Part 1: Outputs of Commands

1. pwd

The `pwd` command displays the present working directory.

```
$ pwd
/home/user
```

2. cd

The `cd` command is used to change directories.

```
$ cd Documents/
$ pwd
/home/user/Documents
```

3. ls

The `ls` command lists the files and directories in the current directory.

```
$ ls
file1.txt  file2.txt  Documents  Downloads
```

4. mkdir

The `mkdir` command is used to create a new directory.

```
$ mkdir NewFolder
$ ls
NewFolder  file1.txt  file2.txt
```

5. rm

The `rm` command is used to remove a file.

```
$ rm file1.txt
$ ls
file2.txt  NewFolder
```

6. touch

The `touch` command creates a new empty file.

```
$ touch newfile.txt
$ ls
newfile.txt  file2.txt  NewFolder
```

7. hostname

The `hostname` command displays the system's hostname.

```
$ hostname
my-laptop
```

8. cat

The `cat` command is used to view the contents of a file.

```
$ cat newfile.txt
Hello, this is a test file.
```

9. chmod

The `chmod` command changes file permissions.

```
$ chmod 777 newfile.txt
```

10. echo

The `echo` command prints a message to the terminal.

```
$ echo "Hello, World!"
Hello, World!
```

11. grep

The `grep` command searches for a pattern in a file.

```
$ grep "Hello" newfile.txt
Hello, this is a test file.
```

12. fgrep

The `fgrep` command searches for a fixed string in a file.

```
$ fgrep "Hello" newfile.txt
Hello, this is a test file.
```

13. mv

The `mv` command moves or renames a file.

```
$ mv newfile.txt oldfile.txt
$ ls
oldfile.txt
```

14. cp

The `cp` command copies a file.

```
$ cp oldfile.txt copyfile.txt
$ ls
oldfile.txt  copyfile.txt
```

15. more

The `more` command displays file content page by page.

```
$ more largefile.txt
```

16. less

The `less` command is similar to `more`, allowing backward navigation.

```
$ less largefile.txt
```

17. wc

The `wc` command counts words, lines, and characters in a file.

```
$ wc oldfile.txt
5  10  50 oldfile.txt
```

18. awk

The `awk` command is used for pattern scanning and processing.

```
$ awk '{print $1}' oldfile.txt
```

19. sed

The `sed` command is used for stream editing.

```
$ sed 's/Hello/Hi/' oldfile.txt
```

20. tail

The `tail` command shows the last lines of a file.

```
$ tail -n 5 oldfile.txt
```

Part 2: Answering Questions with Commands

1. **Navigate to a Specific Directory**:

```
cd /path/to/directory
```

2. **See detailed information about files and directories using `ls`**:

```
ls -l
```

3. **Create multiple directories using `mkdir`**:

```
mkdir dir1 dir2 dir3
```

4. **Remove multiple files at once**:

```
rm file1.txt file2.txt file3.txt
```

5. **Delete directories using `rm`**:

```
rm -r directory_name
```

6. **Copy files and directories**:

```
cp file1.txt /destination/path/
cp -r directory_name /destination/path/
```

7. **Rename a file using `mv`**:

```
mv oldname.txt newname.txt
```

8. **Move multiple files using `mv`**:

```
mv file1.txt file2.txt /destination/path/
```

9. **Create multiple empty files using `touch`**:

```
touch file1 file2 file3
```

10. **View content of multiple files**:

```
cat file1.txt file2.txt
```

11. **Create a file and add content using `cat`**:

```
cat > newfile.txt
Hello, this is new content.
(Ctrl+D to save)
```

12. **Append contents of one file to another**:

```
cat file1.txt >> file2.txt
```

13. **View large files with `cat` and paging**:

```
cat largefile.txt | less
```

14. **Merge multiple files using `cat`**:

```
cat file1.txt file2.txt > merged.txt
```

15. **Append to an existing file using `cat`**:

```
Cat >> existingfile.txt
Additional content here.
(Ctrl+D to save)
```

16. **Explanation of `chmod` commands**:

- `chmod 777` – Gives full permissions (read, write, execute to all users).
- `chmod 755` – Owner has full permissions, others have read and execute.
- `chmod +x` – Adds execute permission to a file.

17. **Find the number of lines matching a pattern**:

```
grep -c "pattern" filename.txt
```

18. **Display files containing a specific string**:

```
grep -l "pattern" *.txt
```

19. **Show line numbers of matched lines**:

```
grep -n "pattern" filename.txt
```

20.**Match lines starting with a string**:

```
grep "^pattern" filename.txt
```

21.**Sort files in descending order**:

```
 sort -r filename.txt
```

22.**Sort a file based on a specific column**:

```
sort -k 2 filename.txt
```
