Final Exam Study Notes

How to Clone a GitHub Repository

Definition: Cloning downloads a copy of a remote GitHub repository to your computer.

Syntax:

```
git clone <repo_url>
```

Examples:

```
git clone https://github.com/username/repository.git

```

What it does: Creates a new folder containing the repository files and a hidden `.git` folder for version control.

How to Use Git Commands

Definition: Git is a version control system used to track changes and collaborate using repositories.

Common commands:

- `git add .` — stages *all* changes in the current repo folder
- `git commit -m "message"` — saves a snapshot (commit) with a message
- `git push` — uploads commits to GitHub
- `git pull` — downloads remote updates and merges them locally

Example workflow:

```
git add .
git commit -m "Add final exam study notes"
git push
```

How to Write a Markdown File With Images and Proper Formatting

Definition: Markdown is a lightweight markup language that formats text using simple symbols.

Formatting examples:

Headings

```
# Heading 1
## Heading 2
### Heading 3
```

Bold / Italic / Inline code

```
**bold**
*italic*
`inline code`
```

Lists

```
- item 1
- item 2

1. step 1
2. step 2
```

Images Definition: An image in Markdown uses the format `![alt text](path/to/image)`.

```
![Screenshot of terminal](screenshot.png)
```

- **Alt text** is inside `[]` (describes the image)
- **File path** is inside `()` (where the image is located)

How to Convert a Markdown File to PDF

Definition: Converting Markdown to PDF turns your `.md` notes into a printable document.

Right click and select Markdown PDF:Export (pdf)

---

5. How to Compress (Zip) a Directory/Folder in Debian

Definition: Compression reduces size and bundles multiple files/folders into one archive.

Syntax:

```
zip -r <archive_name.zip> <folder_name>
```

Examples:

```
zip -r final_exam_study_notes.zip final_exam_study_notes
zip -r notes_backup.zip final_exam_study_notes/
```

---

6. Absolute Paths and Relative Paths

Definition (Absolute path): A full path that starts from the root directory `/`.
Definition (Relative path): A path that starts from your *current* directory (does not begin with `/`).

Examples (absolute):

```
touch /home/miguel/Documents/test.txt
ls /var/log/apache2/
```

Examples (relative):

```
touch Documents/test.txt
ls ../Downloads
```

---

7. How to Work With Manual Pages (man Command)

Definition: `man` opens the manual (documentation) page for a command.

Syntax:

```
man <command>
```

Examples:

```
man ls
man grep
```

---

8. How to Parse (Search) for Specific Words in the Manual Page

Definition: Searching inside `man` lets you find a word quickly.

Steps:

1. Open a man page:

   ```
   man ls
   ```

2. Press `/` and type the word (example: `sort`), then press Enter.
3. Press `n` to find the next match, `N` to go to the previous match.

---

9. How to Redirect Output (>, >>, and |)

Definition (redirection): Sending command output somewhere else (like into a file) instead of the screen.

`>` overwrite redirection

Definition: Writes output to a file and overwrites it if it already exists.

```
ls > files.txt
```

`>>` append redirection

Definition: Adds output to the end of a file (does not erase existing contents).

```
ls >> files.txt
```

`|` pipe

Definition: Sends the output of one command as the input to another command.

```
ls | grep ".txt"
```

---

10. How to Append the Output of a Command to a File

Definition: Appending means adding new output to the end of an existing file.

Examples:

```
date >> log.txt
echo "New entry" >> log.txt
ls -l >> log.txt
```

---

11. How and When to Redirect Output of One Command to Another (Pipes)

Definition: A pipe `|` is used when you want to *filter* or *process* output using another command.

Examples:

```
ps aux | grep apache
cat /etc/passwd | cut -d: -f1
man grep | head -n 20
```

---

12. Use `echo` and Output Redirection to Create a New File With Text

Definition: `echo` prints text; using `>` writes it into a file.

Examples:

```
echo "Hello World" > hello.txt
echo "Miguel Ponce" > name.txt
echo "Line 1" > notes.txt
```

---

13. Use Wildcards (Copy/Move Multiple Files at the Same Time)

Definition: Wildcards are special characters that match multiple filenames.

Common wildcards:

- `*` = matches any characters
- `?` = matches one character

Examples:

```
cp *.txt backup/
mv *.jpg images/
ls file?.txt
```

---

14. Use Brace Expansion (Create Directory Structures in One Command)

Definition: Brace expansion generates multiple strings/paths at once, saving time.

