matching lines.

```
grep "search_term" filename.txt
```

Common Options:

- -i : Ignore case.
- -r : Search recursively in directories.

Pop Quiz

Q. Which command would you use to search for a specific pattern in a file?

A

grep

B

find

Pop Quiz

Q. Which command would you use to search for a specific pattern in a file?

A

grep

B

find

SKILLS

Purpose of users & groups in Linux

Overview of Purposes

Users

Definition: A user in Linux is an account created for an individual or a process that interacts with the system. Each user has a unique username and is assigned a User ID (UID).

Purpose:

- Identity
- Access control
- Resource Management

Overview of Purposes

Groups

Definition: A group is a collection of users who share common access privileges. Each group has a unique Group ID (GID).

Purpose:

- Access Control
- Simplified administration
- Collaboration
- Resource allocation

Pop Quiz

Q. What is the primary purpose of a user account in Linux?

A

To manage system processes

B

To provide individual access to the system

Pop Quiz

Q. What is the primary purpose of a user account in Linux?

A

To manage system processes

B

To provide individual access to the system

PW SKILLS

Demonstrating user creation, modification, and deletion

User Creation

Command: adduser

The ‘adduser’ command is used to create a new user account in a user-friendly manner.

Example:

```
sudo adduser johnny
```

This command creates a new user named johnny and sets up a home directory at /home/johnny.

User Modification

Command: usermod

The usermod command modifies existing user accounts. You can change attributes like group memberships or home directories.

Example:

```
sudo usermod -aG sudo johnny
```

This adds johnny to the sudo group, granting him administrative privileges.

User Deletion

Command: deluser

The deluser command removes a user account from the system. You can also delete the user's home directory if needed.

Example:

```
sudo deluser johnny
```

This deletes the user johnny but retains his home directory.

To delete the user and their home directory:

```
sudo deluser --remove-home johnny
```

Pop Quiz

Q. Which command is used to create a new user in Linux?

A

useradd

B

adduser

Pop Quiz

Q. Which command is used to create a new user in Linux?

A

useradd

B

adduser

PW SKILLS

How to manage groups

Group Creation

Command: groupadd

Creates a new group in the system.

Example:

```
sudo groupadd developers
```

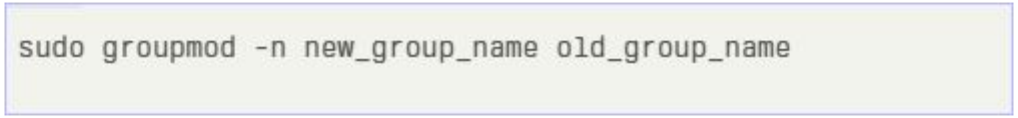
This command creates a new group named developers.

Group Modification

Command: groupmod

Modifies an existing group’s attributes, such as its name or Group ID (GID).

Example:



```
sudo groupmod -n new_group_name old_group_name
```

This renames old-group-name to new-group-name.

Group Deletion

Command: groupdel

Deletes an existing group from the system.

Example:

```
sudo groupdel developers
```

This command deletes the developers group.

Pop Quiz

Q. What does the 'groupmod -n new-group old-group' command do?

A

Creates a new group with the new name

B

Renames the old group to the new group name

Pop Quiz

Q. What does the 'groupmod -n new-group old-group' command do?

A

Creates a new group with the new name

B

Renames the old group to the new group name

Assigning users to groups & Changing group membership

Assigning Users to Groups

Adding a User to a Group

Use the usermod command with the -a (append) and -G (group) flags.

Example:

```
sudo usermod -a -G sudo john
```

This adds the user john to the sudo group, granting him administrative privileges.

Assigning Users to Groups

Creating a New User and Assigning Groups:

You can create a new user and add them to a group simultaneously using the useradd command.

```
sudo useradd -g staff -G developers,test cktutorials
```

This creates the user ‘cktutorials’ with ‘staff’ as the primary group and adds them to the ‘developers’ and ‘test’ groups.

Changing Group Memberships

1. Modifying Group Memberships:

- To change a user's group memberships, use the ‘usermod’ command again, ensuring to use the ‘-a’ option when adding them to additional groups.
- To remove a user from a specific group, you typically need to specify their new group memberships without including the group you want to remove them from.

