

# Module 2 Cheat Sheet - Introduction to Linux Commands

## Getting information

---

Return your user name:

```
whoami
```

Return your user and group id:

```
id
```

Return operating system name, username, and other info:

```
uname -a
```

Display reference manual for a command:

```
man top
```

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

```
man -k .
```

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

```
curl --help
```

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

**Return the current date and time:**

```
date
```

## Navigating and working with directories

---

**List files and directories by date, newest to last:**

```
ls -lrt
```

**Find files in directory tree that end in `.sh`:**

```
find -name "*.sh"
```

**Return path to present working directory:**

```
pwd
```

**Make a new directory:**

```
mkdir new_folder
```

**Change the current directory:**

**Up one level:**

```
cd ../
```

**To home:**

```
cd ~` or `cd
```

**To some other directory:** `cd path_to_directory`

**Remove directory verbosely:**

```
rmdir temp_directory -v
```

## Monitoring system performance and status

---

**List selection of/all running processes and their PIDs:**

```
ps
```

```
ps -e
```

**Display resource usage:**

```
top
```

**List mounted file systems and usage:**

```
df
```

## Creating, copying, moving, and deleting files:

---

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

```
touch a_new_file.txt
```

#### Copy a file:

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

#### Change file name or path:

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

#### Remove a file verbosely:

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

## Working with file permissions

---

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

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

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

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

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

```
chmod go-r
```

## Displaying file and string contents

---

### Display file contents:

```
cat my_shell_script.sh
```

### Display file contents page-by-page:

```
more ReadMe.txt
```

### Display first 10 lines of file:

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

### Display last 10 lines of file:

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

### Display string or variable value:

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

## Basic text wrangling

---

### Sorting lines and dropping duplicates:

#### Sort and display lines of file alphanumerically:

```
sort text_file.txt
```

**In reverse order:**

```
sort -r text_file.txt
```

**Drop consecutive duplicated lines and display result:**

```
uniq list_with_duplicated_lines.txt
```

**Displaying basic stats:**

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

**Lines:**

```
wc -l table_of_data.csv
```

**Words:**

```
wc -w my_essay.txt
```

**Characters:**

```
wc -m some_document.txt
```

**Extracting lines of text containing a pattern:**

Some frequently used options for grep:

Option	Description
-n	Print line numbers along with matching lines

Option	Description
-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:

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

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

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

**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:

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

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

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

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

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

Extract first names, line-by-line:

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

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

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

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

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

## Compression and archiving

---

**Archive a set of files:**

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

**Compress a set of files:**

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

**Extract files from a compressed zip archive:**

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

## Working with networking commands

---

**Print hostname:**

```
hostname
```

**Send packets to URL and print response:**

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

**Display or configure system network interfaces:**

```
ip
```

**Display contents of file at a URL:**

```
curl <url>
```

**Download file from a URL:**

```
wget <url>
```

## Authors

Jeff Grossman  
Sam Propupchuk

## Other Contributors

Rav Ahuja



# Skills Network