

File Handling Utilities

Linux file handling commands are an essential part of daily system administration tasks. Get familiar with these Linux utilities and their usage with this comprehensive guide.

```
ver.pem ec2-user@13.112.191.175  
191.175 (13.112.191.175)' can't be established.  
60v2VvZXAxCU3kWJ21/DthHPY1xRhr7SN0jJtFzagS0.  
e connecting (yes/no)? yes  
12.191.175' (ECDSA) to the list of known hosts.  
ion denied (publickey).  
ver.pem ubuntu@13.112.191.175  
NU/Linux 4.4.0-1074-aws x86_64)
```

ubuntu.com
cape.canonical.com
u.com/advantage

Advantage Cloud Guest:
ss/services/cloud

untu system are free software;
each program are described in the
oc/*copyright.

WARRANTY, to the extent permitted by

(user "root"), use "sudo <command>".

1. ls: List Files and Directories

```
ravichandra@DESKTOP-HQRA3L1:~$ ls
dir1 dir2 file1 file2 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -r
file3 file2 file1 dir2 dir1
ravichandra@DESKTOP-HQRA3L1:~$ ls -l
total 20
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:58 dir1
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:59 dir2
-rw-r--r-- 1 ravichandra ravichandra 50 Sep 15 05:42 file1
-rw-r--r-- 1 ravichandra ravichandra 72 Sep 15 05:41 file2
-rw-r--r-- 1 ravichandra ravichandra 122 Sep 15 05:46 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -rl
total 20
-rw-r--r-- 1 ravichandra ravichandra 122 Sep 15 05:46 file3
-rw-r--r-- 1 ravichandra ravichandra 72 Sep 15 05:41 file2
-rw-r--r-- 1 ravichandra ravichandra 50 Sep 15 05:42 file1
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:59 dir2
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:58 dir1
ravichandra@DESKTOP-HQRA3L1:~$ ls -t
dir2 dir1 file3 file1 file2
ravichandra@DESKTOP-HQRA3L1:~$ ls -lt
total 20
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:59 dir2
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:58 dir1
-rw-r--r-- 1 ravichandra ravichandra 122 Sep 15 05:46 file3
-rw-r--r-- 1 ravichandra ravichandra 50 Sep 15 05:42 file1
-rw-r--r-- 1 ravichandra ravichandra 72 Sep 15 05:41 file2
```

Description: The ls command in Linux lists the contents of directories, providing information about files and directories. It can display file names, permissions, sizes, and timestamps. By default, ls lists the contents of the current directory in alphabetical order.

Syntax: ls [options] [file/directory]

```
ravichandra@DESKTOP-HQRA3L1:~$ ls -a
. .bash_logout .bashrc .lessht .motd_shown .profile dir1 dir2 file1 file2 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -i
29405 dir1 29407 dir2 29398 file1 29399 file2 29402 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -h
dir1 dir2 file1 file2 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -hl
total 20K
drwxr-xr-x 2 ravichandra ravichandra 4.0K Sep 15 06:58 dir1
drwxr-xr-x 2 ravichandra ravichandra 4.0K Sep 15 06:59 dir2
-rw-r--r-- 1 ravichandra ravichandra 50 Sep 15 05:42 file1
-rw-r--r-- 1 ravichandra ravichandra 72 Sep 15 05:41 file2
-rw-r--r-- 1 ravichandra ravichandra 122 Sep 15 05:46 file3
ravichandra@DESKTOP-HQRA3L1:~$ ls -ltr
total 20
-rw-r--r-- 1 ravichandra ravichandra 72 Sep 15 05:41 file2
-rw-r--r-- 1 ravichandra ravichandra 50 Sep 15 05:42 file1
-rw-r--r-- 1 ravichandra ravichandra 122 Sep 15 05:46 file3
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:58 dir1
drwxr-xr-x 2 ravichandra ravichandra 4096 Sep 15 06:59 dir2
```

-I (Long Format): Displays detailed information about files and directories, including permissions, owner, group, size, and modification time.

-a (All Files): Lists all files, including hidden files that start with a dot.

-t (Sort by Time): Sorts files and directories by their last modification time, displaying the most recently modified ones first.

-r (Reverse Order): Lists files and directories in reverse order.

-S (Sort by Size): Sorts files and directories by their sizes, listing the largest ones first.

-R (Recursive): Lists files and directories recursively, including subdirectories.

-i (Inode): Displays the index number (inode) of each file and directory.

