

## Shell Scripting

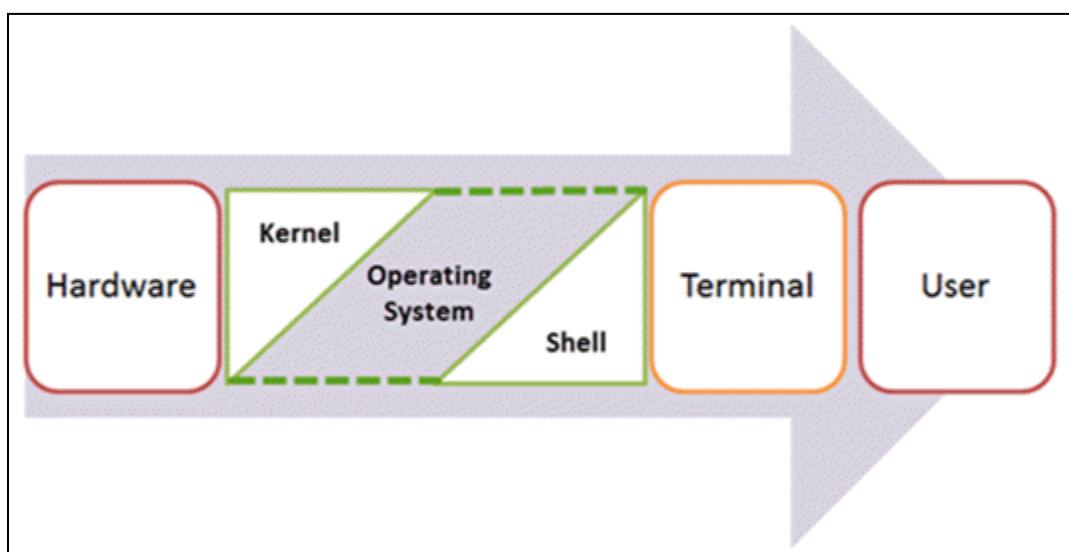
Shell Scripting is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime which reduces programming efforts.

### What is Shell?

Shell is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program's output in a shell script.

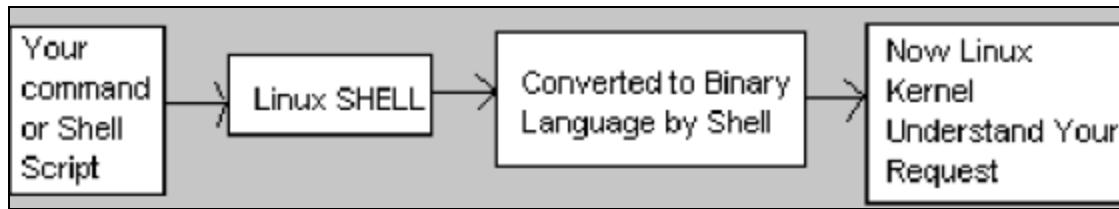
An Operating is made of many components, but its two prime components are –

- Kernel
- Shell



Components of Shell Program

Computer understand the language of 0's and 1's called binary language, In earlydays of computing, instruction are provided using binary language, which is difficult for all of us, to read and write. So in O/s there is special program called Shell. Shell accepts your instruction or commands in English and translates it into computers native binary language.



**A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.**

**A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.**

**When you run the terminal, the Shell issues a command prompt (usually \$), where you can type your input, which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.**

### Types of Shell

**There are two main shells in Linux:**

**1. The Bourne Shell:** The prompt for this shell is \$ and its derivatives are listed below:

- POSIX shell also is known as sh
- Korn Shell also knew as sh
- Bourne Again SHell also knew as bash (most popular)

**2. The C shell:** The prompt for this shell is %, and its subcategories are:

- C shell also is known as csh
- Tops C shell also is known as tcsh

### How to Write Shell Script in Linux/Unix

**Shell Scripts are written using text editors. On your Linux system, open a text editor program, open a new file to begin typing a shell script or shell programming, then give the shell permission to execute your shell script and put your script at the location from where the shell can find it.**

**Let us understand the steps in creating a Shell Script:**

1. Create a file using a vi editor(or any other editor). Name script file with extension .sh
2. Start the script with #! /bin/sh
3. Write some code.
4. Save the script file as filename.sh
5. For executing the script type bash filename.sh

**“#!” is an operator called shebang which directs the script to the interpreter location. So, if we use “#! /bin/sh” the script gets directed to the bourne-shell.**

Let's create a small script –

```
#!/bin/sh  
ls
```

Creating a new script file scriptsample.sh

```
home@VirtualBox:~$ vi scriptsample.sh
```

