Basic Linux Commands

Note: If we are running our commands in our personal username use sudo before all commands
The **sudo command** allows you to **run** programs with the security privileges of another user (by default, as the superuser). It prompts you for your personal password and confirms your request to execute a **command** by checking a file, called **sudoers**, which the system administrator configures.

Creating empty file: (touch)

$ touch "filename"

```
[ankireddy@localhost root]$ sudo touch abc.txt
[ankireddy@localhost root]$ sudo ls
abc.txt
```

```
[ankireddy@localhost root]$ sudo touch abcd.txt abcdefg.txt
[ankireddy@localhost root]$ sudo ls
abcdefg.txt  abcd.txt  abc.txt
```

Listing files: (ls)

$ ls

This command lists all the files present in the system

```
[ankireddy@localhost root]$ sudo ls
abcdefg.txt  abcd.txt  abc.txt
```

$ ls -R

'ls -R' to shows all the files not only in directories but also subdirectories

```
[ankireddy@localhost root]$ sudo ls -R
.:
abcdefg.txt  abcd.txt  abc.txt
```

NOTE: The command is case-sensitive. If you enter, "**ls - r**" you will get an error.

'ls -al' gives detailed information of the files. The command provides information in a columnar format. The columns contain the following information:

```
[ankireddy@localhost root]$ sudo ls -al
total 24
dr-xr-x---.  2 root root 162 Jul  9 09:16 .
dr-xr-xr-x. 17 root root 224 May  2  2018 ..
-rw-r--r--.  1 root root   0 Jul  9 09:16 abcdefg.txt
-rw-r--r--.  1 root root   0 Jul  9 09:16 abcd.txt
-rw-r--r--.  1 root root   0 Jul  9 09:02 abc.txt
-rw-------.  1 root root 157 Jul  8 12:58 .bash_history
-rw-r--r--.  1 root root  18 Dec 28  2013 .bash_logout
-rw-r--r--.  1 root root 176 Dec 28  2013 .bash_profile
-rw-r--r--.  1 root root 176 Dec 28  2013 .bashrc
-rw-r--r--.  1 root root 100 Dec 28  2013 .cshrc
-rw-r--r--.  1 root root 129 Dec 28  2013 .tcshrc
```

'ls -a' gives detailed information of the hidden files.

```
[ankireddy@localhost root]$ sudo ls -a
.  ..  abcdefg.txt  abcd.txt  abc.txt  .bash_history  .bash_logout  .bash_profile  .bashrc  .cshrc  .tcshrc
```

Deleting files:

$ rm filename

```
[ankireddy@localhost root]$ sudo ls
abcdefg.txt  abcd.txt  abc.txt
[ankireddy@localhost root]$ sudo rm abcd.txt
[ankireddy@localhost root]$ sudo ls
abcdefg.txt  abc.txt
```

Making Directories:

$ mkdir directoryname creates directory

$ rmdir directoryname removes directory

Let's see how it works

```
[ankireddy@localhost root]$ sudo mkdir ankireddy
[ankireddy@localhost root]$ sudo mkdir ashwini
[ankireddy@localhost root]$ sudo ls
ankireddy  ashwini
```

```
[ankireddy@localhost root]$ sudo rmdir ankireddy
rmdir: failed to remove 'ankireddy': Directory not empty
```

```
[ankireddy@localhost root]$ sudo rmdir ankireddy/MUSIC
[ankireddy@localhost root]$ sudo ls
ankireddy  ashwini
[ankireddy@localhost root]$ sudo rmdir ankireddy
[ankireddy@localhost root]$ sudo ls
ashwini
```

Renaming Directory:

The 'mv' (move) command can be used for renaming directories. Use the below-given format:

