**Linux Command**

**Ctrl**+**A** Move cursor to the beginning of the command line.

**Ctrl**+**C** End a running program and return the prompt,

**Ctrl**+**D** Log out of the current shell session, equal to typing **exit** or **logout**.

**Ctrl**+**E** Move cursor to the end of the command line.

**Ctrl**+**H** Generate backspace character.

**Ctrl**+**L** Clear this terminal.

**Ctrl**+**R** Search command history,

**Ctrl**+**Z** Suspend a program,

**Ctrl**+**W** Erase one word in the current line

**!!** Repeat the Last Command

**Exit** log out of current session

**ArrowLeft** and **ArrowRight** Move the cursor one place to the left or right on the command line, so that you caninsert characters at other places than just at the beginning and the end.

**ArrowUp** and**ArrowDown** Browse history. Go to the line that you want to repeat, edit details if necessary,and press **Enter** to save time.

**Shift**+**PageUp** and**Shift**+**PageDown** Browse terminal buffer (to see text that has "scrolled off" the screen).

**echo $SHELL Which shell**

**echo $HOME Home directory**

· a partition for user programs (*/usr*)

· a partition containing the users' personal data (*/home*)

· a partition to store temporary data like print- and mail-queues (*/var*)

· a partition for third party and extra softw

emmy:~> **cd /**

emmy:/> **ls**

bin/ dev/ home/ lib/ misc/ opt/ root/ tmp/ var/

boot/ etc/ initrd/ lost+found/ mnt/ proc/ sbin/ usr/

**Directory Content**

/bin Common programs, shared by the system, the system administrator and the users.

/boot he startup files and the kernel, vmlinuz. In some recent distributions also grub data. Grub is the GRand Unified Boot loader and is an attempt to get rid of the many different boot-loaders we

know today.

/dev Contains references to all the CPU peripheral hardware, which are represented as files with

special properties.

/etc Most important system configuration files are in /etc, this directory contains data similar to

those in the Control Panel in Windows

/home Home directories of the common users.

/initrd (on some distributions) Information for booting. Do not remove!

/lib Library files, includes files for all kinds of programs needed by the system and the users.

/lost+found Every partition has a lost+found in its upper directory. Files that were saved during failures are here.

/misc For miscellaneous purposes.

/mnt Standard mount point for external file systems, e.g. a CD-ROM or a digital camera.

/net Standard mount point for entire remote file systems

/opt Typically contains extra and third party software.

/usr Programs, libraries, documentation etc. for all user-related programs.

/var Storage for all variable files and temporary files created by users, such as log files, the mail

queue, the print spooler area, space for temporary storage of files downloaded from the Internet

/proc A virtual file system containing information about system resources. More information about the meaning of the files in proc is obtained by entering the command **man *proc*** in a terminal

window. The file proc.txt discusses the virtual file system in detail.

/root The administrative user's home directory. Mind the difference between /, the root directory and

/root, the home directory of the *root* user.

/sbin Programs for use by the system and the system administrator.

/tmp Temporary space for use by the system, cleaned upon reboot, so don't use this for saving any work.

**Cheat**

*Note*  ***l*** *means long and* ***a*** *means hidden*

**quota -v :** File directory limitation in servor

**ls :** directory listing

ls -l

**ls - R :**listall the files in subdirectories as well

**ls -a** : will show hidden files

**ls -al** : shows a long list of files as well as the destinations that leads symbolic links point to.

Ls -ld : to list the properties of directories not the properties of file

**ls -latr :**displays the same files, only now in reversed order of the last change, so

that the file changed most recently occurs at the bottom of the list. **Ls- ltra** Dipangar’s famous

**ls -l | grep *part\_of\_file\_name :***To find a file in a directory

**grep pattern files1\_** : search for pattern in files1\_

**grep blue notepad.txt :** search for the word blue in the notepad file. Lines that contain the searched word will be displayed fully

**grep -r pattern dir :** search recursively for pattern in dir

**command | grep pattern** : search for pattern in the outpout command

**ls -la | sort -nk *5 :***directory content is sorted smallest files first, biggest files last

**ls -la | less :**To display output of a directory listing one page at a time

**locate -i school\*note :** search for any file that contains the word “school” and “note”, whether it is uppercase or lowercase. grep -l “school” \*

**find** **/home/ -name notes.txt :** search for a file called **notes.txt** within the home directory and its subdirectories.