-g (Group): Displays the group ownership of files and directories instead of the owner

```
andra@DESKTOP-HQRA3L1:~$ cp file1 file4
andra@DESKTOP-HQRA3L1:~$ ls
dir2 file1 file2 file3 file4
andra@DESKTOP-HQRA3L1:~$ cat file4
am new to Linux
start

ing data
andra@DESKTOP-HQRA3L1:~$ cp file3 file4 di
andra@DESKTOP-HQRA3L1:~$ ls dir2
file4
andra@DESKTOP-HQRA3L1:~$ cp -r dir2 dir3
andra@DESKTOP-HQRA3L1:~$ ls
dir2 dir3 file1 file2 file3 file4
andra@DESKTOP-HQRA3L1:~$ ls dir3
file4
andra@DESKTOP-HQRA3L1:~$ cp -f file1 file5
andra@DESKTOP-HQRA3L1:~$ ls
dir2 dir3 file1 file2 file3 file4 fi
andra@DESKTOP-HQRA3L1:~$ cat file5
am new to Linux
start

ing data
andra@DESKTOP-HQRA3L1:~$ cp -i file2 file5
Overwrite 'file5'? y
andra@DESKTOP-HQRA3L1:~$ cat file5
ing files and accessing are very easy in li
s cat command
```

cp Command

Description: The cp command in Linux is used for copying files and directories. It duplicates the source file to the specified destination, which can be a file or directory. It offers various options for preserving attributes, renaming, and more.

Syntax: cp [options] source destination

mv Command

```
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir3 dir9 file1 file2 file3 file9
sudhakaratchala@MSI:~$ mv file1 file4
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir3 dir9 file2 file3 file4 file9
sudhakaratchala@MSI:~$ cat file4
hai
sudhakaratchala@MSI:~$ ls dir9
file1 file2
sudhakaratchala@MSI:~$ mv dir9 dir99
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir3 dir99 file2 file3 file4 file9
sudhakaratchala@MSI:~$ ls dir99
file1 file2
sudhakaratchala@MSI:~$ mkdir dir4
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir3 dir4 dir99 file2 file3 file4 file9
sudhakaratchala@MSI:~$ mv file4 file9 dir4
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir3 dir4 dir99 file2 file3
```

```
sudhakaratchala@MSI:~$ mv -i file1 file3
mv: cannot stat 'file1': No such file or directory
sudhakaratchala@MSI:~$ mv -i file2 file3
mv: overwrite 'file3'? y
sudhakaratchala@MSI:~$ mv dir3 dir4
```

Description: The mv command in Linux is used for moving or renaming files and directories. It allows files to be relocated from one location to another within the file system, and it can also be used to rename files.

Syntax: mv [options] source destination

Rename a File: Rename a file by specifying the old and new names. mv old_file_name new_file_name

Interactive Move: Prompt for confirmation before overwriting existing files in the destination directory.

```
mv -i source_file destination_directory/
```

rm Command

```
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir4 dir99 file2 file3 file4
sudhakaratchala@MSI:~$ rm file2 file3
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir4 dir99 file4
sudhakaratchala@MSI:~$ cp file4 file3
sudhakaratchala@MSI:~$ cp file3 file2
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir4 dir99 file2 file3 file4
sudhakaratchala@MSI:~$ rm *
rm: cannot remove 'dir1': Is a directory
rm: cannot remove 'dir2': Is a directory
rm: cannot remove 'dir4': Is a directory
rm: cannot remove 'dir99': Is a directory
sudhakaratchala@MSI:~$ ls
dir1 dir2 dir4 dir99
sudhakaratchala@MSI:~$ rm -r dir1
sudhakaratchala@MSI:~$ ls
dir2 dir4 dir99
sudhakaratchala@MSI:~$ rm -r *
sudhakaratchala@MSI:~$ ls
sudhakaratchala@MSI:~$
```

Description: The rm command in Linux is used to remove (delete) files and directories. It permanently deletes the specified files and directories, and it does not move them to the trash or recycle bin.

Syntax: rm [options] file/directory

Remove a Directory: Delete an empty directory.

```
rm -r directory_name
```

Interactive Remove: Prompt for confirmation before removing each file. rm -i file_name

mkdir Command

```
sudhakaratchala@MSI:~$ ls  
sudhakaratchala@MSI:~$ mkdir dir1  
sudhakaratchala@MSI:~$ ls  
dir1  
sudhakaratchala@MSI:~$ cd dir1  
sudhakaratchala@MSI:~/dir1$ cat > file1  
hai  
hello  
sudhakaratchala@MSI:~/dir1$ ls  
file1  
sudhakaratchala@MSI:~/dir1$ mkdir sub1  
sudhakaratchala@MSI:~/dir1$ ls  
file1 sub1  
sudhakaratchala@MSI:~/dir1$ -
```

Description: The `mkdir` command in Linux is used to create new directories or folders. It allows users to specify the names of the directories they want to create and their location within the file system.

