

## Unix and Shell Programming Lab Course

### Objectives:

1. Learn UNIX Filters related to text processing, communication and search utilities
2. Learn programming filters and interactive shell scripting
3. Learn shell programming constructs writing advanced scripts
4. Learn kernel programming on file operations and managing processes

### Course Outcomes:

CO1: **Develop** chained commands to extract the required information from the system (**apply**)

CO2: **Navigate** in any Unix-flavoured operating system ( **Apply**)

CO3: **Develop** scripts compatible with different shells available under UNIX environment ( **Create**)

CO4: **Develop** scripts for automating the tasks of programmer during deployment and maintenance (**Create** )

CO5: **Develop** scripts to automate task using programmable filters (**Create**)

### List of Shell Scripts:

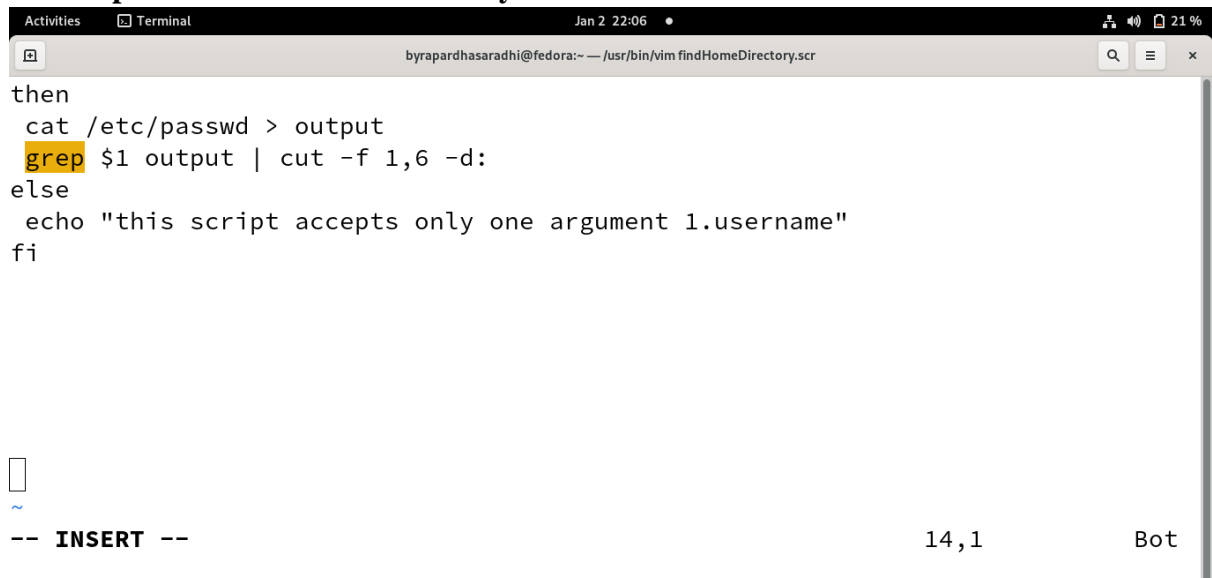
1. Create a script that, given a user name, finds the home directory of the user using the /etc/passwd file.

#### Preparation:

- None

#### Script:

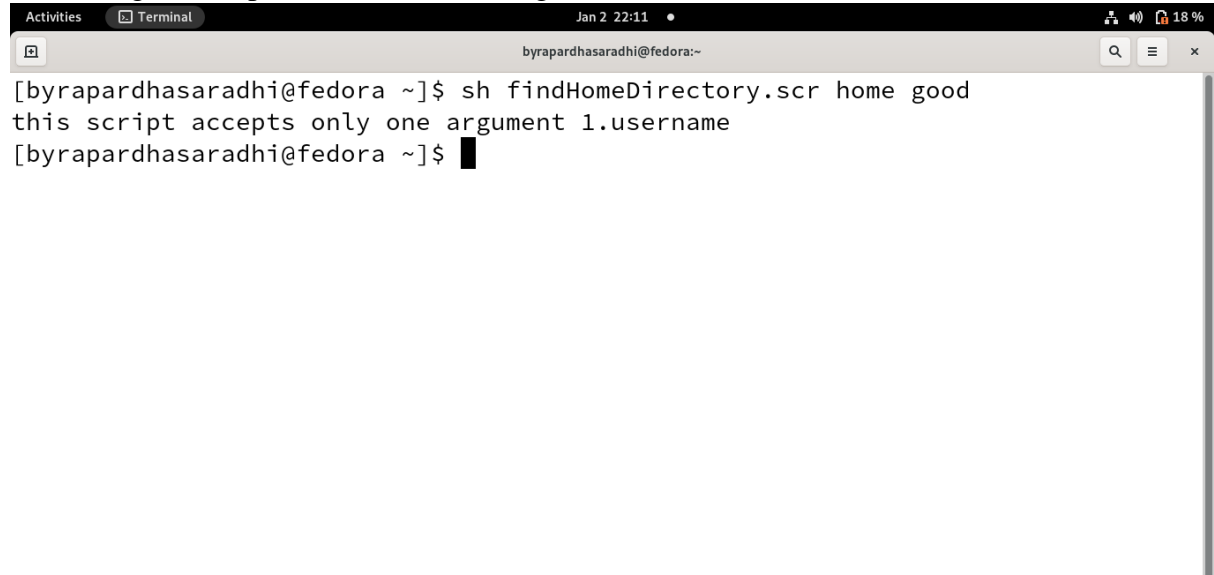
- **Script Name:** findHomeDirectory.scr



```
then
  cat /etc/passwd > output
  grep $1 output | cut -f 1,6 -d:
else
  echo "this script accepts only one argument 1.username"
fi
```

- **Arguments:** One, The user name.
- **Validation:** The minimum validation requirements are :
  - i. Ensure that there is only one argument.
- **Body Section:** Create a script that, given the name of a user (as the only argument), prints the absolute pathname of the user's home directory

- **Testing the Script:**
- Testing the script with two or more arguments.



A terminal window titled 'byrapardhasaradhi@fedora:~' with a search icon, menu icon, and close button. The terminal shows the command `sh findHomeDirectory.scr home good` being executed. The output is `this script accepts only one argument 1.username`. The prompt `[byrapardhasaradhi@fedora ~]$` is followed by a black cursor block.

```
[byrapardhasaradhi@fedora ~]$ sh findHomeDirectory.scr home good
this script accepts only one argument 1.username
[byrapardhasaradhi@fedora ~]$
```

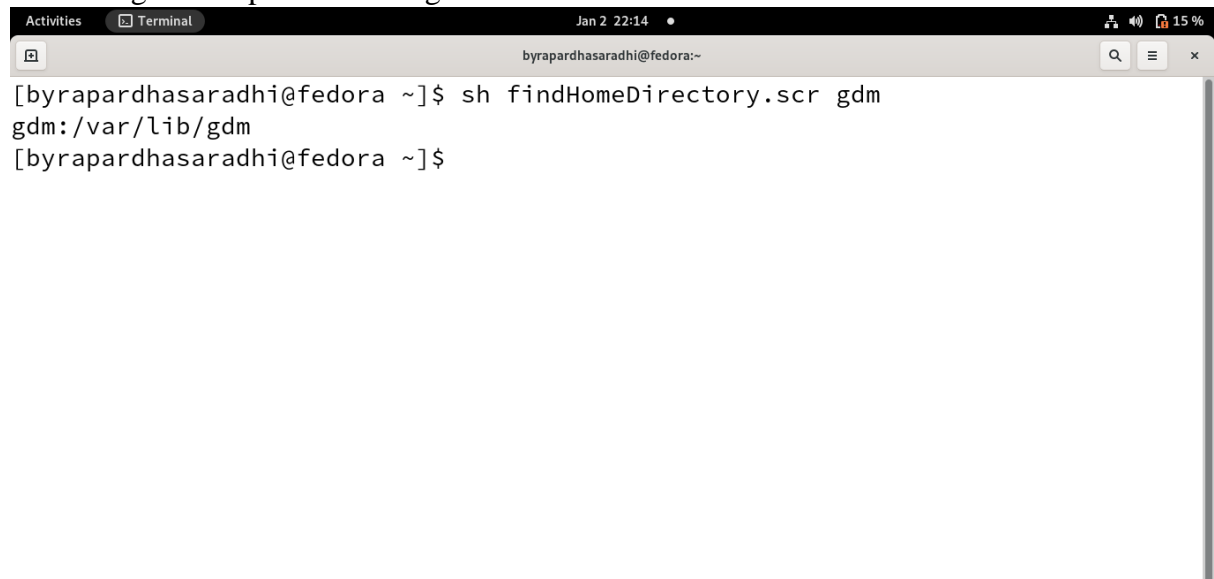
- Testing the script with no arguments.



A terminal window titled 'byrapardhasaradhi@fedora:~' with a search icon, menu icon, and close button. The terminal shows the command `sh findHomeDirectory.scr` being executed. The output is `this script accepts only one argument 1.username`. The prompt `[byrapardhasaradhi@fedora ~]$` is followed by a black cursor block.

```
[byrapardhasaradhi@fedora ~]$ sh findHomeDirectory.scr
this script accepts only one argument 1.username
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument.

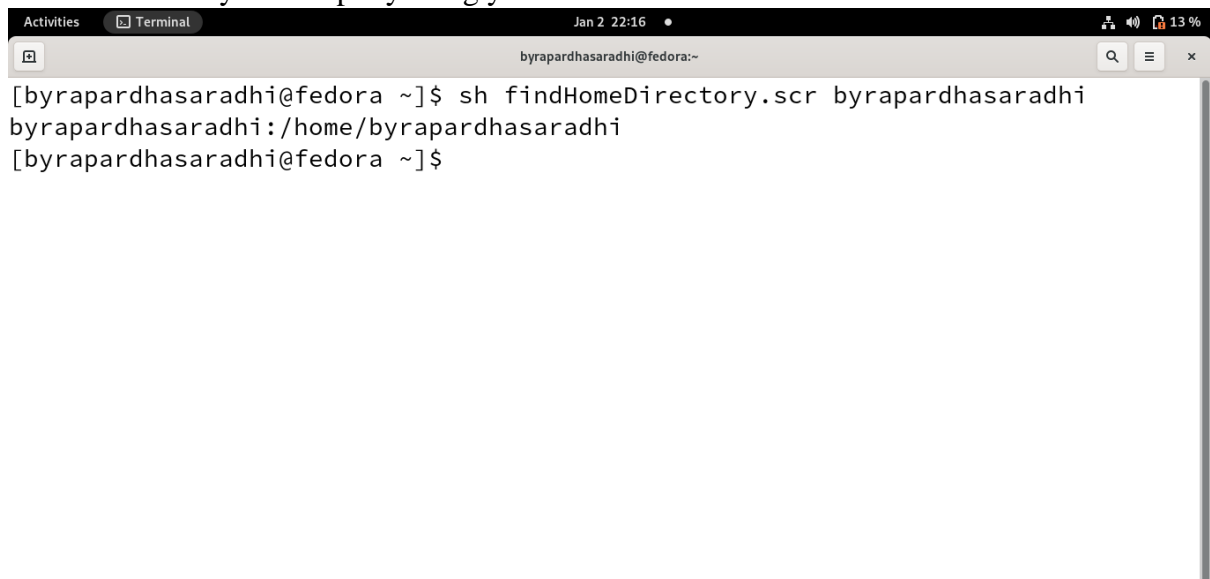


A terminal window titled 'byrapardhasaradhi@fedora:~' with a search icon, menu icon, and close button. The terminal shows the command `sh findHomeDirectory.scr gdm` being executed. The output is `gdm:/var/lib/gdm`. The prompt `[byrapardhasaradhi@fedora ~]$` is followed by a black cursor block.

