



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

**NAAC  
GRADE A+**  
Accredited University

# **Disk Usage Analyzer - Linux CLI Tool**

**Linux: Project Report**

***Submitted by***

Gaurav Raj (24MCA20025)

***in partial fulfillment for the award of the degree of***

**MASTER OF COMPUTER APPLICATION**

**IN**

**COMPUTER SCIENCE**



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

Chandigarh University  
OCTOBER, 2024



**CHANDIGARH  
UNIVERSITY**

Discover. Learn. Empower.

**NAAC  
GRADE A+**  
Accredited University

## **BONAFIDE CERTIFICATE**

Certified that this project report “**Disk Usage Analyzer**” is the Bonafide work of “**Gaurav Raj**” who carried out the project work under my/our supervision.

### **SIGNATURE**

Gitanjali Mam  
**SUPERVISOR**  
**(MCA)**

### **SIGNATURE**

DR. Abdullah  
**HEAD OF THE  
DEPARTMENT**  
**(MCA)**

Submitted for the project viva-voice examination held on\_6.11.2024

INTERNAL EXAMINER

EXTERNAL EXAMINER

## **Acknowledgement**

We would like to express our special thanks of gratitude to our supervisor Gitanjali Mam as well as to our head of the department who gave us this opportunity to work on the project “ Disk Usage Analyzer “ which also helped us in doing lot of research work and we came to know lot of new things.

Gaurav Raj [24MCA20025]

## 1.Introduction

Managing disk space is a common challenge, particularly on Linux systems where numerous files and directories accumulate over time. The Disk Usage Analyzer is a command-line tool designed specifically for Linux users, providing a comprehensive view of storage usage and helping users quickly identify which directories and files consume the most space. By employing standard Linux commands like `df`, `du`, and `find`, this tool delivers critical insights for effective disk management.

The `df` command is used to display an overall summary of disk usage across different mounted filesystems, helping users monitor free and occupied space at a glance. For a more detailed analysis, the `du` command assesses storage consumption on a directory and file level, offering a deeper understanding of specific areas of the system that require attention. Additionally, the `find` command assists in locating large files within specific directories, enabling users to prioritize which files to review or delete when space runs low.

The Disk Usage Analyzer is especially valuable for both system administrators and regular users who want to avoid disk overflow issues. Its outputs are formatted for clarity, making it easy to interpret storage patterns and track down excessive usage quickly. Whether managing a personal computer or an enterprise server, this tool provides a straightforward and efficient approach to maintaining optimal disk usage. By combining the capabilities of `df`, `du`, and `find`, the Disk Usage Analyzer becomes a lightweight yet powerful solution for effective storage management on Linux systems.

## 2. Abstract

The Disk Usage Analyzer is a command-line tool for Linux that provides users with a comprehensive overview of their storage usage, making it easier to identify large directories and files and manage disk space effectively. Storage management can become challenging, especially on systems with limited disk space or when dealing with a vast number of files. The Disk Usage Analyzer simplifies this task by leveraging essential Linux commands, such as `df`, `du`, and `find`, to gather data on storage allocation and usage patterns.

The tool scans the file system to deliver insights into the amount of disk space used by specific directories and files, offering both overall summaries and targeted analyses of selected directories. The `df` command provides a quick summary of disk space availability and usage on mounted filesystems, while `du` dives deeper into the disk usage by each directory and file. To locate and sort large files, the tool integrates the `find` command, allowing users to focus on particular directories, sort files by size, and pinpoint storage-heavy locations quickly.

In addition to its informative output, the Disk Usage Analyzer is user-friendly, with clear formatting that presents data in an accessible way. This clarity is valuable for system administrators and everyday Linux users who need quick insights into their disk usage. By combining these fundamental Linux utilities, the Disk Usage Analyzer is a lightweight, efficient solution for anyone seeking to optimize storage use, prevent disk space shortages, and maintain better overall system performance.

### 3.Objectives

The objective of the Disk Usage Analyzer is to provide Linux users and administrators with a convenient, command-line-based tool to monitor and manage disk space usage efficiently. This tool aims to:

1. **Offer a Disk Usage Overview:** Provide a quick summary of the total disk usage across all mounted filesystems, helping users understand overall space consumption.
2. **Identify Largest Storage Consumers:** Quickly list the top 10 largest directories and files on the system, enabling users to identify areas with significant storage usage and potentially free up space.
3. **Enable Specific Directory Analysis:** Allow users to analyze the disk usage of specific directories, providing insights into their impact on overall disk space.
4. **Streamline Storage Management:** Simplify the process of identifying storage-hogging directories and files, making it easier to clean up unnecessary data and optimize disk space.