Syntax: `mkdir [options] directory_name...`

```
sudhakaratchala@MSI:~$ ls  
dir1  
sudhakaratchala@MSI:~$ mkdir dir2 dir3 dir4 dir5  
sudhakaratchala@MSI:~$ ls  
dir1 dir2 dir3 dir4 dir5  
sudhakaratchala@MSI:~$ mkdir -p a/b/c/d  
sudhakaratchala@MSI:~$ ls -R  
.:  
a dir1 dir2 dir3 dir4 dir5  
  
.a:  
b  
  
.a/b:  
c  
  
.a/b/c:  
d  
  
.a/b/c/d:  
  
.dir1:
```

rmdir command

```
sudhakaratchala@MSI:~$ ls
a dir1 dir2 dir3 dir4 dir5
sudhakaratchala@MSI:~$ rmdir dir2
sudhakaratchala@MSI:~$ rmdir a
rmdir: failed to remove 'a': Directory not empty
sudhakaratchala@MSI:~$ rmdir dir1
rmdir: failed to remove 'dir1': Directory not empty
sudhakaratchala@MSI:~$ rmdir dir3 dir4 dir5
sudhakaratchala@MSI:~$ ls
a dir1
sudhakaratchala@MSI:~$ rm -r a
sudhakaratchala@MSI:~$ ls
dir1
sudhakaratchala@MSI:~$ rm dir1
rm: cannot remove 'dir1': Is a directory
sudhakaratchala@MSI:~$ rm -r dir1
sudhakaratchala@MSI:~$ ls
sudhakaratchala@MSI:~$ mkdir -p a/b/c
sudhakaratchala@MSI:~$ ls
a
```

```
sudhakaratchala@MSI:~$ mkdir -p a/b/c
sudhakaratchala@MSI:~$ ls
a
sudhakaratchala@MSI:~$ ls -R
.:
a

./a:
b

./a/b:
c

./a/b/c:
sudhakaratchala@MSI:~$ rmdir -p a/b/c
sudhakaratchala@MSI:~$ ls
sudhakaratchala@MSI:~$
```

Description: The `rmdir` command in Linux is used to remove empty directories. It deletes a directory only if it is empty, meaning it contains no files or subdirectories.

Syntax: `rmdir [options] directory_name`

cat Command

```
sudhakaratchala@MSI:~$ cat -n file1
 1  file1 line1
 2  file1 line2
 3
 4  file1 linen
sudhakaratchala@MSI:~$ cat -b file1
 1  file1 line1
 2  file1 line2

 3  file1 linen
sudhakaratchala@MSI:~$ cat file1 file2
file1 line1
file1 line2

file1 linen
file2 line1
file2 line2

file2 linen
```

```
sudhakaratchala@MSI:~$ cat file1 file2 > file4
sudhakaratchala@MSI:~$ ls
file1  file2  file3  file4
sudhakaratchala@MSI:~$ cat file4
file1 line1
file1 line2

file1 linen
file2 line1
file2 line2

file2 linen
```

```
sudhakaratchala@MSI:~$ cat>>file1
hai i am appending some data
sudhakaratchala@MSI:~$ cat file1
file1 line1
file1 line2

file1 linen
hai i am appending some data
sudhakaratchala@MSI:~$
```

Description: The `cat` command in Linux is used to display the contents of a file in the terminal. It can also be used to concatenate and display the contents of multiple files. `cat` is short for "concatenate."

Syntax: `cat [options] [file]`

Append to a File: Append data to an existing file.

```
cat >> file.txt
```

Number Lines: Display line numbers with the file content.

```
cat -n file.txt
```

Number Non-Blank Lines: Number only non-blank lines.

```
cat -b file.txt
```

wc command

```
sudhakaratchala@MSI:~$ cat file
hai hello
how r u
i am fine
sudhakaratchala@MSI:~$ wc file
 3  8 28 file
sudhakaratchala@MSI:~$ wc -l file
3 file
sudhakaratchala@MSI:~$ wc -w file
8 file
sudhakaratchala@MSI:~$ wc -c file
28 file
sudhakaratchala@MSI:~$ wc -lw file
 3  8 file
sudhakaratchala@MSI:~$ wc -wl file
 3  8 file
sudhakaratchala@MSI:~$ wc -cw file
```

Description: The `wc` command in Linux is used to count the number of lines, words, and characters in a text file. It provides information about the file's size and word count, making it useful for analyzing text data.