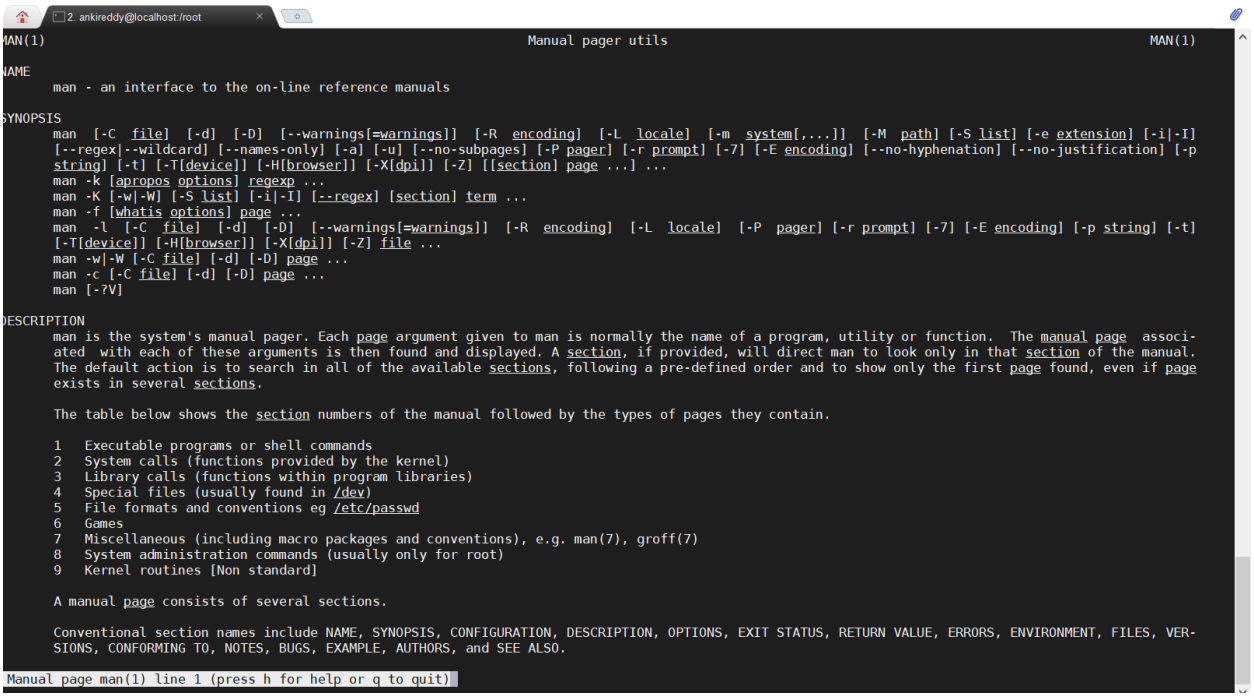
$ mv directoryname newdirectoryname

```
[ankireddy@localhost root]$ sudo ls
ankireddy
[ankireddy@localhost root]$ sudo mv ankireddy ankireddy1
[ankireddy@localhost root]$ sudo ls
ankireddy1
```

The ‘man’ command:

To get help on any command that you do not understand, you can type

```
[ankireddy@localhost root]$ man
What manual page do you want?
[ankireddy@localhost root]$ man man
```



The History command:

History command shows all the commands that you have used in the past for the current terminal session. This can help you refer to the old commands you have entered and re-used them in your operations again.

```
[ankireddy@localhost root]$ history
    1  sudo ls -lrt /root
    2  usermod -a -G wheel ankireddy
    3  exit
    4  sudo ls -lrt /root
    5  sudo ls -la /root
    6  exit
    7  sudo ls -la /root
    8  ls
    9  ls -lr
   10  ls
   11  sudo ls
   12  sudo touch abc.txt
   13  sudo ls
   14  sudo touch abcd.txt abcdefg.txt
   15  sudo ls
   16  sudo ls -R
   17  sudo ls -al
   18  sudo ls -a
   19  cat sample1
   20  sudo cat sample 1
   21  sudo cat sample1
   22  sudo cat abc.txt
   23  sudo cat abcd.txt
   24  sudo cat abc.txt abcd.txt > combined_abc.txt
   25  sudo cat abc.txt abcd.txt > combined.txt
   26  sudo rm abcd.txt
   27  sudo ls
   28  sudo touch abcd.txt
   29  sudo ls
   30  sudo rm abcd.txt
   31  sudo ls
   32  mkdir ankireddy
   33  sudo mkdir ankireddy
   34  sudo ls
   35  sudo mkdir/temp/MUSIC
   36  sudo mkdir /temp/MUSIC
```

Clear Command:

```
[ankireddy@localhost root]$ clear
```

```
2. ankireddy@localhost:/root
[ankireddy@localhost root]$
```

Linux head command:

head command syntax:

head [option] [filename]…[filename]

Using option in ‘head’ command is optional. You can apply ‘head’ command for one or more files.

