

# OS Lab Report – Week 2

PES1201800366

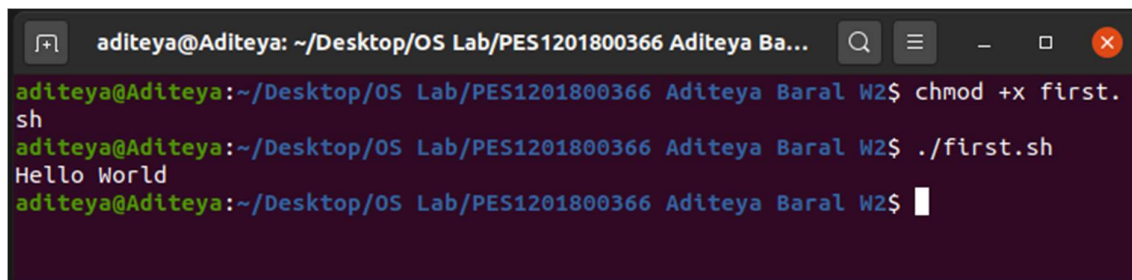
Aditeya Baral

## 1. First.sh

### 1.1 Code

```
1. #!/bin/sh
2. # This is a comment!
3. echo Hello World      # This is a comment, too!
```

### 1.2 Output

A terminal window titled 'aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...' with search, menu, and window control icons. The terminal shows the user running 'chmod +x first.sh' and then './first.sh', which outputs 'Hello World'.

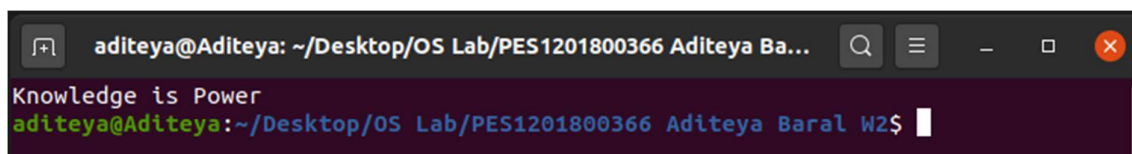
```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ chmod +x first.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./first.sh
Hello World
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 2. Second.sh

### 2.1 Code

```
1. #!/bin/sh
2. clear
3. echo "Knowledge is Power"
```

### 2.2 Output

A terminal window titled 'aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...' with search, menu, and window control icons. The terminal shows the user running 'Second.sh', which outputs 'Knowledge is Power' after clearing the screen.

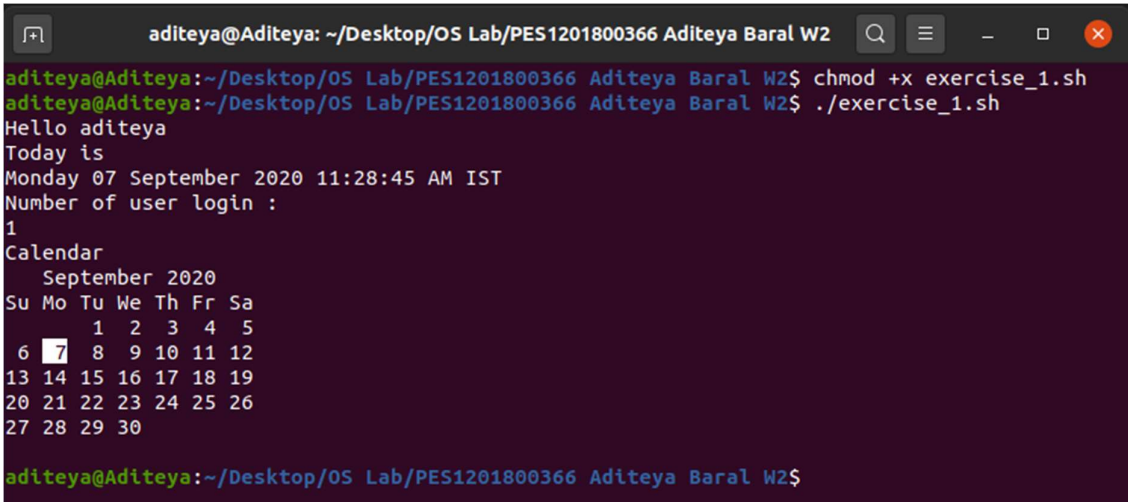
```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
Knowledge is Power
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

