

Shell Script :

A shell script is a text file that contains sequence of commands for unix based operating system. It combines the commands in “script” and execute them, otherwise would have to execute those commands individually on terminal .The extension for shell script file is **.sh**. eg **filename.sh**

Step to do shell script program.

- 1) First create a text file by using any text editor,here I used gedit. So on terminal type **\$gedit example1.sh** it will create a shell script file with name **example1.sh**
- 2) now start to write shell script program in file , for we must need to write first line in file is **#!/bin/bash**
- 3) after writing script save the file and exit.
- 4) To run the file we first need to made it executable for that we change its mode to be executable by using chmod command as **\$chmod +x example1.sh**
- 5) last step to execute script file do **\$bash exampl1.sh**

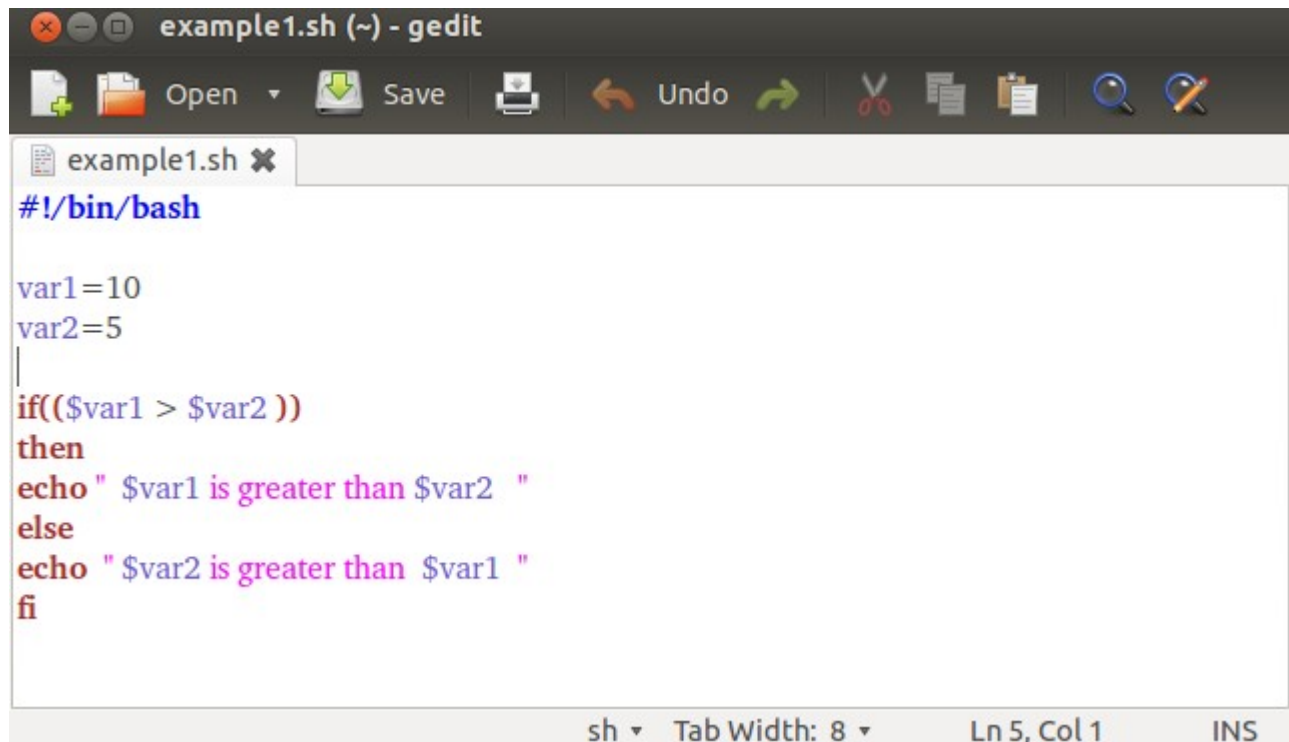
Here are some syntax :

```
if ((condition))
then
    statements (when condition is true)
else
    statements (when condition is false)
fi
```

```
for((var intial_value ; var < max_value ; var++))
do
    statements
done
```

```
while((condition))
do
    statements
done
```

Shell script to find largest between two numbers ?



