BASH ASSIGNMENT- Module 4

Gokula Madhavan B

Sri Sairam Engineering College

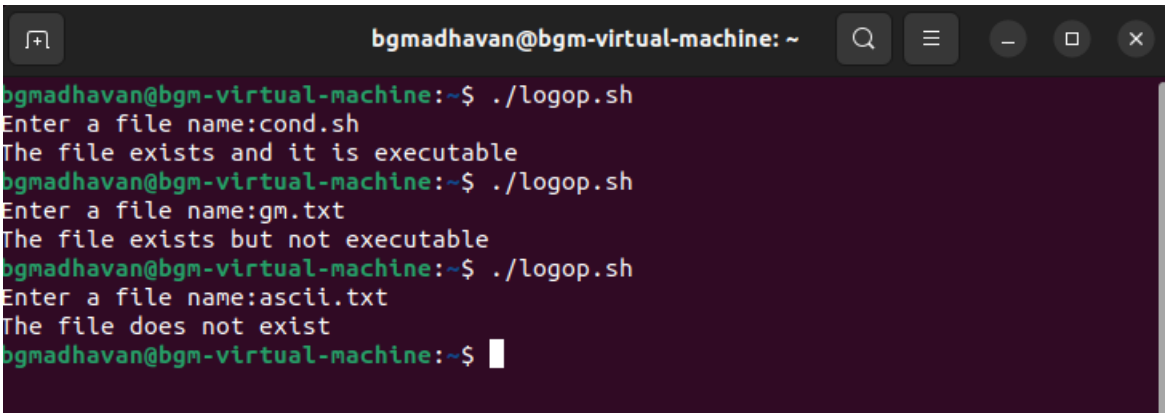
Logical Operators

1) Check whether the file exists and is executable using logical operators.
   Hint:man test

Open

logop.sh
~/

```
#!/bin/bash
read -p "Enter a file name:" fname
if [ -e "$fname" -a -x "$fname" ];then
        echo "The file exists and it is executable"
elif [ -e "$fname" ];then
        echo "The file exists but not executable"
else
        echo "The file does not exist"
fi
```



Arithmetic operators

1)Write a program to demonstrate the use of not equal to operator.
   Hint: -ne

Open

arithop.sh
~/

```
#!/bin/bash
if [ "$1" -ne 64 ]; then
        echo "The entered number  is $1 and it is not equal to 64"
else
        echo "The entered number is equal to 64"
fi
```

```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ ./arithop.sh 64
The entered number is equal to 64
bgmadhavan@bgm-virtual-machine:~$ ./arithop.sh 72
The entered number  is 72 and it is not equal to 64
bgmadhavan@bgm-virtual-machine:~$
```

String and file attributes

1)Explore some more attributes
   -r
   -x
   -o

```
-r FILE
       FILE exists and read permission is granted
```

```
-x FILE
       FILE exists and execute (or search) permission is granted
```

```
-O FILE
       FILE exists and is owned by the effective user ID
```

Open strfile.sh ~/ Save

```
1 #!/bin/bash
2 read -p "Enter a file name : " fname
3 if [ -r "$fname" -a -O "$fname" -a -x "$fname" ]; then
4         echo "The file $fname is readable and owned by with user ID $(id -u) and is executable"
5 fi
```

```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ ./strfile.sh
Enter a file name : logop.sh
The file logop.sh is readable and owned by with user ID 1000 and is executable
bgmadhavan@bgm-virtual-machine:~$
```

Conditional Loops

1) Find the sum of first n prime numbers.

```
#!/bin/bash
read -p "Enter number of prime numbers to print sum: " lim
num=2
ans=0
echo -n "The sum of first $lim prime numbers is : "
while [ $lim -gt 0 ]; do
        isprime=1
        for ((i=2;(i*i)<= $num;i++)); do
                if [ $(($num % i)) == 0 ]; then
                        isprime=0
                        break
                fi
        done
        if [ $isprime -eq 1 ]; then
                ans=$[$ans+$num]
                lim=$[$lim-1]
        fi
        num=$[$num+1]
done
echo "$ans"
```

```
bgmadhavan@bgm-virtual-machine:~$ ./prime.sh
Enter number of prime numbers to print sum: 10
The sum of first 10 prime numbers is : 129
bgmadhavan@bgm-virtual-machine:~$ ./prime.sh
Enter number of prime numbers to print sum: 5
The sum of first 5 prime numbers is : 28
bgmadhavan@bgm-virtual-machine:~$
```

More on Loops

1) Retype nested-for.sh bash script using nested while loop
2) Save your program with the name: nested-while.sh

Open

filelisting.sh
~/

```
#!/bin/bash
ls -d test*/ | while read -r dir; do
        echo "Files in the directory $dir"
        echo ""
        ls -1 $dir | while read -r file;do
                echo "$file"
        done
        echo "---------------------------------"
done
```

```
bgmadhavan@bgm-virtua
bgmadhavan@bgm-virtual-machine:~$ ./filelisting.sh
Files in the directory test1/

file1
file2
file3
---------------------------------
Files in the directory test2/

file1
file2
file3
---------------------------------
Files in the directory test3/

file1
file2
file3
---------------------------------
```

Case statement

1) Write a menu driven program for mathematical calculation
   a. It should take user inputs a and b
   b. It should ask for mathematical operator (+, -, / and *).
   c. Do the calculation
   d. Print the output

