

```
mayank@mayank-Vostro-3446: /media/mayank
mayank@mayank-Vostro-3446:/media$ sudo mount /dev/sda3 /media/mayank
mayank@mayank-Vostro-3446:/media$ cd mayank/
mayank@mayank-Vostro-3446:/media/mayank$ ls -l
total 5084
drwxrwxrwx 1 root root 4096 Aug 23 23:20 131026
-rw-rw-rwx 1 root root 4502143 Aug 23 23:22 131026.zip
drwxrwxrwx 1 root root 4096 Jan 5 2015 7007-0ff1301519c50505f12d
-rw-rw-rwx 1 root root 18912 Jun 10 04:58 config.bin
drwxrwxrwx 1 root root 8192 Aug 28 11:41 Entertainment
drwxrwxrwx 1 root root 0 Mar 9 21:06 Games
drwxrwxrwx 1 root root 0 Aug 19 2014 GREYC'sLab
drwxrwxrwx 1 root root 4096 Jul 28 21:08 Sem-1
drwxrwxrwx 1 root root 0 Jul 28 16:19 Sem-2
drwxrwxrwx 1 root root 4096 Jul 25 2015 Sem-3
drwxrwxrwx 1 root root 4096 May 19 08:21 Sem-4
drwxrwxrwx 1 root root 4096 Sep 7 18:08 Sem-5
-rw-rw-rwx 1 root root 312486 Mar 4 2015 Sets.pdf
drwxrwxrwx 1 root root 4096 Aug 10 23:02 Software
drwxrwxrwx 1 root root 4096 Jul 20 10:42 System
drwxrwxrwx 1 root root 4096 Sep 7 19:29 Tools
drwxrwxrwx 1 root root 0 Aug 19 2014 Utilities
drwxrwxrwx 1 root root 4096 Aug 23 07:36 vcredist
-rw-rw-rwx 2 root root 311050 Jan 5 2015 vcredist-MSI_vc_red.msi.txt
mayank@mayank-Vostro-3446:/media/mayank$
```

Mastering File Handling in Linux

If you're new to Linux or just looking to refresh your file handling skills, this guide is for you. Below are 22 powerful and versatile commands to help you navigate and manipulate your files and directories with ease.



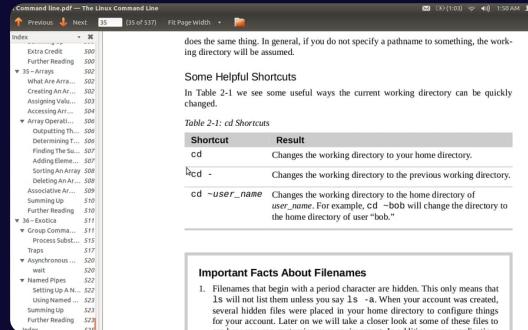
by Sai Rohith

List and Navigate with ls, pwd, and cd

```
Administrator:~$ ls
dbus-uuidgen  javac  mt-gnu  pidof  sync
dd           javadoc  mv     ping  tailf
df           javadoc  nano   ping6  tar
dir          kbd_mode nc    Plymouth  true
dmesg        kill    netcat ps    tempfi
dnsdomainname knod   netstat pwd   touch
dumpkeys      less   nroff  ralink  unctrl
echo         lesscho nroff_laname rbash  unctrl
ed           lesskey  nroff_pg  remlink  unctrl
egrep        lesspipe ntfs-3g_probe red   unctrl
false        ln     ntfsck  running-in-container which
fgconsole    loadkeys ntfscluster run-parts yptab
grep         login   ntfscluster  set  zmodload
findmnt      loginctl ntfscluster run-parts which
fusermount   ls     ntfsdump_logfile setfacl  zcat
getfact      lsblk  ntfsfix  setfont  zcmp
grep         lsmod  ntfsinfo  setupcon  zdiff
gunzip       mkdir  ntftsls  sh    zgrep
gzexe        nknod  ntfsntalloc sh_distrib  zfgrep
gzip         nktemp  ntfsmove  sleep   zforce
hostname    more   ntfstruncate ss    zgrep
ip           mount  ntfswipe static-sh  zless
java        mountpoint open   stty   zmore
                     mt   openvt  su    znew
```

List Files and Directories

The **ls** command is your go-to tool for listing the contents of a directory. Use options like **-a**, **-l**, and **-h** to customize the output.



