# Linux and Bash Command Cheat Sheet: The Basics

## Getting information

---

# return your user name

```
1. 1
1. whoami
```

Copied! Executed!

# return your user and group id

```
1. 1
1. id
```

Copied! Executed!

# return operating system name, username, and other info

```
1. 1
1. uname -a
```

Copied! Executed!

# display reference manual for a command

```
1. 1
1. man top
```

Copied! Executed!

# get help on a command

```
1. 1
1. curl --help
```

Copied! Executed!

# return the current date and time

```
1. 1
1. date
```

Copied! Executed!

## Monitoring performance and status

---

# list selection of or all running processes and their PIDs

```
1. 1
```

```
2. 2
```

```
1. ps
2. ps -e
```

Copied! Executed!

# display resource usage

```
1. 1
```

```
1. top
```

Copied! Executed!

# list mounted file systems and usage

```
1. 1
```

```
1. df
```

Copied! Executed!

## Working with files

---

# copy a file

```
1. 1
```

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

Copied! Executed!

# change file name or path

```
1. 1
```

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

Copied! Executed!

# remove a file verbosely

```
1. 1
```

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

Copied! Executed!

# create an empty file, or update existing file's timestamp

```
1. 1
```

```
1. touch a_new_file.txt
```

Copied! Executed!

# change/modify file permissions to 'execute' for all users

```
1. 1
```

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

Copied! Executed!

# get count of lines, words, or characters in file

```
1. 1
2. 2
3. 3
```

```
1. wc  -l table_of_data.csv
2. wc  -w my_essay.txt
3. wc  -m some_document.txt
```

Copied! Executed!

# return lines matching a pattern from files matching a filename pattern - case insensitive and whole words only

```
1. 1
```

```
1. grep  -iw hello  \*.txt
```

Copied! Executed!

# return file names with lines matching the pattern 'hello' from files matching a filename pattern

```
1. 1
```

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

Copied! Executed!

## Navigating and working with directories

---

# list files and directories by date, newest last

```
1. 1
```

```
1. ls -lrt
```

Copied! Executed!

# find files in directory tree with suffix 'sh'

```
1. 1
```

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

Copied! Executed!

# return present working directory

```
1. 1
```

```
1. pwd
```

Copied! Executed!

# make a new directory

```
1. 1
```

```
1. mkdir new_folder
```

Copied! Executed!

# change the current directory: up one level, home, or some other path

```
1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
```

```
1. cd ../
2. cd ~ or cd
3. cd another_directory
4.
5. `\# remove directory, verbosely`
6. rmdir temp_directory -v
```

Copied! Executed!

## Printing file and string contents

---

# print file contents

```
1. 1
```

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

Copied! Executed!

# print file contents page-by-page

```
1. 1
```

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

Copied! Executed!

# print first N lines of file

```
1. 1
```

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

Copied! Executed!

# print last N lines of file

```
1. 1
```

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

Copied! Executed!

# print string or variable value

```
1. 1
2. 2
```

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

Copied! Executed!

## Compression and archiving

---

# archive a set of files

```
1
```

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

Copied! Executed!

# compress a set of files

```
1
2
```

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

Copied! Executed!

# extract files from a compressed zip archive

```
1
2
```

```
unzip my_zipped_file.zip
unzip my_zipped_file.zip -d extract_to_this_direcory
```

Copied! Executed!

## Performing network operations

---

# print hostname

```
1
```

```
hostname
```

Copied! Executed!

# send packets to URL and print response

```
1
```

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

Copied! Executed!

# display or configure system network interfaces

```
1
2
```

```
ifconfig
ip
```

Copied! Executed!

# display contents of file at a URL

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

Copied! Executed!

# download file from a URL

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

Copied! Executed!

## Bash shebang

---

```
#!/bin/bash
```

## Pipes and Filters

---

# chain filter commands using the pipe operator

```
1. 1
1. ls | sort -r
```

Copied! Executed!