### 3. Exercise 1

#### 3.1 Code

```
1. #!/bin/sh
2. # Script to print user information who currently login, current date
   & time
3. # Enter the following commands in a file
4. echo "Hello $USER"
5. echo "Today is ";date
6. echo "Number of user login : " ; who | wc -l
7. echo "Calendar"
8. cal
9. exit 0
```

#### 3.2 Output

A terminal window titled 'aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2'. The user runs 'chmod +x exercise\_1.sh' and then './exercise\_1.sh'. The script outputs: 'Hello aditeya', 'Today is', 'Monday 07 September 2020 11:28:45 AM IST', 'Number of user login :', '1', 'Calendar', and a calendar for September 2020. The calendar shows the 7th as the current date. The terminal prompt returns to 'aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2\$'.

#### 3.3 Reason behind `exit 0`

Every Linux or Unix command executed by the shell script or user has an exit status which is an integer number. `exit 0` status means the command was successful without any errors. A non-zero (1-255 values) exit status means the command was a failure.

#### 3.4 Significance of `$`

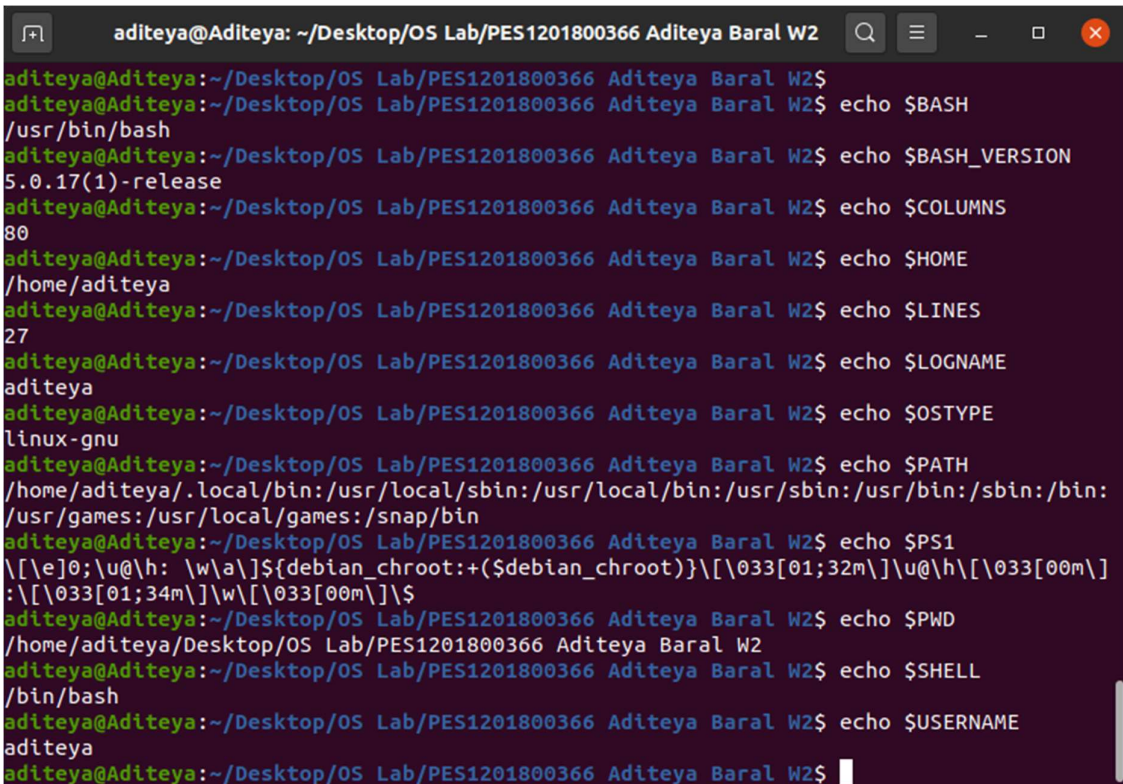
In a shell script, the `$` symbol is used to access the value stored in an identifier or variable. If the `$` symbol is not used as a prefix before the variable name, the shell will just display the name of a variable since the shell will read it as an ordinary string to be displayed.