**find . -name notes.txt :** To find files in the current directory

Perform a case-insensitive search for the word ‘bar’ in Linux and Unix: **grep -i 'bar' file1**

Look for all files in the current directory and in all of its subdirectories in Linux for the word ‘httpd ’**grep -R 'httpd' .**

|  |  |
| --- | --- |
| **Linux grep command options** | **Description** |
| -i | Ignore case distinctions on Linux and Unix |
| -w | Force PATTERN to match only whole words |
| -v | Select non-matching lines |
| -n | Print line number with output lines |
| -h | Suppress the Unix file name prefix on output |
| -r | Search directories recursivly on Linux |
| -R | Just like -r but follow all symlinks |
| -l | Print only names of FILEs with selected lines |
| -c | Print only a count of selected lines per FILE |
| --color | Display matched pattern in colors |

Jlab code.

[if you are the owner of a file or directory, you may use the chmod (change file access permissions) Unix command to change permissions.  When you use the ls -l Unix command, the owner and group owner of a file or directory will be listed after the permissions.  If you wanted to give the world permission to read and write to your home directory, then you would use the command below:

chmod o+rw /home/yourusername

You can also take away the read and write permissions by doing the following:

chmod o-rw /home/yourusername

The syntax of the chmod command is very simple.  By using the abbreviations given above, and the plus and minus symbols, you can change the add or remove permissions on files and directories you own.]

***Note (searching locate= file, find=files and directories , grep =text in a given file)***

**head -n 5 filename.ext**. view the first five lines of any text file

**tail -n filename.ext. : last ten lines of any text files**

**useradd** is used to create a new user, while **passwd** is adding a password to that user’s account. To add a new person named John type, **useradd John** and then to add his password type, **passwd 123456789.**

**cd** : change to home

**cd .. :** (with two dots) to move one directory up

**cd-** :(with a hyphen) to move to your previous directory

**pwd** : show current directory

**cd ~** : home directory

**echo $HOME :** home directory

**mkdir dir** : create directory dir

**mkdir Music/Newfile :** To generate a new directory inside another directory

**mkdir -p 2001/reports/Restaurants-Michelin/ :**Creating directories and subdirectories in one step is done using the -p option

**grep *pattern1* file | grep -v *pattern2 :***To find a word within some text, display all lines matching "pattern1", and exclude lines also matching

"pattern2" from being displayed:

**rm file :** delete file

rmdir : remove empty directory

**rm -r dir :** delete directory dir with content

**rm-f file :** force remove file

**rm -rf dir** : force remove directory dir with caution

**mv everest sagarmatha**  :To rename Everest into Sagarmatha. It is clear that only the name of the file changes. All other properties remain the same.

**mv file1 file2** : rename or move file1 to file2 if file 2 is an existing directory,moves file1 into directory file2

**Cp coconut butterfly :**Makes the duplicate of file cocon and give it name butterfly.

**Cp file1 file2** : copy file1 to file2

**Cp /home/text/contract ../contract.bak :**Makes a copy of files contract found in the directory taylor and put one step above working directory in a file called contract.bak