```
[byrapardhasaradhi@fedora ~]$ sh findHomeDirectory.scr gdm
gdm:/var/lib/gdm
[byrapardhasaradhi@fedora ~]$
```

## Testing the Effect of the Script:

- Verify the script by using your user name.



```
byrapardhasaradhi@fedora ~]$ sh findHomeDirectory.scr byrapardhasaradhi
byrapardhasaradhi:/home/byrapardhasaradhi
[byrapardhasaradhi@fedora ~]$
```

**Result:-** The Home directory of the given user displayed on the screen.

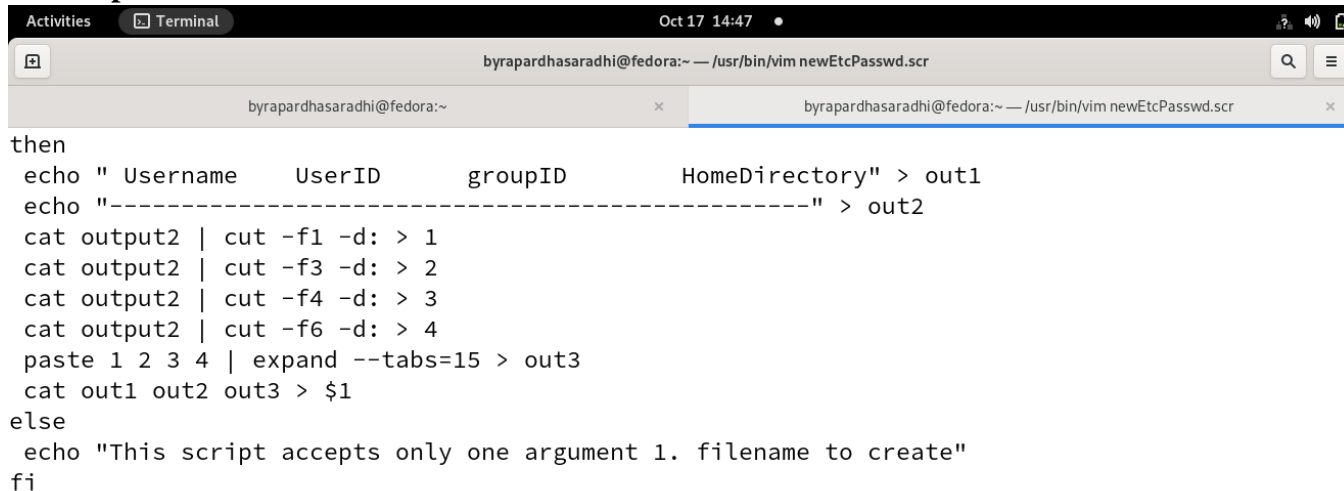
2. Write a script that creates a file out of the /etc/passwd file.

### Preparation:

- None

### Script:

- **Script Name:** newEtcPasswd.scr



```
then
echo " Username      UserID      groupID      HomeDirectory" > out1
echo "-----" > out2
cat output2 | cut -f1 -d: > 1
cat output2 | cut -f3 -d: > 2
cat output2 | cut -f4 -d: > 3
cat output2 | cut -f6 -d: > 4
paste 1 2 3 4 | expand --tabs=15 > out3
cat out1 out2 out3 > $1
else
echo "This script accepts only one argument 1. filename to create"
fi
```

■  
"newEtcPasswd.scr" 18L, 453B 18,1 Bo

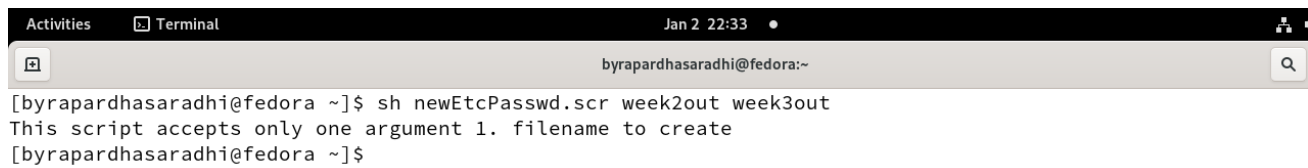
- **Arguments:** One, The name of the file.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that there is only one argument.
- **Body Section:** Create a script that makes a file out of the information in the /etc/passwd file using the following format.

User Name User Id Group ID Home Directory

-----  
ram 234 23 /etc/usr/student/ram  
----

## Testing the Script:

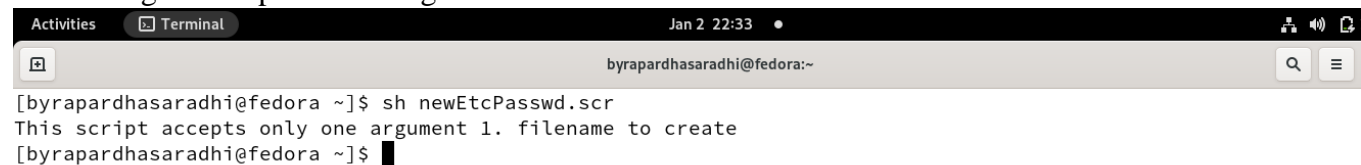
- Testing the script with two or more arguments.



A terminal window titled 'Terminal' with a timestamp of 'Jan 2 22:33'. The prompt is 'byrapardhasaradhi@fedora:~'. The user enters the command 'sh newEtcPasswd.scr week2out week3out'. The script outputs 'This script accepts only one argument 1. filename to create' and then returns to the prompt.

```
byrapardhasaradhi@fedora:~  
[byrapardhasaradhi@fedora ~]$ sh newEtcPasswd.scr week2out week3out  
This script accepts only one argument 1. filename to create  
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with no arguments.



A terminal window titled 'Terminal' with a timestamp of 'Jan 2 22:33'. The prompt is 'byrapardhasaradhi@fedora:~'. The user enters the command 'sh newEtcPasswd.scr'. The script outputs 'This script accepts only one argument 1. filename to create' and then returns to the prompt.

```
byrapardhasaradhi@fedora:~  
[byrapardhasaradhi@fedora ~]$ sh newEtcPasswd.scr  
This script accepts only one argument 1. filename to create  
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument that is not the name of a file.

```
Activities Terminal Jan 2 22:35 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh newEtcPasswd.scr !notfile
bash: !notfile: event not found
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument that is the name of a file.

```
Activities Terminal Jan 2 22:36 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh newEtcPasswd.scr week2out
Username      UserID      groupID      HomeDirectory
-----
root          0           0            /root
bin           1           1            /bin
daemon       2           2            /sbin
adm           3           4            /var/adm
lp           4           7            /var/spool/lpd
sync         5           0            /sbin
shutdown     6           0            /sbin
halt         7           0            /sbin
mail         8           12           /var/spool/mail
operator     11          0            /root
games        12          100          /usr/games
ftp          14          50           /var/ftp
nobody       65534       65534        /
apache       48          48           /usr/share/httpd
systemd-network 192        192          /
systemd-coredump 999        997          /
systemd-resolve 193        193          /
systemd-oom   998        996          /
systemd-timesync 997        995          /
tss          59          59           /dev/null
```

### Testing the Effect of the Script:

- Verify the file was created and contains the correct information and format.

```
Activities Terminal Jan 2 22:31 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh newEtcPasswd.scr week2out
Username      UserID      groupID      HomeDirectory
-----
root          0           0            /root
bin           1           1            /bin
daemon       2           2            /sbin
adm           3           4            /var/adm
lp           4           7            /var/spool/lpd
sync         5           0            /sbin
shutdown     6           0            /sbin
halt         7           0            /sbin
mail         8           12           /var/spool/mail
operator     11          0            /root
games        12          100          /usr/games
ftp          14          50           /var/ftp
nobody       65534       65534        /
apache       48          48           /usr/share/httpd
systemd-network 192        192          /
systemd-coredump 999        997          /
systemd-resolve 193        193          /
systemd-oom   998        996          /
systemd-timesync 997        995          /
tss          59          59           /dev/null
```

**Result:-** The Required file was created and the result is as shown above.

3. In a C Program, there is only one comment format. All comments must start with an open comment token, /\*, and end with a close comment token, \*/. C++ programs use the C tokens for comments

that span several lines. Single-line comments start with two slashes (//). In either case, the start token can be anywhere on the line.

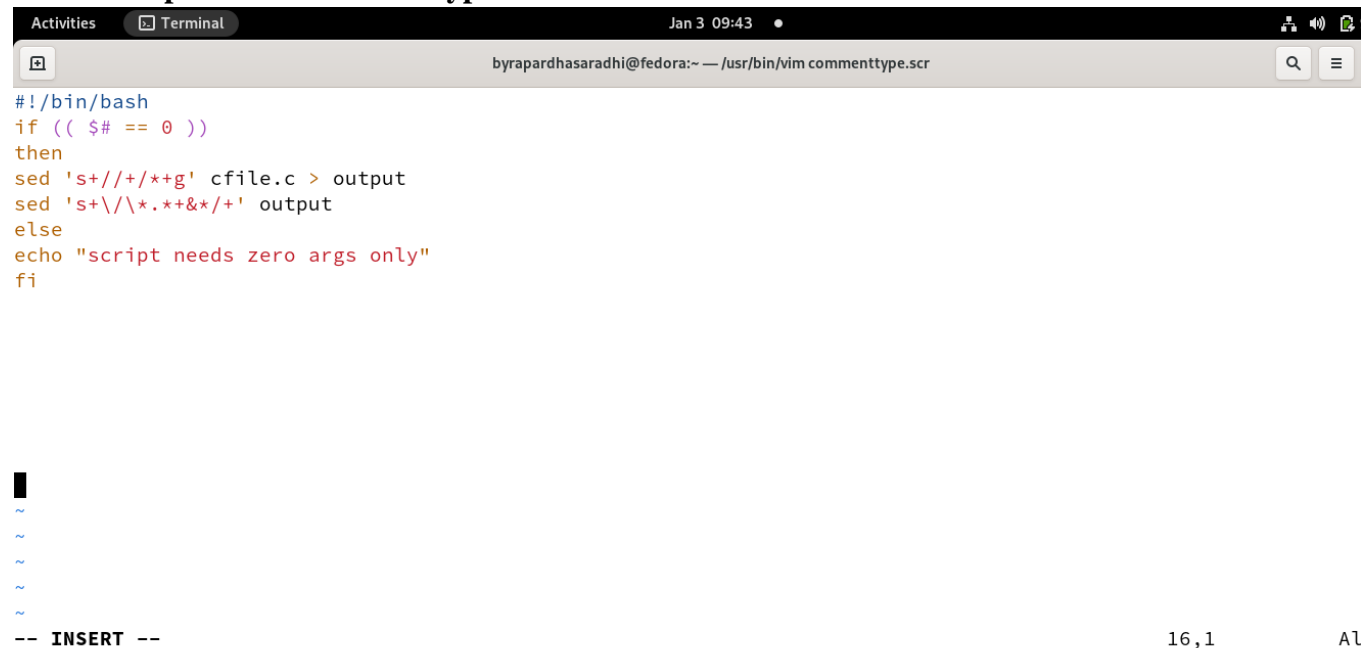
Write a script to change every single-line comment in a C++ source file that uses C program start and end comment tokens to a single-line comment starting with a C++ single-line token. The comment itself is to be unchanged.

