

# Module 2 Cheat Sheet - Introduction to Linux Commands

## Getting information

---

Return your user name:

1. 1
1. whoami

Copied!

Return your user and group id:

1. 1
1. id

Copied!

Return operating system name, username, and other info:

1. 1
1. uname -a

Copied!

Display reference manual for a command:

1. 1
1. man top

Copied!

List available man pages, including a brief description for each command:

1. 1
1. man -k .

Copied!

Get help on any command (for eg: curl):

1. 1
1. curl --help

Copied!

This provides a brief overview of the curl command's usage and options.

Return the current date and time:

1. 1
1. date

Copied!

## Navigating and working with directories

---

List files and directories by date, newest to last:

1. 1
1. ls -ltr

Copied!

Find files in directory tree that end in .sh:

1. 1
1. find -name '\*.sh'

Copied!

Return path to present working directory:

- 1. 1
- 1. pwd

Copied!

Make a new directory:

- 1. 1
- 1. mkdir new\_folder

Copied!

Change the current directory:

Up one level:

- 1. 1
- 1. cd ../

Copied!

To home:

- 1. 1
- 1. cd ~` or `cd

Copied!

To some other directory: cd path\_to\_directory

Remove directory verbosely:

- 1. 1
- 1. rmdir temp\_directory -v

Copied!

Monitoring system performance and status

---

List selection of/all running processes and their PIDs:

- 1. 1
- 1. ps

Copied!

- 1. 1
- 1. ps -e

Copied!

Display resource usage:

- 1. 1
- 1. top

Copied!

List mounted file systems and usage:

- 1. 1
- 1. df

Copied!

Creating, copying, moving, and deleting files:

---

Create an empty file or update existing file's timestamp:

- 1. 1
- 1. touch a\_new\_file.txt

Copied!

**Copy a file:**

```
1. 1
1. cp file.txt new_path/new_name.txt
```

Copied!

**Change file name or path:**

```
1. 1
1. mv this_file.txt that_path/that_file.txt
```

Copied!

**Remove a file verbosely:**

```
1. 1
1. rm this_old_file.txt -v
```

Copied!

**Working with file permissions**

---

**Change/modify file permissions to 'execute' for all users:**

```
1. 1
1. chmod +x my_script.sh
```

Copied!

**Change/modify file permissions to 'execute' only for you, the current user:**

```
1. 1
1. chmod u+x my_file.txt
```

Copied!

**Remove 'read' permissions from group and other users:**

```
1. 1
1. chmod go-r
```

Copied!

**Displaying file and string contents**

---

**Display file contents:**

```
1. 1
1. cat my_shell_script.sh
```

Copied!

**Display file contents page-by-page:**

```
1. 1
1. more ReadMe.txt
```

Copied!

**Display first 10 lines of file:**

```
1. 1
1. head -10 data_table.csv
```

Copied!

**Display last 10 lines of file:**

```
1. 1
```

```
1. tail -10 data_table.csv
```

Copied!

Display string or variable value:

```
1. 1
2. 2

1. echo "I am not a robot"
2. echo "I am $USERNAME"
```

Copied!

Basic text wrangling

Sorting lines and dropping duplicates:

Sort and display lines of file alphanumerically:

```
1. 1

1. sort text_file.txt
```

Copied!

In reverse order:

```
1. 1

1. sort -r text_file.txt
```

Copied!

Drop consecutive duplicated lines and display result:

```
1. 1

1. uniq list_with_duplicated_lines.txt
```

Copied!

Displaying basic stats:

Display the count of lines, words, or characters in a file:

Lines:

```
1. 1

1. wc -l table_of_data.csv
```

Copied!

Words:

```
1. 1

1. wc -w my_essay.txt
```

Copied!

Characters:

```
1. 1

1. wc -m some_document.txt
```

Copied!

Extracting lines of text containing a pattern:

Some frequently used options for grep:

Option	Description
-n	Print line numbers along with matching lines
-c	Get the count of matching lines
-i	Ignore the case of the text while matching
-v	Print all lines which do not contain the pattern
-w	Match only if the pattern matches whole words

Extract lines containing the word "hello", case insensitive and whole words only:

```
1. 1
1. grep -iw hello a_bunch_of_hellos.txt
```

Copied!

Extract lines containing the pattern "hello" from all files in the current directory ending in .txt:

```
1. 1
1. grep -l hello *.txt
```

Copied!

**Merge two or more files line-by-line, aligned as columns:**

Suppose you have three files containing the first and last names of your customers, plus their phone numbers.

Use paste to align file contents into a Tab-delimited table, one row for each customer:

```
1. 1
1. paste first_name.txt last_name.txt phone_number.txt
```

Copied!

Use a comma as a delimiter instead of the default Tab delimiter:

```
1. 1
1. paste -d "," first_name.txt last_name.txt phone_number.txt
```

Copied!

**Use the cut command to extract a column from a table-like file:**

Suppose you have a text file whos rows consist of first and last names of customers, delimited by a comma.

Extract first names, line-by-line:

```
1. 1
1. cut -d "," -f 1 names.csv
```

Copied!

Extract the second to fifth characters (bytes) from each line of a file:

```
1. 1
1. cut -b 2-5 my_text_file.txt
```

Copied!

Extract the characters (bytes) from each line of a file, starting from the 10th byte to the end of the line:

```
1. 1
1. cut -b 10- my_text_file.txt
```

Copied!

## Compression and archiving

---

Archive a set of files:

```
1. 1
1. tar -cvf my_archive.tar.gz file1 file2 file3
```

Copied!

Compress a set of files:

```
1. 1
2. 2
1. zip my_zipped_files.zip file1 file2
2. zip my_zipped_folders.zip directory1 directory2
```

Copied!

Extract files from a compressed zip archive:

```
1. 1
2. 2

1. unzip my_zipped_file.zip
2. unzip my_zipped_file.zip -d extract_to_this_directory
```

Copied!

## Working with networking commands

---

### Print hostname:

```
1. 1
1. hostname
```

Copied!

### Send packets to URL and print response:

```
1. 1
1. ping www.google.com
```

Copied!

### Display or configure system network interfaces:

```
1. 1
2. 2

1. ifconfig
2. ip
```

Copied!

### Display contents of file at a URL:

```
1. 1
1. curl <url>
```

Copied!

### Download file from a URL:

```
1. 1
1. wget <url>
```

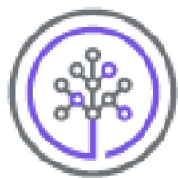
Copied!

## Authors

Jeff Grossman  
Sam Propupchuk

## Other Contributors

Rav Ahuja



# Skills Network