

Basic Linux Commands

Part2



Day 3 Task: Basic Linux Commands



Kunal Maurya · Feb 24, 2023 · 📖 5 min read

Cat Command

The `cat` command is a Unix and Linux utility that stands for "concatenate." It is used to display the contents of one or more files to the terminal, and it can also be used to combine files or create new files.

- Display the contents of a single file:

```
$ cat file1.txt
This is the contents of file1.txt.
```

COPY 

- Display the contents of multiple files:

```
$ cat file1.txt file2.txt file3.txt
This is the contents of file1.txt.
This is the contents of file2.txt.
This is the contents of file3.txt.
```

COPY 

- Combine two or more files and display the output:

```
$ cat file1.txt file2.txt > combined.txt
$ cat combined.txt
This is the contents of file1.txt.
This is the contents of file2.txt.
```

COPY 

- Add the contents of one file to the end of another file:

```
$ cat file1.txt >> file2.txt
$ cat file2.txt
```

COPY 

```
This is the contents of file1.txt.  
This is the contents of file1.txt.
```

- Display the contents of a file with line numbers:

```
$ cat -n file1.txt  
1 This is the contents of file1.txt.
```

COPY 

- Display the contents of a file in reverse order:

```
$ tac file1.txt  
.txt1 elif fo stnetnoc eht si sihT
```

COPY 

chmod Command

The `chmod` command is a command-line utility used in Unix-based operating systems to change the access permissions of files and directories. It stands for "change mode" and is used to modify the read, write, and execute permissions for a file or directory.

```
chmod 777 foldername
```

COPY 

History Command

The `history` command is used to display a list of previously executed commands from the user's command history.

- `history`: Display the list of previously executed commands along with their line numbers

```
$ history  
1 ls  
2 cd ..  
3 mkdir test  
4 cd test  
5 touch file.txt  
6 ls  
7 rm file.txt  
8 ls  
9 history
```

COPY 

- `history n`: Display the last `n` commands from the command history.

```
$ history 3  
7 rm file.txt  
8 ls  
9 history 3
```

COPY 

- `!n`: Execute the command with line number `n`.

- ```
$!5
touch file.txt
```

- `!!`: Execute the last command in the command history.

- ```
$ ls -l
total 0
-rw-r--r-- 1 user user 0 Feb 24 10:42 file.txt

$ !!
ls -l
total 0
-rw-r--r-- 1 user user 0 Feb 24 10:42 file.txt
```

- `!string`: Execute the most recent command that starts with `string`.

```
$ ls -l
total 0
-rw-r--r-- 1 user user 0 Feb 24 10:42 file.txt

$ !ls
ls -l
total 0
-rw-r--r-- 1 user user 0 Feb 24 10:42 file.txt
```

Remove Command

- `rm file.txt`: Remove a single file named `file.txt`.

- ```
$ ls
file.txt

$ rm file.txt

$ ls
(no output)
```

- `rm -r directory`: Remove a directory and its contents recursively.

- ```
$ ls -R test/
test/:
subdir  file.txt

test/subdir:
file.txt

$ rm -r test/

$ ls -R test/
ls: cannot access 'test/': No such file or directory
```

- `rm -i file.txt`: Remove a file and prompt for confirmation before removing it.

```
$ ls
file.txt

$ rm -i file.txt
rm: remove regular file 'file.txt'? y

$ ls
(no output)
```

- `rm -f file.txt`: Remove a file without prompting for confirmation.

```
$ ls
file.txt

$ rm -f file.txt

$ ls
(no output)
```

Touch Command

- `touch file.txt`: Create a new file named `file.txt`.

```
$ ls
(no output)

$ touch file.txt

$ ls
file.txt
```

- `touch *.txt`: Update the modification timestamp of all files in the current directory that end with the `.txt` extension.

```
$ ls -l
-rw-r--r-- 1 user user 0 Feb 24 11:00 file1.txt
-rw-r--r-- 1 user user 0 Feb 24 11:00 file2.txt
-rw-r--r-- 1 user user 0 Feb 24 10:58 file3.txt

$ touch *.txt

$ ls -l
-rw-r--r-- 1 user user 0 Feb 24 11:01 file1.txt
-rw-r--r-- 1 user user 0 Feb 24 11:01 file2.txt
-rw-r--r-- 1 user user 0 Feb 24 11:01 file3.txt
```

Head Command

- `head file.txt`: Display the first 10 lines of the file `file.txt`.
- `head -n 5 file.txt`: Display the first 5 lines of the file `file.txt`.

```
$ cat file.txt
line 1
line 2
line 3
line 4
line 5
line 6
line 7
line 8
line 9
line 10
line 11
line 12

$ head -n 5 file.txt
line 1
line 2
line 3
line 4
line 5
```

Tail Command

- `tail file.txt`: Display the last 10 lines of the file `file.txt`.
- `tail -n 5 file.txt`: Display the last 5 lines of the file `file.txt`.
- `tail -f file.txt`: Display the last few lines of the file `file.txt` and then monitor the file for changes, displaying any new lines as they are added. The `-f` option is particularly useful for monitoring log files that are constantly being updated.

```
$ tail -f file.txt
line 11
line 12
line 13
line 14
line 15

$ echo "line 16" >> file.txt # Add a new line to the file.

$ tail -f file.txt
line 12
line 13
line 14
line 15
line 16
```

vi Command

The `vi` command is a Unix-based text editor that is commonly used on the command line.

- `vi file.txt`: Open the file `file.txt` in the `vi` editor.

```
$ vi Colors.txt
```

- Once the file is open in the `vi` editor, you can enter editing mode by pressing the `i` key. In editing mode, you can make changes to the file.
-
- To save your changes and exit the editor, first press the `Esc` key to exit editing mode. Then, type `:wq` and press `Enter`. This will write your changes to the file and quit the editor.

diff Command

- `diff file1.txt file2.txt`: Compare the files `file1.txt` and `file2.txt` and display the differences between them.

```
$ cat file1.txt
hello world
this is a line
this is another line
this is a fourth line

$ cat file2.txt
hello world
this is a line
this is a new line
this is another line
this is a fourth line

$ diff file1.txt file2.txt
3a4
> this is a new line
```

Thank you for reading! Hope you find this article helpful.

~Kunal



Subscribe to my newsletter

Read articles from directly inside your inbox.
Subscribe to the newsletter, and don't miss out.

SUBSCRIBE

Linux

linux for beginners

linux-basics

linux-commands

DevOps Journey

MORE ARTICLES

Kunal Maurya

Kunal Maurya

Day 2 Task: Basics Linux command

Listing commands Here are some examples of different ls commands in Linux with explanations: ls - T...

Day 1: Getting Started with DevOps

Introduction In today's world, software development has become an integral part of businesses of all...

©2023 Kunal Maurya's Blog

[Archive](#) · [Privacy policy](#) · [Terms](#)



Publish with Hashnode

Powered by [Hashnode](#) - Home for tech writers and readers