Example – 1: 'head' command without any option

products.txt file has 11 lines with heading. The following command will display the first 10 lines of products.txt file because no option is used with 'head' command.

$ head products.txt

```
ubuntu@ubuntu-VirtualBox:~/code$ head products.txt
ID      Type            Brand           Size            Price
01      HDD             Samsung         1TB             $70
02      Monitor         DELL            15"             $60
03      Mouse           A4              N/A             $04
04      Keyboard        Atech           Normal          $10
05      Scanner         HP              N/A             $50
06      Printer         Samsung         N/A             $100
07      Adapter         A4              N/A             $10
08      Monitor         Samsung         17"             $90
09      HDD             Toshiba         500GB           $45
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2: 'head' command with -n option and positive value

'-n' option with 5 is used in the following 'head' command. The first five lines of products.txt file will be shown in the output.

$ head -n 5 products.txt

```
ubuntu@ubuntu-VirtualBox:~/code$ head -n 5 products.txt
ID      Type            Brand           Size            Price
01      HDD             Samsung         1TB             $70
02      Monitor         DELL            15"             $60
03      Mouse           A4              N/A             $04
04      Keyboard        Atech           Normal          $10
ubuntu@ubuntu-VirtualBox:~/code$
```

Linux Tail Command:

Use of 'tail' command:

By default, **'tail'** command reads last 10 lines of the file. If you want to read more or less than 10 lines from the ending of the file then you have to use '-n' option with **'tail'** command.

tail command syntax:

tail [option] [filename]…[filename]

Example – 1: 'tail' command without any option

employee.txt file has only 6 lines which is less than 10. So, the following command will display the full content of employee.txt file.

$ tail employee.txt

```
ubuntu@ubuntu-VirtualBox:~/code$ tail employee.txt
ID      Name            Department              Post
S001    John Paul       Sales                   Marketing Officer
S002    Willliam Bobv   Sales                   Sales Reprentative
E003    Jason           HR                      Manager
E004    Jullie          HR                      Assistant Manager
E005    Jannifer        HR                      Programmer
ubuntu@ubuntu-VirtualBox:~/code$
```

Example – 2: 'tail' command with -n option and positive value

When you want to read particular lines from the ending of the file then you have to use '-n' option with positive value. The following command will display the last 2 lines of employee.txt file.

$ tail -n 2 employee.txt