### Preparation:

- Create at least five C++ source files in your home directory. The files do not have to be real C++ source files; they can contain only a few lines of comments, some with C program tokens and some with C++ single-line tokens. Each program should have at least one multiple comment and at least one single-line comment that uses the C program tokens. Use one or more blank lines between comments. The name of the files should have C++ extension (.c++), such as file1.c++.

### Script:

- **Script Name: commentType.scr**



```
#!/bin/bash
if (( $# == 0 ))
then
sed 's+//+/*+g' cfile.c > output
sed 's+\\/*+&*+/' output
else
echo "script needs zero args only"
fi
```

- **Arguments:** None
- **Validation:** The minimum validation requirements are:
  - i. Ensure that there is no argument.
- **Body Section:** Create a script that finds all files with extension (.c++) under your directory and change only the lines with comments. The name of the files should be preserved. If a file has the name file1.c++, the name still should be file1.c++ after the change.

### Testing the Script:

- Testing the script with one or two arguments.

```
Activities Terminal Jan 3 09:33
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh commenttype.scr myfile.c
script needs zero args only
[byrapardhasaradhi@fedora ~]$ sh commenttype.scr myfile.cc file.c
script needs zero args only
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with no arguments.

```
Activities Terminal Jan 3 09:34
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh commenttype.scr
/*hi this is first comment*/
#include<stdio.h>
#include<conio.h>
/* hi this is main*/
int main()
{
    /*this is the body*/
}
/* this is the end*/
[byrapardhasaradhi@fedora ~]$
```

### Testing the Effect of the Script:

- Check to see if the comments are changed in the files.

```
Activities Terminal Jan 3 09:34 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh commenttype.scr
/*hi this is first comment*/
#include<stdio.h>
#include<conio.h>
/* hi this is main*/
int main()
{
    /*this is the body*/
}
/* this is the end*/
[byrapardhasaradhi@fedora ~]$
```

**Result:** It is observed that the comments in c++ source file has been changed to C source file comments.

4. Write a script to backup and archive a list of files.

**Preparation:**

- Create a file and type in it the list of files (in your home directory) that you want to back and archive
- Create a directory in which you will store the backed-up files and archive file.

**Script**

- **Script Name:** backup.scr

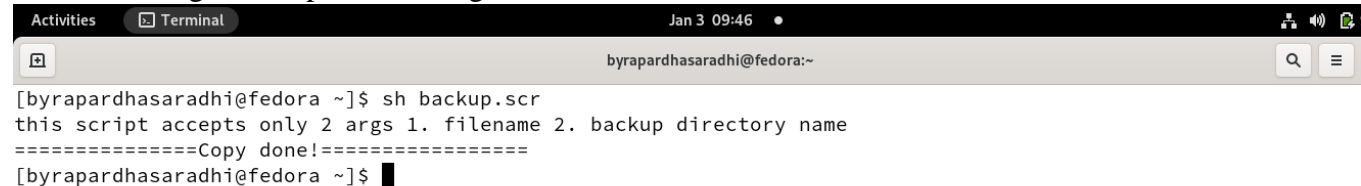
```
Activities Terminal Jan 3 09:45 byrapardhasaradhi@fedora:~ — /usr/bin/vim backup.scr
if (( $# == 2 ))
then
    if [[ -s $1 ]]
    then
        if [[ -d $2 ]]
        then
            for filename in $(cat $1)
            do
                cp $filename /home/byrapardhasaradhi/$2
            done
        else
            echo "the directory not exist"
        fi
    else
        echo "The file not exist or it is empty"
    fi
else
    echo "this script accepts only 2 args 1. filename 2. backup directory name"
fi
echo "=====Copy done!=====
-- INSERT --
```



- **Arguments:** A filename and a directory. The filename holds the list of the files that should be backed-up. The directory is where the backed-up files should be stored.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that exactly two arguments are entered.
  - ii. Check that the first argument is the name of a file exists
  - iii. Check that the second argument is the name of the directory that exists
- **Body Section:** Create backup files for files listed in the first argument. The backup files should have the same name as the original file with the extension bak. They should be copied to the directory given as the second argument.

### Testing the Script:

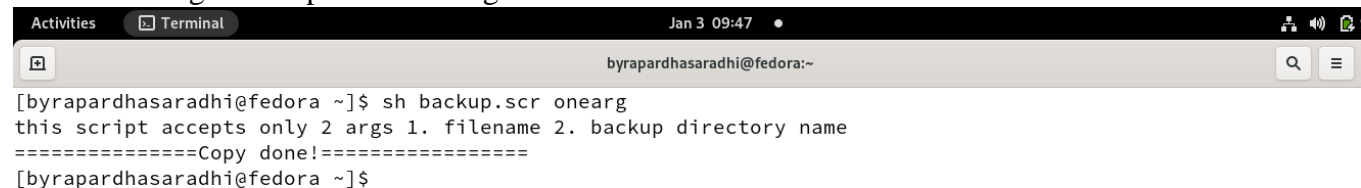
- Testing the script with no arguments



```

[byrapardhasaradhi@fedora ~]$ sh backup.scr
this script accepts only 2 args 1. filename 2. backup directory name
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$
  
```

- Testing the script with one argument



```

[byrapardhasaradhi@fedora ~]$ sh backup.scr onearg
this script accepts only 2 args 1. filename 2. backup directory name
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$
  
```

- Testing the script with three arguments

```
Activities Terminal Jan 3 09:48 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh backup.scr onearg twoarg
The file not exist or it is empty
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$
```

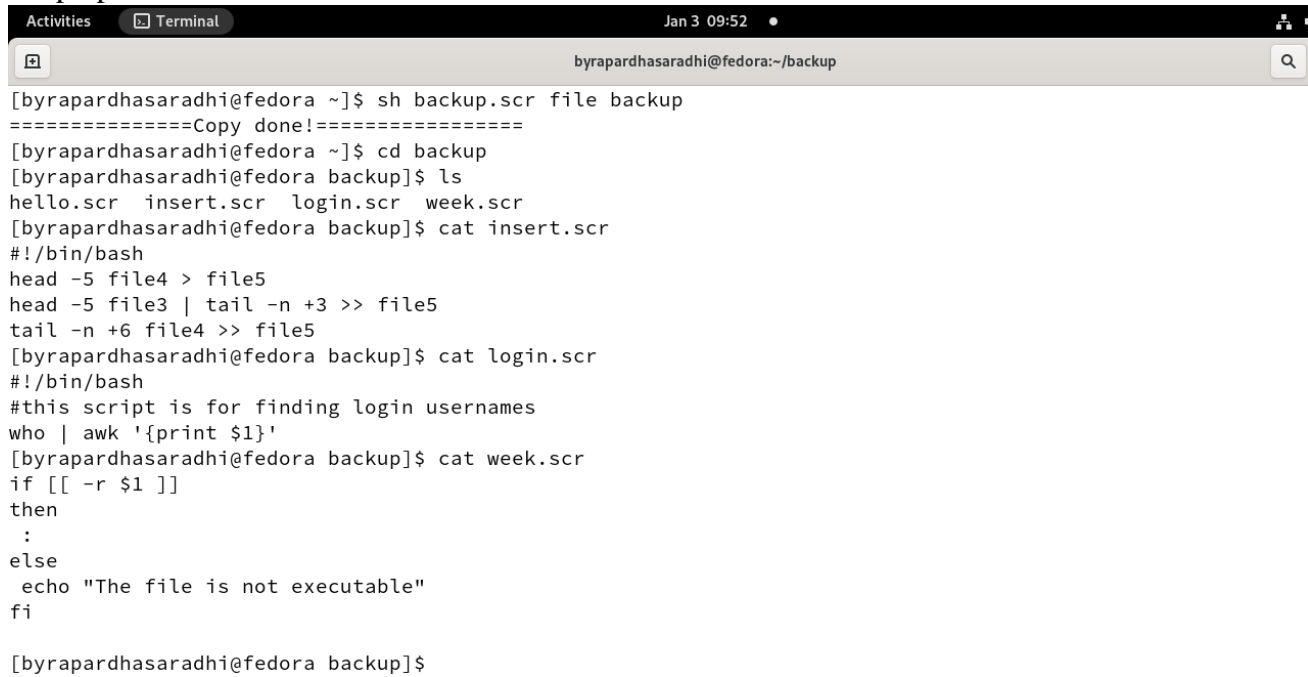
- Testing the script with two arguments in which the first one is not the name of the file.

```
Activities Terminal Jan 3 09:48 96 % byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh backup.scr onearg twoarg
The file not exist or it is empty
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with two arguments in which the second one is the name of a file rather than a directory.

```
Activities Terminal Jan 3 09:50 byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh backup.scr file notdirectory
the directory not exist
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with name of the file and the name of the directory you created in the preparation section.