### 4. System Variables

## 4.1 Code

```
1. #!/bin/sh
2. echo $BASH
3. echo $BASH_VERSION
4. echo $COLUMNS
5. echo $HOME
6. echo $LINES
7. echo $LOGNAME
8. echo $OSTYPE
9. echo $PATH
10. echo $PS1
11. echo $PWD
12. echo $SHELL
13. echo $USERNAME
```

## 4.2 Output



The screenshot shows a terminal window titled "aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2". The terminal displays the output of the script executed in section 4.1. The output is as follows:

```
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $BASH
/usr/bin/bash
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $BASH_VERSION
5.0.17(1)-release
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $COLUMNS
80
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $HOME
/home/aditeya
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $LINES
27
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $LOGNAME
aditeya
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $OSTYPE
linux-gnu
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $PATH
/home/aditeya/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $PS1
\[ \e]0;\u@\h: \w\a\]${debian_chroot:+($debian_chroot)}\[ \033[01;32m\]\u@\h\[ \033[00m\]:\[ \033[01;34m\]\w\[ \033[00m\] \]$
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $PWD
/home/aditeya/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $SHELL
/bin/bash
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $USERNAME
aditeya
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 5. Exercise 2

### 5.1 Code

```
1. echo $HOME
```

```
1. echo HOME
```

## 5.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo $HOME
/home/aditeya
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo HOME
HOME
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 5.3 Explanation

Hence, `echo $HOME` is the right command to display the home directory, since `HOME` is a system variable, and values stored in variables in a shell script can only be accessed using a `$` symbol as a prefix. Without the `$` symbol, the shell will read it as a normal string and display `HOME`.

## 6. User Defined Variables

### 6.1 Code

```
1. #!/bin/sh
2. name="Aditeya Baral"
3. srn=PES1201800366
4. age=20
5. echo "Name\t$name"
6. echo "SRN\t$srn"
7. echo "Age\t$age"
```

### 6.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...$ chmod +x third.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./third.sh
Name    Aditeya Baral
SRN     PES1201800366
Age     20
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 7. Echo

### 7.1 Code

```
1. #!/bin/sh
2. echo -e "hello \\ bye"
3. echo -e "hello \r bye"
4. echo -e "hello \n bye"
5. echo -e "hello \c bye"
6. echo -e "hello \b bye"
```

```
7. echo -e "hello \a bye"
8. echo -e "An apple a day keeps away \a\t\tdoctor\n"
```

## 7.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \\ bye"
hello \ bye
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \r bye"
bye
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \n bye"
hello
bye
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \c bye"
hello
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \b bye"
hello bye
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "hello \a bye"
hello  bye
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo -e "An apple a da
y keeps away \a\t\tdoctor\n"
An apple a day keeps away          doctor
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 8. Shell Arithmetic

### 8.1 Code

```
1. expr 1 - 2
2. expr 1 \* 2
3. expr 1 + 2
```

### 8.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ expr 1 - 2
-1
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ expr 1 \* 2
2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ expr 1 + 2
3
```

## 9. Exercise 3

### 9.1 Code

```
1. #!/bin/sh
2. expr 1 + 3
3. expr 2 - 1
4. expr 10 / 2
5. expr 20 % 3
6. expr 10 \* 3
7. echo `expr 6 + 3`
```

## 9.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./exercise_03.sh
4
1
5
2
30
9
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 10. Exercise 4

**Question -** What is the meaning of Single quote (‘), Double quote (“) and Back quote (`) in shell?

**Answer –** The shell understands special characters (such as escape sequences) with special meanings. For example, `$variable` is used to expand and obtain the value stored in variable. It also expands wildcards (such as `*` and `?`). However, sometimes, we need to display them as is. In such cases, we can use the various quoting methods.