Changing Group Memberships

2. Viewing Group Memberships:

- To check which groups a user belongs to, use:

```
groups username
```

- This command lists all groups associated with the specified user.

Pop Quiz

Q. Which command would you use to create a new user and assign them to a primary and secondary group?

A

```
useradd -g primarygroup -G secondarygroup username
```

B

```
adduser username --groups primarygroup,secondarygroup
```

Pop Quiz

Q. Which command would you use to create a new user and assign them to a primary and secondary group?

A

useradd -g primarygroup -G secondarygroup username

B

adduser username --groups primarygroup,secondarygroup

SKILLS

Take A 5-Minute **Break!**

- Stretch and relax
- Hydrate
- Clear your mind
- Be back in 5 minutes

Package Managers

APT (Advanced Package Tool)

Overview:

APT is a package management system used primarily in Debian-based distributions such as Ubuntu. It simplifies the process of handling software packages by automating the retrieval, configuration, and installation of software from repositories.

Key Features:

- Dependency Resolution
- User-Friendly Interface

APT (Advanced Package Tool)

Common Commands:

- Update Package Lists:

```
sudo apt update
```

- Install a Package:

```
sudo apt install <package-name>
```

- Remove a Package:

```
sudo apt remove <package-name>
```

YUM (Yellowdog Updater Modified) and DNF (Dandified YUM)

Overview

- YUM is a package manager for RPM-based distributions like CentOS and Fedora. It was designed to manage .rpm packages and facilitate software installation and updates.

- DNF is the successor to YUM, introduced in Fedora 22 and RHEL 8, offering improved performance and additional features.

Key Features:

YUM:

- Automatic Dependency Management
- Command-Line Interface

YUM (Yellowdog Updater Modified) and DNF (Dandified YUM)

DNF:

- Enhanced Performance
- Plugin Support

Common Commands:

- Install a Package:

```
sudo dnf install <package-name>
```

- Update Packages:

```
sudo dnf update
```

Pop Quiz

Q. What is the primary function of package managers like APT and YUM?

A

To manage system hardware

B

To handle software installation and updates

Pop Quiz

Q. What is the primary function of package managers like APT and YUM?

A

To manage system hardware

B

To handle software installation and updates

PW SKILLS

Demonstrating installing, updating, and removing packages using package management tools.

Installing, Updating, and Removing Packages

Installing a Package:

To install a package using APT, you first need to update the package index to ensure you have the latest information about available packages. Then, you can install the desired package.

1. Update Package Index:

```
sudo apt update
```

2. Install a Package:

```
sudo apt install <package-name>
```

```
sudo apt install curl
```

Installing, Updating, and Removing Packages

Updating Packages:

To update all installed packages to their latest versions, use the following command:

```
sudo apt upgrade
```

If you want to ensure that any obsolete packages are removed and new dependencies are installed as needed, you can use:

```
sudo apt full-upgrade
```

This command is more comprehensive as it handles changing dependencies with new versions of packages.

Installing, Updating, and Removing Packages

Removing a Package:

When you need to remove a package, you can choose whether to keep its configuration files or remove them as well.

1. Remove a Package (keeping configuration files):

```
sudo apt remove <package-name>
```

This command removes the specified package but leaves its configuration files intact. For example:

```
sudo apt remove curl
```

Installing, Updating, and Removing Packages

Removing a Package:

2. Purge a Package (removing configuration files):

```
sudo apt remove curl
```

This command removes both the specified package and its configuration files. For example:

```
sudo apt purge curl
```

Installing, Updating, and Removing Packages

Removing a Package:

3. Remove Unused Dependencies:

After removing packages, it's good practice to clean up any dependencies that are no longer needed:

```
sudo apt autoremove
```

This command automatically removes packages that were installed as dependencies but are no longer required by any installed software.

Pop Quiz

Q. Which command will completely remove a package along with its configuration files?

A

sudo apt purge
<package-name>

B

sudo apt remove
<package-name>

Pop Quiz

Q. Which command will completely remove a package along with its configuration files?

A

sudo apt purge <package-name>

B

sudo apt remove <package-name>

Basics Of System Administration With Systemctl

Basics of System Administration with systemctl

What is systemctl ?

‘systemctl’ is a command-line utility that allows administrators to manage systemd services, which are essential for starting, stopping, and controlling the behavior of services on a Linux system.

It provides a unified way to manage services and their dependencies.