```

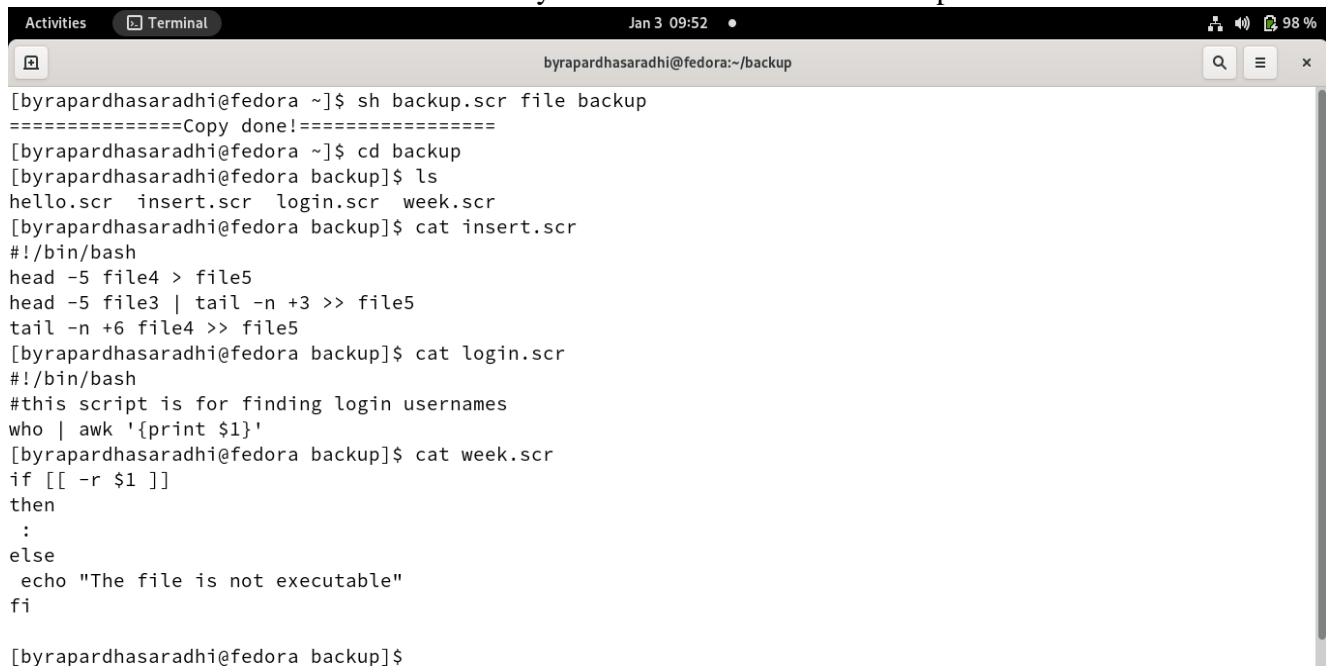
[byrapardhasaradhi@fedora ~]$ sh backup.scr file backup
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$ cd backup
[byrapardhasaradhi@fedora backup]$ ls
hello.scr insert.scr login.scr week.scr
[byrapardhasaradhi@fedora backup]$ cat insert.scr
#!/bin/bash
head -5 file4 > file5
head -5 file3 | tail -n +3 >> file5
tail -n +6 file4 >> file5
[byrapardhasaradhi@fedora backup]$ cat login.scr
#!/bin/bash
#this script is for finding login usernames
who | awk '{print $1}'
[byrapardhasaradhi@fedora backup]$ cat week.scr
if [[ -r $1 ]]
then
:
else
echo "The file is not executable"
fi

[byrapardhasaradhi@fedora backup]$

```

## Testing the Effect of the Script:

- Check the contents of the directory to be sure that the files are copied



```

[byrapardhasaradhi@fedora ~]$ sh backup.scr file backup
=====Copy done!=====
[byrapardhasaradhi@fedora ~]$ cd backup
[byrapardhasaradhi@fedora backup]$ ls
hello.scr insert.scr login.scr week.scr
[byrapardhasaradhi@fedora backup]$ cat insert.scr
#!/bin/bash
head -5 file4 > file5
head -5 file3 | tail -n +3 >> file5
tail -n +6 file4 >> file5
[byrapardhasaradhi@fedora backup]$ cat login.scr
#!/bin/bash
#this script is for finding login usernames
who | awk '{print $1}'
[byrapardhasaradhi@fedora backup]$ cat week.scr
if [[ -r $1 ]]
then
:
else
echo "The file is not executable"
fi

[byrapardhasaradhi@fedora backup]$

```

**Result:** The backup of list of files given in “file” was created in the given directory.

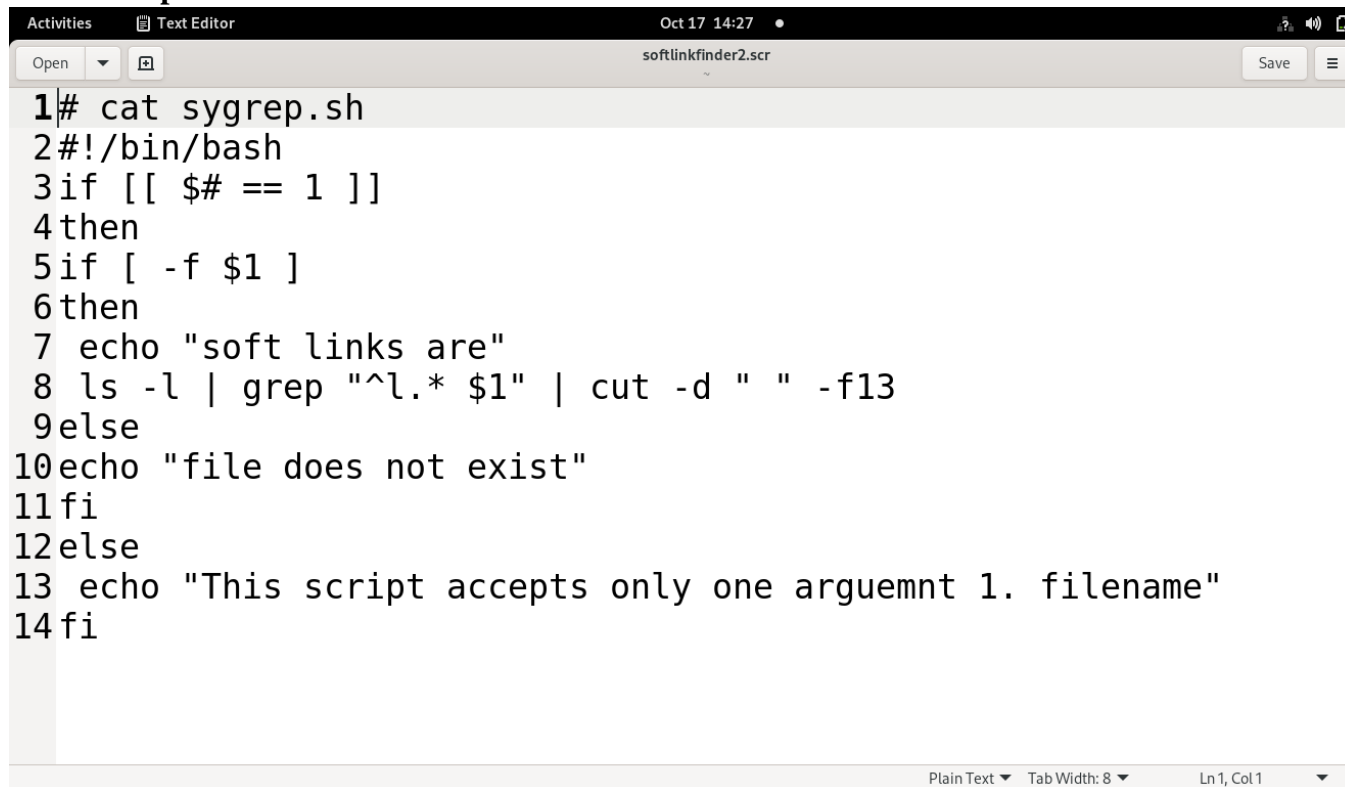
5. Write a script that finds all soft links to a specific file.

**Preparation:**

- Create a file and type some junk in it.
- Make at least five soft links to this file using completely arbitrary names.

**Script:**

- **Script Name: softLinkFinder.scr**



```
1# cat sygrep.sh
2#!/bin/bash
3if [[ $# == 1 ]]
4then
5if [ -f $1 ]
6then
7 echo "soft links are"
8 ls -l | grep "^l.* $1" | cut -d " " -f13
9else
10echo "file does not exist"
11fi
12else
13 echo "This script accepts only one arguemnt 1. filename"
14fi
```

- **Arguments:** A filename. The file for which we want to find the soft links.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that exactly one argument is entered. ii. Check that only argument is the name of a file and that the specified file exists.
- **Body Section:** Use ls -l and grep command to find all the soft links attached to \$1 positional parameter. Note that a file of type soft link is distinguished by lower case l. Be sure to find the soft links to the file defined in \$1 and not other files.

**Testing the Script:**

- Testing the script with no arguments

```
Activities Terminal Jan 3 09:58
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh softlinkfinder.scr
this script accepts only one argument 1. filename
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument

```
Activities Terminal Oct 17 14:30
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ ./softlinkfinder2.scr one
file does not exist
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument that is not a file

```
Activities Terminal Jan 3 09:59
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh softlinkfinder.scr nofile
file not exist or empty
[byrapardhasaradhi@fedora ~]$
```

- Test the script with one valid argument.

```

Activities Terminal Oct 17 14:33
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ ln -s sample.scr link1
[byrapardhasaradhi@fedora ~]$ ln -s sample.scr link2
[byrapardhasaradhi@fedora ~]$ ln -s sample.scr link3
[byrapardhasaradhi@fedora ~]$ ln -s sample.scr link4
[byrapardhasaradhi@fedora ~]$ ln -s sample.scr link5
[byrapardhasaradhi@fedora ~]$ ls -l sample.scr link1 link2 link3 link4 link5
lrwxrwxrwx. 1 byrapardhasaradhi byrapardhasaradhi 10 Oct 17 14:32 link1 -> sample.scr
lrwxrwxrwx. 1 byrapardhasaradhi byrapardhasaradhi 10 Oct 17 14:32 link2 -> sample.scr
lrwxrwxrwx. 1 byrapardhasaradhi byrapardhasaradhi 10 Oct 17 14:32 link3 -> sample.scr
lrwxrwxrwx. 1 byrapardhasaradhi byrapardhasaradhi 10 Oct 17 14:32 link4 -> sample.scr
lrwxrwxrwx. 1 byrapardhasaradhi byrapardhasaradhi 10 Oct 17 14:32 link5 -> sample.scr
-rwxrwxr-x. 1 byrapardhasaradhi byrapardhasaradhi 58 Oct 17 13:30 sample.scr
[byrapardhasaradhi@fedora ~]$ ./softlinkfinder2.scr sample.scr
soft links are
->
->
->
->
->
[byrapardhasaradhi@fedora ~]$

```

```

Activities Terminal Oct 17 14:37
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ ./softlinkfinder2.scr sample.scr
soft links are
link1
link2
link3
link4
link5
[byrapardhasaradhi@fedora ~]$

```

## Testing the Effect of the Script:

- Check to make sure all the soft links you created are included in the list of soft links.

