- date
- uname
- du
- free
- echo
- apt
- pwd
- cd
- ls
- tree
- man
- mkdir
- touch
- rm
- cp
- mv
- stat
- Wildcards (*,?,[])
- Brace expansion
- cat
- head
- tail
- cut
- tr
- paste
- wc
- grep
- output redirection
- Saving the output of a command
- vim or nano (basic stuff: open a file, close a file, edit a file)
- tar
- gz, bzip2, or xz
- chmod

Command name

`date`

Description

Display date and time in the given FORMAT. With -s, or with [MMD- Dhhmm[[CC]YY][.ss]], set the date and time.

Syntax

```
date OPTION + FORMAT
```

Example

- Description of example:
  - `date --date='TZ="America/Los_Angeles" 09:00 next Fri`
- Description of example:
  - `TZ='America/Los_Angeles' date`
- Description of example:
  - `date --date='@2147483647'`

Command name

`uname`

Description

Print certain system information. With no OPTION, same as -s.

Syntax

`uname [OPTION]...`

Example

- Description of example:
  - `uname -a`
- Description of example:
  - `uname -r`
- Description of example:
  - `uname -m`

Command name

`du`

Description

- Summarize device usage of the set of FILEs, recursively for directo- ries.

Syntax

`du [OPTION]... [FILE]...`

Example

- Description of example:
  - `du -h /home/cis106/Downloads`
- Description of example:
  - `du -a -h /home/cis106/Downloads`

- Description of example:
  - `du -c -h /home/cis106/Downloads`

Command name

`free`

Description

- free displays the total amount of free and used physical and swap mem- ory in the system, as well as the buffers and caches used by the ker- nel. The information is gathered by parsing /proc/meminfo. The dis- played columns

Syntax

`free [options]`

Example

- Description of example:
  - `free -h`
- Description of example:
  - `free -b`
- Description of example:
  - `free --mega`

Command name

`echo`

Description

- display a line of text

Syntax

`echo [SHORT-OPTION]... [STRING]...`

Example

- Description of example:
  - `echo -e 'Hello, GPU! \c This is PNAP!'`
- Description of example:
  - `echo -e 'Hello, \tGPU!'`
- Description of example:
  - `echo -e 'Hello, \vGPU, \vthis \vis \vPNAP!'`

Command name

`apt`

Description

- apt is a command-line utility for installing, updating, removing, and otherwise managing deb packages on Ubuntu, Debian, and related Linux distributions. It combines the most frequently used commands from the apt-get and apt-cache tools with different default values of some options.

Syntax

```
apt [-h] [-o=config_string] [-c=config_file] [-t=target_release] [-
a=architecture] {list | search | show | update | install pkg
[{=pkg_version_number | /target_release}]... | remove pkg... | upgrade | full-
upgrade | edit-sources | {-v | --version} | {-h | --help}}
```

Example

- Description of example:
  - `sudo apt full-upgrade`
- Description of example:
  - `apt list --upgradable`
- Description of example:
  - `sudo apt install nginx`

Command name

`pwd`

Description

- Print the full filename of the current working directory.

Syntax

```
pwd [OPTION]...
```

Example

- Description of example:
  - `pwd -P`
- Description of example:
  - `pwd -L`
- Description of example:
  - `pwd`

Command name

`cd`

Description

- change the working directory

Syntax

```
cd [-L|-P] [directory]
```

Example

- Description of example:
  - `cd ..`
- Description of example:
  - `cd /Downloads`
- Description of example:
  - `cd /Downloads/cis106`

Command name

`ls`

Description

- list directory contents

Syntax

```
ls [OPTION]... [FILE]...
```

Example

- Description of example:
  - `ls -l`
- Description of example:
  - `ls -ah /repollo`
- Description of example:
  - `ls -F /Documents`

Command name

`tree`

Description

- list contents of directories in a tree-like format.

Syntax

```
tree [-acdfghilnpqrstuvxACDFJQNSUX] [-L level [-R]] [-H baseHREF] [-T title] [-o filename] [-P pattern] [-I pattern] [--gitignore] [--matchdirs] [--metafirst] [--ignore-case] [--nolinks] [--inodes] [--device] [--sort[=]name] [--dirsfirst] [--filesfirst] [--filelimit #] [--si] [--du] [--prune] [--timefmt[=]format] [--fromfile] [--info] [--noreport] [--version] [--help] [--] [directory ...]
```

Example

- Description of example:
  - `tree -df`
- Description of example:
  - `tree -f -L 2`
- Description of example:
  - `tree -f -P cata*`

Command name

`man`

Description

- man is the system's manual pager. Each page argument given to man is normally the name of a program, utility or function. The manual page associated with each of these arguments is then found and displayed. A section, if provided, will direct man to look only in that section of the manual. The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the first page found, even if page exists in several sections.

