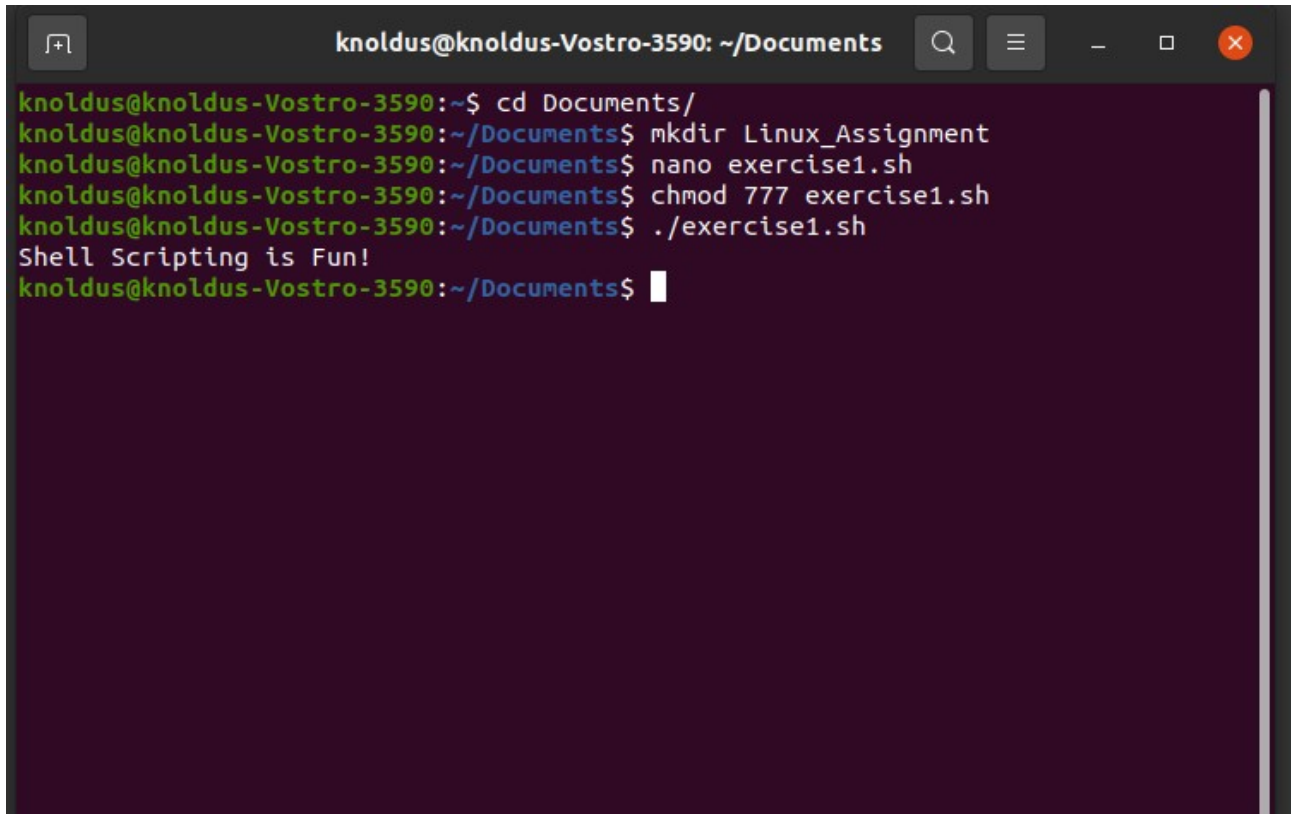


LINUX ASSIGNMENT

Exercise_1 - Write a shell script that prints "Shell Scripting is Fun!" on the screen


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

```
echo "Shell Scripting is Fun!"
```



A terminal window titled "knoldus@knoldus-Vostro-3590: ~/Documents" with search, menu, and window control icons. The terminal shows the following commands and output:

```
knoldus@knoldus-Vostro-3590:~$ cd Documents/  
knoldus@knoldus-Vostro-3590:~/Documents$ mkdir Linux_Assignment  
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise1.sh  
Shell Scripting is Fun!  
knoldus@knoldus-Vostro-3590:~/Documents$
```



A terminal window titled "knoldus@knoldus-Vostro-3590: ~/Documents" with search, menu, and window control icons. The window shows the GNU nano 4.8 editor editing the file "exercise1.sh". The editor's status bar at the top indicates "GNU nano 4.8", "exercise1.sh", and "Modified". The file content is:

```
#!/bin/bash  
echo "Shell Scripting is Fun!"
```

