

## LINUX PROGRAMMING WEEK 1

NAME : S KAILASH  
REG. NO : 18MIS1074  
SLOT : B1 + TB1  
COURSE CODE : SWE4009

=====

1. Write a shell script that deletes all lines containing the specified word in one or more files, supplied as arguments to it.

**AIM:** To delete all the lines containing a specified word in one or more files which are supplied as argument.

### ALGORITHM:

Enter file name/s as argument (positional parameters)

Enter word to delete

Use sed command with /d option and -i flag

Use cat command to list updated file

### PROGRAM:

```
#!/bin/bash
```

```
#we are going to use sed command
```

```
#short for stream editor
```

```
#used to list,replace,delete the files in a stream, etc
```

```
#here we are going to use /d option, to delete
```

```
#-i flag is used to delete in-place.
```

```
echo 'Delete words in file'
```

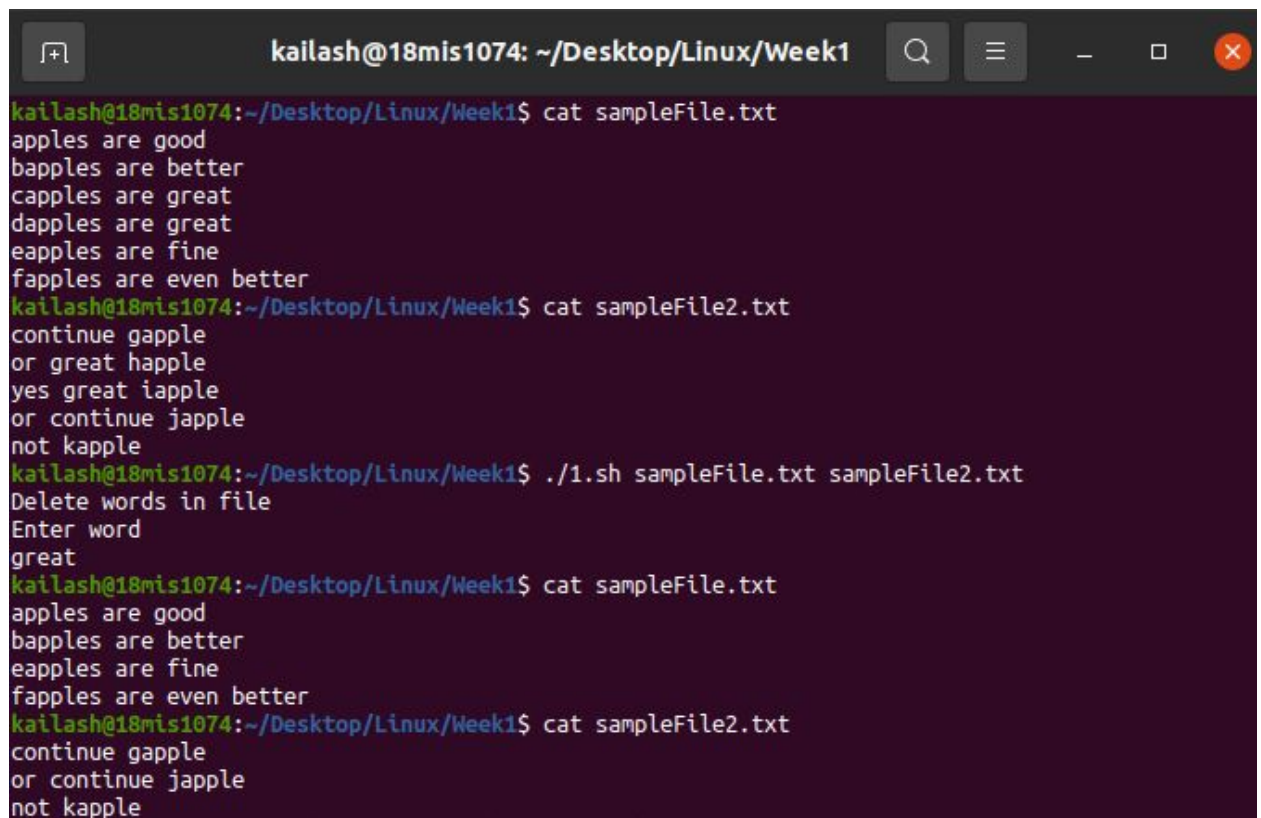
```
echo 'Enter word'
```

```
read word
```

`sed -i "$word/d" $*`

```
kailash@18mis1074:~/Desktop/Linux/Week1$ cat 1.sh
#!/bin/bash
#we are going to use sed command
#short for stream editor
#used to list,replace,delete the files in a stream, etc
#here we are going to use /d command, to delete
#-i flag is used to delete in-place.
echo 'Delete words in file'
echo 'Enter word'
read word
sed -i "$word/d" $*
kailash@18mis1074:~/Desktop/Linux/Week1$
```

**OUTPUT:** 2 files are **sampleFile.txt** and **sampleFile2.txt**



The terminal window shows the execution of the script on two files. The first file, sampleFile.txt, contains the following text:

```
apples are good
bapples are better
capples are great
dapples are great
eapples are fine
fapples are even better
```

The second file, sampleFile2.txt, contains the following text:

```
continue gapple
or great happle
yes great iapple
or continue japple
not kapple
```

The script is then executed on both files using the command `./1.sh sampleFile.txt sampleFile2.txt`. The output shows that the word "great" has been deleted from both files.

```
kailash@18mis1074:~/Desktop/Linux/Week1$ cat sampleFile.txt
apples are good
bapples are better
eapples are fine
fapples are even better
kailash@18mis1074:~/Desktop/Linux/Week1$ cat sampleFile2.txt
continue gapple
or continue japple
not kapple
```

**RESULT:**

Thus, we deleted the word great from 2 files consecutively.