- **Single Quotes (‘):**
  - The single quote is used to remove the special meaning of **any character** enclosed within them i.e. the special characters are treated as ordinary strings
  - Variables, wildcards as well as command substitutions are disabled
  - Example: `echo ‘$SHELL’` will display `$SHELL`

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo '$SHELL'
$SHELL
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

- **Double Quotes (“):**
  - The double quote is used to remove the special meaning of **most characters** enclosed within them i.e. most special characters are treated as ordinary strings
  - Only wildcards are disabled
  - This does not apply to `$`, `‘`, `“`, ```, `\$`, and `\`
  - Example: `echo “$SHELL”` will display `/bin/bash`



```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo "$SHELL"
/bin/bash
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

- **Back Quotes (`):**

- The back quote is used to **execute any command** enclosed within back quotes
- Example: `echo `expr 6 + 3`` will display 9

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ echo `expr 6 + 3`
9
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 11. Wildcards

### 11.1 Code

```
1. #!/bin/sh
2. ls *.sh
3. ls exercise_0?.sh
4. ls [ex]*.sh
```

### 11.2 Output

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls *.sh
exercise_03.sh exercise_1.sh first.sh fourth.sh second.sh third.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls exercise_0?.sh
exercise_03.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls [ex]*.sh
exercise_03.sh exercise_1.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 12. Redirection of Standard IO

### 12.1 Output Redirection

```
1. ls > filenames.txt
2. cat filenames.txt
```

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls > filenames.txt
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ cat filenames.txt
exercise_03.sh
exercise_1.sh
filenames.txt
first.sh
fourth.sh
second.sh
third.sh
WEEK 2 - OS LAB manual for Shell Scripting.docx
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 12.2 Output Redirection with Redirector

```
1. date >> filenames.txt
2. cat filenames.txt
```

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ date >> filenames.txt
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ cat filenames.txt
exercise_03.sh
exercise_1.sh
filenames.txt
first.sh
fourth.sh
second.sh
third.sh
WEEK 2 - OS LAB manual for Shell Scripting.docx
Monday 07 September 2020 08:24:28 PM IST
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 12.3 Input Redirection

```
1. wc -l filenames.txt
```

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ wc -l < filenames.txt
9
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 13. Exercise 5

Question – What does the following command do?

```
1. sort < myfile > sorted_file
```

**Answer** – The given command will first use input redirection to obtain the contents (lines) of myfile, which is sent to the sort command. After sorting the lines in alphabetical order, the output is redirected to sorted\_file and stored.



```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ sort filenames.txt > sorted_filenames.txt
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ cat sorted_filenames.txt
exercise_1.sh
exercise_3.sh
filenames.txt
first.sh
fourth.sh
Monday 07 September 2020 08:35:33 PM IST
second.sh
sorted_filenames.txt
third.sh
WEEK 2 - OS LAB manual for Shell Scripting.docx
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 14. Pipes

### 14.1 Code

```
1. ls | more
2. who | sort
3. who | sort > user_list.txt
4. who | wc -l
5. ls -l | wc -l
6. who | grep aditeya
```

