LAB 1: UNIX SHELL COMMANDS

1. Execute and write output of all the commands explained so far in this manual

1.1 Special Characters

Many of these characters have special meaning when used in shell commands.

Character	Example	Output/Effect
\	touch filename*	Refers to filename* literally
/	/usr/src/unix	Directory path separator
.	ls -a (lists .hidden files)	.filename shows hidden file
..	cd ..	Goes up one directory
~	cd ~	Goes to home directory
*	ls *.txt	Lists all .txt files
?	ls file?.txt	Files: file1.txt, fileA.txt etc
[...]	ls hello[1-2].txt	Lists hello1.txt, hello2.txt
|	ls | more	Paginate output
>	ls > out.txt	Output to (overwrites) out.txt
>>	echo "Hello" >> out.txt	Append to out.txt
<	more < out.txt	File redirected to program
<<	tr a-z A-Z << EOF	Inline text processing (see below)
<<<	bc <<< 2+2	Processes direct string
;	cd /tmp ; ls	Execute sequentially
&&	mkdir test && cd test	Executes cd only if mkdir succeeds
&	sleep 10 &	Background process, get shell prompt

1.2 Shell Commands and Getting Help

```
$ date
Fri Aug  8 17:29:12 IST 2025

$ echo $PATH
/usr/local/bin:/usr/bin:/bin:/usr/local/sbin:/usr/sbin:/sbin

$ chmod +x script.sh
$ ./script.sh
# runs script.sh (if executable)
```

Getting Help:

- `$ man ls` – display manual page for `ls`
- `$ grep --help` – display help for grep

1.3 Navigating the UNIX File System

```
$ pwd
/home/user

$ cd /usr/bin
$ pwd
/usr/bin

$ cd ~
$ pwd
/home/user

$ cd ..
$ pwd
/home

$ cd -
# returns to previous directory
```

1.4 Listing Files

```
$ ls
file1.txt  file2.txt  folderA

$ ls /var/log
messages  boot.log  dmesg

$ ls -l
-rw-r--r-- 1 user user 34 Aug  8 17:29 file1.txt
drwxr-xr-x 3 user user 99 Aug  8 16:00 folderA

$ ls -a
.  ..  .bashrc  file1.txt  folderA

$ ls -ld folderA
drwxr-xr-x 3 user user 99 Aug  8 16:00 folderA
```

1.5 File Patterns / Wildcards

```
$ ls *.txt
file1.txt  data.txt

$ ls file?.txt
file1.txt

$ ls [a-c]*.txt
a.txt  b.txt  cfile.txt

# Explanation:
# '*' matches any string, '?' matches single character, '[abc]' matches a, b, or c
```

1.6 Working With Files and Directories

```
$ touch abc xyz mno
$ ls
abc  xyz  mno

$ file /bin/ls
/bin/ls: ELF 64-bit LSB executable, ...

$ cat abc
# (shows contents of file abc; if empty, nothing displays)

$ head abc
# Displays first few lines (default: 10)

$ tail abc
# Displays last few lines

$ cp abc /tmp/
$ mv xyz xyz_renamed

$ rm mno

$ mkdir tmpdir
$ rmdir tmpdir
# (removes directory if empty)
```

1.7 Finding Files

```
$ which grep
/usr/bin/grep

$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz

$ locate mozilla
/etc/mozilla/firefox/profile

$ find . -name "*.sh"
./script.sh
```

1.8 Piping and Redirection

```
$ ls -la /usr/bin | less
# Output paged by less

$ ls -l *.mp3 > mp3files.txt
# Output redirected to mp3files.txt

$ ls -l *.mp3 >> mp3files.txt
# Appends to the file
```

1.9 Shortcuts

- **Ctrl+C:** Halts current command
- **Ctrl+Z:** Stops the current command, backgrounds it
- **Ctrl+D:** Logout/Exit shell
- **Ctrl+W:** Erases one word
- **Ctrl+U:** Erases whole line
- **!!:** Repeats last command
- **exit:** Logout

2. Lab Exercises Answers With Sample Output

1. Execute and write output of all the commands explained so far.

Example session:

```
$ pwd
/home/user

$ ls -l
total 4
-rw-r--r-- 1 user user    0 Aug  8 17:29 abc
drwxr-xr-x 2 user user 4096 Aug  8 17:22 folderA

$ echo "Hello World" > hello.txt
$ cat hello.txt
Hello World

$ cp hello.txt hello_copy.txt
$ mv hello_copy.txt sample.txt
$ rm sample.txt

$ mkdir testdir
$ rmdir testdir
```

2. Explore the following commands along with their various options:

a. cat

```
$ cat > newfile.txt
Hello OS Lab Manual
# (Ctrl+D to save)
$ cat newfile.txt
Hello OS Lab Manual

$ cat file1.txt >> file2.txt
# Appends content of file1.txt to file2.txt
```

b. head and tail

```
$ head -n 3 hello.txt
Hello World
Line2
Line3

$ tail -c 4 hello.txt
rld
```

```
# -n for number of lines, -c for bytes
```

c. cp

```
$ cp -n file1.txt file2.txt
# Does not overwrite file2.txt if it exists

$ cp -i file1.txt file2.txt
# Prompts before overwriting

$ cp -f file1.txt file2.txt
# Forces overwrite
```

d. mv

```
$ mv -i file1.txt file2.txt
# Prompts before overwriting

$ mv -f file1.txt file2.txt
# No prompt, overwrites

# Move files to directory
$ mv abc folderA/
```

e. rm

```
$ rm -r folderA
# Removes directory and its contents

$ rm -i file1.txt
# Prompts before deleting

$ rm -f file1.txt
# Forces delete, no prompt
```

f. rmdir

```
$ rmdir testdir
# Works only if empty

$ rm -r testdir
# Removes recursively
```

g. find

```
$ find . -name "*.sh"
./script.sh

$ find . -type d
# Lists directories
```

3. List all the file names satisfying following criteria

Suppose files: alpha.txt, 1beta.txt, omega123.txt, txtfile, cry.txt, my_file

```
# a. Has the extension .txt
$ ls *.txt
alpha.txt 1beta.txt omega123.txt cry.txt

# b. Containing at least one digit
$ ls *[0-9]*
1beta.txt omega123.txt

# c. Minimum length of 4
$ find . -name '????*'
alpha.txt omega123.txt

# d. Does not contain any of the vowels as the start letter
$ ls [^aeiouAEIOU]*
cry.txt my_file 1beta.txt
```