```

Activities Terminal Oct 17 14:37
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ ./softlinkfinder2.scr sample.scr
soft links are
link1
link2
link3
link4
link5
[byrapardhasaradhi@fedora ~]$

```

**Result:** The Softlinks for the given file displayed.

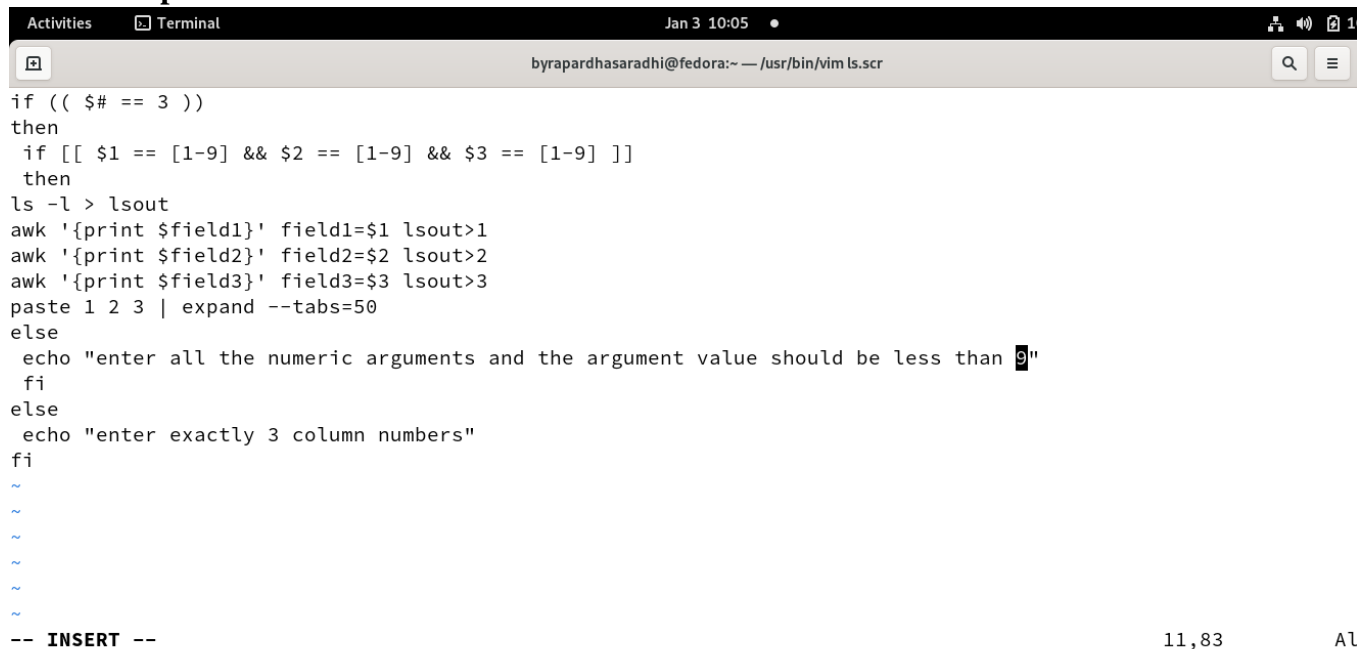
6. Create a script that simulates the `ls -l` command but prints only three columns of our choice.

**Preparation:**

- None

**Script:**

- **Script Name: ls.scr**



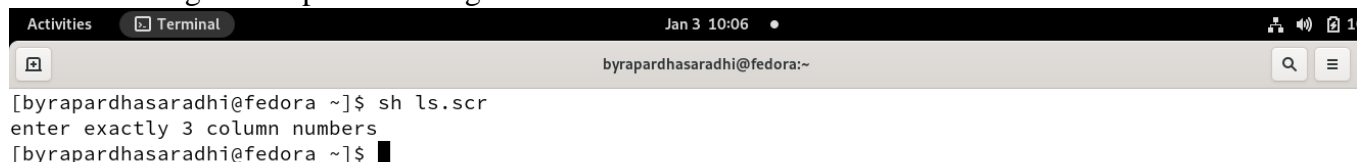
```
Activities Terminal Jan 3 10:05
byrapardhasaradhi@fedora:~ — /usr/bin/vim ls.scr

if (( $# == 3 ))
then
  if [[ $1 == [1-9] && $2 == [1-9] && $3 == [1-9] ]]
  then
    ls -l > lsout
    awk '{print $field1}' field1=$1 lsout>1
    awk '{print $field2}' field2=$2 lsout>2
    awk '{print $field3}' field3=$3 lsout>3
    paste 1 2 3 | expand --tabs=50
  else
    echo "enter all the numeric arguments and the argument value should be less than 9"
  fi
else
  echo "enter exactly 3 column numbers"
fi
~
~
~
~
~
-- INSERT -- 11,83
```

- **Arguments:** Three numeric arguments defining the column number of the `ls -l` output to be printed in the order we specify.
- **Validation:** The minimum validation requirements are :
  - i. Ensure that exactly three arguments are entered.
  - ii. Ensure that all three arguments are numeric
  - iii. Ensure that each argument is less than or equal to the actual number of columns in the `ls -l` command output.
- **Body Section:** Creates a new command that shows the output of the `ls -l` command to be printed in three columns in the order we like.

**Testing the Script:**

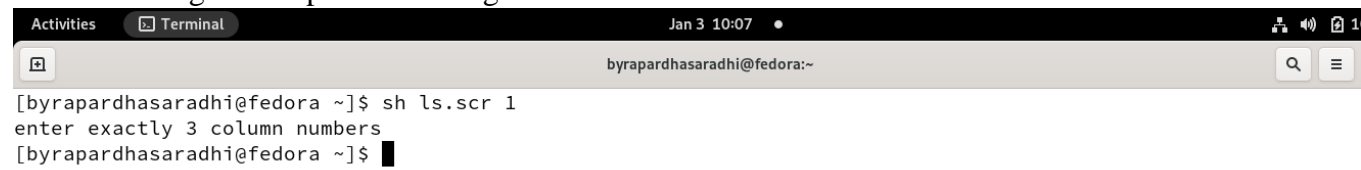
- Testing the script with no arguments.



```
Activities Terminal Jan 3 10:06
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ sh ls.scr
enter exactly 3 column numbers
[byrapardhasaradhi@fedora ~]$
```

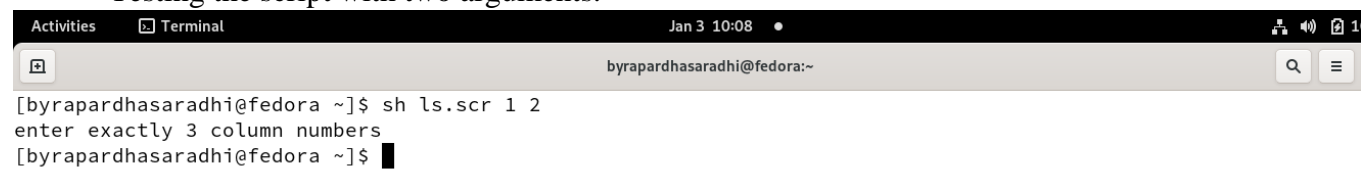
- Testing the script with one argument.



A terminal window titled 'Terminal' showing the execution of a script. The prompt is [byrapardhasaradhi@fedora ~]. The command sh ls.scr 1 is entered. The script outputs 'enter exactly 3 column numbers'. The prompt returns to [byrapardhasaradhi@fedora ~].

```
[byrapardhasaradhi@fedora ~]$ sh ls.scr 1
enter exactly 3 column numbers
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with two arguments.



A terminal window titled 'Terminal' showing the execution of a script. The prompt is [byrapardhasaradhi@fedora ~]. The command sh ls.scr 1 2 is entered. The script outputs 'enter exactly 3 column numbers'. The prompt returns to [byrapardhasaradhi@fedora ~].

```
[byrapardhasaradhi@fedora ~]$ sh ls.scr 1 2
enter exactly 3 column numbers
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with three arguments, one of them nonnumeric.



```
Activities Terminal Jan 3 10:09 • 100 %
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh ls.scr 1 2 nonnumber
enter all the numeric arguments and the argument value should be less than 9
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with three arguments, two of them nonnumeric.

```
Activities Terminal Jan 3 10:10 • 1
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh ls.scr 1 nonnumber nonnumber
enter all the numeric arguments and the argument value should be less than 9
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with three arguments, one of them too large.

```
Activities Terminal Jan 3 10:10 • 1
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh ls.scr 1 2 12
enter all the numeric arguments and the argument value should be less than 9
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with three arguments, 1 4 5

```

Activities Terminal Jan 3 10:11
byrapardhasaradhi@fedora:~$ sh ls.scr 1 4 5
total
-rw-rw-r--. byrapardhasaradhi 392
-rw-rw-r--. byrapardhasaradhi 164
-rw-rw-r--. byrapardhasaradhi 162
-rw-rw-r--. byrapardhasaradhi 466
-rw-rw-r--. byrapardhasaradhi 1328
-rw-rw-r--. byrapardhasaradhi 35
-rw-rw-r--. byrapardhasaradhi 49
drwxrwxr-x. byrapardhasaradhi 24
-rw-rw-r--. byrapardhasaradhi 118
-rwxrwxr-x. byrapardhasaradhi 25344
-rwxrwxr-x. byrapardhasaradhi 14
-rw-rw-r--. byrapardhasaradhi 186
drwxrwxr-x. byrapardhasaradhi 24
drwxrwxr-x. byrapardhasaradhi 72
drwxrwxr-x. byrapardhasaradhi 0
drwxrwxr-x. byrapardhasaradhi 72
-rw-rw-r--. byrapardhasaradhi 372
-rw-rw-r--. byrapardhasaradhi 75
drwxrwxr-x. byrapardhasaradhi 4
-rw-rw-r--. byrapardhasaradhi 676
-rw-rw-r--. byrapardhasaradhi 183

```

- Test the script with three arguments, 3 7 1

```

Activities Terminal Jan 3 10:12
byrapardhasaradhi@fedora:~$ sh ls.scr 3 4 7
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 2
byrapardhasaradhi byrapardhasaradhi 6
byrapardhasaradhi byrapardhasaradhi 22
byrapardhasaradhi byrapardhasaradhi 22
byrapardhasaradhi byrapardhasaradhi 31
byrapardhasaradhi byrapardhasaradhi 22
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 4
byrapardhasaradhi byrapardhasaradhi 20
byrapardhasaradhi byrapardhasaradhi 31
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 17
byrapardhasaradhi byrapardhasaradhi 17
byrapardhasaradhi byrapardhasaradhi 3
byrapardhasaradhi byrapardhasaradhi 22
byrapardhasaradhi byrapardhasaradhi 31
byrapardhasaradhi byrapardhasaradhi 14
byrapardhasaradhi byrapardhasaradhi 14

```

### Testing the Effect of the Script:

- None

**Result:-** The required fields in the ls-l command were displayed.


7. Create a script that sends contents of a message file to everybody who logged in.

### Preparation:

- Create a file of a short friendly message and mention that this is a test message that should be discarded by the receiver

## Script:

- **Script Name: message.scr**



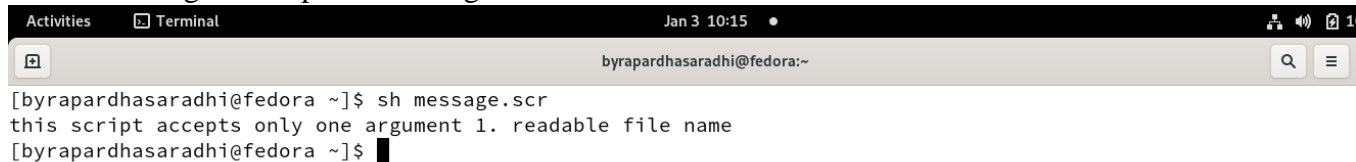
```
if (($# == 1))
then
  if [[ -r $1 ]]
  then
    who | awk '{print $1}' > users
    cat $1 > message
    wall message
    echo "successfully message sent to all users"
  else
    echo "the file is not readable"
  fi
else
  echo "this script accepts only one argument 1. readable file name"
fi
```

-- INSERT -- 21,1 All

- **Arguments:** One argument, a message file.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that exactly one argument is entered.
  - ii. Ensure that the argument is a readable filename.
- **Body Section:** Create a script that uses awk to create a temporary file containing the usernames of those users who are logged into the system at this moment. Then send the message contained in the first argument to every logged-in user. Note that a user who has logged in more than once should receive only one message.

