

1. Introduction to Linux Shell and Shell Scripting

What is Shell

A shell is a special user program which provides an interface to users to use operating system services. Shell accept human readable commands from the user and convert them into something which the kernel can understand. It is a command language interpreter that executes commands read from input devices such as keyboards or from files. The shell gets started when the user logs in or starts the terminal.

Shell is broadly classified into two categories –

- Command Line Shell
- Graphical shell
- BASH (Bourne Again SHell) – It is the most widely used shell in Linux systems. It is used as the default login shell in Linux systems and in macOS. It can also be installed on Windows OS.

Shell Scripting

Usually shells are interactive, that means, they accept commands as input from users and execute them. However, we want to execute a bunch of commands routinely, so we have to type in all commands each time in the terminal.

As shell can also take commands as input from file we can write these commands in a file and can execute them in shell to avoid this repetitive work. These files are called Shell Scripts or Shell Programs. Shell scripts are similar to the batch file in MS-DOS. Each shell script is saved with .sh file extension eg. myscript.sh

2. Program to perform arithmetic operations on two numbers

IN f.sh

```
read -p 'Enter a: ' a
read -p 'Enter b: ' b
add=$((a + b))
echo "Sum is: $add"
sub=$((a - b))
echo "Difference is: $sub"
mul=$((a * b))
```

```

echo "Product is: $mul"
div=$((a / b))
echo "Quotient is: $div"
mod=$((a % b))
echo "Modulus is: $mod"
((++a))
echo $a
((--b))
echo $b

```

IN terminal:

```

student@t2:~/Desktop/Script$ ./f.sh
Enter a: 4
Enter b: 2
Sum is: 6
Difference is: 2
Product is: 8
Quotient is: 2
Modulus is: 0
5
1

```

3. Write a program to pass arguments through command line and print first argument, second argument, total number of arguments passed and entire argument list.

```

read -a name
echo "First argument: ${name[0]} Second argument: ${name[1]} Total number of arguments:
${#name[@]} Entire argument list: ${name[@]}"

```

```

student@t2:~/Desktop/Script$ ./demo.sh
Kim Jim
First argument: Kim Second argument: Jim Total number of arguments: 2 Entire argument list: Kim
Jim
student@t2:~/Desktop/Script$

```

4. Program to find the largest of two numbers

```

read -p 'Enter value of a ' a
read -p 'Enter value of b ' b
if [ $a -gt $b ]
then
echo "Value of a is greater than b"
else

```

```
echo "Value of b is greater than a"
fi
```

```
student@t2:~/Desktop/Script$ ./demo.sh
Enter value of a 2
Enter value of b 4
Value of b is greater than a
```

5. Program to check the given no is odd or even [pass argument through command line]

```
num=$1
if (( num % 2 == 0 )); then
    echo "$num is even."
else
    echo "$num is odd."
fi
```

```
student@t2:~/Desktop/Script$ ./demo.sh 5
5 is odd.
student@t2:~/Desktop/Script$ ./demo.sh 4
4 is even.
```

6. Program to print day of week name using case.

- IN day.sh:

```
read -p "Enter a day number: " day
case "$day" in
    1) echo "SUNDAY";;
    2) echo "MONDAY";;
    3) echo "TUESDAY";;
    4) echo "WEDNESDAY";;
    5) echo "THURSDAY";;
    6) echo "FRIDAY";;
    7) echo "SATURDAY";;
    *) echo "Enter a number less than 8 !!";;
esac
```

IN terminal:

```
student@t2:~/Desktop/Script$ ./day.sh
Enter a day number: 1
SUNDAY
student@t2:~/Desktop/Script$ ./day.sh
```

Enter a day number: 9
Enter a number less than 8 !!
student@t2:~/Desktop/Script\$

7. Program to check the entered character is digit, alphabet or other character using case

IN alpha.sh:

```
read -p "Enter input " i
case "$i" in
    [1-9]*) echo "Number";;
    [a-zA-Z]*) echo "Alphabet";;
    *) echo "Special Character";;
esac
```