# pipe the output of manual page for ls to head to display the first 20 lines

```
1. 1
1. man ls | head -20
```

Copied! Executed!

## Shell and Environment Variables

---

# list all shell variables

```
1. 1
1. set
```

Copied! Executed!

# define a shell variable called my_planet and assign value Earth to it

```
1. 1
1. my_planet=Earth
```

Copied! Executed!

# display shell variable

```
1
```

```
echo $my_planet
```

Copied! Executed!

# list all environment variables

```
1
```

```
env
```

Copied! Executed!

# environment vars: define/extend variable scope to child processes

```
1
2
```

```
export my_planet
export my_galaxy='Milky Way'
```

Copied! Executed!

## Metacharacters

---

# comments

```
# The shell will not respond to this message
```

# command separator

```
1
```

```
echo 'here are some files and folders'; ls
```

Copied! Executed!

# file name expansion wildcard

```
1
```

```
ls *.json
```

Copied! Executed!

# single character wildcard

```
1
```

```
ls file_2021-06-??.json
```

Copied! Executed!

## Quoting

---

# single quotes - interpret literally

```
1
```

```
echo 'My home directory can be accessed by entering: echo $HOME'
```

Copied! Executed!

# double quotes - interpret literally, but evaluate metacharacters

```
1. 1
1. echo "My home directory is $HOME"
```

Copied! Executed!

# backslash - escape metacharacter interpretation

```
1. 1
1. echo "This dollar sign should render: \$"
```

Copied! Executed!

## I/O Redirection

---

# redirect output to file

```
1. 1
1. echo 'Write this text to file x' > x
```

Copied! Executed!

# append output to file

```
1. 1
1. echo 'Add this line to file x' >> x
```

Copied! Executed!

# redirect standard error to file

```
1. 1
1. bad_command_1 2> error.log
```

Copied! Executed!

# append standard error to file

```
1. 1
1. bad_command_2 2>> error.log
```

Copied! Executed!

# redirect file contents to standard input

```
1. 1
1. $ tr “[a-z]” “[A-Z]” < a_text_file.txt
```

Copied! Executed!

# the input redirection above is equivalent to

```
1
```

```
$cat a_text_file.txt | tr “[a-z]” “[A-Z]”
```

Copied! Executed!

### Command Substitution

---

# capture output of a command and echo its value

```
1
2
```

```
THE_PRESENT=$(date)
echo "There is no time like $THE_PRESENT"
```

Copied! Executed!

### Command line arguments

---

```
1
```

```
./My_Bash_Script.sh arg1 arg2 arg3
```

Copied! Executed!

### Batch vs. concurrent modes

---

# run commands sequentially

```
1
```

```
start=$(date); ./MyBigScript.sh ; end=$(date)
```

Copied! Executed!

# run commands in parallel

```
1
```

```
./ETL_chunk_one_on_these_nodes.sh  & ./ETL_chunk_two_on_those_nodes.sh
```

Copied! Executed!

## Scheduling jobs with Cron

---

# open crontab editor

```
1
```

```
crontab -e
```

Copied! Executed!

# job scheduling syntax

```
1
```

```
m  h  dom  mon  dow   command
```

Copied! Executed!

*minute, hour, day of month, month, day of week*
* means any

# append the date/time to file every Sunday at 6:15 pm

```
1
```

```
15 18 * * 0 date >> sundays.txt
```

Copied! Executed!

# run a shell script on the first minute of the first day of each month

```
1
```

```
1  0 1 * * ./My_Shell_Script.sh
```

Copied! Executed!

# back up your home directory every Monday at 3 am

```
1
```

```
0 3 * * 1  tar -cvf my_backup_path\my_archive.tar.gz $HOME\
```

Copied! Executed!

# deploy your cron job
*Close the crontab editor and save the file*

# list all cron jobs

```
1
```

```
crontab -l
```

Copied! Executed!

---