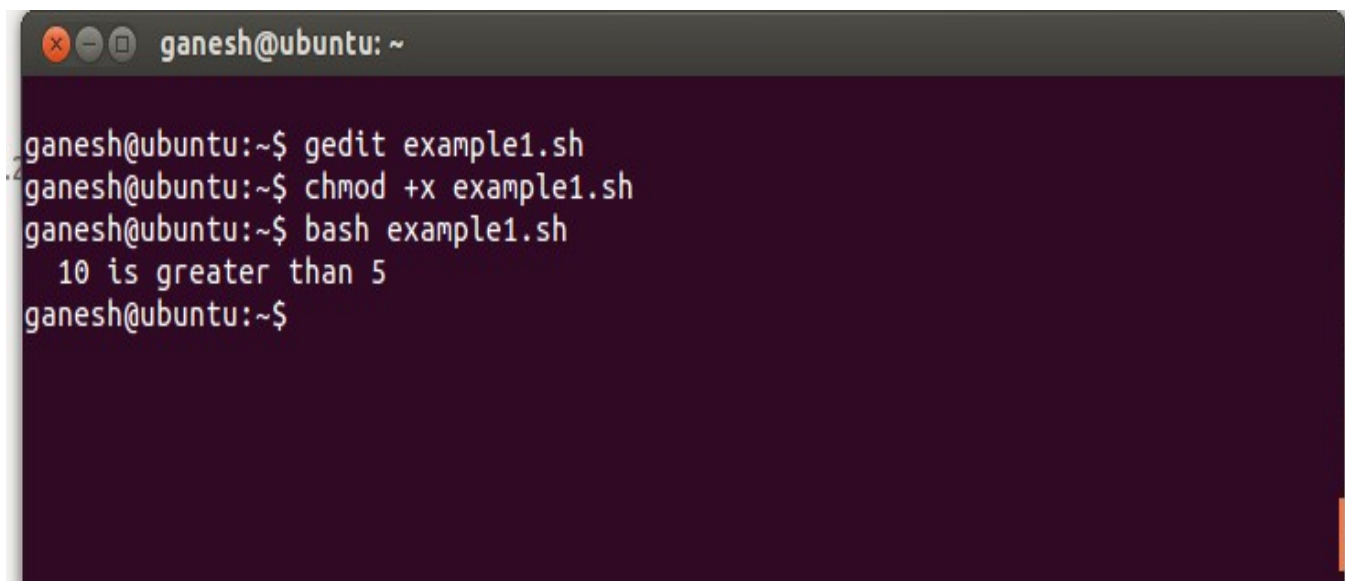
The screenshot shows a gedit editor window titled "example1.sh (~) - gedit". The window has a menu bar with "Open", "Save", "Undo", and other standard editing icons. The script content is as follows:

```
#!/bin/bash

var1=10
var2=5

if(($var1 > $var2 ))
then
echo " $var1 is greater than $var2 "
else
echo " $var2 is greater than $var1 "
fi
```