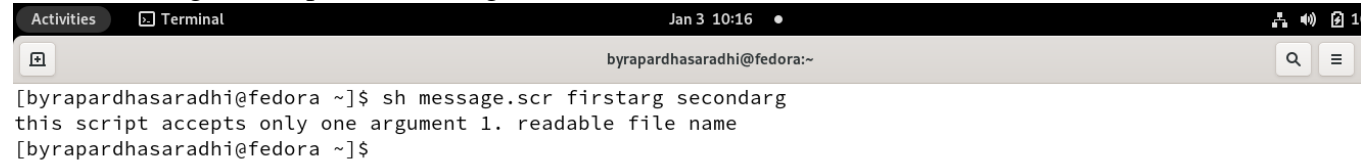
## Testing the Script:

- Testing the script with no arguments.



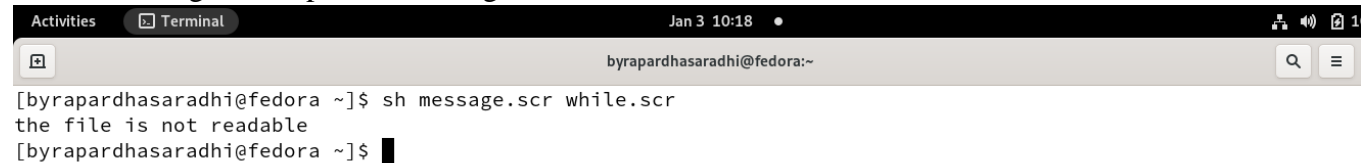
```
[byrapardhasaradhi@fedora ~]$ sh message.scr
this script accepts only one argument 1. readable file name
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with two arguments.

A terminal window titled 'Terminal' with a dark theme. The prompt is 'byrapardhasaradhi@fedora:~'. The user enters 'sh message.scr firstarg secondarg'. The script outputs 'this script accepts only one argument 1. readable file name'.

```
byrapardhasaradhi@fedora:~  
[byrapardhasaradhi@fedora ~]$ sh message.scr firstarg secondarg  
this script accepts only one argument 1. readable file name  
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument that is not a readable file.

A terminal window titled 'Terminal' with a dark theme. The prompt is 'byrapardhasaradhi@fedora:~'. The user enters 'sh message.scr while.scr'. The script outputs 'the file is not readable'.

```
byrapardhasaradhi@fedora:~  
[byrapardhasaradhi@fedora ~]$ sh message.scr while.scr  
the file is not readable  
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one valid argument.

```
Activities Terminal Jan 3 10:18 • byrapardhasaradhi@fedora:~  
[byrapardhasaradhi@fedora ~]$ sh message.scr file  
successfully message sent to all users  
[byrapardhasaradhi@fedora ~]$
```

### Testing the Effect of the Script:

- You should include yourself in the recipient list. Check to see if you have received the message.

```
Activities Terminal Jan 3 10:18 • byrapardhasaradhi@fedora:~ 100 %  
[byrapardhasaradhi@fedora ~]$ sh message.scr file  
successfully message sent to all users  
[byrapardhasaradhi@fedora ~]$
```

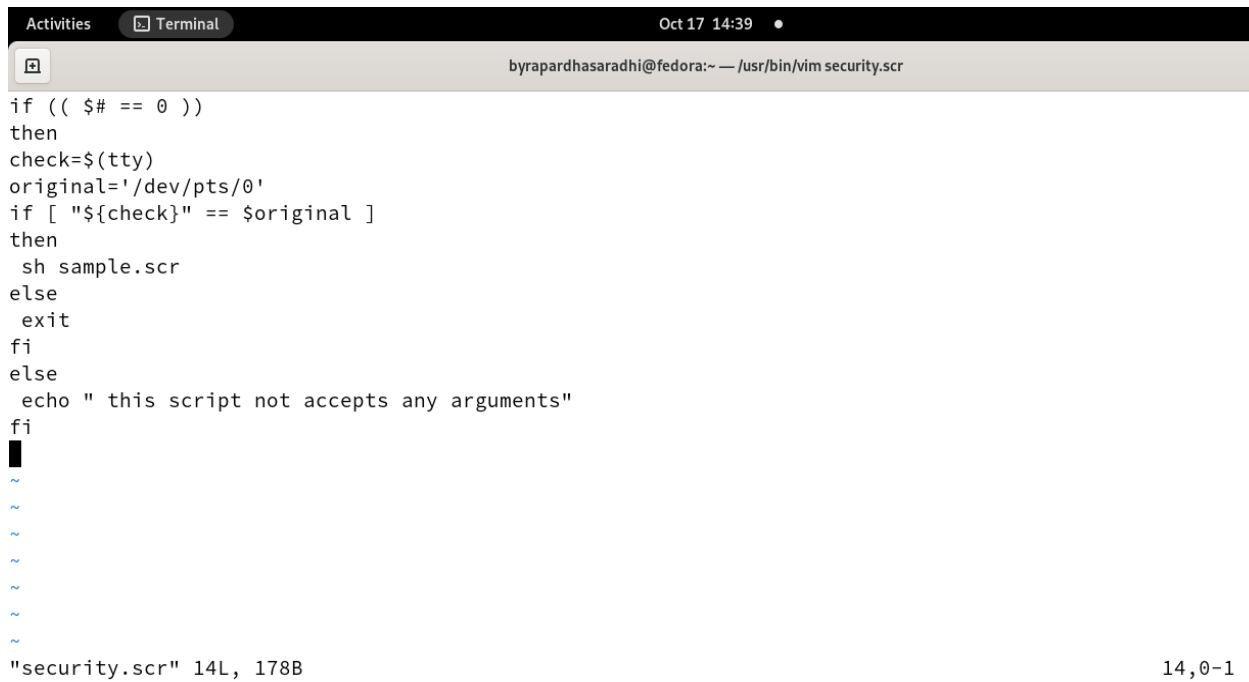
8. Create a script that can be executed only from a specific terminal. This is done for security purposes. For example, a superuser may write scripts that can only be executed from his or her office and nowhere else.

#### Preparation:

- None

#### Script:

- **Script Name: security.scr**

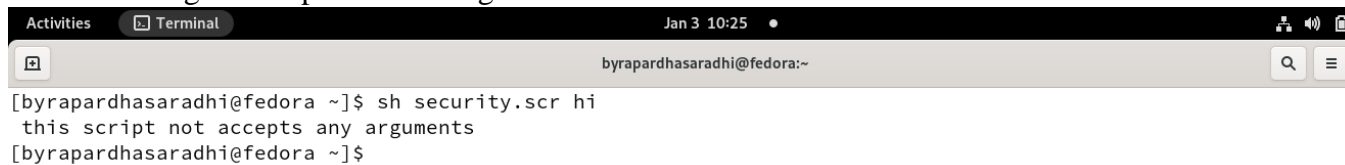


```
if (( $# == 0 ))
then
check=$(tty)
original='/dev/pts/0'
if [ "${check}" == $original ]
then
sh sample.scr
else
exit
fi
else
echo " this script not accepts any arguments"
fi
~
~
~
~
~
~
~
"security.scr" 14L, 178B
```

- **Arguments:** None.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that no argument is entered.
- **Body Section:** Create a script that prints a friendly message. However, the script can be executed only for one terminal. You can use the name of the terminal you are using when you write the script. If somebody uses the script from a terminal that is not authorized, the script is to exit immediately. Hint: Use the tty command to show your current terminal.

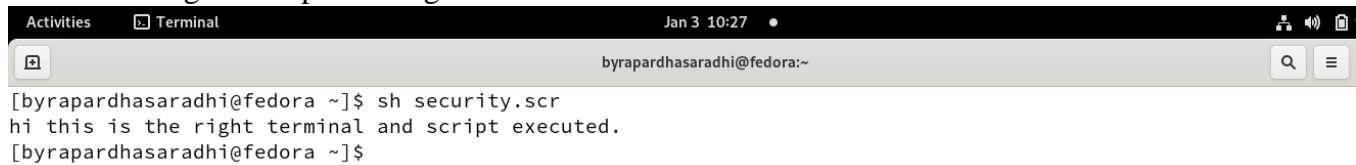
### Testing the Script:

- Testing the script with one argument.



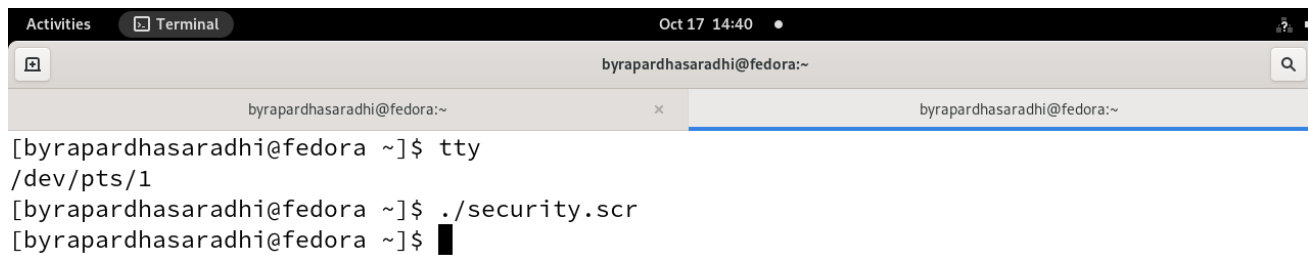
```
[byrapardhasaradhi@fedora ~]$ sh security.scr hi
this script not accepts any arguments
[byrapardhasaradhi@fedora ~]$
```

- Testing the script from right terminal.



A terminal window titled 'Terminal' with a dark header bar showing 'Activities', 'Terminal', and the date 'Jan 3 10:27'. The terminal prompt is 'byrapardhasaradhi@fedora:~'. The user enters 'sh security.scr' and the output is 'hi this is the right terminal and script executed.' followed by a new prompt 'byrapardhasaradhi@fedora ~]\$'.

- Log into the system using another terminal and test the script.



A terminal window titled 'Terminal' with a dark header bar showing 'Activities', 'Terminal', and the date 'Oct 17 14:40'. The terminal prompt is 'byrapardhasaradhi@fedora:~'. The user enters 'tty' and the output is '/dev/pts/1'. The user then enters './security.scr' and the output is a blank line followed by a new prompt 'byrapardhasaradhi@fedora ~]\$'.

### Testing the Effect of the Script:

- None

**Result:-** Script tested in other terminals and is not executed and it's only executing in the right terminal only.

9. Create a script that finds each line in a file that contains a specified string.

### Preparation:

- Create a file of at least 20 lines and insert a double quoted string, such as "hello," in several lines.

### Script:

- **Script Name: search.scr**

```
Activities Terminal Jan 3 10:29
byrapardhasaradhi@fedora:~ — /usr/bin/vim search.scr

if (( $# == 2 ))
then
  if [[ -s "$2" ]]
  then
    grep -n $1 $2
  else
    echo " The file not exist or it is empty"
  fi
else
  echo "This program requires 2 arguments 1. string to be matched 2. file"
fi
```

-- INSERT --

21,1

- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are:
  - i. Ensure that exactly two arguments are entered.
  - ii. Ensure that the second argument is the name of the file that exists and is not empty.
- **Body Section:** Create a script that uses grep and loops to find the line numbers in which the string is found. Note that grep should be applied to each line, not the whole file. The script should print the result in the following format:

**Line Number: [Line contents] Testing the**

### Script:

- Testing the script with no arguments.