Print Working Directory

Need to know which directory you're currently in? The **pwd** command will show you the full path of your current working directory.

```
Administrator:~$ pwd
/home/antonius
Administrator:~$ cd /home/santos/.local/share/Trash/expunged
Administrator:~$ ls -l
total 12
wx----- 2 santos santos 4096 Mar 21 21:18 expunged
wx----- 2 santos santos 4096 Mar 21 21:18 files
wx----- 2 santos santos 4096 Mar 21 21:18 hidden
Administrator:~$ chmod -R ugrw+expunged .
Administrator:~$ cd /home/santos/.local/share/Trash/expunged
Administrator:~$ ls -l
total 12
wx----- 2 santos santos 4096 Mar 21 21:18 expunged
wx----- 2 santos santos 4096 Mar 21 21:18 files
wx----- 2 santos santos 4096 Mar 21 21:18 hidden
Administrator:~$ chmod -R ugrw+expunged .
Administrator:~$ cd /home/santos/.local/share/Trash/expunged/2873016728
Administrator:~$ ls -l
total 36
wxr--r-- 3 santos santos 4096 Mar 21 21:27 3062667192
wxr--r-- 3 santos santos 14348 Mar 21 21:18 395584556
Administrator:~$ chmod -R ugrw+expunged 2873016728
Administrator:~$ ls -l
total 36
wxr--r-- 3 santos santos 4096 Mar 21 21:27 3062667192
wxr--r-- 36 santos santos 14348 Mar 21 21:18 395584556
wxr--r-- 36 santos santos 4096 Mar 21 21:18 406783164
wxr--r-- 36 santos santos 4096 Mar 21 21:18 839558455
wxr--r-- 3 santos santos 4096 Mar 21 15:21 899331308
Administrator:~$ chmod -R ugrw+expunged 2873016728/395584556
Administrator:~$ ls -l
total 36
wxr--r-- 3 santos santos 4096 Mar 21 21:27 3062667192
wxr--r-- 36 santos santos 14348 Mar 21 21:18 395584556
Administrator:~$ chmod -R ugrw+expunged 2873016728/395584556
```

Change Your Directory

To move around in the file system, use the **cd** command followed by the path of the target directory. Use **".."** to go back to the parent directory.

Create and Delete Directories with `mkdir` and `rmdir`

Create a New Directory

Use the `mkdir` command to create a new directory with the specified name.

Remove an Empty Directory

If you need to delete a directory, use the `rmdir` command. Note that this command only works on empty directories.

The Power of Directories

Directories (folders) are a fundamental part of the file system. They provide a way to organize your files and keep your system tidy. A good folder structure can make your life much easier!



Made with Gamma

All Linux Commands Using **ls**

In addition to listing files and directories, the **ls** command has many powerful options that allow you to filter, sort, and customize your output. Here are some useful commands:

- **ls -a**: list all files, including hidden ones
- **ls -l**: list files in long format
- **ls -h**: list files with human-readable file sizes
- **ls -t**: list files by modification time
- **ls -S**: list files by size
- **ls -r**: list files in reverse order
- **ls -R**: list files recursively

Use these commands to take your file management skills to the next level.

LS COMMANDS

```
sai@DESKTOP-S5AQR80:~$ ls -b
ravi rc
sai@DESKTOP-S5AQR80:~$ ls
ravi rc
sai@DESKTOP-S5AQR80:~$ ls -n
total 8
-rw-r-xrw- 1 1000 1000 218 Sep 15 12:14 ravi
drwxr-xr-x 2 1000 1000 4096 Sep 15 10:47 rc
sai@DESKTOP-S5AQR80:~$ ls -a
. . . bash_logout .bashrc .motd_shown .profile ravi rc
sai@DESKTOP-S5AQR80:~$ ls -r
rc ravi
sai@DESKTOP-S5AQR80:~$ ls -Ls
total 8
4 ravi 4 rc
sai@DESKTOP-S5AQR80:~$ man man
sai@DESKTOP-S5AQR80:~$ ls -l
total 8
-rw-r-xrw- 1 sai sai 218 Sep 15 12:14 ravi
drwxr-xr-x 2 sai sai 4096 Sep 15 10:47 rc
sai@DESKTOP-S5AQR80:~$ ls -R
.:
ravi rc

./rc:
sai@DESKTOP-S5AQR80:~$ ls -lt
total 8
-rw-r-xrw- 1 sai sai 218 Sep 15 12:14 ravi
drwxr-xr-x 2 sai sai 4096 Sep 15 10:47 rc
sai@DESKTOP-S5AQR80:~$ ls -ltr
total 8
drwxr-xr-x 2 sai sai 4096 Sep 15 10:47 rc
-rw-r-xrw- 1 sai sai 218 Sep 15 12:14 ravi
```

Description: The `ls` command in Linux is used to list files and directories in the current directory or a specified location. It provides information like file names, sizes, permissions, and timestamps. It is a fundamental tool for navigating and managing files in a terminal.