Basics of System Administration with systemctl

Common systemctl Commands:

1. Starting a Service

```
sudo systemctl start <service-name>
```

Example: To start the Apache web server service (usually named httpd or apache2), you would run:

```
sudo systemctl start apache2
```

Basics of System Administration with systemctl

Common systemctl Commands:

2. Stopping a Service

```
sudo systemctl stop <service-name>
```

Example: To stop the Apache web server service:

```
sudo systemctl stop apache2
```

Basics of System Administration with systemctl

Common systemctl Commands:

3. Restarting a Service

```
sudo systemctl restart <service-name>
```

Example: To restart the Apache web server:

```
sudo systemctl restart apache2
```

Basics of System Administration with systemctl

Common systemctl Commands:

4. Checking the status of a Service

```
sudo systemctl status <service-name>
```

Example: To check the status of the Apache web server:

```
sudo systemctl status apache2
```

Basics of System Administration with systemctl

Common systemctl Commands:

5. Enabling a Service at Boot

```
sudo systemctl enable <service-name>
```

Example: To enable Apache to start at boot:

```
sudo systemctl enable apache2
```

Basics of System Administration with systemctl

Common systemctl Commands:

6. Disabling a Service at Boot

```
sudo systemctl disable <service-name>
```

Example: To disable Apache from starting at boot:

```
sudo systemctl disable apache2
```

Pop Quiz

Q. What is the primary purpose of using systemctl in Linux?

A

To manage user accounts

B

To control and manage services

Pop Quiz

Q. What is the primary purpose of using systemctl in Linux?

A

To manage user accounts

B

To control and manage services

File Permissions & Its Numeric (octal) Format

File Permissions

Overview:

Types of Permissions:

1. Read (r)
2. Write (w)
3. Execute (x)

User Categories:

Permissions are assigned to three categories of users:

- Owner: The user who created the file.
- Group: A set of users who share access to the file.
- Others: All other users not in the owner or group categories.

File Permissions

Numeric (Octal) Format:

File permissions can also be represented in numeric (octal) format, where each permission type is assigned a specific value:

- Read = 4
- Write = 2
- Execute = 1

The numeric representation consists of three digits:

- The first digit represents the owner's permissions.
- The second digit represents the group's permissions.
- The third digit represents others' permissions.

File Permissions

Numeric (Octal) Format:

Examples:

‘chmod 755 file.txt’:

- Owner: Read (4) + Write (2) + Execute (1) = 7
- Group: Read (4) + Execute (1) = 5
- Others: Read (4) + Execute (1) = 5
- Resulting permissions: ‘rwxr-xr-x’

Pop Quiz

Q. In octal format, what does the number '7' represent in terms of file permissions?

A

Read, Write, and Execute

B

Read and Write

Pop Quiz

Q. In octal format, what does the number '7' represent in terms of file permissions?

A

Read, Write, and Execute

B

Read and Write

SKILLS

‘chmod’ for changing file permissions

Changing File Permissions with chmod

The chmod command is used to change file permissions.

1. Numeric (Octal) Format:

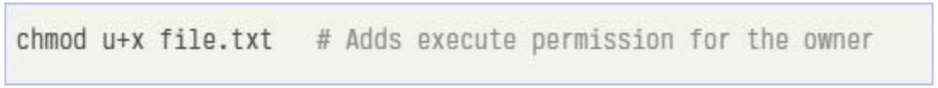
```
 chmod 644 file.txt   # Sets read/write for owner and read-only for group
and others
```

This command grants full permissions to the owner and read/execute permissions to the group and others.

Changing File Permissions with chmod

2. Symbolic Format:

In symbolic format, you can add (+), remove (-), or set (=) specific permissions for the user categories: user (u), group (g), and others (o).



```
chmod u+x file.txt   # Adds execute permission for the owner
```

This command adds execute permission for the user (owner) of the file filename.

Changing File Permissions with chmod

3. Changing Permissions Recursively:

To change permissions for a directory and all its contents, use the -R option.

```
chmod -R 755 directoryname
```

- This command sets the permissions of directoryname and all files and subdirectories within it to allow full access for the owner and read/execute access for group and others.

Changing File Permissions with chmod

4. Removing Permissions:

You can also remove specific permissions using symbolic notation.

```
chmod g-w filename
```

- This command removes write permission from the group for filename.