Syntax

```
man [OPTION]... [COMMAND NAME]...
```

Example

- Description of example:
  - `man printf`
- Description of example:
  - `man -f ls`
- Description of example:
  - `man -k cd`

Command name

`mkdir`

Description

- make directories

Syntax

```
mkdir [OPTION]... DIRECTORY...
```

Example

- Description of example:
  - `mkdir Linux`
- Description of example:
  - `mkdir {test1,test2,test3}`
- Description of example:
  - `mkdir -P {one,two,three}/deva{andalai,maniu,manuki}`

Command name

```
touch
```

Description

- The touch command is a standard program for Unix/Linux operating systems, that is used to create, change and modify timestamps of a file. Before heading up for touch command examples, please check out the following options.

Syntax

```
touch <options> <file or directory name>
```

Example

- Description of example:
  - `touch bismuto1 bismuto2 bismuto3`
- Description of example:
  - `touch -m bismuto3`
- Description of example:
  - `touch -c bismuto4`

Command name

```
rm
```

Description

- rm command is used to remove objects such as files, directories, symbolic links and so on from the file system like UNIX.

Syntax

```
rm [OPTION]... [FILE]...
```

Example

- Description of example:
  - `rm -i d.txt`
- Description of example:
  - `rm -f e.txt`
- Description of example:
  - `rm -r *`

Command name

`cp`

Description

- Copy SOURCE to destination, or multiple SOURCE(s) to DIRECTORY.

Syntax

```
cp [OPTION]... SOURCE... DIRECTORY
```

Example

- Description of example:
  - `cp -i a.txt b.txt`
- Description of example:
  - `cp -b alimony.txt juvenile.txt`
- Description of example:
  - `cp -f repulsion.txt lamusa.txt`

Command name

`mv`

Description

- Copy SOURCE to destination, or multiple SOURCE(s) to DIRECTORY.

Syntax

```
mv [OPTION]... SOURCE... DIRECTORY
```

Example

- Description of example:
  - `mv sample1.txt sample2.txt sample3.txt ~/Documents/`
- Description of example:
  - `mv testdir1/ testdir2/`
- Description of example:
  - `mv -i sample.png ~/Documents/`

Command name

`stat`

Description

- Display file or file system status.

Syntax

```
stat [OPTION]... FILE...
```

Example

- Description of example:
  - `stat /etc/resolv.conf`
- Description of example:
  - `stat /etc/`
- Description of example:
  - `stat locale.conf login.defs`

Command name

`Wildcards (*,?,[])`

Description

- Wildcards (also referred to as meta characters) are symbols or special characters that represent other characters. You can use them with any command such as ls command or rm command to list or remove files matching a given criteria, receptively.

Syntax

```
* + option or option and *
```

Example

- Description of example:
  - `ls -l l*`
- Description of example:
  - `ls users-0*`

- Description of example:
  - `ls l?st.sh`

Command name

`Brace expansion`

Description

- Brace expansion is a mechanism by which arbitrary strings may be generated. This mechanism is similar to filename expansion (see Filename Expansion), but the filenames generated need not exist. Patterns to be brace expanded take the form of an optional preamble, followed by either a series of comma-separated strings or a sequence expression between a pair of braces, followed by an optional postscript. The preamble is prefixed to each string contained within the braces, and the postscript is then appended to each resulting string, expanding left to right.

Syntax

`{}`

Example

- Description of example:
  - `echo {one,two,three,four}`
- Description of example:
  - `echo {1..10}`
- Description of example:
  - `echo {q..v}`

Command name

`cat`

Description

- concatenate files and print on the standard output

Syntax

`cat [OPTION]... [FILE]...`

Example

- Description of example:
  - `cat`
- Description of example:
  - `cat /etc/issue`
- Description of example:

    - `cat -e test`

Command name

`head`

Description

- output the first part of files

Syntax

`head [OPTION]... [FILE]...`

Example

- Description of example:
    - `head -v cars.txt`
- Description of example:
    - `head --lines 5 cars.txt`
- Description of example:
    - `head cars.txt names.txt`

Command name

`head`

Description

- output the first part of files

Syntax

`head [OPTION]... [FILE]...`

Example

- Description of example:
    - `head -v cars.txt`
- Description of example:
    - `head --lines 5 cars.txt`
- Description of example:
    - `head cars.txt names.txt`

Command name

`tail`

Description

- output the last part of files

Syntax

```
tail [OPTION]... [FILE]...
```

Example

- Description of example:
  - `tail -n 3 state.txt`
- Description of example:
  - `tail +25 state.txt`
- Description of example:
  - `tail -c -253 cereal.csv`

Command name

`cut`

Description

- Remove sections from each line of files

Syntax

```
cut OPTION... [FILE]...
```

Example

- Description of example:
  - `cut -b 1,2,3 cereal.csv`
- Description of example:
  - `cut -d " " -f 1 cereal.csv`
- Description of example:
  - `cut --complement -d " " -f 1 cereal.csv`

Command name

`tr`

Description

- Translate, squeeze, and/or delete characters from standard input, writ- ing to standard output. STRING1 and STRING2 specify arrays of charac- ters ARRAY1 and ARRAY2 that control the action.