```
Activities Terminal Jan 3 10:30
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ sh search.scr
This program requires 2 arguments 1. string to be matched 2. file
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument.



```
Activities Terminal Jan 3 10:31
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh search.scr file
This program requires 2 arguments 1. string to be matched 2. file
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with two argument but the second one is not a file.

```
Activities Terminal Jan 3 10:33
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh search.scr string notfile
The file not exist or it is empty
[byrapardhasaradhi@fedora ~]$
```

- Test the script with two correct arguments.

```
Activities Terminal Jan 3 10:42 90 %
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh search.scr hello hello
1:hello This is line 1
2:hello this is line2
3:hello this is line3
4:hello
5:hello
[byrapardhasaradhi@fedora ~]$
```

### Testing the Effect of the Script:

- Compare the results of your script with a printout of the file.

**Result:-** Results were compared with printout of the file and the line numbers are the exact matched lines.

10. Create a script that compiles all C source files in your home directory and create executable files.

### Preparation:

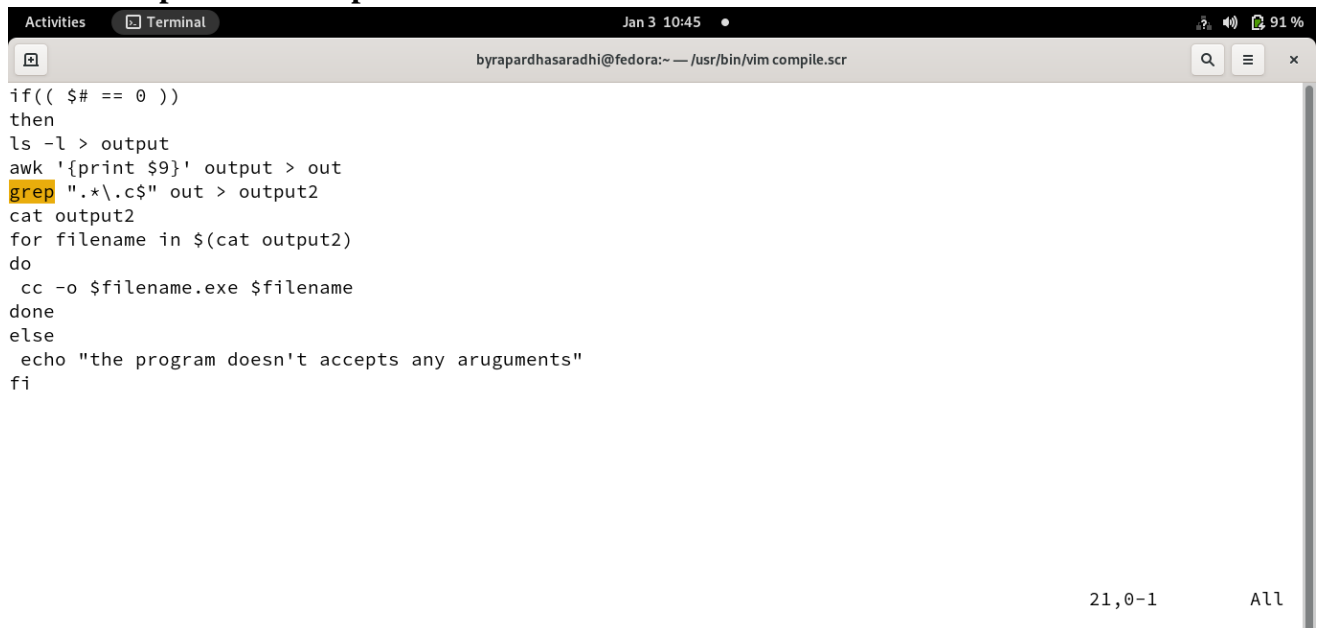
- Create at least five C source files in your home directory. The files do not have to be real C source files; at a minimum they should contain a comment line that contain a unique program name such as the following example:

/\* .....file1.c .....\*/

The name of the files should have a C source file extension (.c), such as file1.c.

### Script:

- **Script Name: compile.scr**



```
if(( $# == 0 ))
then
ls -l > output
awk '{print $9}' output > out
grep "\.c$" out > output2
cat output2
for filename in $(cat output2)
do
cc -o $filename.exe $filename
done
else
echo "the program doesn't accepts any aruguments"
fi
```

21,0-1 All

- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are :
  - i. Ensure that there is no argument
- **Body Section:** Create a script that finds all files with extension (.c) under your home directory and compiles them one by one. Each executable file should have the same name as the source file except that the extension should be (.exe). For example, if the source filename is file1.c, the executable filename should be file1.exe. Use the following command to compile: `cc -o executable_filename source_filename`

### Testing the Script:

- Test the script with one or two arguments.



```
[byrapardhasaradhi@fedora ~]$ sh compile.scr h i
the program doesn't accepts any aruguments
[byrapardhasaradhi@fedora ~]$
```

- Test the script with no arguments.

```

Activities Terminal Jan 3 10:50
byrapardhasaradhi@fedora:~

[byrapardhasaradhi@fedora ~]$ vi cfile.c
[byrapardhasaradhi@fedora ~]$ sh compile.scr
cfile.c
file1.c
file2.c
file3.c
file4.c
file5.c
file6.c
sample.c
small.c
smalln.c
[byrapardhasaradhi@fedora ~]$ ls
1 date.scr file6.c.exe Music small.c.exe
2 Desktop fileextr.scr new smalln.c
3 dir1 file.scr newEtcPasswd.scr smalln.c.exe
4 dir2 findHomeDirectory.scr notread softlinkfinder.scr
456 Documents for null special.scr
5lines.dat doublespace.awk forex.scr out sprintf.awk
7lines.dat Downloads forout.out output stringsub.awk
a duplicatename.scr foroutput.out output2 sum
afterblkln.awk elif.scr for.scr outputpipe.scr sumavg
a.out evalu.scr grep param.scr Templates

Activities Terminal Jan 3 10:51
byrapardhasaradhi@fedora:~

a.out args.sh backup backup2 backup3 backup.scr blankln.awk c case case.scr cfile.c cfile.c.exe check commenttype.scr compile.scr compsaleavg.awk countwdslns.awk cprogs date
evalu.scr extract.scr file file1 file1.c file1.c.exe file1.exe file2 file2.c file2.c.exe file3 file3.c file3.c.exe file4 file4.c file4.c.exe file4.sh file5 file5.c file5.c.exe file6.c
grep grepx he hello hello.scr if inputpipe.scr insert.scr login.scr loopcatfiles.dat loopred.scr lsawk.scr lsl.scr lsout ls.scr ls.sh merge.awk message message.scr midraven.txt midRaven.txt
param.scr phones.awk phones.dat Pictures Public report saleszerodiv.awk sample.c sample.c.exe sample.scr sample.sh script.scr search.scr security.scr sedex.sed select select.scr selecttwo.scr set.scr shift.scr small.c
Templates theraven TheRaven TheRaven1 TheRavenv1 time.scr toal total total.awk until.scr 'UP record.doc' users Videos week1.scr week2out week2.scr week2.sed week2.sh week6ex week.scr while.scr

[byrapardhasaradhi@fedora ~]$

```

### Testing the Effect of the Script:

- Verify that executable files were created under your home directory.

**Result:-** It is verified that the required executable files are created as shown above in snapshot.

11. Create a script that finds all files in subdirectories that have the same filename.

### Preparation:

- Make several directories, at different levels, under your home directory. For example, make ~/A, ~/B, ~/C, ~/A/AA, ~/A/BB, ~/A/AA/AAA, and so on until you have at least 15 directories. Copy a small junk file named file1 under some of these directories; do not change its name. Copy another small junk file named file2 under some other directories. Copy a third junk file under several directories. Be sure that some directories get a combination of file1 and file2 or file1 and file3. In at least three of the directories, create a junk file with a unique name.

## Script:

- **Script Name: duplicateName.scr**

```
Activities Terminal Jan 3 20:43
byrapardhasaradhi@fedora:~ — /usr/bin/vim week11.scr
awk -F '/' '{
f=$NF
a[f] = f in a? a[f] RS $0 : $0
b[f]++}
END{for(x in b)
  if(b[x]>1)
    printf "Duplicate filename: %s\n%s\n",x,a[x]}' <(find . -type f)
```

```
-- INSERT -- 21,1 AU
```

- **Arguments:** None
- **Validation:** The minimum validation requirements are :
  - i. Ensure that there is no argument.
- **Body Section:** Create a script that uses find and awk commands to create a list of files that are duplicated; use the full pathname for the duplicated filenames. Hint: Use a basename command and an array in awk. The output should look like the following example:

```
file1: ~/A/file1 ~/A/AA/file1 ~/A/B/BB/BBB/file1
file2: ~/B/file2 ~/C/file2
...
```

## Testing the Script:

- Test the script with one argument.
- Testing the script with no arguments.

```
Activities Terminal Jan 3 20:45
byrapardhasaradhi@fedora:~
json
./config/google-chrome/Default/Extensions/nmmhkkegccagdldgiimedpiccmgmieda/1.0.0.6_0/_metadata/computed_hashes
json
Duplicate filename: file1
./file1
./a/aa/aaa/file1
./a/file1
./b/bb/bbb/file1
./b/file1
Duplicate filename: file2
./file2
./a/aa/file2
./a/file2
./b/bb/bbb/file2
./b/bb/file2
./b/file2
Duplicate filename: file3
./file3
./a/aa/aaa/file3
./a/aa/file3
./b/bb/bbb/file3
./b/bb/file3
Duplicate filename: pkcs11.txt
./pkcs11.txt
./a/pkcs11.txt
./b/pkcs11.txt
```

## Testing the Effect of the Script:

- Use a recursive long list command to list the complete contents of your home directory. Verify the output of your script against the list command output.