### 14.2 Output

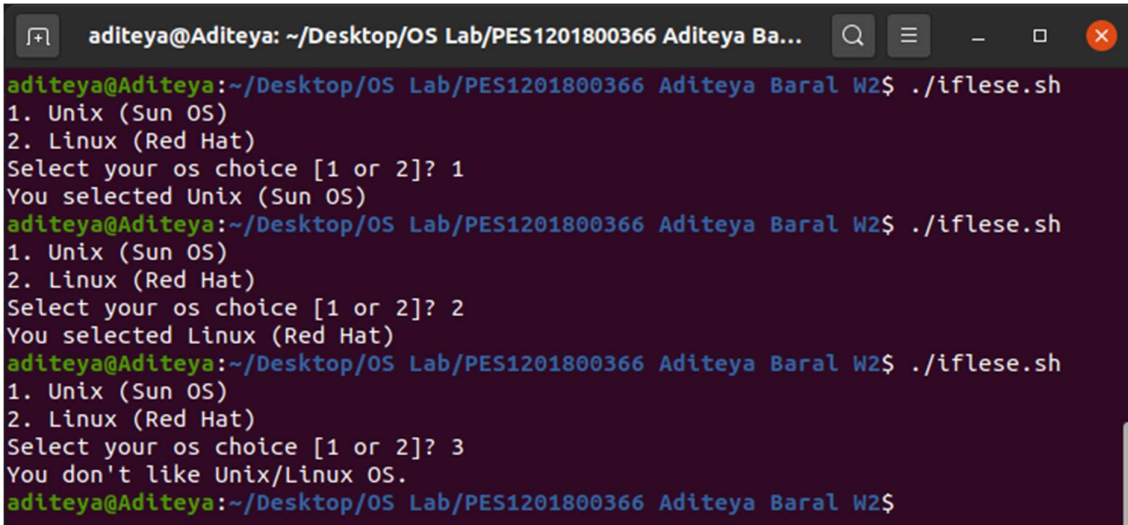
```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls | more
exercise_1.sh
exercise_3.sh
filenames.txt
hello_world.sh
knowledge_is_power.sh
op
pipe.sh
sorted_filenames.txt
user_list
user_list.txt
variables.sh
WEEK 2 - OS LAB manual for Shell Scripting.docx
wildcard.sh
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ who | sort
aditeya :0                2020-09-07 19:24 (:0)
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ who | sort > user_list
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ who | wc -l
1
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ls -l | wc -l
14
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ who | grep aditeya
aditeya :0                2020-09-07 19:24 (:0)
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 15. if-else-fi Construct

## 15.1 Code

```
1. #!/bin/sh
2. osch=0
3.
4. echo "1. Unix (Sun OS)"
5. echo "2. Linux (Red Hat)"
6. echo -n "Select your os choice [1 or 2]? "
7. read osch
8.
9. if [ $osch -eq 1 ]
10. then
11.     echo "You selected Unix (Sun OS)"
12.
13. else
14.     if [ $osch -eq 2 ]
15.     then
16.         echo "You selected Linux (Red Hat)"
17.     else
18.         echo "You don't like Unix/Linux OS."
19.     fi
20. fi
```

## 15.2 Output



```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./ifleese.sh
1. Unix (Sun OS)
2. Linux (Red Hat)
Select your os choice [1 or 2]? 1
You selected Unix (Sun OS)
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./ifleese.sh
1. Unix (Sun OS)
2. Linux (Red Hat)
Select your os choice [1 or 2]? 2
You selected Linux (Red Hat)
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./ifleese.sh
1. Unix (Sun OS)
2. Linux (Red Hat)
Select your os choice [1 or 2]? 3
You don't like Unix/Linux OS.
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 16. Loops

### 16.1 For Loop

```
1. #!/bin/sh
2. for ((i = 0; i <= 4; i++))
3. do
4.     echo "Welcome $i times"
5. done
```

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./for.sh
Welcome 0 times
Welcome 1 times
Welcome 2 times
Welcome 3 times
Welcome 4 times
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 16.2 While Loop

```
1. #!/bin/sh
2. a=0
3. while [ $a -lt 10 ]
4. do
5.     echo $a
6.     a=`expr $a + 1`
7. done
```

```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./while.sh
0
1
2
3
4
5
6
7
8
9
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$
```