The bottom of the window displays a list of nano editor shortcuts:

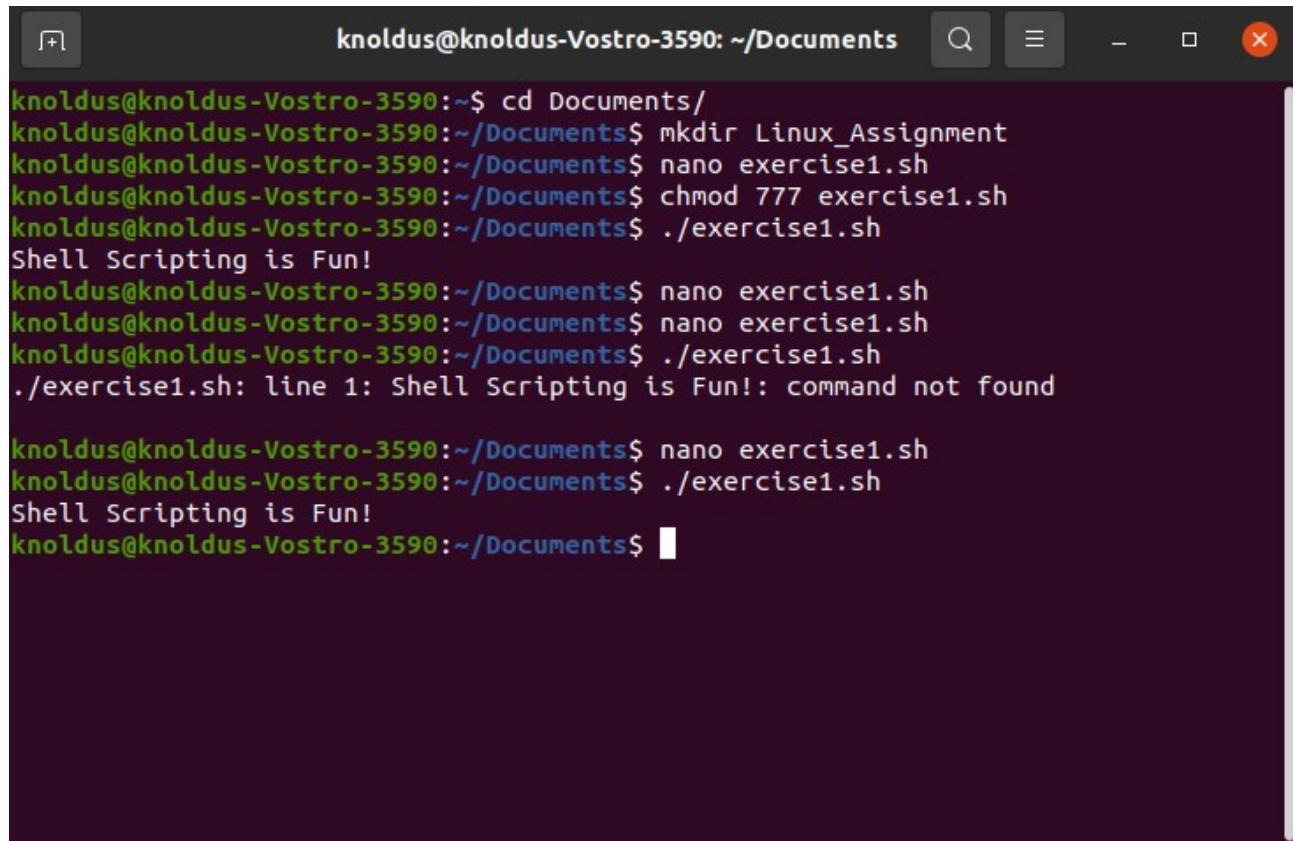
^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^I Replace	^U Paste Text	^T To Spell	^_ Go To Line

Exercise_2 - Modify the shell script from exercise 1 to include a variable. The variable will hold the contents of the message "Shell Scripting is Fun!"