Syntax: `ls [OPTION]... [FILE]...`

- `[OPTION]`: Various options for customizing the output.
- `[FILE]...`: Optional, specify one or more file or directory names to list their contents.



Create and Manipulate Files with touch, cp, mv, and rm

Copy or Move Files

The **cp** and **mv** commands let you copy and move files and directories as needed. These commands can be customized with various options to suit your needs.

1

Create a New File

The **touch** command can be used to create a new, empty file. It can also be used to update the timestamps of an existing file.

2

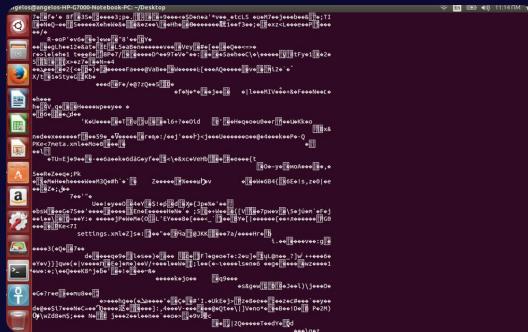
Delete Files You Don't Need

Sometimes you need to clean up your directory and get rid of files you no longer need. The **rm** command can help you do just that.

3

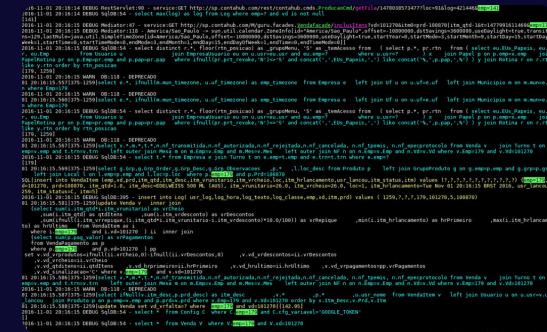


View File Contents with cat, more, less, head, and tail



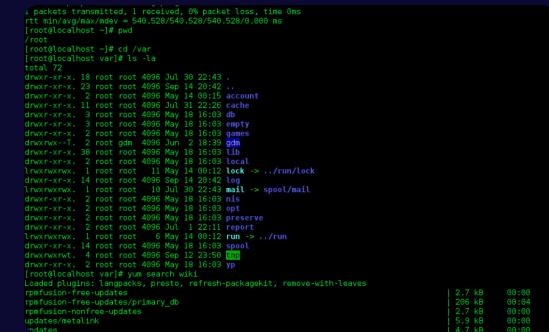
Display File Contents

The **cat** command is a quick way to display the contents of a file. For longer files, try using the **more** or **less** command to view the contents one screen at a time.



View the Beginning or End of a File

If you only need to see the beginning or end of a file, save yourself some time and use the **head** or **tail** command.



The Power of Piping

One of the strengths of Linux is its ability to combine commands using pipes. With pipes, you can take the output of one command and use it as input to another command. This technique can be incredibly powerful and save you a ton of time.

Manage Permissions and Ownership with chmod and chown

Change File Permissions

The **chmod** command lets you change the read, write, and execute permissions on a file. You can modify permissions for the owner, group, and others. Experiment with different permissions to find the right configuration for your needs.

- Read (**r**): allows you to see the contents of the file
- Write (**w**): allows you to modify the contents of the file
- Execute (**x**): allows you to run the file as a program

Change File Ownership

The **chown** command lets you change the ownership of a file. You can set the owner and group, as well as a few other attributes. This is useful if you need to give someone else access to a file or take ownership of someone else's file.

Remember that changing ownership can be dangerous if you don't know what you're doing. Be careful!

Change File Permissions with chmod

The **chmod** command is used to change file permissions in Linux. Use numbers or letters to specify the permissions. Here are some useful commands:

- **chmod u+x FILE**: add execute permission for the owner
- **chmod g+w FILE**: add write permission for the group
- **chmod o-r FILE**: remove read permission for others
- **chmod 755 FILE**: set read, write, and execute permission for the owner, and read and execute permission for others
- **chmod -R u=rwX,go=rX DIRECTORY**: set permissions recursively for a directory and its contents

Use these commands to control access to your files and directories.