Syntax: `wc [options] file_name`

Count lines in a file: `wc -l file.txt`

Count words in a file: `wc -w file.txt`

Count characters in a file: `wc -c file.txt`



cd Command

```
sudhakaratchala@MSI:~$ pwd  
/home/sudhakaratchala  
sudhakaratchala@MSI:~$ cd dir1/dir2/dir3/dir4/dir5  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4/dir5$ pwd  
/home/sudhakaratchala/dir1/dir2/dir3/dir4/dir5  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4/dir5$ cd .  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4/dir5$ pwd  
/home/sudhakaratchala/dir1/dir2/dir3/dir4/dir5  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4/dir5$ cd ..  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4$ pwd  
/home/sudhakaratchala/dir1/dir2/dir3/dir4  
sudhakaratchala@MSI:~/dir1/dir2/dir3$ cd ..  
sudhakaratchala@MSI:~/dir1$ pwd  
/home/sudhakaratchala/dir1  
sudhakaratchala@MSI:~/dir1$ cd ..  
sudhakaratchala@MSI:~$
```

```
sudhakaratchala@MSI:/$ cd ~  
sudhakaratchala@MSI:~$ pwd  
/home/sudhakaratchala  
sudhakaratchala@MSI:~$ cd dir1/dir2/dir3/dir4/dir5  
sudhakaratchala@MSI:~/dir1/dir2/dir3/dir4/dir5$ cd ~  
sudhakaratchala@MSI:~$ pwd  
/home/sudhakaratchala  
sudhakaratchala@MSI:~$ cd dir1/dir2/dir3  
sudhakaratchala@MSI:~/dir1/dir2/dir3$ pwd  
/home/sudhakaratchala/dir1/dir2/dir3
```

Description: The `cd` command in Linux is used for changing the current working directory in the command-line interface. It allows users to navigate through the directory structure to access and work with files and directories.

Syntax: `cd [directory_path]`

Move Up One Directory Level:

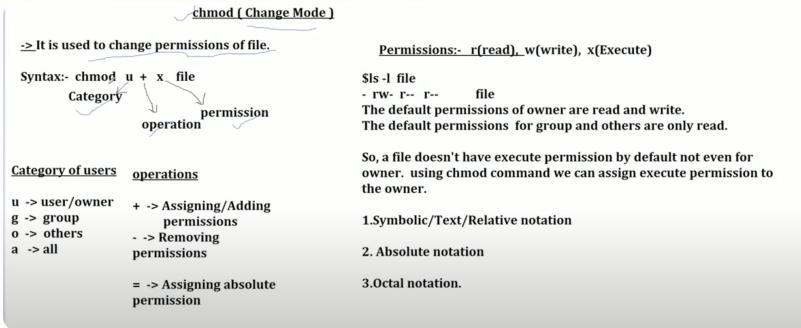
```
cd..
```

Move to the Previous Directory: `cd-`

Navigate Using Relative Paths:

```
cd ../another_directory
```

chmod Command



Description: The chmod command in Linux is used to change file permissions, allowing you to control who can read, write, or execute a file. It can modify permissions for the owner, group, and others.

Syntax: `chmod [options] mode file`

```
sudhakaratchala@MSI:~$ ls -l
total 4
-rw-r--r-- 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
-rw-r--r-- 1 sudhakaratchala sudhakaratchala 10 Jun 21 00:05 hai
-rw-r--r-- 1 sudhakaratchala sudhakaratchala 16 Jun 21 00:05 hello
sudhakaratchala@MSI:~$ ls -l file9
-rw-r--r-- 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
sudhakaratchala@MSI:~$ chmod u+x file9
sudhakaratchala@MSI:~$ ls -l file9
-rwxr--r-- 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
sudhakaratchala@MSI:~$ chmod u-x file9
sudhakaratchala@MSI:~$ ls -l file9
-rw-r--r-- 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
sudhakaratchala@MSI:~$ chmod ugo+x file9
sudhakaratchala@MSI:~$ ls -l file9
-rwxr-xr-x 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
sudhakaratchala@MSI:~$ chmod a=w file9
sudhakaratchala@MSI:~$ ls -l file9
--w--w--w- 1 sudhakaratchala sudhakaratchala 692 Jun 22 13:23 file9
sudhakaratchala@MSI:~$
```

Conclusion

These are some of the essential Linux file handling commands that every sysadmin should know. With these tools' help, you can navigate through files, rename and copy them, view their contents, and locate files and directories. Explore each command and get comfortable with it, and you'll be a pro in no time.

```
/Users/yuu/.pyenv/version)  
5.0
```

```
on
```

```
jupyter-2.6.0
```

```
on
```

```
5ee98b69288471b0fcf2e0ede82ce5209eb90b, Jun 01  
GCC 4.9.2]
```

```
treasuredata/jupyter
```

```
0 (set by /Volumes/treasuredata/.python-version)
```

```
on
```

```
Continuum Analytics, Inc.
```