Pop Quiz

Q. If you want to remove execute permission for all users from a file, which command would you use?

A

chmod u-x filename

B

chmod a-x filename

Pop Quiz

Q. If you want to remove execute permission for all users from a file, which command would you use?

A

chmod u-x filename

B

chmod a-x filename

Ownership of File & Use of ‘chown’

File Ownership

- User: The individual who owns the file, typically the creator.
- Group: A collection of users that can share permissions for the file.
- Others: All other users on the system who are not the owner or part of the group.

Changing Ownership with chown:

The chown command is used to change the owner and/or group of a file or directory. The basic syntax is:

```
chown [new_owner]:[new_group] [file_or_directory]
```

File Ownership

Examples:

1. Change Owner Only:

```
sudo chown newuser filename
```

This command changes the owner of filename to newuser.

File Ownership

Examples:

2. Change Owner and Group:

```
sudo chown newuser:newgroup filename
```

This command changes the owner to new user and the group to newgroup.

File Ownership

Examples:

3. Change Group Only:

```
sudo chown :newgroup filename
```

This command changes only the group of filename to newgroup, leaving the owner unchanged.

File Ownership

Viewing Ownership:

To view the current owner and group of a file, you can use:

```
ls -l filename
```

This command displays detailed information about the file, including its permissions, owner, and group.

Pop Quiz

Q. What does the command chown user:group filename do?

A

Changes the owner to 'user' and the group to 'group'.

B

Changes the permissions of the file.

Pop Quiz

Q. What does the command chown user:group filename do?

A

Changes the owner to 'user' and the group to 'group'.

B

Changes the permissions of the file.

‘umask’ for setting default permissions

‘umask’ for default permissions

Umask:

Umask, or user file-creation mode, is a command used in Linux and UNIX-like systems to set default permissions for newly created files and directories. It determines the permissions that will not be assigned to new files, effectively controlling the default access rights.

How Umask works:

- Default Permissions: By default, files are created with permissions of 666 (read and write for user, group, and others), while directories are created with permissions of 777 (read, write, and execute for all).

‘umask’ for default permissions

Calculation: The umask value is subtracted from these defaults to determine the actual permissions. For example, a umask of 022 results in:

- Files: 666−022=644 (read and write for owner; read for group and others)
- Directories: 777−022=755 (read, write, and execute for owner; read and execute for group and others)

‘umask’ for default permissions

Setting Umask:

Umask can be set globally in system configuration files or locally in user profiles. To check the current umask value, simply type umask in the terminal. To change it temporarily, use:

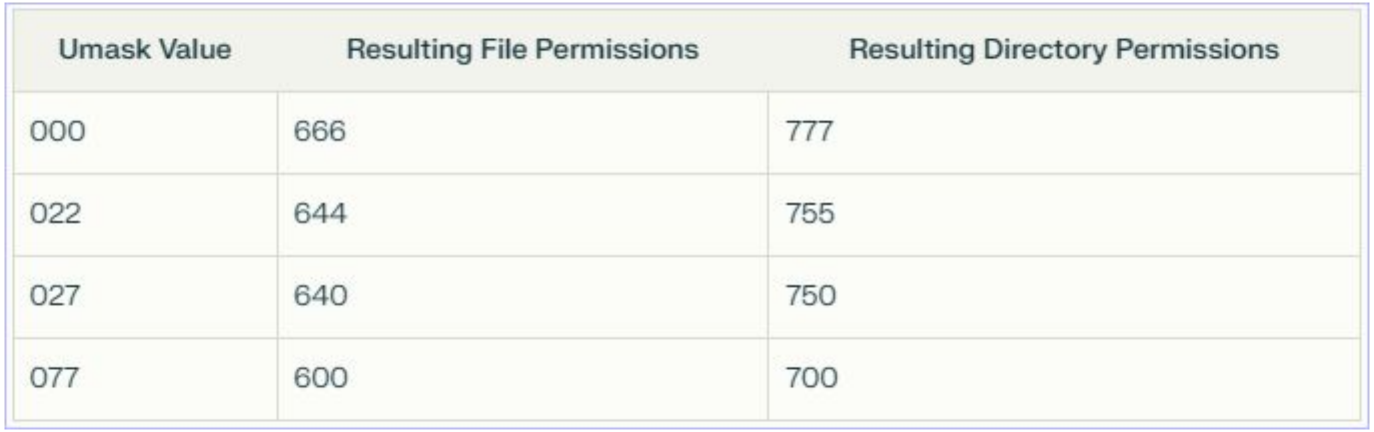
```
umask <value>
```

```
umask 027
```

This would deny write permission to group members and others for newly created files and directories.