Syntax

```
tr [OPTION]... STRING1 [STRING2]
```

Example

- Description of example:
  - `echo "Welcome To Fedora" | tr [:space:] '\t'`
- Description of example:
  - `echo "Welcome To Universe Transformers" | tr -d 'w'`
- Description of example:
  - `echo "my ID is 73535" | tr -d [:digit:]`

Command name

`paste`

Description

- merge lines of files

Syntax

```
paste [OPTION]... [FILE]...
```

Example

- Description of example:
  - `paste film.csv cereal.csv`
- Description of example:
  - `paste -d '_' film.csv cereal.csv`
- Description of example:
  - `paste -d '%|' film.csv cereal.csv`

Command name

`wc`

Description

- Print newline, word, and byte counts for each FILE, and a total line if more than one FILE is specified. A word is a non-zero-length sequence of printable characters delimited by white space.

Syntax

```
wc [OPTION]... [FILE]...
```

Example

- Description of example:

    - `wc film.csv cereal.csv`
- Description of example:
    - `wc -l film.csv cereal.csv`
- Description of example:
    - `wc -w film.csv cereal.csv`

Command name

`grep`

Description

- grep searches for PATTERNS in each FILE. PATTERNS is one or more patterns separated by newline characters, and grep prints each line that matches a pattern. Typically PATTERNS should be quoted when grep is used in a shell command.

Syntax

```
grep [OPTION...] PATTERNS [FILE...]
```

Example

- Description of example:
    - `grep -i 'trix' cereal.csv`
- Description of example:
    - `lspci | grep -i --color 'vga\|3d\|2d'`
- Description of example:
    - `ngrep -d any port 25`

Command name

`output redirection`

Description

- Input/Output (I/O) redirection in Linux refers to the ability of the Linux operating system that allows us to change the standard input (stdin) and standard output (stdout) when executing a command on the terminal.

By default, the standard input device is your keyboard and the standard output device is your screen.

Syntax

```
input command + option > output file
```

Example

- Description of example:

    - `ls -al > listings`
- Description of example:
    - `cat music.mp3 > /dev/audio`
- Description of example:
    - `myprogram 2>errorsfile`

Command name

`Saving the output of a command`

Description

- When you run a command or script in the Linux terminal, it prints the output on the screen for your immediate viewing.

There will be times when you need to save the output to a file for future references. Now, you can surely copy and paste in Linux terminal but there are better ways to save the output of a shell script or command in Linux command line. Let me show them to you.

Syntax

`command > file`

Example

- Description of example:
    - `ls * .c >> output.txt`
- Description of example:
    - `ls -l | tree output.txt`
- Description of example:
    - `ls -A | output.txt`

Command name

`nano`

Description

- nano is a small and friendly editor. It copies the look and feel of Pico, but is free software, and implements several features that Pico lacks, such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping overlong lines.

Syntax

`nano [options] [[+line[,column]] file]...`

Example

- Description of example:
  - `nano /etc/nanorc`
- Description of example:
  - `nano -w /etc/apache2/apache2.conf`
- Description of example:
  - `nano +2,5 abc.txt`

Command name

`tar`

Description

- an archiving utility

Syntax

```
tar -A [OPTIONS] ARCHIVE ARCHIVE
```

Example

- Description of example:
  - `tar cvf file.tar *.c`
- Description of example:
  - `tar xvf file.tar`
- Description of example:
  - `tar cvzf file.tar.gz *.c`

Command name

`gzip`

Description

- The gzip command reduces the size of the named files using Lempel-Ziv coding (LZ77). Whenever possible, each file is replaced by one with the extension .gz, while keeping the same ownership modes, access and modification times. (The default extension is z for MSDOS, OS/2 FAT, Windows NT FAT and Atari.) If no files are specified, or if a file name is "-", the standard input is compressed to the standard output. The gzip command will only attempt to compress regular files. In par- ticular, it will ignore symbolic links.

Syntax

```
gzip [ -acdfhklLnNrtvV19 ] [-S suffix] [ name ... ] gunzip [ -acfhklLnNrtvV ] [-S suffix] [ name ... ] zcat [ -fhLV ] [ name ... ]
```

Example

- Description of example:
  - `gzip -k cereal.csv`
- Description of example:
  - `gzip -c cereal.csv`
- Description of example:
  - `gzip -v cereal.csv`

Command name

`chmod`

Description

- chmod changes the file mode bits of each given file according to mode, which can be ei- ther a symbolic representation of changes to make, or an octal number representing the bit pattern for the new mode bits.

Syntax

```
chmod [OPTION]... MODE[,MODE]... FILE...
```

Example

- Description of example:
  - `chmod u=rw,og=r papiro.txt`
- Description of example:
  - `chmod a+x papiro.sh`
- Description of example:
  - `chmod +x papiro.sh`