```
ubuntu@ubuntu-VirtualBox:~/code$ tail -n 2 employee.txt
E004    Jullie          HR                      Assistant Manager
E005    Jannifer        HR                      Programmer
ubuntu@ubuntu-VirtualBox:~/code$
```

Printing in Linux:

'pr' command

This command helps in formatting the file for printing on the terminal. There are many options available with this command which help in making desired format changes on file. The most used '**pr'** options are listed below.

Option	Function
-x	Divides the data into 'x' columns
-h "header"	Assigns "header" value as the report header
-t	Does not print the header and top/bottom margins
-d	Double spaces the output file
-n	Denotes all line with numbers
-l page length	Defines the lines (page length) in a page. Default is 56
-o margin	Formats the page by the margin number

Linux Commands List:

Below is a Cheat Sheet of Linux commands we have learned in this tutorial

Command	Description
ls	Lists all files and directories in the present working directory
ls - R	Lists files in sub-directories as well
ls - a	Lists hidden files as well
ls - al	Lists files and directories with detailed information like permissions, size, owner, etc.
cat > filename	Creates a new file
cat file1 file2 > file3	Joins two files (file1, file2) and stores the output in a new file (file3)
mv file "new file path"	Moves the files to the new location
mv filename new_file_name	Renames the file to a new filename
sudo	Allows regular users to run programs with the security privileges of the superuser or root

rm filename	Deletes a file
man	Gives help information on a command
history	Gives a list of all past commands typed in the current terminal session
clear	Clears the terminal
mkdir directoryname	Creates a new directory in the present working directory or a at the specified path
rmdir	Deletes a directory
mv	Renames a directory
pr -x	Divides the file into x columns
pr -h	Assigns a header to the file
pr -n	Denotes the file with Line Numbers

lp -nc lpr c	Prints "c" copies of the File
lp -d lp -P	Specifies name of the printer
apt-get	Command used to install and update packages
mail -s 'subject' -c 'cc-address' -b 'bcc-address' 'to-address'	Command to send email
mail -s "Subject" to-address < Filename	Command to send email with attachment

makeuseof.com

Unix/Linux Command Reference

File commands

ls	Directory listing
ls -al	Formatted listing with hidden files
cd dir	Change directory to dir
cd	Change to home
pwd	Show current directory
mkdir dir	Create a directory dir
rm file	Delete file
rm -r dir	Delete directory dir
rm -f file	Force remove fire
rm -rf dir	For remove directory dir
cp file1 file2	Copy file1 to file2
cp -r dir1 dir2	Copy dir1 to dir2; create dir2 if it doesn't exit
mv file1 file2	Rename or move file1 to file2. If tiled 2 is an existing directory, moves file1 into directory file2
ln -s file link	Create symbolic link link to file
touch file	Create or update file
cat > file	Places standard input into file
more file	Output the contents of file
head file	Output the firest 10 lines of file
tail file	Output the last 10 lines of file
tail -f file	Output the contents of file as it grows, starting with the last 10 lines

Process Management

ps	display all currently active processes
top	display all running processes
kill pid	kill process id pid
killall proc	kill all processes named proc *
bg	lists stopped or background jobs; resume a stopped job in the background
fg	Brings the most recent job to the foreground
fg a	brings job a to the foreground

File Permissions

chmod octal file	change the permissions of file to octal, which can be found separately for user, group, and world by adding: • 4 – read (r) • 2 – write (w) • 1 – execute (x) **Examples:** ***chmod 777*** *– read, write, execute for all* ***chmod 755*** *– rwx for owner, rx for group and world. For more options, see* **man chmod.**

SSH

ssh user@host	connect to host as user
ssh -p port user@host	connect to host on port port as user
ssh-copy-id user@host	add your key to host for user to enable a keyed or passwordless login

Searching

grep pattern files	search for pattern in files
grep -r pattern dir	search recursively for pattern in dir
command | grep pattern	search for pattern in the output of command
locate file	find all instances of file

System Info

date	show the current date and time
cal	show this month's calendar
uptime	show current uptime
w	display who is online
whoami	who you are logged in as
finger user	display information about user
uname -a	show kernel information
cat /proc /cpuinfo	cpu information
cat /proc /meminfo	memory information
man command	show the manual for command
df	show disk usage
du	show directory space usage
free	show memory and swap usage
whereis app	show possible locations of app
which app	show which app will be run by default

Compression

tar cf file.tar files	create a tar named file.tar containing files
tar xf file.tar	extract the files from file.tar
tar czf file.tar.gz files	create a tar with Gzip compression
tar xzf file.tar.gz	extract a tar using Gzip