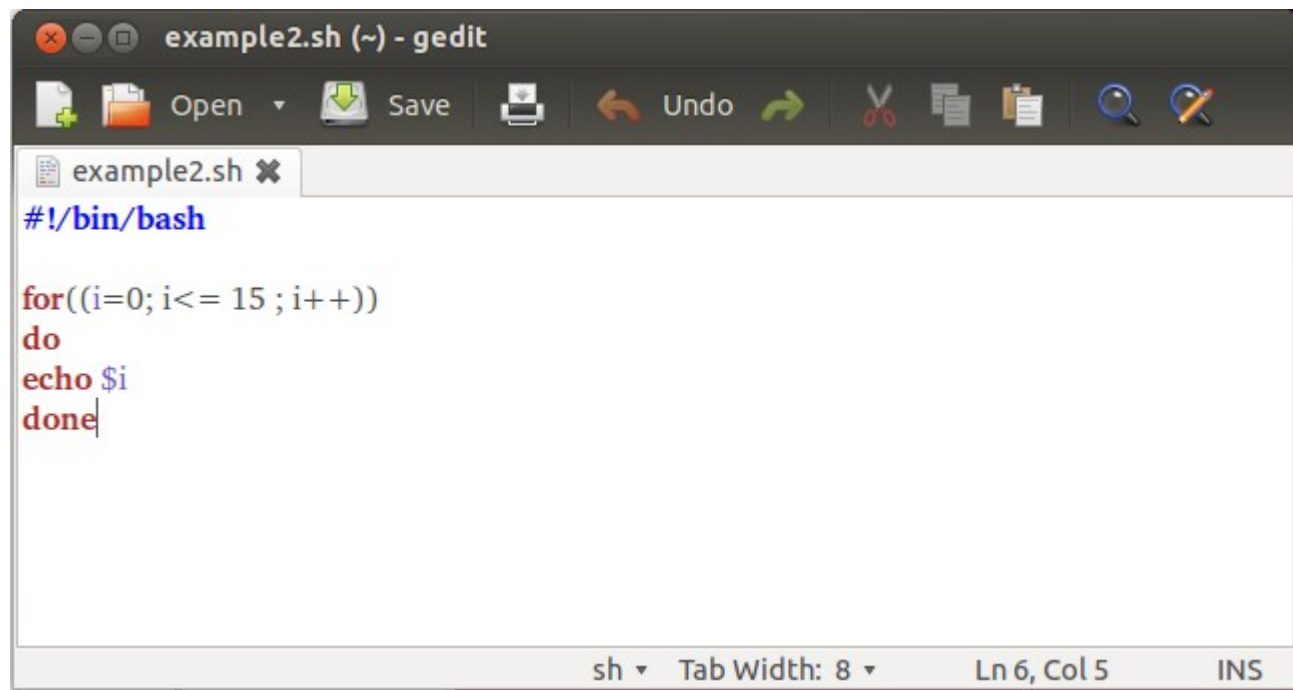
The status bar at the bottom indicates "sh", "Tab Width: 8", "Ln 5, Col 1", and "INS".



The screenshot shows a terminal window with the prompt "ganesh@ubuntu: ~". The user has executed the following commands:

```
ganesh@ubuntu:~$ gedit example1.sh
ganesh@ubuntu:~$ chmod +x example1.sh
ganesh@ubuntu:~$ bash example1.sh
  10 is greater than 5
ganesh@ubuntu:~$
```

Shell script to print first 25 numbers using for loop ?

