Linux and Bash Command Cheat Sheet: The Basics

Getting information

# return your user name

```
whoami
```

Copied! Executed!

# return your user and group id

```
id
```

Copied! Executed!

# return operating system name, username, and other info

```
uname -a
```

Copied! Executed!

# display reference manual for a command

```
man top
```

Copied! Executed!

# get help on a command

```
curl --help
```

Copied! Executed!

# return the current date and time

```
date
```

Copied! Executed!

Monitoring performance and status

# list selection of or all running processes and their PIDs

```
ps
ps -e
```

Copied! Executed!

# display resource usage

```
top
```

Copied! Executed!

# list mounted file systems and usage

```
df
```

Copied! Executed!

Working with files

# copy a file

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

Copied! Executed!

# change file name or path

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

Copied! Executed!

# remove a file verbosely

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

Copied! Executed!

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

```
touch a_new_file.txt
```

Copied! Executed!

# change/modify file permissions to ‘execute’ for all users

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

Copied! Executed!

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

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

Copied! Executed!

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

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

Copied! Executed!

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

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

Copied! Executed!

Navigating and working with directories

# list files and directories by date, newest last

```
ls -lrt
```

Copied! Executed!

# find files in directory tree with suffix ‘sh’

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

Copied! Executed!

# return present working directory

```
pwd
```

Copied! Executed!

# make a new directory

```
mkdir new_folder
```

Copied! Executed!

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

```
cd ../
cd ~ or cd
cd another_directory

`\# remove directory, verbosely`
rmdir temp_directory -v
```

Copied! Executed!

Printing file and string contents

# print file contents

```
cat my_shell_script.sh
```

Copied! Executed!

# print file contents page-by-page

```
more ReadMe.txt
```

Copied! Executed!

# print first N lines of file

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

Copied! Executed!

# print last N lines of file

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

Copied! Executed!

# print string or variable value

```
echo "I am not a robot"
echo "I am $USERNAME"
```

Copied! Executed!

Compression and archiving

# archive a set of files

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

Copied! Executed!

# compress a set of files

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

Copied! Executed!

# extract files from a compressed zip archive

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

Copied! Executed!

Performing network operations

# print hostname

```
hostname
```

Copied! Executed!

# send packets to URL and print response

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

Copied! Executed!

# display or configure system network interfaces

```
ifconfig
ip
```

Copied! Executed!

# display contents of file at a URL

```
curl  <url>
```

Copied! Executed!

# download file from a URL

```
wget  <url>
```

Copied! Executed!

Bash shebang

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

Pipes and Filters

# chain filter commands using the pipe operator

```
ls | sort -r
```

Copied! Executed!

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

```
man ls | head -20
```

Copied! Executed!

Shell and Environment Variables

# list all shell variables

```
set
```

Copied! Executed!

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

```
my_planet=Earth
```

Copied! Executed!

# display shell variable

```
echo $my_planet
```

Copied! Executed!

# list all environment variables

```
env
```

Copied! Executed!

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

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

Copied! Executed!

Metacharacters

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

# command separator

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

Copied! Executed!

# file name expansion wildcard

```
ls *.json
```

Copied! Executed!

# single character wildcard

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

Copied! Executed!

Quoting

# single quotes - interpret literally

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

Copied! Executed!

# double quotes - interpret literally, but evaluate metacharacters

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

Copied! Executed!

# backslash - escape metacharacter interpretation

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

Copied! Executed!

I/O Redirection

# redirect output to file

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

Copied! Executed!

# append output to file

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

Copied! Executed!

# redirect standard error to file

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

Copied! Executed!

# append standard error to file

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

Copied! Executed!

# redirect file contents to standard input

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

Copied! Executed!

# the input redirection above is equivalent to

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

Copied! Executed!

Command Substitution

# capture output of a command and echo its value

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

Copied! Executed!

Command line arguments

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

Copied! Executed!

Batch vs. concurrent modes

# run commands sequentially

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

Copied! Executed!

# run commands in parallel

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

Copied! Executed!

Scheduling jobs with Cron

# open crontab editor

```
crontab -e
```

Copied! Executed!

# job scheduling syntax

```
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

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

Copied! Executed!

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

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

Copied! Executed!

# back up your home directory every Monday at 3 am

```
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

```
crontab -l
```

Copied! Executed!