tar cjf file.tar.bz2	create a tar with Bzip2 compression
tar xjf file.tar.bz2	extract a tar using Bzip2
gzip file	compresses file and renames it to file.gz
gzip -d file.gz	decompresses file.gz back to file

Network

ping host	ping host and output results
whois domain	get whois information for domain
dig domain	get DNS information for domain
dig -x host	reverse lookup host
wget file	download file
wget -c file	continue a stopped download

Installation

Install from source:	
./configure	
make	
make install	
dpkg -i pkg.deb	install a package (Debian)
rpm -Uvh pkg.rpm	install a package (RPM)

Shortcuts

Ctrl+C	halts the current command
Ctrl+Z	stops the current command, resume with fg in the foreground or bg in the background
Ctrl+D	log out of current session, similar to exit
Ctrl+W	erases one word in the current line
Ctrl+U	erases the whole line
Ctrl+R	type to bring up a recent command
!!	repeats the last command
exit	log out of current session
*	use with extreme caution

Unix/Linux Command Reference

FOSSwire.com

File Commands

ls – directory listing
ls -al – formatted listing with hidden files
cd *dir* - change directory to *dir*
cd – change to home
pwd – show current directory
mkdir *dir* – create a directory *dir*
rm *file* – delete *file*
rm -r *dir* – delete directory *dir*
rm -f *file* – force remove *file*
rm -rf *dir* – force remove directory *dir* *
cp *file1 file2* – copy *file1* to *file2*
cp -r *dir1 dir2* – copy *dir1* to *dir2*; create *dir2* if it doesn't exist
mv *file1 file2* – rename or move *file1* to *file2*
if *file2* is an existing directory, moves *file1* into directory *file2*
ln -s *file link* – create symbolic link *link* to *file*
touch *file* – create or update *file*
cat > *file* – places standard input into *file*
more *file* – output the contents of *file*
head *file* – output the first 10 lines of *file*
tail *file* – output the last 10 lines of *file*
tail -f *file* – output the contents of *file* as it grows, starting with the last 10 lines

Process Management

ps – display your currently active processes
top – display all running processes
kill *pid* – kill process id *pid*
killall *proc* – kill all processes named *proc* *
bg – lists stopped or background jobs; resume a stopped job in the background
fg – brings the most recent job to foreground
fg *n* – brings job *n* to the foreground

File Permissions

chmod *octal file* – change the permissions of *file* to *octal*, which can be found separately for user, group, and world by adding:

- 4 – read (r)
- 2 – write (w)
- 1 – execute (x)

Examples:
chmod 777 – read, write, execute for all
chmod 755 – rwx for owner, rx for group and world
For more options, see **man chmod**.

SSH

ssh *user@host* – connect to *host* as *user*
ssh -p *port user@host* – connect to *host* on port *port* as *user*
ssh-copy-id *user@host* – add your key to *host* for *user* to enable a keyed or passwordless login

Searching

grep *pattern files* – search for *pattern* in *files*
grep -r *pattern dir* – search recursively for *pattern* in *dir*
command **|** **grep *pattern*** – search for *pattern* in the output of *command*
locate *file* – find all instances of *file*

System Info

date – show the current date and time
cal – show this month's calendar
uptime – show current uptime
w – display who is online
whoami – who you are logged in as
finger *user* – display information about *user*
uname -a – show kernel information
cat /proc/cpuinfo – cpu information
cat /proc/meminfo – memory information
man *command* – show the manual for *command*
df – show disk usage
du – show directory space usage
free – show memory and swap usage
whereis *app* – show possible locations of *app*
which *app* – show which *app* will be run by default

Compression

tar cf *file.tar files* – create a tar named *file.tar* containing *files*
tar xf *file.tar* – extract the files from *file.tar*
tar czf *file.tar.gz files* – create a tar with Gzip compression
tar xzf *file.tar.gz* – extract a tar using Gzip
tar cjf *file.tar.bz2* – create a tar with Bzip2 compression
tar xjf *file.tar.bz2* – extract a tar using Bzip2
gzip *file* – compresses *file* and renames it to *file.gz*
gzip -d *file.gz* – decompresses *file.gz* back to *file*

Network

ping *host* – ping *host* and output results
whois *domain* – get whois information for *domain*
dig *domain* – get DNS information for *domain*
dig -x *host* – reverse lookup *host*
wget *file* – download *file*
wget -c *file* – continue a stopped download

Installation

Install from source:
./configure
make
make install
dpkg -i *pkg.deb* – install a package (Debian)
rpm -Uvh *pkg.rpm* – install a package (RPM)

Shortcuts

Ctrl+C – halts the current command
Ctrl+Z – stops the current command, resume with **fg** in the foreground or **bg** in the background
Ctrl+D – log out of current session, similar to **exit**
Ctrl+W – erases one word in the current line
Ctrl+U – erases the whole line
Ctrl+R – type to bring up a recent command
!! - repeats the last command
exit – log out of current session

* use with extreme caution.