### 4.Prerequisites

Before using the Disk Usage Analyzer script, ensure the following prerequisites are met:

1. **Linux Operating System:** The script is designed to run on Linux-based systems with bash shell support.
2. **Basic Knowledge of Command Line:** Familiarity with using the terminal and basic command-line operations is helpful for executing the script and interpreting its output.
3. **Sudo or Root Access (Optional):** For complete system-wide analysis, root privileges may be required to access certain directories and files. Running the script as a regular user may limit access to certain areas of the filesystem.
4. **Installations:** The script relies on standard Linux utilities (df, du, find, sort, and head), which are commonly available on most Linux distributions by default. Ensure these
5. utilities are installed on the system.
6. **Script File:** Save the provided code as `disk_usage_analyzer.sh` and make it executable. Use the following command to make the script executable:

```
[root@MiWiFi-R4CM-srv ~]# ls
anaconda-ks.cfg      Downloads
Desktop              edit.txt
disk_usage_analyzer.sh file1.txt
Documents            file2.txt
```

```
[root@MiWiFi-R4CM-srv ~]# ./disk_usage_analyzer.sh
```

## 5.Implementation

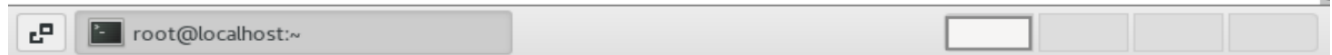
Here is the implementation of the **Disk Usage Analyzer** script. You can copy this code into a file, make it executable, and run it to analyze disk usage on your Linux system.

```
#!/bin/bash
display usage Summary
echo "Disk Usage Summary:"
df -h --total
echo

#List top 10 largest Directories
echo"Top 10 Largest Directories:"
du -ah / | sort -rh | head -n 10
echo

#check disk usage
read -p "Enter a directory path to analyze (or press enter to skip): " dir
if [-d "$dir" ]; then
    echo "Disk Usage for directory $dir: "
    du -sh "$dir"
else
    echo "No Directory Specified or invalid directory."
fi
echo

#find and list the top 10 largest files
echo "Top 10 Largest Files:"
find / -type f -exec du -h {} + 2>/dev/null | sort -rh | head -n 10
echo
```



### 5.1 Usage Instructions

1. Save the script as disk\_usage\_analyzer.sh:

```
nano disk_usage_analyzer.sh:
```

2. Make the script executable:

```
chmod +x disk_usage_analyzer.sh
```

3. Run the script:

```
./disk_usage_analyzer.sh
```

## 6.Explanation of the Script

1. `find / -type f`: Searches for all files (`-type f`) under the root directory (`/`).
2. `-exec du -h {} +`: For each file found, `du -h` is executed to display its size in human-readable format.

3. `2>/dev/null`: Redirects permission errors to `/dev/null`, effectively suppressing them, so the script only shows files it has access to.
4. `sort -rh | head -n 10`: Sorts the files by size in reverse order and displays the top 10 largest files.
5. This section is useful for quickly identifying large, individual files that may be taking up unnecessary space.

### 1.Display Disk Usage Summary

```
bash

echo "Disk Usage Summary:"
df -h --total
echo
```

- **Command:** `df -h --total`

➤ **Explanation:**

- `df` shows the amount of disk space used and available on mounted filesystems.
- `-h` makes it human-readable (shows sizes in MB, GB).
- `--total` adds a summary line showing the total disk usage across all filesystems.
- **Purpose:** Provides an overview of total disk space usage.

### 2.List the Top 10 Largest Directories

```
bash

echo "Top 10 Largest Directories:"
du -ah / | sort -rh | head -n 10
echo
```

- **Command:** `du -ah / | sort -rh | head -n 10`

➤ **Explanation:**

- `du -ah /`: Calculates the size of each file and directory from the root (`/`) directory.
- `-a` includes files as well as directories.
- `-h` makes it human-readable.
- `sort -rh`: Sorts entries by size in reverse order (largest first).
- `head -n 10`: Shows only the top 10 entries.
- **Purpose:** Identifies the directories and files using the most space.

### 3. Check Disk Usage for a Specific Directory

```
read -p "Enter a directory path to analyze (or press enter to skip): " dir
if [ -d "$dir" ]; then
    echo "Disk Usage for directory $dir: "
    du -sh "$dir"
else
    echo "No Directory Specified or invalid directory."
fi
echo
```

- **Commands:** read, du -sh "\$dir"

#### ➤ Explanation:

- read -p: Prompts the user to enter a directory path.
- if [ -d "\$dir" ]; then: Checks if the entered path is a valid directory.
- du -sh "\$dir": Calculates and displays the total size of the specified directory.
- -s gives a summary for the whole directory.
- -h makes it human-readable.
- **Purpose:** Allows the user to analyze a specific directory's storage usage.

