

Handling Text Files

1. Introduction to Working with Text Files

Linux systems heavily rely on text files for configuration, logging, and data storage. Almost every aspect of the system, from user settings to system behavior, is defined in plain text files, making them a crucial part of Linux administration.

For example, some common types of text files include:

- Log files
 - Configuration files
 - Data files
-

2. Viewing Files in Terminal

Command: `cat`

The `cat` command prints the entire contents of one or more files:

```
cat file.txt
```

Note: Avoid using `cat` for large files as it prints the entire content at once.

Viewing Partial File Contents

- **Head:** View the first lines of a file

```
head file.txt          # Default first 10 lines
head -n 3 file.txt     # First 3 lines only
```

- **Tail:** View the last lines of a file

```
tail file.txt          # Default last 10 lines
tail -n 3 file.txt     # Last 3 lines only
```

Real-Time File Monitoring

Use `tail -f` to monitor changes to log files in real-time:

```
sudo tail -f /var/log/syslog
```

- Press **Ctrl+C** to exit.
-

3. Editing Text Files (Nano Editor)

Linux has several text editors. `Nano` is user-friendly and recommended for beginners:

- Opening files with `Nano` :

```
nano file.txt
```

Nano Basic Commands:

Shortcut	Description
Ctrl + O	Save (write out) file
Ctrl + X	Exit Nano editor
Ctrl + F	Search within file
Ctrl + K	Cut line
Ctrl + U	Paste line
Ctrl + C	Show current cursor position

- **Saving to a new file:** Press `Ctrl + O` , enter new filename, then confirm with `Enter` .

4. Working with Large Files Efficiently

Checking File Size and Contents (`wc`)

Use `wc` (word count) to analyze file size and contents quickly:

```
wc file.txt      # Prints lines, words, bytes
```

- Specific info:
 - Lines → `wc -l file.txt`
 - Words → `wc -w file.txt`
 - Bytes → `wc -c file.txt`
- Multiple files at once:

```
wc -l file1.txt file2.txt
```

Efficient Navigation with `less`

The `less` command is optimized for viewing large files without loading them entirely into memory:

```
less file.txt
```

Navigation in `less` :

Shortcut	Description
Arrow Keys	Move line-by-line
f	Move forward (page down)

b	Move backward (page up)
/search_term	Search forward for search_term
n / N	Move to next/previous search match
50%	Jump to the middle (50%) of file
q	Exit less

5. Calculating Disk Usage (du)

Check **disk usage** with the `du` command.

- **Check file size on disk (allocated space):**

```
du file.txt
```

- **Check real ("apparent") file size:**

```
du -b file.txt
```

- **Check folder size (with summary):**

```
du -sb directory/
```

- **List folder size recursively:**

```
du -b directory/
```

Note:

Disk space may slightly differ from apparent file size due to file system allocation blocks.