Open ▾ switchcase.sh ~/

```
1 #!/bin/bash
2
3 read -p "Enter two numbers seperated by space : " a b
4 echo "Choose an operation to perform:"
5 echo "+ for addition"
6 echo "- for Subtraction"
7 echo "x for Multiplication"
8 echo "/ for division"
9 read -p "Choose an operation: " op
10
11 case $op in
12         '+') echo "The sum of numbers $a and $b is : $(($a+$b))"
13         ;;
14         '-') echo "The difference of numbers $a and $b is : $(($a-$b))"
15         ;;
16         'x') echo "The product of numbers $a and $b is : $(($a*$b))"
17         ;;
18         '/') if [ $b -ne 0 ]; then
19                 echo "The quotient of numbers $a and $b is : $(($a / $b)) and the remainder is $(($a%$b)) "
20              else
21                 echo "Cannot divide by zero"
22              fi
23         ;;
24 esac
```

```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ ./switchcase.sh
Enter two numbers seperated by space : 20 4
Choose an operation to perform:
+ for addition
- for Subtraction
x for Multiplication
/ for division
Choose an operation: /
The quotient of numbers 20 and 4 is : 5 and the remainder is 0
bgmadhavan@bgm-virtual-machine:~$ ./switchcase.sh
Enter two numbers seperated by space : 6 7
Choose an operation to perform:
+ for addition
- for Subtraction
x for Multiplication
/ for division
Choose an operation: x
The product of numbers 6 and 7 is : 42
bgmadhavan@bgm-virtual-machine:~$
```

Using File Descriptors

1) Try to append few lines to a file test.txt using file descriptor.
2) Display the content of the file using file descriptor.

Open | filedesc.sh ~/ | Save

```
1 #!/bin/bash
2 exec 3> fd.txt
3 echo "This is a shell script to demonstrate file descriptors" >&3
4 echo "Here both input and display file descriptora are demonstrated" >&3
5 exec 3<&-
6 exec 3< fd.txt
7 cat <&3
8 exec 3<&-
```

```
bgmadhavan@bgm-virtual-machine: ~
bgmadhavan@bgm-virtual-machine:~$ gedit filedesc.sh &
[2] 37941
bgmadhavan@bgm-virtual-machine:~$ chmod +x filedesc.sh
[2]+  Done                    gedit filedesc.sh
bgmadhavan@bgm-virtual-machine:~$ ./filedesc.sh
cat: -: Bad file descriptor
bgmadhavan@bgm-virtual-machine:~$ ./filedesc.sh
This is a shell script to demonstrate file descriptors
Here both input and display file descriptora are demonstrated
bgmadhavan@bgm-virtual-machine:~$
```

Basics of functions

1) Write a program with two functions:

a. The first function should display diskspace usage in human readable form.

(Hint: df -h)

b. The second function should display filesystem usage in human readable form.

(Hint: du -h)

Open

fu

```
1 #!/bin/bash
2 disk_space()
3 {
4         echo "Disk space information";
5         df -h | more
6 }
7 file_sys()
8 {
9         echo "File System status";
10        du -h | more
11 }
12
13 echo "MAIN PROGRAM"
14 disk_space
15 file_sys
16 echo "End of execution"
```

```
bgmadhavan@bgm-virtual-machine:~$ gedit func.sh &
[1] 38035
bgmadhavan@bgm-virtual-machine:~$ chmod +x func.sh
bgmadhavan@bgm-virtual-machine:~$ ./func.sh
MAIN PROGRAM
Disk space information
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           388M  2.1M  386M   1% /run
/dev/sda3        20G   13G  5.5G  70% /
tmpfs           1.9G     0  1.9G   0% /dev/shm
tmpfs           5.0M  4.0K  5.0M   1% /run/lock
/dev/sda2       512M  6.1M  506M   2% /boot/efi
tmpfs           388M  120K  388M   1% /run/user/1000
/dev/sr0        156M  156M     0 100% /media/bgmadhavan/CDROM
/dev/sr1        4.7G  4.7G     0 100% /media/bgmadhavan/Ubuntu 22.04.3 LTS amd64
File System status
8.0K    ./.config/gedit
4.0K    ./.config/gnome-session/saved-session
8.0K    ./.config/gnome-session
4.0K    ./.config/enchant
```

More on functions

1) Write a program,

a. where the function accepts two arguments.

b. The function should multiply the two arguments.

c. Make 3 function calls with arguments - (1, 2), (2, 3) and (3, 4)

Open

func2.sh
~/

```
1 #!/bin/bash
2 func_demo()
3 {
4         echo "The product of the numbers $1 and $2 is : $(($1*$2))"
5 }
6
7 func_demo 1 2
8 func_demo 2 3
9 func_demo 3 4
10
```

bgmadhavan@bgm-virtual-machine:

```
bgmadhavan@bgm-virtual-machine:~$ gedit func2.sh &
[2] 38092
[1]   Done                    gedit func.sh
bgmadhavan@bgm-virtual-machine:~$ chmod +x func2.sh
bgmadhavan@bgm-virtual-machine:~$ ./func2.sh
The product of the numbers 1 and 2 is : 2
The product of the numbers 2 and 3 is : 6
The product of the numbers 3 and 4 is : 12
bgmadhavan@bgm-virtual-machine:~$
```

Arrays and functions

1) Write a program,
    a. Where a function adds all the elements in an array.
    b. The function should display the sum of elements.
    c. Make 2 function calls with array elements- (1, 2, 3) and (4, 5, 6).

Open

array_func.sh
~/

```
#!/bin/bash
array_func()
{       local add=0
        array=($@)
        len=(${#array[@]})
        for((i=0;i<len;i++)); do
                add=$[$add+${array[i]}]
        done
        echo "The sum of elements in the array (${array[@]}) is $add"
}
array=(1 2 3)
array_func ${array[@]}
array2=(4 5 6)
array_func ${array2[@]}
```

```
bgmadhavan@bgm-virtual-machine:~$ ./array_func.sh
The sum of elements in the array (1 2 3) is 6
The sum of elements in the array (4 5 6) is 15
bgmadhavan@bgm-virtual-machine:~$
```