### 4. Find and List the Top 10 Largest Files

```
bash

echo "Top 10 Largest Files:"
find / -type f -exec du -h {} + 2>/dev/null | sort -rh | head -n 10
echo
```

- **Command:** find / -type f -exec du -h {} + 2>/dev/null | sort -rh | head -n 10

#### ➤ Explanation:

- find / -type f: Finds all files starting from the root directory.
- -exec du -h {} +: For each file found, du -h displays its size.
- 2>/dev/null: Suppresses permission errors by redirecting them to /dev/null.
- sort -rh | head -n 10: Sorts files by size (largest first) and displays the top 10.
- **Purpose:** Lists the largest files on the system to help identify space hogs.



## 7.How to Run the Script

Follow these steps to save, make executable, and run the script:

### Save the Script:

- Open a text editor (like nano) and create a file named disk\_usage\_analyzer.sh

```
nano disk_usage_analyzer.sh
```

- Copy and paste the script into this file.
- Save and close the file by pressing Ctrl+X, then Y, then Enter.

### Make the Script Executable:

- Run the following command to grant execute permissions to the script:

```
chmod +x disk_usage_analyzer.sh
```

### Run the Script:

- Execute the script using the following command:

```
./disk_usage_analyzer.sh
```

## Conclusion

The **Disk Usage Analyzer** script is a straightforward yet powerful tool for managing disk space on Linux systems. By utilizing fundamental commands like `df`, `du`, and `find`, the script provides a comprehensive view of storage usage, making it easier for users to identify the largest directories and files, check specific directory sizes, and ultimately manage storage resources more effectively. The tool is designed with ease of use in mind, offering human-readable outputs and useful summaries without requiring extensive knowledge of Linux commands.

This script is particularly helpful for system administrators and users who need to keep a close eye on disk space, especially on servers or systems with limited storage. With the ability to pinpoint the largest directories and files, users can make informed decisions about which files to clean up, resulting in a more optimized and efficient system.

The Disk Usage Analyzer is a versatile solution that can be easily customized and extended to suit individual needs, and it serves as a solid foundation for anyone looking to build upon disk usage analysis tools in Linux.

## OUTPUT

```
[root@MiWiFi-R4CM-srv ~]# ./disk_usage_analyzer.sh
./disk_usage_analyzer.sh: line 1: splay: command not found
Disk Usage Summary:
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        903M   0  903M   0% /dev
tmpfs           919M   0  919M   0% /dev/shm
tmpfs           919M  9.4M  910M   2% /run
tmpfs           919M   0  919M   0% /sys/fs/cgroup
/dev/mapper/centos-root 17G  6.3G  11G  37% /
/dev/sda1       1014M  188M  827M  19% /boot
tmpfs           184M   40K  184M   1% /run/user/0
total           22G  6.5G   16G  30% -

./disk_usage_analyzer.sh: line 7: echoTop 10 Largest Directories:: command not found
du: cannot access '/proc/16613/task/16613/fd/3': No such file or directory
du: cannot access '/proc/16613/task/16613/fdinfo/3': No such file or directory
du: cannot access '/proc/16613/fd/4': No such file or directory
du: cannot access '/proc/16613/fdinfo/4': No such file or directory
6.4G    /
4.9G    /usr
2.0G    /usr/share
```

```
2.0G    /usr/share
1.3G    /usr/lib64
1.3G    /usr/lib
1022M   /swapfile
513M    /usr/lib/firmware
453M    /usr/share/locale
323M    /usr/share/xml
321M    /usr/share/xml/scap/ssg/content
```

```
Enter a directory path to analyze (or press enter to skip): /
./disk_usage_analyzer.sh: line 13: [-d: command not found
No Directory Specified or invalid directory.
```

```
Top 10 Largest Files:
1022M   /swapfile
141M    /var/lib/rpm/Packages
139M    /usr/lib64/firefox/libxul.so
102M    /usr/lib/locale/locale-archive
78M     /boot/initramfs-0-rescue-5d0428da0520244fa8910c9cca6b981b.img
71M     /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.332.b09-1.el7_9.x86_64/jre/lib/rt.jar
66M     /usr/lib/jvm/java-1.7.0-openjdk-1.7.0.261-2.6.22.2.el7_8.x86_64/jre/lib/rt.jar
50M     /usr/lib64/libwebkit2gtk-4.0.so.37.44.4
41M     /usr/lib64/libLLVM-7-rhel.so
40M     /usr/lib64/firefox/browser/omni.ja
```

```
[root@MiWiFi-R4CM-srv ~]# vi disk_usage_analyzer.sh
```