IN terminal:

```
student@t2:~/Desktop/Script$ ./alpha.sh
Enter input A
Alphabet
student@t2:~/Desktop/Script$ ./alpha.sh
Enter input 4
Number
student@t2:~/Desktop/Script$ ./alpha.sh
Enter input *
Special Character
```

8. Program to print array elements and sum of elements in the array.

9. Program to Print the odd numbers between 1 to 100

```
for i in {1..100..2}
do
echo $i
done
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
1
3
5
7
9
```

11
13
15
17
19
21
23
25
27
29
31
33
35
37
39
41
43
45
47
49
51
53
55
57
59
61
63
65
67
69
71
73
75
77
79
81
83
85
87
89
91
93
95
97

99

10. Program to Print Fibonacci series up to 50

```
i=1
j=1
echo $i
s=1
while [ $s -le 50 ]
do
echo $s
i=$j
j=$s
s=$((i + $j))
done
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
```

```
1
1
2
3
5
8
13
21
34
```

11. Program to print prime numbers between 1 to 100

```
#!/bin/bash
echo "prime numbers upto 100 are : "
echo 1
i=2
while [ $i -le 100 ]
do
    flag=1
    j=2
    while [ $j -lt $i ]
    do
        rem=$(( $i % $j ))
        if [ $rem -eq 0 ]
        then
            flag=0
        fi
    done
    if [ $flag -eq 1 ]
    then
        echo $i
    fi
    i=$((i + 1))
done
```

```
        break
    fi
    j=$(( $j+1 ))
done
if [ $flag -eq 1 ]
then
    echo "$i"
fi
i=$(( $i+1 ))
done
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
prime numbers upto 100 are :
```

```
1
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
53
59
61
67
71
73
79
83
89
97
```

12. Program to check whether the given number is Armstrong or not

```
read -p "Enter any number : " no
rev=0
```

```

temp=$no
while [ $no -ne 0 ]
do
dig=$(( $no%10 ))
rev=$(( $rev+$dig*$dig*$dig ))
no=$(( $no/10 ))
done
if [ $rev -eq $temp ]
then
echo "Armstrong"
else
echo "Not Armstrong"
fi

```

```

student@t2:~/Desktop/Script$ ./loop1.sh
Enter any number : 152
Not Armstrong
student@t2:~/Desktop/Script$ ./loop1.sh
Enter any number : 153
Armstrong

```

13. Program to Check whether the given number is Palindrome or not.

```

echo "Enter a Number: "
read n
temp=$n
while ((temp>0))
do
ans=$((ans*10))
mod=$((temp%10))
ans=$((ans+mod))
temp=$((temp/10))
done
if (($n == $ans))
then
echo "It's a Palindrome Number"
else
echo "It's not a Palindrome Number"
fi

```



```
student@t2:~/Desktop/Script$ ./loop1.sh
Enter a Number:
121
It's a Palindrome Number
student@t2:~/Desktop/Script$ ./loop1.sh
Enter a Number:
122
It's not a Palindrome Number
```

14. Program to print the reverse of a number

```
read -p "Enter a number: " no
n=$no
s=0
while [ $no -gt 0 ]
do
    r=$(( $no % 10 ))
    s=$(( $s * 10 + $r ))
    no=$(( ($no - $r) / 10 ))
done
echo "The reverse is "$s
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
Enter a number: 123
The reverse is 321
```

15. Program to print the pattern

```
*
**
***
****
*****
```

```
for i in {1..5}
do
    for j in $(seq 1 $i)
    do
        echo -n " "
```

```
done
echo
done
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
*
**
***
****
*****
```

16. Program to print the following pattern

```
1
22
333
4444
55555
```

```
for i in {1..5}
do
for j in $(seq 1 $i)
do
echo -n $i
done
echo
done
```

```
student@t2:~/Desktop/Script$ ./loop1.sh
1
22
333
4444
55555
```