2. Write a shell script that displays a list of all files in the current directory to which the user has read, write and execute permissions.

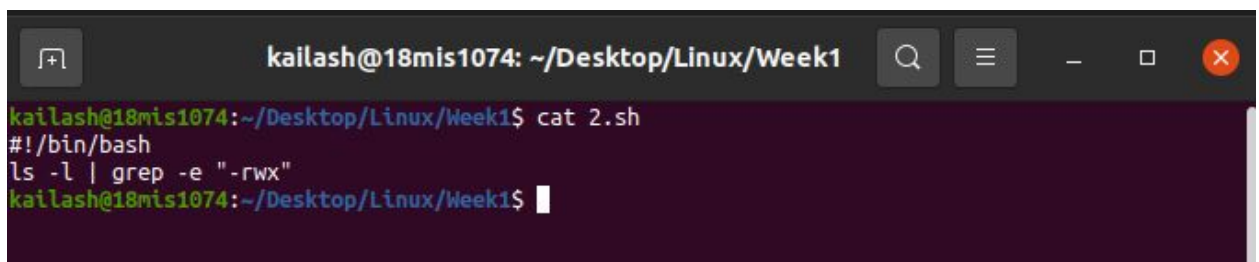
**AIM:** To list all files in the directory, where the user has read, write **and** execute permissions.

### ALGORITHM:

Use ls command with -l flag to long list information and pipeline with grep (global regular expression print) set to -rwx, with -e flag to read all patterns.

This lists all the files with read, write and execute permissions.

### PROGRAM:

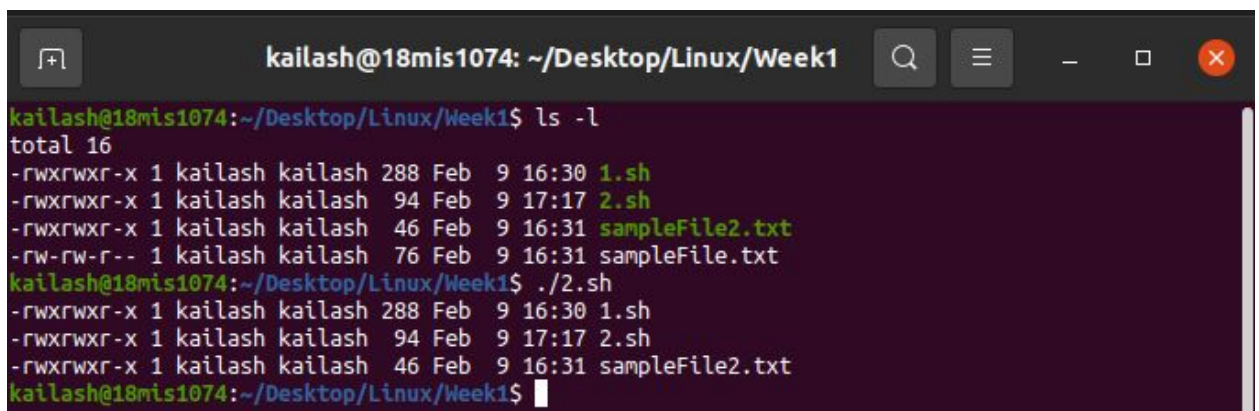


```
kailash@18mis1074: ~/Desktop/Linux/Week1
kailash@18mis1074:~/Desktop/Linux/Week1$ cat 2.sh
#!/bin/bash
ls -l | grep -e "-rwx"
kailash@18mis1074:~/Desktop/Linux/Week1$
```

```
#!/bin/bash
```

```
ls -l | grep -e "-rwx"
```

### OUTPUT:



```
kailash@18mis1074: ~/Desktop/Linux/Week1
kailash@18mis1074:~/Desktop/Linux/Week1$ ls -l
total 16
-rwxrwxr-x 1 kailash kailash 288 Feb  9 16:30 1.sh
-rwxrwxr-x 1 kailash kailash  94 Feb  9 17:17 2.sh
-rwxrwxr-x 1 kailash kailash  46 Feb  9 16:31 sampleFile2.txt
-rw-rw-r-- 1 kailash kailash  76 Feb  9 16:31 sampleFile.txt
kailash@18mis1074:~/Desktop/Linux/Week1$ ./2.sh
-rwxrwxr-x 1 kailash kailash 288 Feb  9 16:30 1.sh
-rwxrwxr-x 1 kailash kailash  94 Feb  9 17:17 2.sh
-rwxrwxr-x 1 kailash kailash  46 Feb  9 16:31 sampleFile2.txt
kailash@18mis1074:~/Desktop/Linux/Week1$
```

**RESULT:** Thus, the files with read, write and execute permissions were listed with detailed information.

3. Write a shell script that computes the gross salary of a employee according to the following
- if basic salary is <1500 then HRA 10% of the basic and DA =90% of the basic
  - if basic salary is >=1500 then HRA 500 and DA =98% of the basic

The basic salary is entered interactively through the keyboard.

