LINUX & GIT CHEAT SHEET

Ruksar Nura Shaikh
Sr. Software Engineer

Swipe →

1. ls Command

List files and directories.

```
ls -l
# displays files and directories with detailed information.

ls -a
# shows all files and directories, including

ls -lh
# displays file sizes in a human-readable format.
```

2. cd

Change directory.

```
cd /path/to/directory
# Changes the current directory to the specified path.
```

Swipe →

3. pwd

Print current working directory.

```
pwd
# displays the current working directory.
```

4. mkdir

Create a new directory.

```
mkdir my_directory
creates a new directory named “my_directory”.
```

Swipe →

5. rm

Remove files and directories.

```
rm file.txt
# deletes the file named “file.txt”.

rm -r my_directory
# deletes the directory “my_directory” and its contents.

rm -f file.txt
# forcefully deletes the file “file.txt” without confirmation.
```

6. cp

Copy files and directories.

```
cp -r directory destination
# copies the directory “directory” and its contents to the specified destination.

cp file.txt destination
# copies the file “file.txt” to the specified destination.
```

Swipe →

7. mv

Move/rename files and directories.

```
mv file.txt new_name.txt
# renames the file “file.txt” to “new_name.txt”.

mv file.txt directory
# moves the file “file.txt” to the specified directory.
```

8. touch

Create an empty file or update file timestamps.

```
touch file.txt
creates an empty file named “file.txt”.
```

Swipe →

9. cat

View the contents of a file.

```
cat file.txt
# displays the contents of the file “file.txt”.
```

10. head

Display the first few lines of a file.

```
head file.txt
# shows the first 10 lines of the file “file.txt”.

head -n 5 file.txt
# displays the first 5 lines of the file “file.txt”.
```

Swipe →

11. tail

Display the last few lines of a file.

```
tail file.txt
# shows the last 10 lines of the file “file.txt”.

tail -n 5 file.txt
# displays the last 5 lines of the file “file.txt”.
```

12. ln

Create links between files.

```
ln -s source_file link_name
# creates a symbolic link named “link_name” pointing to “source_file”.
```

Swipe →

13. find

Search for files and directories.

```
find /path/to/search -name “*.txt”
# searches for all files with the extension “.txt” in the specified directory.
```

14. chmod

Change file permissions.

```
chmod u+rwx file.txt
# grants read, write, and execute permissions to the owner of the file.
```

Swipe →

15. chown

Change file ownership.

```
chown user file.txt
# changes the owner of “file.txt” to the specified user.
```

16. chgrp

Change group ownership.

```
chgrp group file.txt
#changes the group ownership of “file.txt” to the specified group.
```

Swipe →

17. umask

Set default file permissions.

```
umask 022
# sets the default file permissions to read and write for the owner,
and read-only for group and others.
```

18. tar

Create or extract archive files.

```
tar -czvf archive.tar.gz files/
# creates a compressed tar archive named “archive.tar.gz”
containing the files in the “files/” directory.
```

Swipe →

19. gzip

Compress files.

```
gzip file.txt
# compresses the file “file.txt” and renames it as “file.txt.gz”.
```

20. zip

Create compressed zip archives.

```
zip archive.zip file1.txt file2.txt
# creates a zip archive named “archive.zip”
containing “file1.txt” and “file2.txt”.
```

Swipe →

21. ps

Display running processes.

```
ps aux
# shows all running processes with detailed information.
```

22. top

Monitor system processes in real-time.

```
top
# displays a dynamic view of system
processes and their resource usage.
```

Swipe →

23. kill

Terminate a process.

```
kill PID
# terminates the process with the specified process ID.
```

24. pkill

Terminate processes based on their name.

```
pkill process_name
# terminates all processes with the specified name.
```

Swipe →

25. pgrep

List processes based on their name.

```
pgrep process_name
# lists all processes with the specified name.
```

26. grep

Used to search for specific patterns or regular expressions in text files or streams and display matching lines.

```
grep -i “hello” file.txt
grep -v “error” file.txt
grep -r “pattern” directory/
grep -l “keyword” file.txt
grep -n “pattern” file.txt
# In these examples we are extracting our desirec output from filename (file.txt)
```

Swipe →

27. uname

Print system information.

```
uname -a
# displays all system information.
```

28. whoami

Display current username.

```
whoami
# Shows the current username.
```

Swipe →

29. df

Show disk space usage.

```
df -h
# displays disk space usage in a human-readable format.
```