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

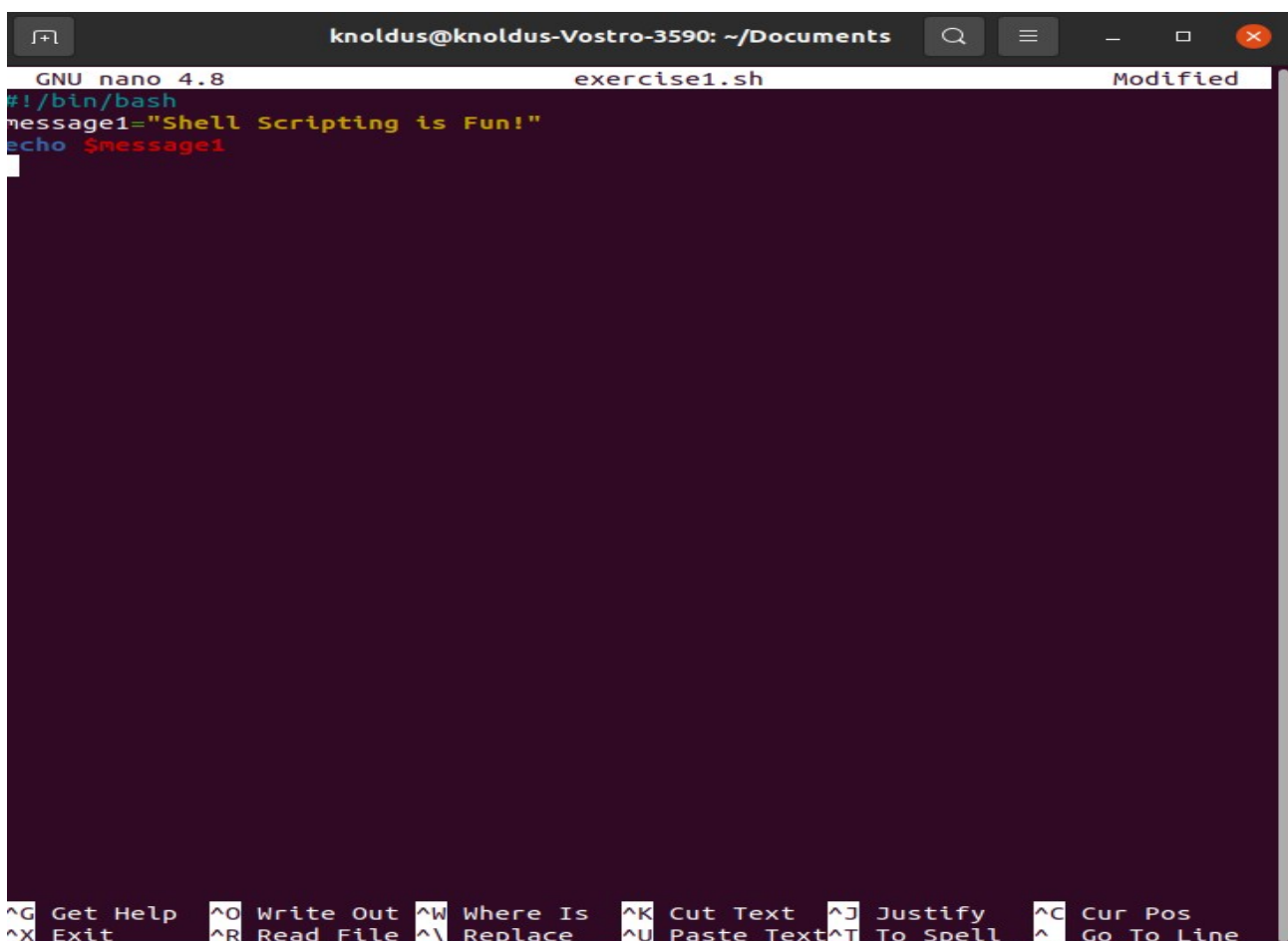
```
message1="Shell Scripting is Fun!"
```

```
echo $message1
```



A terminal window titled "knoldus@knoldus-Vostro-3590: ~/Documents" showing the following commands and output:

```
knoldus@knoldus-Vostro-3590:~$ cd Documents/  
knoldus@knoldus-Vostro-3590:~/Documents$ mkdir Linux_Assignment  
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise1.sh  
Shell Scripting is Fun!  
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise1.sh  
./exercise1.sh: line 1: Shell Scripting is Fun!: command not found  
  
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise1.sh  
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise1.sh  
Shell Scripting is Fun!  
knoldus@knoldus-Vostro-3590:~/Documents$
```



A terminal window titled "knoldus@knoldus-Vostro-3590: ~/Documents" showing the nano editor editing "exercise1.sh". The editor header shows "GNU nano 4.8" and "Modified". The content of the file is:

```
#!/bin/bash  
message1="Shell Scripting is Fun!"  
echo $message1
```

The bottom of the window shows nano editor shortcuts:

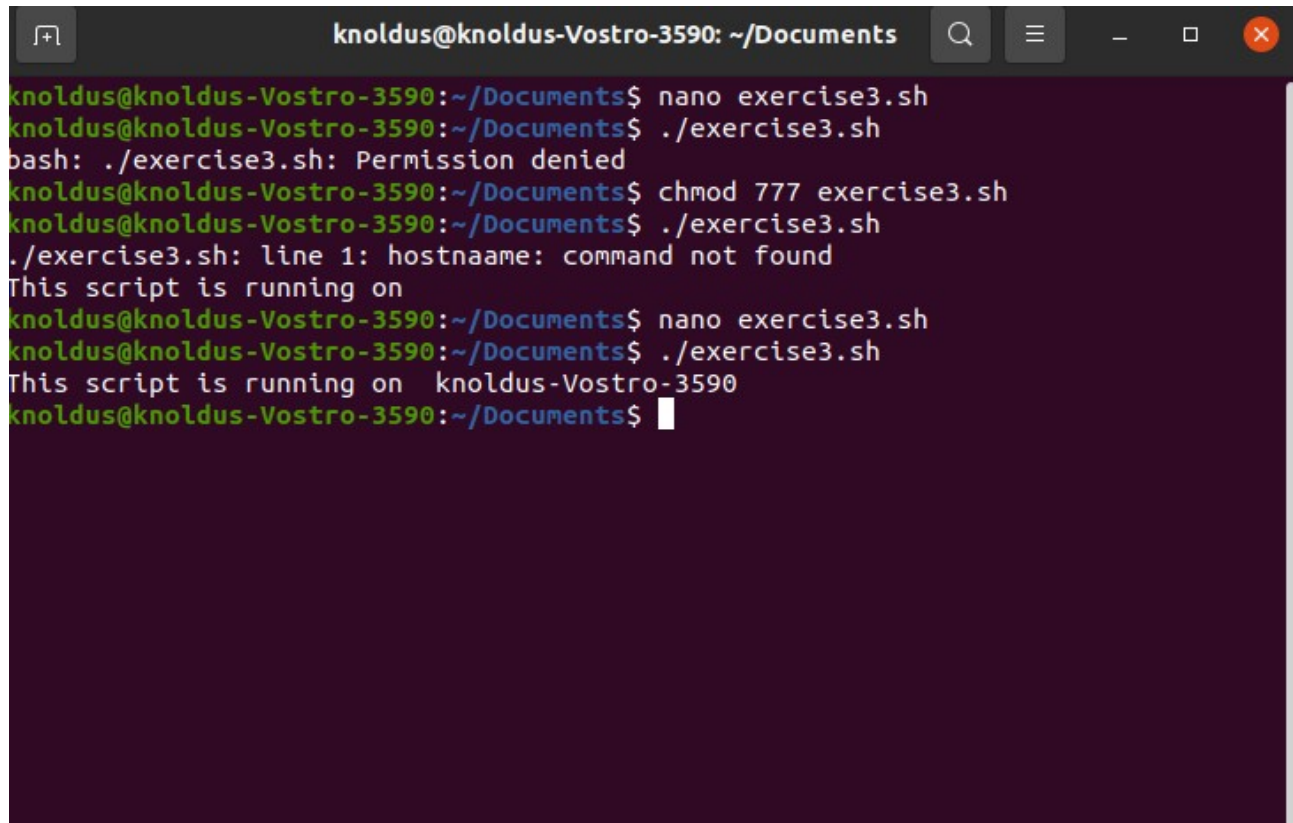
```
^G Get Help    ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos  
^X Exit        ^R Read File  ^\ Replace    ^U Paste Text ^T To Spell   ^_ Go To Line
```

Exercise_3 - Store the output of the command “hostname” in a variable. Display “This script is running on _.” where “_” is the output of the “hostname” command.