‘umask’ for default permissions

Common Umask Values:



Umask Value	Resulting File Permissions	Resulting Directory Permissions
000	666	777
022	644	755
027	640	750
077	600	700

Umask is essential for maintaining security by ensuring that files are not created with overly permissive access rights.

Pop Quiz

Q. How is the final permission of a new file calculated using umask?

A

By adding the umask value to the maximum permission

B

By subtracting the umask value from the maximum permission

Pop Quiz

Q. How is the final permission of a new file calculated using umask?

A

By adding the umask value to the maximum permission

B

By subtracting the umask value from the maximum permission

Pop Quiz

Q. What will be the resulting permissions for a file if the umask is set to 027??

A

rw-r-----

B

rw-r--r--

Pop Quiz

Q. What will be the resulting permissions for a file if the umask is set to 027??

A

rw-r-----

B

rw-r--r--

Demonstrating ‘df’ for checking file system disk usage

Checking file system disk usage

‘df’:

The ‘df’ command in Linux is used to check file system disk usage, providing insights into the total, used, and available disk space across mounted file systems. Here’s a brief overview of how to use it effectively:

Basic Command:

To display disk usage information, simply run:

```
df
```

Checking file system disk usage

This will show a table with columns for:

- **Filesystem:** The name of the mounted storage device.
- **Size:** Total size of the filesystem in 1K blocks.
- **Used:** Amount of space currently occupied.
- **Avail:** Free space available.
- **Use%:** Percentage of the filesystem used.
- **Mounted on:** Directory where the filesystem is mounted.

Checking file system disk usage

Human-Readable Format:

To make the output easier to read, use the -h option:

```
df -h
```

This displays sizes in a human-readable format (e.g., GB, MB).

Example Output:

An example output using df -h might look like this:

```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       20G   5G   15G   25% /
tmpfs           1.9G  200M 1.7G   10% /dev/shm
```

Pop Quiz

Q. What information is NOT typically provided by the df command?

A

Total size of the filesystem

B

Number of files in the filesystem

Pop Quiz

Q. What information is NOT typically provided by the df command?

A

Total size of the filesystem

B

Number of files in the filesystem

‘du’ to identify directory sizes and locate space-consuming files

Basic Usage of ‘du’

The du (disk usage) command in Linux is a useful tool for identifying directory sizes and locating space-consuming files. Here’s a concise overview of how to use it effectively:

1. Check Size of a Specific Directory:

To find the total size of a specific directory in a human-readable format, use:

```
du -sh /path/to/directory
```

Basic Usage of ‘du’

2. Check Sizes of All Subdirectories:
To see the sizes of all first-level subdirectories within a directory, you can use:

```
du -h --max-depth=1 /path/to/directory
```

3. Find the Largest Directories:
To identify the largest directories within a specified path, you can combine ‘du’ with ‘sort’ and ‘head’:

```
du -h /path/to/directory | sort -rh | head -5
```

Pop Quiz

Q. What is the default behavior of the du command when no options are provided?

A

It recursively shows the size of each subdirectory.

B

It displays only the total disk usage.

Pop Quiz

Q. What is the default behavior of the du command when no options are provided?

A

It recursively shows the size of each subdirectory.

B

It displays only the total disk usage.

PW SKILLS

Practical tips for clearing space and managing storage efficiently

Tips

Here are some practical tips for clearing space and managing storage efficiently on a Linux system:

1. Regularly Check Disk Usage
2. Identify and Remove Unnecessary Files
3. Clean Package Cache
4. Use Disk Usage Analyzers
5. Archive and Compress Files
6. Manage Log Files
7. Remove Old Kernels
8. Monitor Disk Usage Continuously

Time for case study!

Important

- Complete the post-class assessment
- Complete assignments (if any)
- Practice the concepts and techniques taught in this session
- Review your lecture notes
- Note down questions and queries regarding this session and consult the teaching assistants

Thanks
!

PW SKILLS