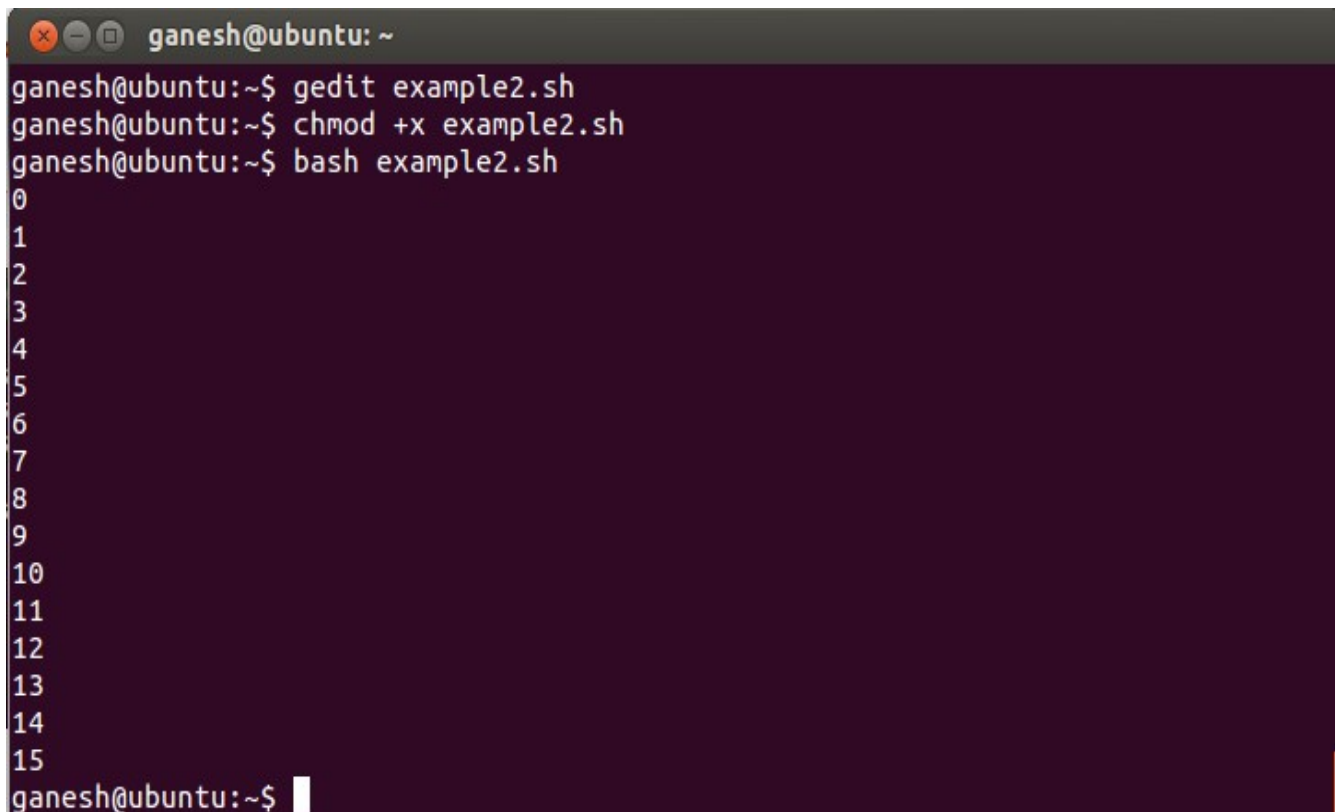


The screenshot shows a gedit editor window titled "example2.sh (~) - gedit". The window has a menu bar with "Open", "Save", "Undo", and other standard editing functions. The main text area contains the following shell script:

```
#!/bin/bash

for((i=0; i<= 15 ; i++))
do
echo $i
done
```