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

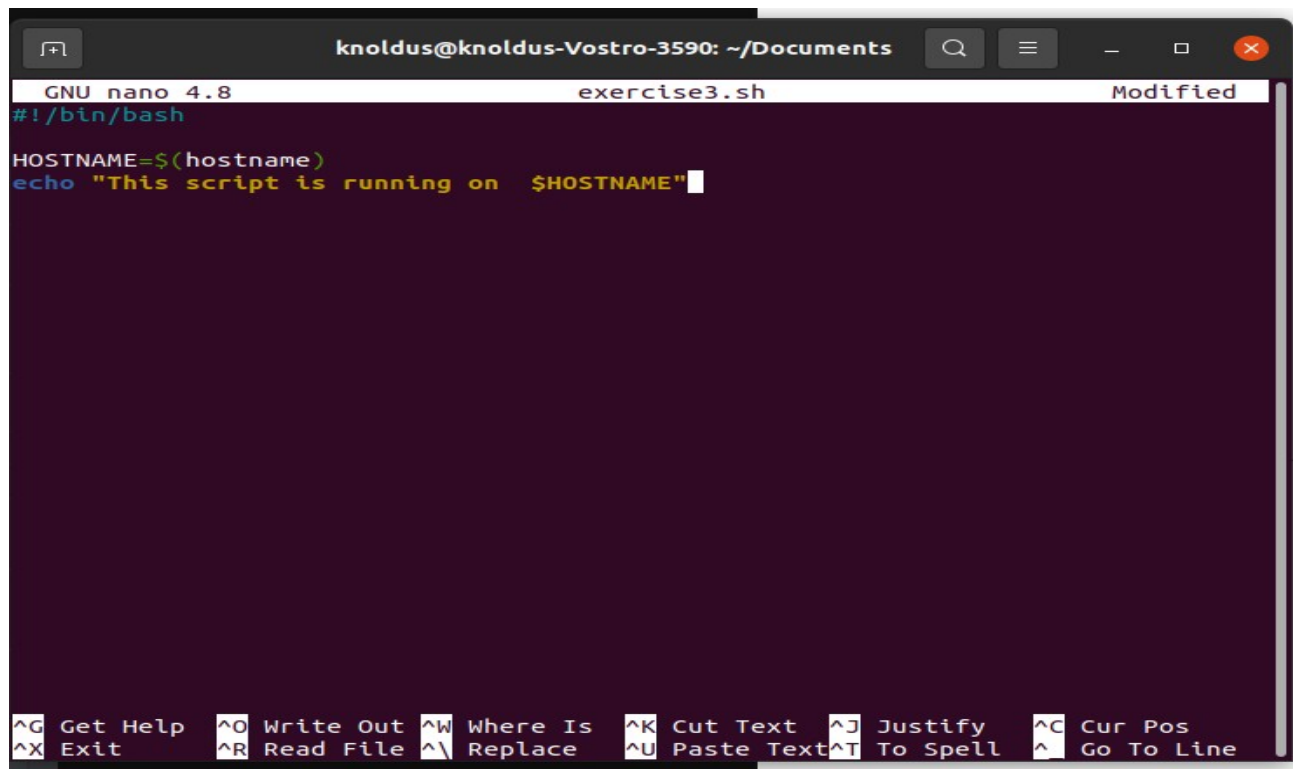
```
HOSTNAME=$(hostname)
```

```
echo "This script is running on $HOSTNAME"
```



A terminal window titled 'knoldus@knoldus-Vostro-3590: ~/Documents' showing the execution of a script. The user runs 'nano exercise3.sh' to create the file. Then they run './exercise3.sh', which fails with 'Permission denied'. They then run 'chmod 777 exercise3.sh' and run './exercise3.sh' again. This time it fails with 'line 1: hostname: command not found'. They then run 'nano exercise3.sh' again to edit the script. Finally, they run './exercise3.sh' and it successfully outputs 'This script is running on knoldus-Vostro-3590'.

```
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise3.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise3.sh
bash: ./exercise3.sh: Permission denied
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise3.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise3.sh
./exercise3.sh: line 1: hostname: command not found
This script is running on
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise3.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise3.sh
This script is running on knoldus-Vostro-3590
knoldus@knoldus-Vostro-3590:~/Documents$
```