30. du

Estimate file and directory sizes.

```
du -sh directory/
# provides the total size of the specified directory.
```

Swipe →

31. free

Display memory usage information.

```
free -h
# displays memory usage in a human-readable format.
```

32. uptime

Show system uptime.

```
uptime
# Shows the current system uptime.
```

Swipe →

33. lscpu

Display CPU information.

```
lscpu
# provides detailed CPU information.
```

34. lspci

List PCI devices.

```
lspci
# List PCI devices.
```

Swipe →

35. lsusb

List USB devices.

```
lsusb
# lists all connected USB devices.
```

36. ifconfig

Display network interface information.

```
ifconfig
# Shows the details of all network interfaces.
```

Swipe →

37. ping

Send ICMP echo requests to a host.

```
ping google.com
# sends ICMP echo requests to “google.com” to check connectivity.
```

38. netstat

Display network connections and statistics.

```
netstat -tuln
#shows all listening TCP and UDP connections.
```

Swipe →

39. ssh

Securely connect to a remote server.

```
ssh user@hostname
# Initiates an SSH connection to the specified hostname.
```

40. scp

Securely copy files between hosts.

```
scp file.txt user@hostname:/path/to/destination
#securely copies “file.txt” to the specified remote host.
```

Swipe →

41. wget

Download files from the web.

```
wget http://example.com/file.txt
# downloads “file.txt” from the specified URL.
```

42. curl

Transfer data to or from a server.

```
curl http://example.com
# Retrieves the content of a webpage
from the specified URL.
```

Swipe →

Git Commands

1. git init

Initializes a new Git repository in the current directory.

```
git init
```

2. git init <directory>

Creates a new Git repository in the specified directory.

```
git init <directory>
```

Swipe →

3. git clone

This Clones a repository from a remote server to your local machine.

```
git clone <repository_url>
```

4. git clone –branch <branch_name> <repository_url>

Clones a specific branch from a repository.

```
git clone -branch <branch_name> <repository_url>
```

Swipe →

5. git add

Adds a specific file to the staging area.

```
git add <file>
```

6. git add . or git add –all

Adds all modified and new files to the staging area.

```
git add .

git add -all
```

Swipe →

7. git status

Shows the current state of your repository, including tracked and untracked files, modified files, and branch information.

```
git status
```

8. git status –ignored

Displays ignored files in addition to the regular status output.

```
git status –ignored
```

Swipe →

9. git diff

Shows the changes between the working directory and the staging area (index).

```
git diff
```

10. git diff <commit1> <commit2>

Displays the differences between two commits.

```
git diff <commit1> <commit2>
```

Swipe →

11. git commit

Creates a new commit with the changes in the staging area and opens the default text editor for adding a commit message.

```
git commit
```

12. git commit -a or git commit –all

Commits all modified and deleted files in the repository without explicitly using git add to stage the changes.

```
git commit -a

git commit –all
```

Swipe →

13. git restore <file>

Restores the file in the working directory to its state in the last commit.

```
git restore <file>
```

14. git reset <commit>

Moves the branch pointer to a specified commit, resetting the staging area and the working directory to match the specified commit.

```
git reset <commit>
```

Swipe →

15. git rm <file>

Removes a file from both the working directory and the repository, staging the deletion.

```
git rm <file>
```

16. git branch

Lists all branches in the repository.

```
git branch
```

Swipe →

19. git merge <branch>

Merges the specified branch into the current branch.

```
git merge <branch>
```

20. git log

Displays the commit history of the current branch.

```
git log
```

Swipe →

21. git fetch

Retrieves change from a remote repository, including new branches and commit.

```
git fetch
```

22. git pull

Fetches changes from the remote repository and merges them into the current branch.

```
git pull
```

Swipe →

23. git push

Pushes local commits to the remote repository.

```
git push
```

24. git revert <commit>

Creates a new commit that undoes the changes introduced by the specified commit.

```
git revert <commit>
```

Swipe →

25. git rebase <branch>

Reapplies commits on the current branch onto the tip of the specified branch.

```
git rebase <branch>
```

26. git remote

Lists all remote repositories.

```
git remote
```

Swipe →

HIT FOLLOW TO STAY UPDATED

Ruksar Nura Shaikh

Sr. Software Engineer