Adding the command 'ls' after #!/bin/sh

```
#!/bin/sh  
ls  
~
```

Executing the script file

```
home@VirtualBox:~$ bash scriptsample.sh  
abc Desktop newfile samp  
ABC Documents newt.txt scr  
ABC~ Downloads Pictures Temp  
abc.bash examples.desktop Public tes  
abcd.sh help sample test
```

## Shell script Programs

Write a shell script to find a greater number out of 3 numbers.  
(June-2013-Old)(May-2018-Old) [NLJET]

Write a Shell Script to find the largest among the 3 given numbers. (May-2019-Old)  
[NLJET]

#shell script to find the greatest of three numbers

```
echo "Enter Num1"  
read num1  
echo "Enter Num2"  
read num2  
echo "Enter Num3"  
read num3  
  
if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]  
then  
    echo $num1  
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]  
then
```

```
echo $num2
else
    echo $num3
fi
```

**read** function used to get the input from the user.

**-gt** stands for greater than.

**&&** represents the logical AND condition.

**Output:**

## Output

```
Enter Num1
1
Enter Num2
34
Enter Num3
2
34
```

```
Enter Num1
500
Enter Num2
5
Enter Num3
6
500
```

**Write a Shell Script to find the factorial of a given number. (Nov-2016-Old)  
(Dec-2013-Old) [NLJIET]**

```
#shell script for factorial of a number
#factorial using while loop
```

```
echo "Enter a number"
read num

fact=1

while [ $num -gt 1 ]
do
    fact=$((fact * num)) #fact = fact * num
    num=$((num - 1))    #num = num - 1
done

echo $fact
```

gt stands for greater than (>).

(or)

```
#shell script for factorial of a number
#factorial using for loop
```

```
echo "Enter a number"
read num

fact=1

for((i=2;i<=num;i++))
{
    fact=$((fact * i)) #fact = fact * i
}

echo $fact
```

**Output:**

## Output

```
Enter a number
```

```
3
```

```
6
```

```
Enter a number
```

```
4
```

```
24
```

```
Enter a number
```

```
5
```

```
120
```

**Write following Shell scripts in Unix/Linux :**

**(i)To find the Factorial of a given number.**

```
#shell script for factorial of a number  
#factorial using while loop  
  
echo "Enter a number"  
read num  
  
fact=1  
  
while [ $num -gt 1 ]  
do  
    fact=$((fact * num)) #fact = fact * num  
    num=$((num - 1))    #num = num - 1  
done  
  
echo $fact
```

**gt stands for greater than (>).**

## Output

```
Enter a number
```

```
3
```

```
6
```

```
Enter a number
```

```
4
```

```
24
```

```
Enter a number
```

```
5
```

```
120
```

(ii)For generating students' mark sheets. Calculate "CLASS" also. (Jan-2016-Old)  
[NLJET]

```
echo "Enter the three subject marks for the student"
read m1 m2 m3
sum1=`expr $m1 + $m2 + $m3`
echo "Sum of 3 subjects are: " $sum1
per=`expr $sum1 / 3`
echo " Percentage: " $per
if [ $per -ge 60 ]
then
echo "You get Distinction"
elif [ $per -ge 50 ]
then
echo "You get First class"
elif [ $per -ge 40 ]
then
echo "You get Second class"
else
    echo "You get Fail"
fi
```

## OUTPUT

Enter the three subject marks for the student

45

35

45

Sum of 3 subjects are: 125

Percentage : 41.67

You get second class

**Write a shell script to create a menu for basic arithmetic operations.  
(June-2016-Old)[NLJET]**

```
#!/bin/bash

clear

sum=0

i="y"

echo " Enter first no."

read n1

echo "Enter second no."

read n2

while [ $i = "y" ]

do

echo "1.Addition"
```

```
echo "2.Subtraction"
echo "3.Multiplication"
echo "4.Division"
echo "Enter your choice"
read ch
case $ch in
    1)sum=`expr $n1 + $n2`
       echo "Sum ="$sum;;
    2)sum=`expr $n1 - $n2`
       echo "Sub ="$sum;;
    3)sum=`expr $n1 \* $n2`
       echo "Mul ="$sum;;
    4)sum=`echo "scale=2;$n1/$n2"|bc`
       echo "div ="$sum;;
    *)echo "Invalid choice";;
esac
echo "Do u want to continue ?[y/n]"
read i
if [ $i != "y" ]
then
    exit
fi
done
```