A terminal window titled 'knoldus@knoldus-Vostro-3590: ~/Documents' showing the nano editor editing the file 'exercise3.sh'. The editor shows the script content: '#!/bin/bash', 'HOSTNAME=\$(hostname)', and 'echo "This script is running on \$HOSTNAME"'. The bottom status bar shows various nano editor shortcuts.

```
GNU nano 4.8 exercise3.sh Modified
#!/bin/bash

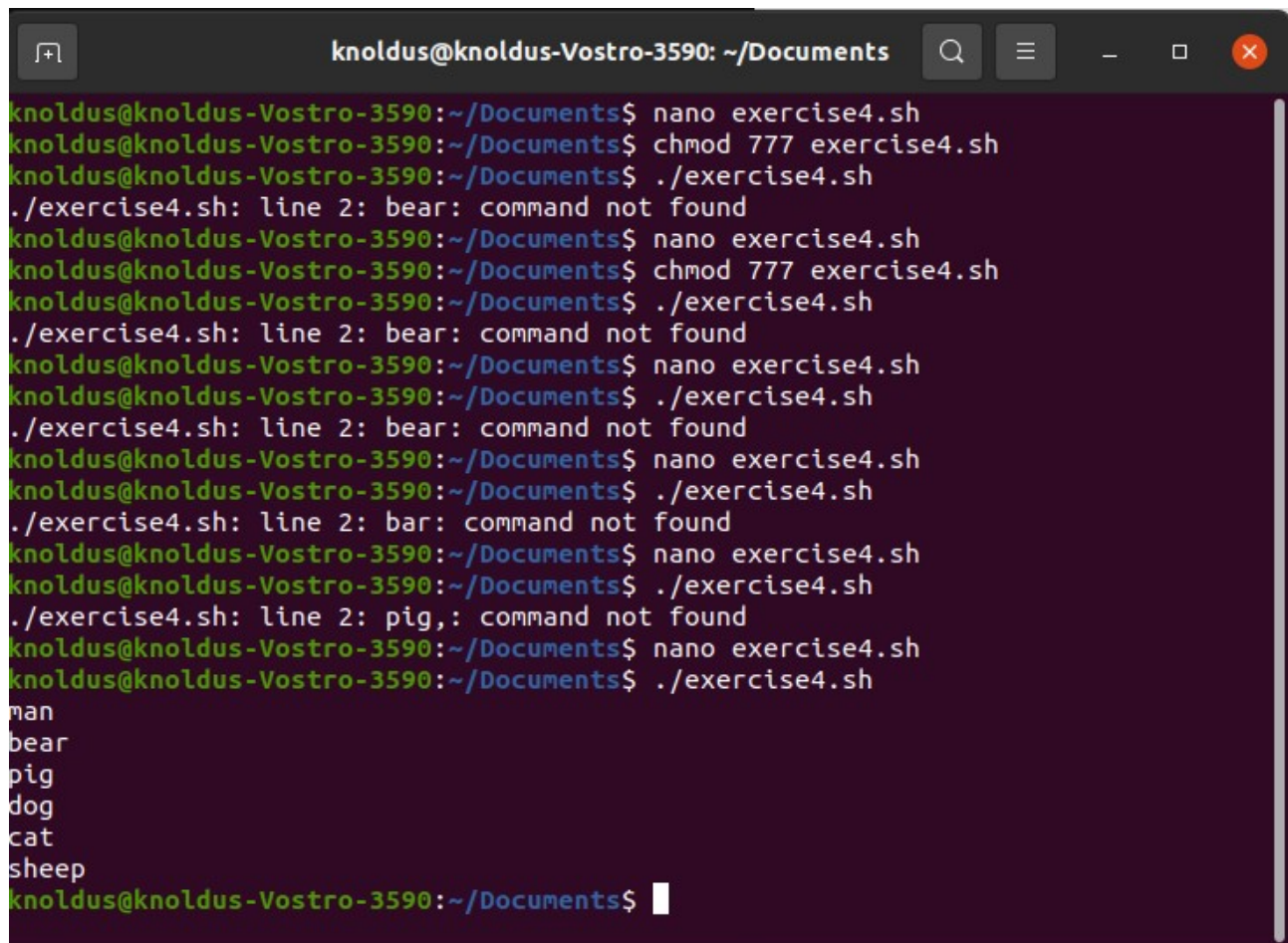
HOSTNAME=$(hostname)
echo "This script is running on $HOSTNAME"
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line

4 - Write a shell script that displays "man","bear","pig","dog","cat",and "sheep" on the screen with each appearing on a separate line. Try to do this in as few lines as possible.

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

```
echo $'man\nbear\npig\ndog\ncat\nsheep'
```



A terminal window titled 'knoldus@knoldus-Vostro-3590: ~/Documents' showing the execution of a script named 'exercise4.sh'. The user runs 'nano exercise4.sh', 'chmod 777 exercise4.sh', and then './exercise4.sh' multiple times. Each execution results in an error: 'line 2: bear: command not found', 'line 2: bar: command not found', and 'line 2: pig,: command not found'. Finally, the script is run again and it successfully outputs the words 'man', 'bear', 'pig', 'dog', 'cat', and 'sheep' on separate lines.

```
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
./exercise4.sh: line 2: bear: command not found
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
./exercise4.sh: line 2: bear: command not found
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
./exercise4.sh: line 2: bear: command not found
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
./exercise4.sh: line 2: bar: command not found
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
./exercise4.sh: line 2: pig,: command not found
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise4.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise4.sh
man
bear
pig
dog
cat
sheep
knoldus@knoldus-Vostro-3590:~/Documents$
```



A terminal window titled 'knoldus@knoldus-Vostro-3590: ~/Documents' showing the nano editor editing 'exercise4.sh'. The editor shows the script content: '#!/bin/bash' and 'echo \$'man\nbear\npig\ndog\ncat\nsheep''. The bottom status bar shows 'GNU nano 4.8 exercise4.sh' and a list of keyboard shortcuts.