Commands

```
sai@DESKTOP-S5AQR80:~$ chmod u+x ravi
sai@DESKTOP-S5AQR80:~$ ls
ravi rc
sai@DESKTOP-S5AQR80:~$ ls -l ravi
-rwxr--r-- 1 sai sai 61 Sep 15 10:50 ravi
sai@DESKTOP-S5AQR80:~$ chmod g+x,o+w+x ravi
sai@DESKTOP-S5AQR80:~$ ls
ravi rc
sai@DESKTOP-S5AQR80:~$ ls -l ravi
-rwxr-xrwx 1 sai sai 61 Sep 15 10:50 ravi
sai@DESKTOP-S5AQR80:~$ chmod g+x,o+x ravi
chmod: invalid mode: 'g+x,ox'
Try 'chmod --help' for more information.
sai@DESKTOP-S5AQR80:~$ chmod g+x,o-x ravi
sai@DESKTOP-S5AQR80:~$ ls -l ravi
-rwrxr-xrw- 1 sai sai 61 Sep 15 10:50 ravi
```

CHMOD

Description:

The **chmod** command in Linux changes file permissions, allowing users to control read, write, and execute access to files and directories.

Syntax: `chmod <permissions> <file_name>`

```
sai@DESKTOP-S5AQR80:~$ cat >> ravi
iam in my 3rd year

i love coding
my nick name is chinku kamichetty
sai@DESKTOP-S5AQR80:~$ cat -n ravi
1 hello
2 iam ravi chandra
3 from aids
4 iam the topper of our class
5 i hold 2 major projects
6 and published a paper on AI
7 builded a application called RCFLIX
8 iam in my 3rd year
9
10 i love coding
11 my nick name is chinku kamichetty
sai@DESKTOP-S5AQR80:~$ cat -b ravi
1 hello
2 iam ravi chandra
3 from aids
4 iam the topper of our class
5 i hold 2 major projects
6 and published a paper on AI
7 builded a application called RCFLIX
8 iam in my 3rd year
9
10 my nick name is chinku kamichetty
```

CAT

Description: The **cat** command in Linux is used to concatenate and display the contents of one or more files. It's commonly used for reading file content and displaying it in the terminal.

Syntax: `cat [OPTION]... [FILE]...`

Find Files and Directories with `find` and `grep`

1 Search for Files

The `find` command is your go-to tool for finding files and directories. You can search based on name, size, permission, and other criteria. This command can be a bit complex, so make sure to consult the manual pages for help if you get stuck.

2 Search for Text Patterns

The `grep` command is a text search tool that lets you search for text patterns within files. This command is incredibly powerful and can save you a ton of time if you need to find a specific piece of information buried deep within a file.

3 Avoid Duplicate Searches

If you find yourself performing the same search frequently, save yourself some time by using history and aliases. The `history` command shows you a list of your past commands, while an `alias` is a shortcut for a longer command. These tools can be customized and are incredibly useful once you get the hang of them.

```
/eggs/" --*-  
from zope.interface import *  
from zope.component import *  
  
from plone.portlets.interfaces import IPortletDataProvider  
from plone.app.portlets import interfaces as portlets  
from zope import schema  
from zope.formlib import form  
from plone.memoize import memoize  
  
from Acquisition import *  
from Products.Five.browser import View  
from Products.CMFCore import utils  
from Products.CMFFormGen import Assignment  
from Products.CMFPlone import ILoginPortlet  
  
class ILoginPortlet(IPortletDataProvider):  
    """A portlet which displays user information.  
    """  
  
    class Assignment(base.Renderer):  
        implements(ILoginPortlet)  
  
        title = _(u'label')  
  
        class Renderer(base.Renderer):  
            def __init__(self, base.Renderer):  
                self.membership = base.membership  
                self.context = base.context  
                self.portal_skins = base.portal_skins  
                self.pas_info = base.pas_info  
  
            def show(self):  
                if not self.portal_skins:  
                    return False  
                if not self.pas_info:  
                    return False  
  
                # ...  
  
                return True  
  
    @memoize.memoize()  
    def update(self):  
        membership = self.membership  
        context = self.context  
        portal_skins = self.portal_skins  
        pas_info = self.pas_info  
  
        # ...  
  
        self.data = {  
            'user': user,  
            'url': url,  
            'group': group,  
            'groups': groups,  
            'skin': skin,  
            'skins': skins,  
            'pas': pas,  
            'pas_info': pas_info,  
            'portal_skins': portal_skins,  
            'context': context,  
            'membership': membership,  
            'url': url  
        }  
  
        return self.data
```