(or)

```
a=10  
b=20  
  
val=`expr $a + $b`  
echo "a + b : $val"  
  
val=`expr $a - $b`  
echo "a - b : $val"  
  
val=`expr $a \* $b`  
echo "a * b : $val"  
  
val=`expr $b / $a`  
echo "b / a : $val"  
  
val=`expr $b % $a`  
echo "b % a : $val"  
  
if [ $a == $b ]  
then  
    echo "a is equal to b"  
fi  
  
if [ $a != $b ]  
then  
    echo "a is not equal to b"  
fi
```

**Write a shell script to display current month calendar.(June-2016-Old) [NLJET]**

*MENU*

*Display calendar of current month*

*Display today's date and time*

*Display usernames those are currently logged in the system*

*Display your name at given x, y position*

*Display your terminal number*

*Exit*

*clear*

```

while(i==0)
do
{
echo"enter choice from menu"
echo"1. Display calendar of current month
    2.Display today's date and time
    3.Display usernames those are currently logged in the system
    4. Display your name at given x, y position
    5. Display your terminal number
    6. Exit "
read choice
case "$choice" in
1)cal;;
2)date;;
3)who;;
4)tput cup 4 5
    whoami;;
5)tty;;
6)exit;;
output:-
enter choice from menu
1. Display calendar of current month
    2.Display today's date and time
    3.Display usernames those are currently logged in the system
    4. Display your name at given x, y position
    5. Display your terminal number
    6. Exit "

```

**Write following Shell scripts in Unix/Linux :**

**I. To find five largest files in the current directory.**

**II. To find Sum & Average of ‘n’ numbers. (Dec-2012-Old) [NLJIET]**

**Shell program to find sum of n numbers**

**#shell script to find sum of n numbers using for loop**

```

echo "Enter Size(N)"
read N

```

```

sum=0

```

```
echo "Enter Numbers"
for((i=1;i<=N;i++))
do
    read num      #get number
    sum=$((sum + num)) #sum+=num
done

echo $sum
```

---

## Output

```
Enter Size(N)
```

```
5
```

```
Enter Numbers
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
15
```

```
Enter Size(N)
```

```
3
```

```
Enter Numbers
```

```
10
```

```
20
```

```
30
```

```
60
```

## Shell program to find average of n numbers

```
#shell script to find average of n numbers
```

```
echo "Enter Size(N)"
read N

i=1
sum=0

echo "Enter Numbers"
while [ $i -le $N ]
do
    read num      #get number
    sum=$((sum + num))  #sum+=num
    i=$((i + 1))
done

avg=$(echo $sum / $N | bc -l)

echo $avg
```

## Output

```
Enter Size(N)  
5  
Enter Numbers  
10  
20  
30  
40  
50  
30.000000000000000000000000000000
```

```
Enter Size(N)  
3  
Enter Numbers  
1  
2  
5  
2.6666666666666666666666
```

## Shell Script to Convert a File to Lowercase

### INPUT:

line 1: filename

```
echo Enter file name  
read file  
a=`cat $file`  
echo $a | tr '[upper:]' '[lower:]' > $file
```

### OUTPUT:

```
$ Enter file name
```

```
$ test
```

### Shell Script to Print all Prime Numbers Between n and m

INPUT:

line 1: n and m

```
echo enter m and n
read m n
for a in $(seq $m $n)
do
    k=0
    for i in $(seq 2 ${expr $a - 1})
    do
        if [ ${expr $a % $i} -eq 0 ]
        then
            k=1
            break
        fi
    done
    if [ $k -eq 0 ]
    then
        echo $a
    fi
done
```

OUTPUT:

```
$ enter m and n
```

```
$ 3 10
```

```
$ 3
```

```
$ 5
```

```
$ 7
```

### Shell Script to Check if the Number is Palindrome or Not

INPUT:

line 1: number

```
echo enter n
read n
num=0
on=$n
while [ $n -gt 0 ]
do
num=$(expr $num \* 10)
k=$(expr $n % 10)
num=$(expr $num + $k)
n=$(expr $n / 10)
done
if [ $num -eq $on ]
then
echo palindrome
else
echo not palindrome
fi
```

Output:

```
$ enter n
```

```
$ 121
```

```
$ palindrome
```

### Shell Script to Reverse a Number

INPUT:

line 1: number

```
echo enter n
```

```
read n
```

```
num=0
while [ $n -gt 0 ]
do
    num=$(expr $num \* 10)
    k=$(expr $n % 10)
    num=$(expr $num + $k)
    n=$(expr $n / 10)
done
echo number is $num
```

**Output :**

```
$ enter n
$ 456
$ number is 654
```

### **Shell Script to Print the Sum of Digits of a Number**

**INPUT:**  
**line 1: number**

```
echo enter n
read n
sum=0
while [ $n -gt 0 ]
do
    r=$(expr $n % 10)
    sum=$(expr $sum + $r)
    n=$(expr $n / 10)
done
echo sum is $sum
```