```
GNU nano 4.8 exercise4.sh
#!/bin/bash
echo $'man\nbear\npig\ndog\ncat\nsheep'
```

Read 3 lines

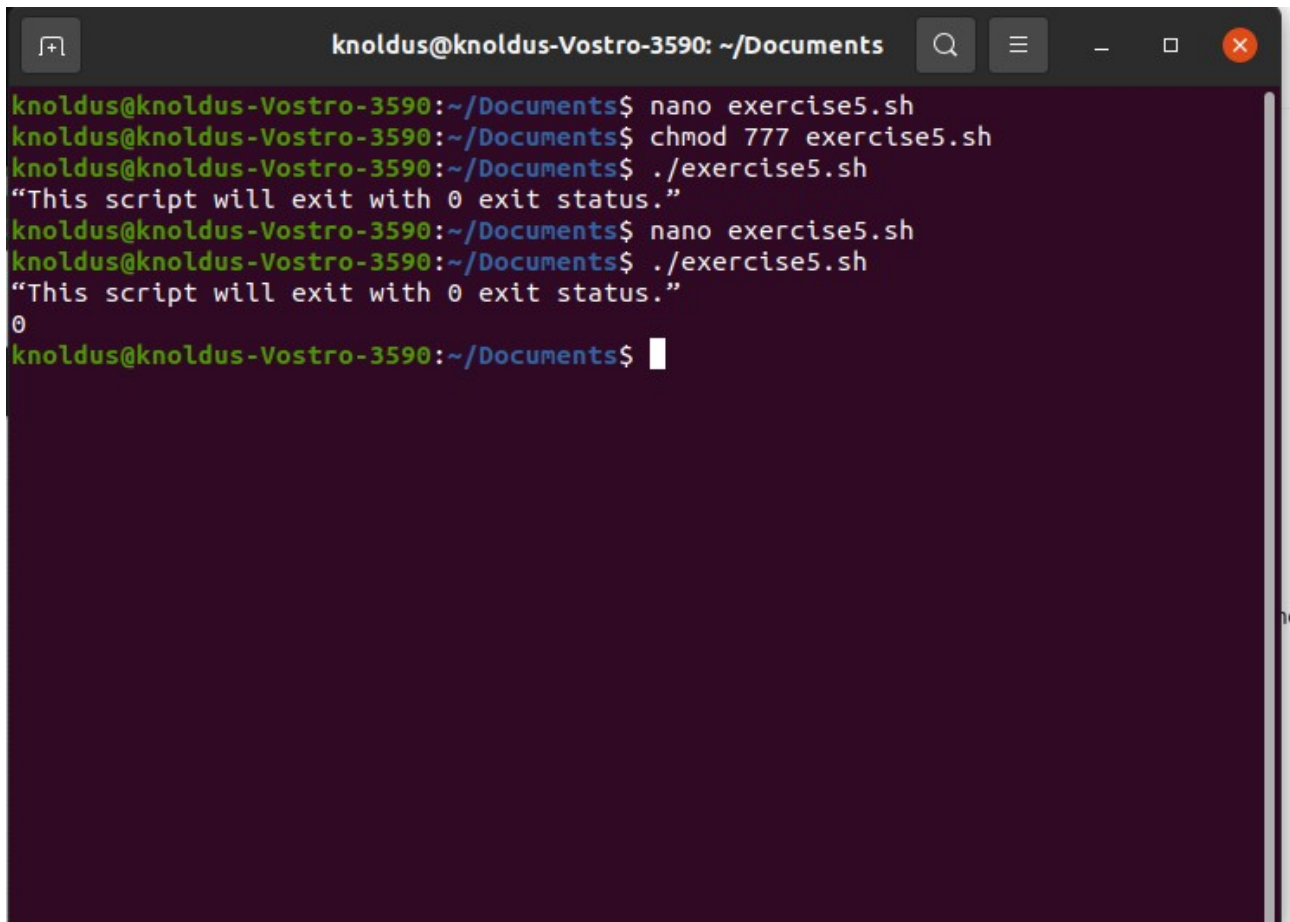
^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Paste Text	^T To Spell	^_ Go To Line

Exercise_5 - Write a shell script that displays, "This script will exit with 0 exit status." Be sure that the script does indeed exit with a 0 exit status.

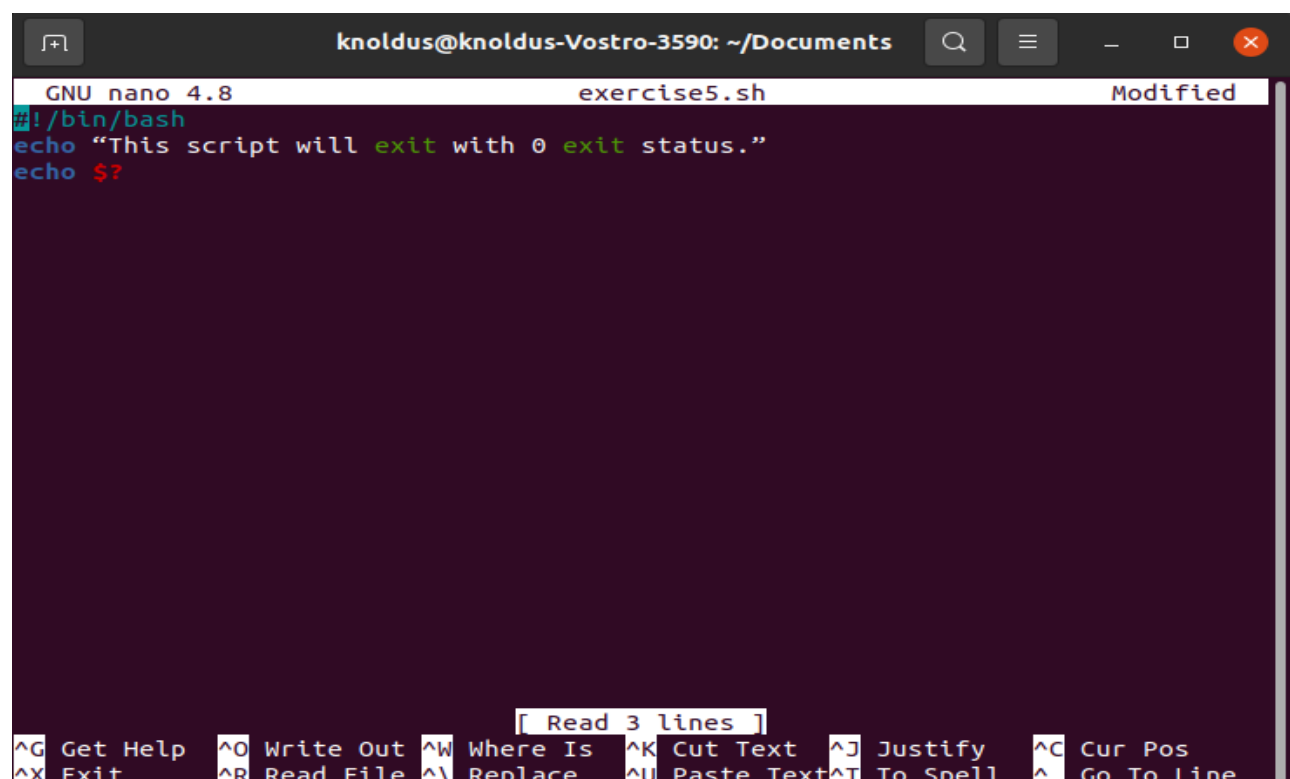
```
#!/bin/bash
```

```
echo "This script will exit with 0 exit status."
```

```
echo $?
```



```
knoldus@knoldus-Vostro-3590: ~/Documents
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise5.sh
knoldus@knoldus-Vostro-3590:~/Documents$ chmod 777 exercise5.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise5.sh
"This script will exit with 0 exit status."
knoldus@knoldus-Vostro-3590:~/Documents$ nano exercise5.sh
knoldus@knoldus-Vostro-3590:~/Documents$ ./exercise5.sh
"This script will exit with 0 exit status."
0
knoldus@knoldus-Vostro-3590:~/Documents$
```



```
GNU nano 4.8 exercise5.sh Modified
#!/bin/bash
echo "This script will exit with 0 exit status."
echo $?
```

Read 3 lines

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Paste Text	^T To Spell	^_ Go To Line

Exercise_6 - Write a shell script that consists of a function that displays the number of files in the present working directory. Name this function “file_count” and call it in your script. If you use variable in your function, remember to make it a local variable.

```
#!/bin/bash

function file_count()
{
    local NUMBER_OF_FILE=$(ls -1 | wc -l)
    echo "$NUMBER_OF_FILE"
}

file_count
```