Examples:

```
mkdir -p project/{css,js,images}
mkdir -p lab/{week1,week2,week3}/{notes,files}
touch file{1..5}.txt
```

---

15. Create a Simple “Hello World” Shell Script

Definition: A shell script is a text file containing commands that the shell runs in order.

Create the script file:

```
touch hello.sh
```

Edit `hello.sh` and add:

```
#!/bin/bash
echo "Hello World"
```

---

16. Use Variables in a Shell Script

Definition: A variable stores a value (text/number) that you can reuse.

Example script (`vars.sh`):

```
#!/bin/bash
name="Miguel"
course="CIS106"
echo "Student: $name"
echo "Course: $course"
```

---

17. Command Reference (Definition + Syntax + 2–5 Examples)

a) `awk`

Definition: Text processing tool that prints/filters fields (columns) based on patterns.
Syntax:

```
awk 'pattern { action }' file
```

Examples:

```
awk '{print $1}' names.txt
awk -F: '{print $1}' /etc/passwd
awk '/root/ {print $0}' /etc/passwd
```

---

b) `cat`

Definition: Displays file contents and can combine files.
Syntax:

```
cat [options] <file>
```

Examples:

```
cat file.txt
cat file1.txt file2.txt
cat -n file.txt
```

---

c) `cp`

Definition: Copies files/directories.
Syntax:

```
cp [options] source destination
```

Examples:

```
cp notes.txt backup_notes.txt
cp notes.txt backup/
cp -r project/ project_backup/
```

---

d) `cut`

Definition: Extracts sections of each line (by delimiter or by character positions).
Syntax:

```
cut -d '<delimiter>' -f <field_number> file
```

Examples:

```
cut -d: -f1 /etc/passwd
cut -d, -f2 data.csv
echo "a:b:c" | cut -d: -f2
```

---

e) `grep`

Definition: Searches text for lines that match a pattern.
Syntax:

```
grep [options] "pattern" file
```

Examples:

```
grep "root" /etc/passwd
grep -i "error" syslog.txt
grep -n "main" program.c
```

---

f) `head`

Definition: Shows the first lines of a file.
Syntax:

```
head [options] file
```

Examples:

```
head file.txt
head -n 5 file.txt
head -n 20 /var/log/apache2/access.log
```

---

g) `ls`

Definition: Lists directory contents.
Syntax:

```
ls [options] [path]
```

Examples:

```
ls
ls -l
ls -a
ls -lh /var/log
```

---

h) `man`

Definition: Opens the manual page for a command.
Syntax:

```
man command
```

Examples:

```
man cp
man awk
man man
```

---

i) `mkdir`

Definition: Creates directories (folders).
Syntax:

```
mkdir [options] directory_name
```

Examples:

```
mkdir test
mkdir -p projects/linux/week1
mkdir -p a/b/c
```

---

j) `mv`

Definition: Moves or renames files/directories.
Syntax:

```
mv [options] source destination
```

Examples:

```
mv old.txt new.txt
mv report.pdf Documents/
mv *.txt backup/
```

---

k) `tac`

Definition: Like `cat`, but prints the file in reverse line order (bottom to top).
Syntax:

```
tac file
```

Examples:

```
tac file.txt
tac /etc/passwd | head
tac log.txt | grep "ERROR"
```

---

l) `tail`

Definition: Shows the last lines of a file (useful for logs).
Syntax:

```
tail [options] file
```

Examples:

```
tail file.txt
tail -n 10 file.txt
tail -f /var/log/apache2/access.log
```

---

m) `touch`

Definition: Creates an empty file or updates file timestamps.
Syntax:

```
touch file
```

Examples:

```
touch notes.txt
touch file{1..3}.txt
touch /home/miguel/Documents/test.txt
```

---

n) `tr`

Definition: Translates or deletes characters from input.
Syntax:

```
tr 'set1' 'set2'
```

Examples:

```
echo "hello" | tr a-z A-Z
echo "abc123" | tr -d 0-9
cat file.txt | tr -s ' '
```

---

o) `tree`

Definition: Displays directories in a tree (hierarchy) format.
Syntax:

```
tree [options] [path]
```

Examples:

```
tree
tree final_exam_study_notes/
tree -L 2 /etc
```