**Output**

```
$ enter n
$ 456
$ sum is 15
```

### **Shell Script to Check if the Current Year is Leap Year or Not**

```
leap=$(date +"%Y")
echo taking year as $leap
if [ `expr $leap % 400` -eq 0 ]
then
echo leap year
```

```
elif [ `expr $leap % 100` -eq 0 ]
then
echo not a leap year
elif [ `expr $leap % 4` -eq 0 ]
then
echo leap year
else
echo not a leap year
fi
```

**Output**

```
$ taking year as 2019
$ not a leap year
```

**Shell Script to Print Good Morning, Good Afternoon, Good Evening and Good Night according to the System Time**

```
h=$(date +"%H")
if [ $h -gt 6 -a $h -le 12 ]
then
echo good morning
elif [ $h -gt 12 -a $h -le 16 ]
then
echo good afternoon
elif [ $h -gt 16 -a $h -le 20 ]
then
echo good evening
else
echo good night
fi
```

**Output**

```
$ good evening
```

**Shell Script to Check if the User is Logged in or Not**

**INPUT:**

The first line contains one string: username

**OUTPUT:**

Print “logged in” the user is logged in else print “not logged in”

```
echo enter username
read name
who > test
if grep $name test
then
echo logged in
else
```

```
echo not logged in
fi

Output
$ enter username
$ test
$ test  tty1      2019-02-21 10:56 (:0)
$ test  pts/0      2019-02-21 10:57 (:0)
$ test  pts/1      2019-02-26 19:19 (:0)
$ logged in
```

**Shell Script to Add Two Float Numbers**

```
echo enter a and b
read a b
c=`echo $a+$b | bc`
echo $c
output
$ enter a and b
$ 4.3 2.6
$ 6.9
```

**Shell Script to Multiply Two Numbers**

```
echo enter a and b
read a b
c=`expr $a \* $b`
echo $c
```

**Output**

```
$ enter a and b
$ 4 5
$ 20
```

**Shell Script to Add Two Numbers**

```
echo enter a and b
read a b
c=`expr $a + $b`
echo $c
```

**OUTPUT:**

```
$ enter a and b
```

```
$ 4 5
```

```
$ 9
```

### **Shell script program to swap two numbers**

```
#!/bin/bash

# Program name: "swap.sh"
# shell script program to swap two numbers.

num1=10
num2=20

echo "Before Swapping"
echo "Num1: $num1"
echo "Num2: $num2"

num3=$num1
num1=$num2
num2=$num3

echo "After Swapping"
echo "Num1: $num1"
echo "Num2: $num2"
```

#### **Output:**

```
$ sh swap.sh
Before Swapping
Num1: 10
Num2: 20
After Swapping
Num1: num2
Num2: num3
```

### **Sample program title: COPY THE CONTENTS OF ONE FILE TO ANOTHER FILE**

#### **ALGORITHM:**

- Step 1- Open vi editor and start the program*
- Step 2- Declare two file Pointer variable p1 and \*p2*
- Step 3- Read the source filename fn1*
- Step 4- Read the target filename fn2*
- Step 5- Assign p1=open the file fn1 in read mode*
- Step 6- Assign p2=open the file fn1 in write mode*
- Step 7- Test the condition if p1 is equal to NULL, true Print file fn1 not exist*

**Step 8- If the condition fails Read a character from File fn1 and assign to c**

**Step 9- While c is not equal to end of file and do Putc on File pointer \*p2 position Read a character from File fn1 and assign to c**

**Step 10-Close the file pointer p1 and p2**

```
#include<stdio.h>
#include<stdlib.h>
void main()
{
FILE *p1,*p2;
int c;
char fn1[40],fn2[40];
printf("\n Enter the source file ");
scanf("%s",fn1);
printf("\n Enter the Target file");
scanf("%s",fn2);
p1=fopen(fn1,"r");
p2=fopen(fn2,"w");
if(p1==NULL)
{
    printf("File %s not exist",fn1);
}
else
{
    c=getc(p1);
    while(c!=EOF)
    {
        putc(c,p2);
        c=getc(p1);
    }
fclose(p1);
fclose(p2);
printf("file %s is successfully copied to file %s",fn1,fn2);
}
```

INPUT	OUTPUT
[com35@localhost ~]\$ cat f1 Varuvan vadivelan Institute of Technology Gundalpatti  [com35@localhost ~]\$ cc exf1.c [com35@localhost ~]\$ ./a.out Enter the source file f1 Enter the Target file f2	file f1 is successfully copied to file f2 [com35@localhost ~]\$ cat f2 Varuvan vadivelan Institute of Technology Gundalpatti