**Cp -r dir1 dir2** :copy dir1 to dir2, create dir2 if not exist

**cp [-R] fromfile tofile :**A useful option is recursive copy (copy all

underlying files and subdirectories), using the -R option to **cp**

**If /home/aditi** is directory then

**Cp aa.cc /home/aditi** will place a copy of aa.cc with directory /**home/aditi** with name **aa.cc**

**cat > filename** : create a new file

**cat filename1 filename2>filename3** joins two files (1 and 2) and stores the output of them in a new file (3)

**/home/username/Documents/Web.html :** create an blank HTML file entitled **Web** under the **Documents** directory.

**chmod *700* dirname :**Make directory Private

**chmod *777* dirname :**

**chmod *700* dirname Hint**( 7= RWX, 6=RW,5=RX,4=R,3=RW,2=W, 1=X,0 =-)

**su - -c "apt-get install xsnow"**

**apt-get *update***

**apt-get *upgrade***

**ssh user@host** : connect to host as a user

**ssh -p port user@host** : connect to host on port port as user

**ssh-copy-id user@host** ; add your key as a host for user to enable a keyed or passwordless login

**w** : display who is online

**whoami** : who you are logged in as

**finger user**: display information about user

**uname -a** : show kerner information

**cat /proc/cpuinfo** : cpu info

**cat /prop/meminfo** : memory info

**df** : show disk usage

**whereis app** : show possible location of app

**which app**: show which app will be run in default

Examples for ls

krissie:~/mp3> **ls**

Albums/ Radio/ Singles/ gene/ index.html

krissie:~/mp3> **ls -a**

./ .thumbs Radio gene/

../ Albums/ Singles/ index.html

krissie:~/mp3> **ls -l Radio/**

total 8

drwxr-xr-x 2 krissie krissie 4096 Oct 30 1999 Carolina/

drwxr-xr-x 2 krissie krissie 4096 Sep 24 1999 Slashdot/

krissie:~/mp3> **ls -ld Radio/**

drwxr-xr-x 4 krissie krissie 4096 Oct 30 1999 Radio/

krissie:~/mp3> **ls -ltr**

total 20

drwxr-xr-x 4 krissie krissie 4096 Oct 30 1999 Radio/

-rw-r--r-- 1 krissie krissie 453 Jan 7 2001 index.html

drwxrwxr-x 30 krissie krissie 4096 Oct 20 17:32 Singles/

drwxr-xr-x 2 krissie krissie 4096 Dec 4 23:22 gene/

drwxrwxr-x 13 krissie krissie 4096 Dec 21 11:40 Albums/

Renaming file

richard:~/archive> **mv ../report[1-4].doc reports/Restaurants-Michelin/**

This command is also applicable when renaming files:

**Color File type for ls**

red compressed archives / directories

white text files \* executable files

pink images @ link

cyan links

yellow devices

green executables

flashing red broken links

The **file** command has a series of options, among others the -z option to look into compressed files.

mike:~> **file me+tux.jpg**

me+tux.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI),

"28 Jun 1999", 144 x 144

Use the **rm** command to remove single files, **rmdir** to remove empty directories. (Use **ls -a** to check whether a directory is empty or not). The **rm** command also has options for removing non-empty directories with alltheir subdirectories, read the Info pages for these rather dangerous options.

To protect the beginning user from this malice, the interactive behavior of the **rm**, **cp** and **mv** commands can be activatedusing the -i option. In that case the system won't immediately act upon request. Instead it will ask forconfirmation, so it takes an additional click on the **Enter** key to inflict the damage:

mary:~> **rm -ri archive/**

rm: descend into directory `archive'? **y**

rm: descend into directory `archive/reports'? **y**

rm: remove directory `archive/reports'? **y**

rm: descend into directory `archive/backup'? **y**

rm: remove `archive/backup/sysbup200112.tar'? **y**

For instance, say that you want to display the file "\*" instead of all the files in a directory, you would have to use **less \\*** The same goes for filenames containing a space:**cat This\ File**

**less** is the GNU version of more and has extra features allowing highlighting of search strings,

scrolling back etc. The syntax is very simple:

**less name\_of\_file**

the first three characters in this series of nine display access rights for the actual user that owns the file. The next three are for the group owner of the file, the last three for other users.The permissions are always in the same order: read, write, execute for the user, the group and the others.

Some examples:

marise:~> **ls -l To\_Do**

-rw-rw-r-- 1 marise users 5 Jan 15 12:39 To\_Do

marise:~> **ls -l /bin/ls**

-rwxr-xr-x 1 root root 45948 Aug 9 15:01 /bin/ls\*

The first file is a regular file (first dash). Users with user name *marise* or users belonging to the group *users* can read and write (change/move/delete) the file, but they can't execute it (second and third dash). All other users are only allowed to read this file, but they can't write or execute it (fourth and fifth dash).