## 16.3 Until Loop

```
1. #!/bin/sh
2. a=0
3. until [ ! $a -lt 10 ]
4. do
5.     echo $a
6.     a=`expr $a + 1`
7. done
```

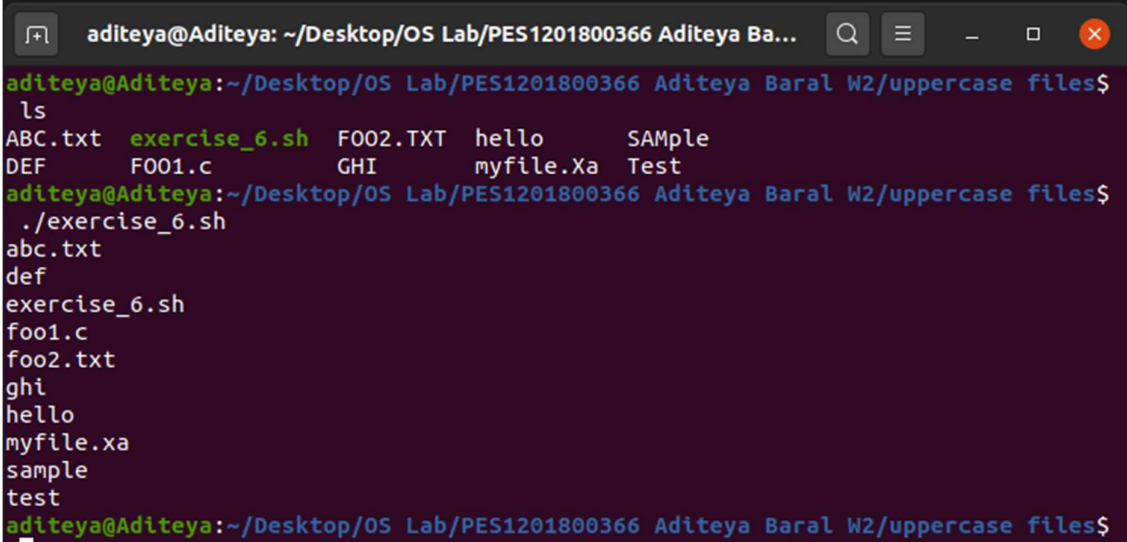
```
aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Ba...
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2$ ./until.sh
0
1
2
3
4
5
6
7
8
9
```

## 17. Exercise 6

### 17.1 Code

```
1. #!/bin/sh
2. ls | tr '[:upper:]' '[:lower:]'
```

### 17.2 Output

A terminal window titled 'aditeya@Aditeya: ~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2/uppercase files...' shows the execution of a script. The prompt is 'aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2/uppercase files\$'. The first command is 'ls', which outputs a list of files: 'ABC.txt', 'exercise\_6.sh', 'F002.TXT', 'hello', 'SAMPLE', 'DEF', 'F001.c', 'GHI', 'myfile.Xa', and 'Test'. The second command is './exercise\_6.sh', which outputs the same list of files in lowercase: 'abc.txt', 'def', 'exercise\_6.sh', 'foo1.c', 'foo2.txt', 'ghi', 'hello', 'myfile.xa', 'sample', and 'test'.

```
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2/uppercase files$
ls
ABC.txt  exercise_6.sh  F002.TXT  hello      SAMPLE
DEF      F001.c        GHI       myfile.Xa  Test
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2/uppercase files$
./exercise_6.sh
abc.txt
def
exercise_6.sh
foo1.c
foo2.txt
ghi
hello
myfile.xa
sample
test
aditeya@Aditeya:~/Desktop/OS Lab/PES1201800366 Aditeya Baral W2/uppercase files$
```

\*\*\*