The status bar at the bottom indicates "sh", "Tab Width: 8", "Ln 6, Col 5", and "INS".



The screenshot shows a terminal window with the prompt "ganesh@ubuntu: ~". The user has executed the following commands:

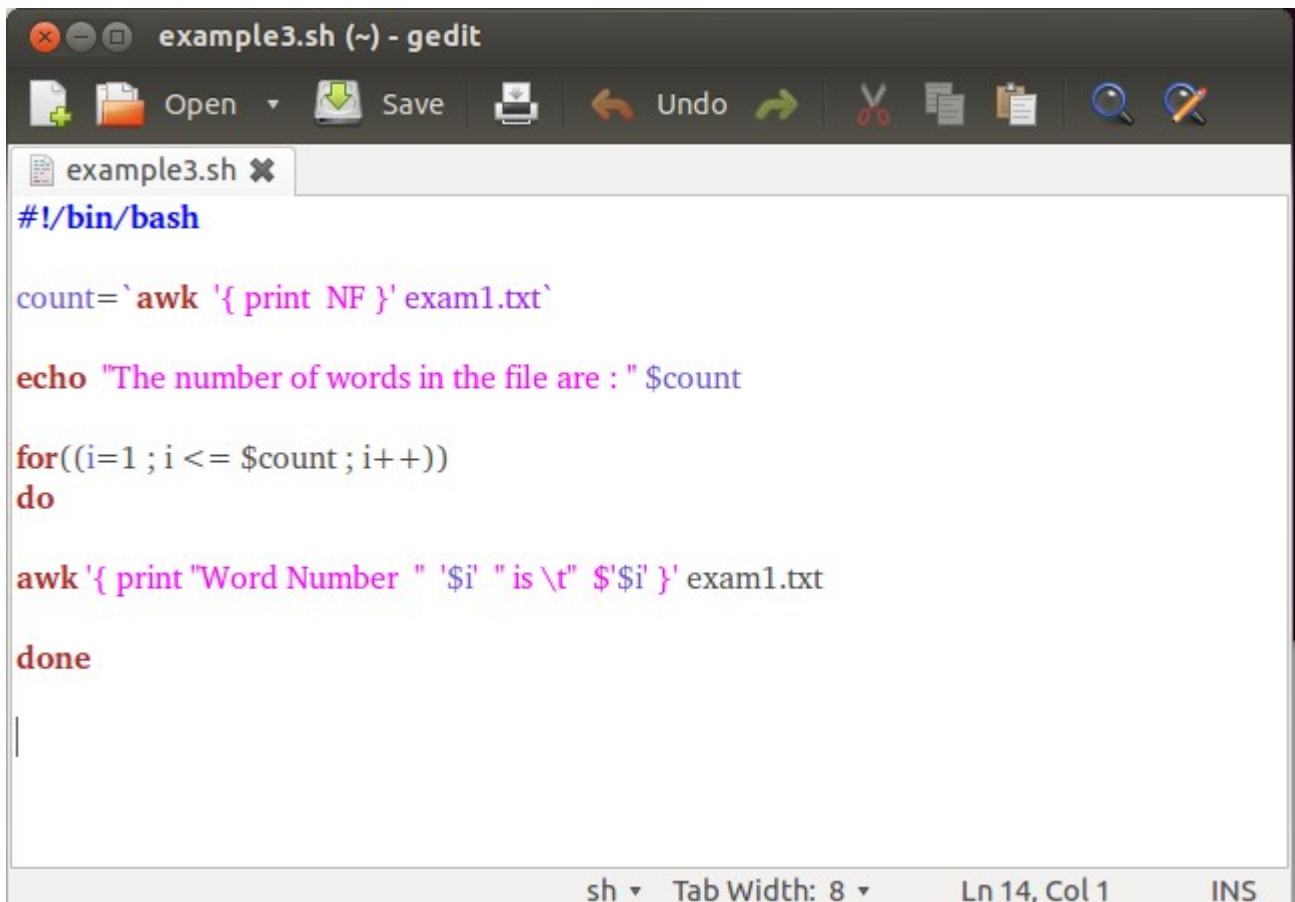
```
ganesh@ubuntu:~$ gedit example2.sh
ganesh@ubuntu:~$ chmod +x example2.sh
ganesh@ubuntu:~$ bash example2.sh
```