### **AIM:**

To calculate the total salary, based on the user input provided and the formulae given.

### **ALGORITHM:**

Get input for basic salary

If basic salary<1500:

Hra = 0.10\*basic

Da = 0.90\*basic

Total = basic + Hra + Da

Print tot value

Endif

If basic salary>=1500

Hra=500

Da=0.98\*basic

Total = basic + Hra + Da

Print tot value

Endif

### **PROGRAM:**

```
kailash@18mis1074: ~/Desktop/Linux/Week1
kailash@18mis1074:~/Desktop/Linux/Week1$ cat 3.sh
#!/bin/bash
echo "Enter basic salary"
read basic

if [ $basic -lt 1500 ]; then
    hra=$(expr 0.1*$basic |bc)
    da=$(expr 0.9*$basic |bc)
    tot=$(expr "$basic"+"$hra"+"$da" |bc)
    echo "Gross salary is $tot"
fi
if [[ $basic -ge 1500 ]]; then
    hra=$(expr 500)
    da=$(expr 0.98*$basic |bc)
    tot=$(expr "$basic"+"$hra"+"$da" |bc)
    echo "Gross salary is $tot"
fi
kailash@18mis1074:~/Desktop/Linux/Week1$
```

```
#!/bin/bash
```

```
echo "Enter basic salary"
```

```
read basic
```

```
if [ $basic -lt 1500 ]; then
```

```
    hra=$(expr 0.1*$basic |bc)
```

```
    da=$(expr 0.9*$basic |bc)
```

```
    tot=$(expr "$basic"+"$hra"+"$da" |bc)
```

```
    echo "Gross salary is $tot"
```

```
fi
```

```
if [[ $basic -ge 1500 ]]; then
```

```
    hra=$(expr 500)
```

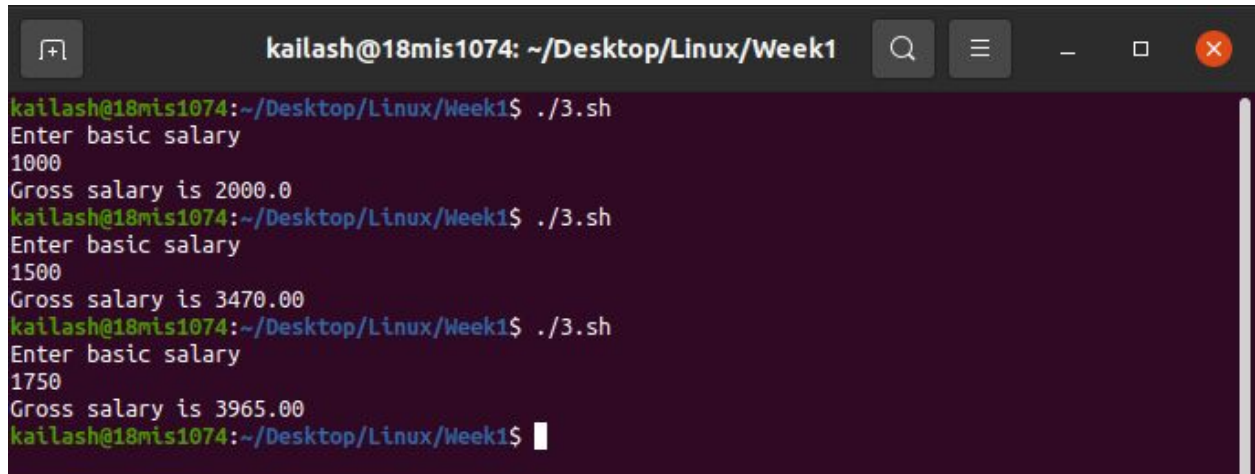
```
    da=$(expr 0.98*$basic |bc)
```

```
    tot=$(expr "$basic"+"$hra"+"$da" |bc)
```

```
    echo "Gross salary is $tot"
```

fi

## OUTPUT:



```
kailash@18mis1074: ~/Desktop/Linux/Week1
kailash@18mis1074:~/Desktop/Linux/Week1$ ./3.sh
Enter basic salary
1000
Gross salary is 2000.0
kailash@18mis1074:~/Desktop/Linux/Week1$ ./3.sh
Enter basic salary
1500
Gross salary is 3470.00
kailash@18mis1074:~/Desktop/Linux/Week1$ ./3.sh
Enter basic salary
1750
Gross salary is 3965.00
kailash@18mis1074:~/Desktop/Linux/Week1$
```

## RESULT:

Thus, the gross salary was calculated and displayed successfully on terminal.

---