The second example is an executable file, the difference: everybody can run this program, but you need to be*root* to change it.

You should know what your user name is. If you don't, it can be displayed using the **id** command, which alsodisplays the default group you belong to and eventually other groups of which you are a member:

tilly:~> **id**

uid=504(tilly) gid=504(tilly) groups=504(tilly),100(users),2051(org)

Your user name is also stored in the environment variable USER:

tilly:~> **echo $USER**

tilly

**Table 4-1. Controlling processes**

**(part of) command Meaning**

**regular\_command** Runs this command in the foreground.

**command &**  Run this command in the background (release the terminal)

**jobs** Show commands running in the background.

**Ctrl**+**Z** Suspend (stop, but not quit) a process running in the foreground (suspend).

**Ctrl**+**C** Interrupt (terminate and quit) a process running in the foreground.

*%n* Every process running in the background gets a number assigned to it. By using the %

expression a job can be referred to using its number, for instance **fg *%2***.

**bg** Reactivate a suspended program in the background.

**Fg** Puts the job back in the foreground.

**kill** End a process (also see Shell Builtin Commands in the Info pages of **bash**)

Some examples: suppose we want to know which files in a certain directory have been modified in February:

jenny:~> **ls -la | grep Feb**

The **grep** command, like most commands, is case sensitive. Use the -i option to make no difference betweenupper and lower case. A lot of GNU extensions are available as well, such as --colour, which is helpful tohighlight searchterms in long lines, and --after-context, which prints the number of lines after the lastmatching line. You can issue a recursive **grep** that searches all subdirectories of encountered directories usingthe -r option. As usual, options can be combined.

With this command,

Make directories private with the command

Examples,

Grep -irn “hadron”

In order to understand these prompts and the escape sequences used, we refer to the Bash Info or man pages.

**export PS1=*"[\t \j] "***

Displays time of day and number of running jobs

**export PS1=*"[\d][\u@\h \w] : "***

Displays date, user name, host name and current working directory. Note that \W displays only base

names of the present working directory.

**export PS1=*"{\!} "***

Displays history number for each command.

**export PS1=*"\[\033[1;35m\]\u@\h\[\033[0m\] "***

Displays user@host in pink.

**export PS1=*"\[\033[1;35m\]\u\[\033[0m\] \[\033[1;34m\]\w\[\033[0m\] "***

Sets the user name in pink and the present working directory in blue.

**export PS1=*"\[\033[1;44m\]$USER is in \w\[\033[0m\] "***

Prompt for people who have difficulties seeing the difference between the prompt and what they type.

**export PS1=*"\[\033[4;34m\]\u@\h \w \[\033[0m\]"***

Underlined prompt.

**export PS1=*"\[\033[7;34m\]\u@\h \w \[\033[0m\] "***

White characters on a blue background.

· **export PS1=*"\[\033[3;35m\]\u@\h \w \[\033[0m\]\a"***

Some of the programs

provided in the core X Consortium distribution include:

· **xterm**: a terminal emulator

· **twm**: a minimalistic window manager

· **xdm**: a display manager

· **xconsole**: a console redirect program

· **bitmap**: a bitmap editor

· **xauth**, **xhost** and **iceauth**: access control programs

· **xset**, **xmodmap** and many others: user preference setting programs

· **xclock**: a clock

**xlsfonts** and others: a font displayer, utilities for listing information about fonts, windows and

displays

·· **xfs**: a font server

Printing from within an application is very easy, selecting the Print option from the menu.

From the command line, use the **lp** or **lpr** command.

**lp file(s)**

**lpr file(s)**

These commands can read from a pipe, so you can print the output of commands using

**command | lp**

**lpr** or **lp** Print file

**lpq** or **lpstat** Query print queue

**lprm** or **cancel** Remove print job

**acroread** PDF viewer

**groff** Formatting tool

**gv** PostScript viewer

**printconf** Configure printers

**xdvi** DVI viewer

**xpdf** PDF viewer

**\*2ps** Convert file to PostScript

<https://linoxide.com/linux-command/essential-linux-basic-commands/>