```
byrapardhasaradhi@fedora:~$ ls -lR
./:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./.:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./a:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./a/aa:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./a/aa/aaa:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./a/aa/aaa/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/aa/aaa/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/aa/aaa/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/aa/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/aa/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/aa/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./a/bb/bbb:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./a/bb/bbb/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb/bbb/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb/bbb/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./a/bb/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./b/aa:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./b/aa/aaa:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./b/aa/aaa/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/aa/aaa/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/aa/aaa/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/aa/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/aa/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/aa/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb:
total 4
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
drwxr-xr-x 3 root root 4096 Jan 3 20:45
./b/bb/bbb:
total 4
drwxr-xr-x 2 root root 4096 Jan 3 20:45
drwxr-xr-x 2 root root 4096 Jan 3 20:45
./b/bb/bbb/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb/bbb/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb/bbb/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/bb/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/file1:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/file2:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./b/file3:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
./pkcs11.txt:
total 4
-rw-r--r-- 1 root root 0 Jan 3 20:45
```

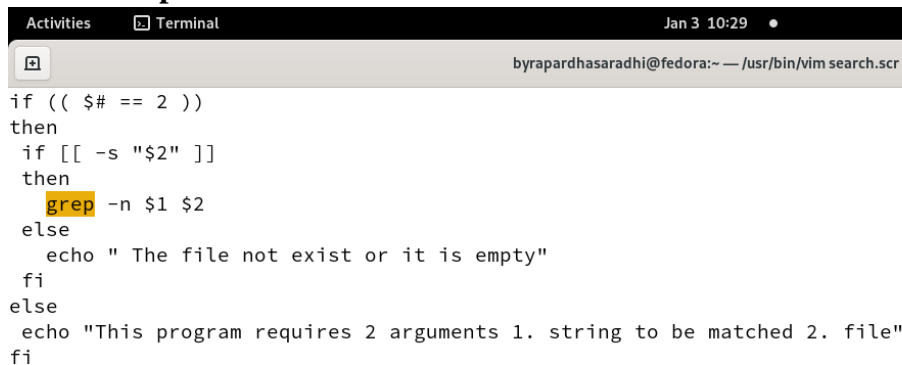
**Result:-** The duplicate files are all found along with path as shown above.

## 12. Create a script that search for multiple occurrences of the specified string in each line. **Preparation:**

- Create a file of at least 20 lines and insert a double quoted string, such as "hello," in several lines.
- Include two or three occurrences of the string in some lines.

## Script:

- **Script Name:** search.scr



```
byrapardhasaradhi@fedora:~$ cat search.scr
if (( $# == 2 ))
then
  if [[ -s "$2" ]]
  then
    grep -n $1 $2
  else
    echo " The file not exist or it is empty"
  fi
else
  echo "This program requires 2 arguments 1. string to be matched 2. file"
fi
```

-- INSERT --

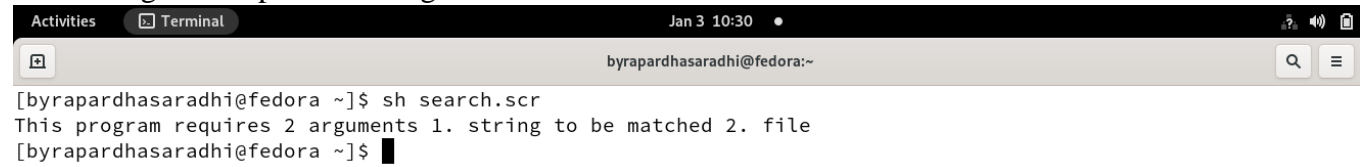
21,1

- **Arguments:** Two arguments, the first is the string to be found; the second is the name of the file.
- **Validation:** The minimum validation requirements are :
  - i. Ensure that exactly two arguments are entered.
  - ii. Ensure that the second argument is the name of the file that exists and is not empty.
- **Body Section:** Create a script that uses grep and loops to find the line numbers in which the string is found. Note that grep should be applied to each line, not the whole file. The script should print the result in the following format:

**Line Number: [Line contents]**

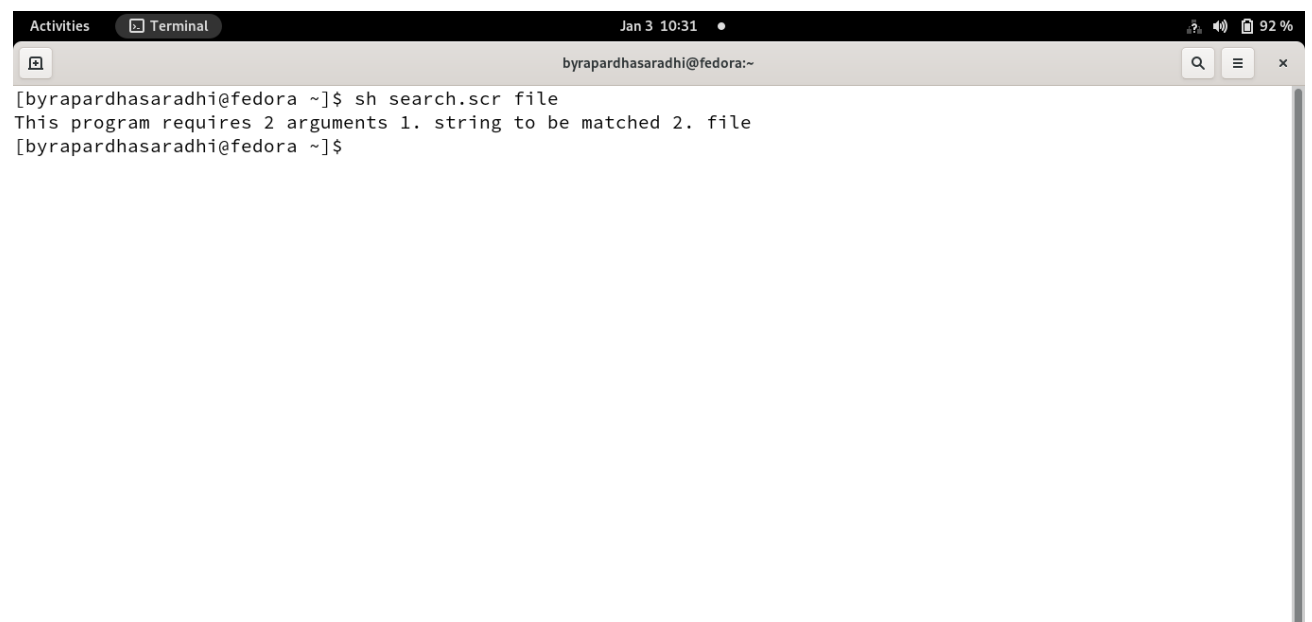
### Testing the Script:

- Testing the script with no arguments.



```
Activities Terminal Jan 3 10:30
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh search.scr
This program requires 2 arguments 1. string to be matched 2. file
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with one argument.



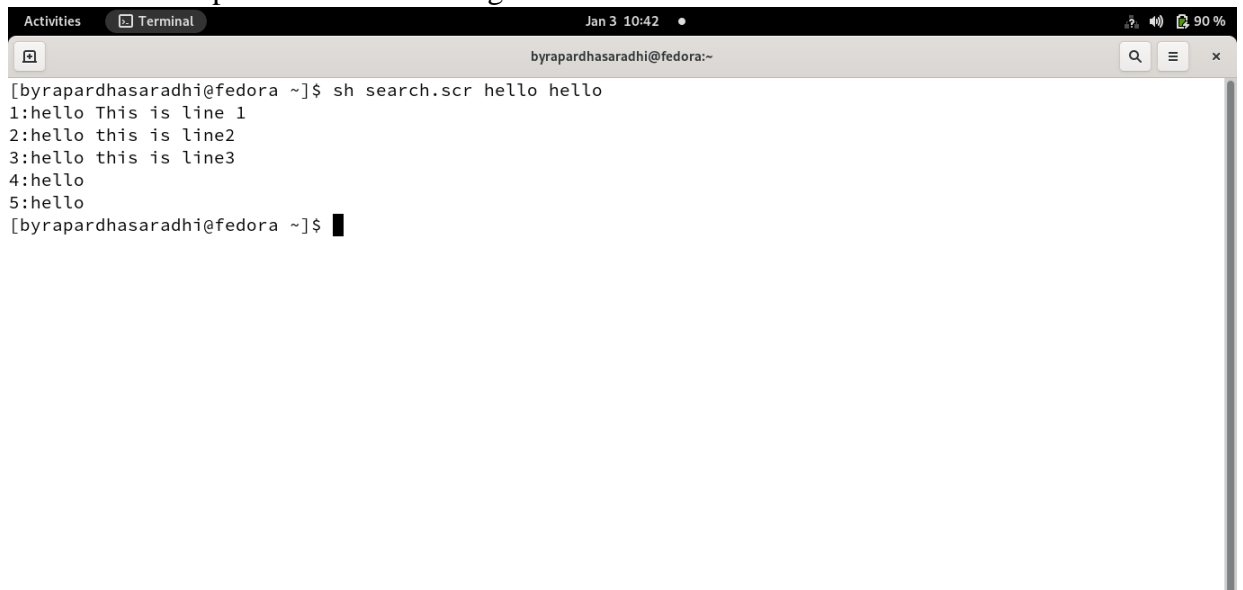
```
Activities Terminal Jan 3 10:31
byrapardhasaradhi@fedora:~
[byrapardhasaradhi@fedora ~]$ sh search.scr file
This program requires 2 arguments 1. string to be matched 2. file
[byrapardhasaradhi@fedora ~]$
```

- Testing the script with two argument but the second one is not a file.

A terminal window titled 'Terminal' with a dark theme. The user 'byrapardhasaradhi@fedora' is at the prompt. They run the command 'sh search.scr string notfile'. The script outputs 'The file not exist or it is empty'.

```
byrapardhasaradhi@fedora:~$ sh search.scr string notfile
The file not exist or it is empty
byrapardhasaradhi@fedora:~$
```

- Test the script with two correct arguments.

A terminal window titled 'Terminal' with a dark theme. The user 'byrapardhasaradhi@fedora' is at the prompt. They run the command 'sh search.scr hello hello'. The script outputs five lines, each starting with a line number followed by 'hello' and a description of the line.

```
byrapardhasaradhi@fedora:~$ sh search.scr hello hello
1:hello This is line 1
2:hello this is line2
3:hello this is line3
4:hello
5:hello
byrapardhasaradhi@fedora:~$
```

### Testing the Effect of the Script:

Compare the results of your script with a printout of the file.

**Result:-** Compared the results of the script with printout of the file and got exact matching results.