The output of the script is a list of numbers from 0 to 15, printed one per line:

```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
```

The terminal prompt "ganesh@ubuntu:~\$" is visible at the bottom, followed by a cursor.

Shell script to count the number of words in line of file and print these words separately ?



```
example3.sh (~) - gedit
#!/bin/bash

count=`awk '{ print NF }' exam1.txt`

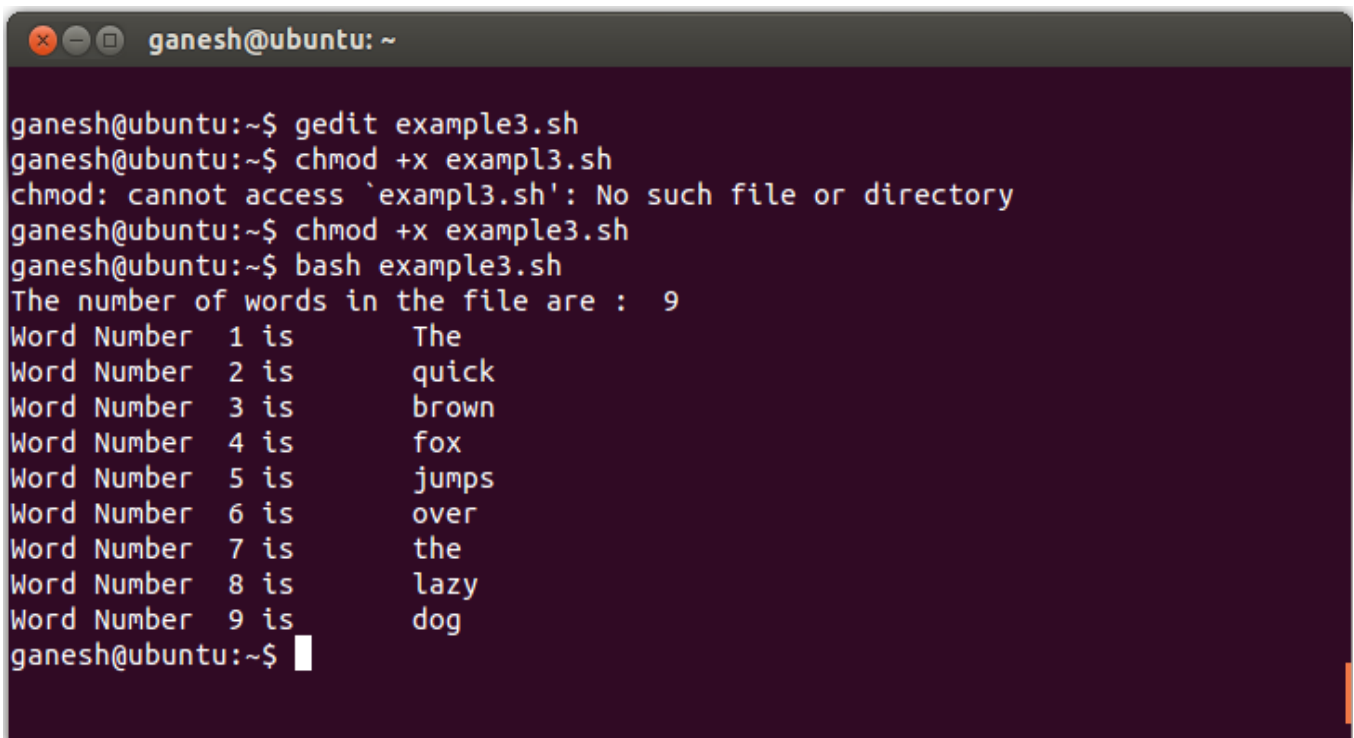
echo "The number of words in the file are : " $count

for((i=1 ; i <= $count ; i++))
do

awk '{ print "Word Number " '$i' " is \t" '$$i' }' exam1.txt

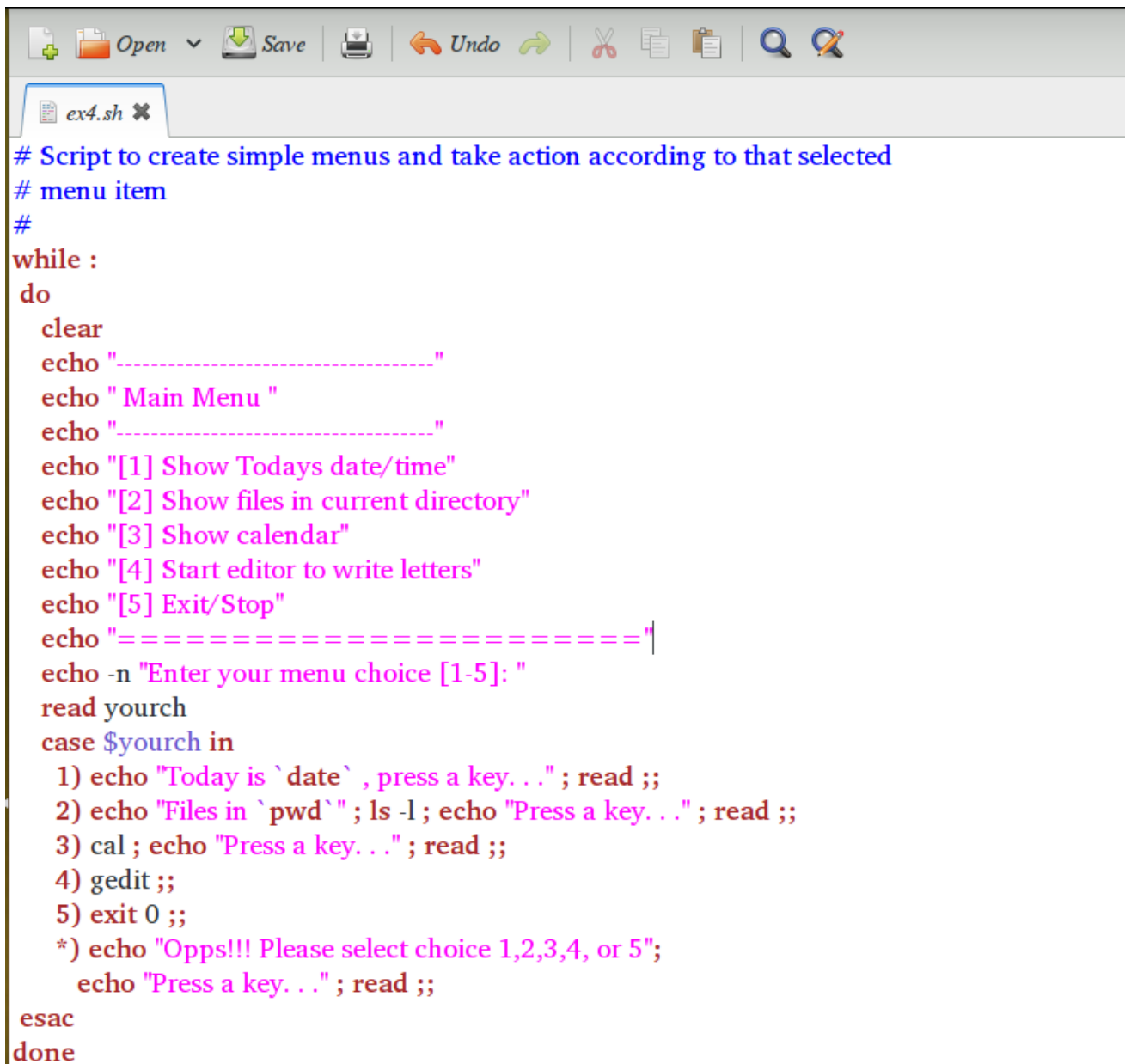
done
```

sh ▾ Tab Width: 8 ▾ Ln 14, Col 1 INS



```
ganesh@ubuntu: ~
ganesh@ubuntu:~$ gedit example3.sh
ganesh@ubuntu:~$ chmod +x exampl3.sh
chmod: cannot access `exampl3.sh': No such file or directory
ganesh@ubuntu:~$ chmod +x example3.sh
ganesh@ubuntu:~$ bash example3.sh
The number of words in the file are : 9
Word Number 1 is      The
Word Number 2 is      quick
Word Number 3 is      brown
Word Number 4 is      fox
Word Number 5 is      jumps
Word Number 6 is      over
Word Number 7 is      the
Word Number 8 is      lazy
Word Number 9 is      dog
ganesh@ubuntu:~$
```

Script to create simple menus and take action according to that selected menu item .



```
# Script to create simple menus and take action according to that selected
# menu item
#
while :
do
clear
echo "-----"
echo " Main Menu "
echo "-----"
echo "[1] Show Todays date/time"
echo "[2] Show files in current directory"
echo "[3] Show calendar"
echo "[4] Start editor to write letters"
echo "[5] Exit/Stop"
echo "=====|
echo -n "Enter your menu choice [1-5]: "
read yourch
case $yourch in
1) echo "Today is `date` , press a key. . ." ; read ;;
2) echo "Files in `pwd`" ; ls -l ; echo "Press a key. . ." ; read ;;
3) cal ; echo "Press a key. . ." ; read ;;
4) gedit ;;
5) exit 0 ;;
*) echo "Oops!!! Please select choice 1,2,3,4, or 5";
echo "Press a key. . ." ; read ;;
esac
done
```

```
ganesh@ubuntu: ~
-----
Main Menu
-----
[1] Show Todays date/time
[2] Show files in current directory
[3] Show calendar
[4] Start editor to write letters
[5] Exit/Stop
=====
Enter your menu choice [1-5]: 1
Today is Sat Sep 29 23:52:38 IST 2012 , press a key. . .
█
```

```
ganesh@ubuntu: ~
-----
Main Menu
-----
[1] Show Todays date/time
[2] Show files in current directory
[3] Show calendar
[4] Start editor to write letters
[5] Exit/Stop
=====
Enter your menu choice [1-5]: 3
    September 2012
Su Mo Tu We Th Fr Sa
      1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30
Press a key. . .
█
```

Learning Outcome:

To test the learning for this lab session, we could ask you to write shell scripts for performing some functions. Here are a few test questions:

- ❏ Write a shell script to find the number of occurrences of a vowel in the file abc.txt
- ❏ Write a shell script to get the minimum number in exam2.txt
- ❏ Write a shell script that adds a.ext extension for each file in a directory (which would be taken as a commandline argument)
- ❏ Write a shell script to create 10 sub-directories in a directory (taken as a first command line argument) and named as 'dir1', 'dir2', 'dir3' and so on if the choice (taken as 2nd command line argument) is 1 or else named as 'dir-a', 'dir-b', 'dir-c' and so on if the choice is 2.

These would test your knowledge of regular expressions, grep, awk, sed and other linux commands and shell programming. Be prepared with them.

References:

SM course slide on Regular Expression.

Some References (It is recommended to read them before coming to the lab)

- 1) Regular expressions : <http://www.grymoire.com/Unix/Regular.html>
- 2) Example Shell Programmes : http://www.arachnoid.com/linux/shell_programming.html

Online Shell Script Executer : http://www.compileonline.com/execute_bash_online.php