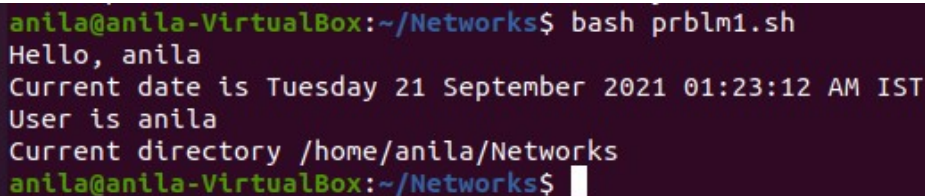


Date: 06-08-2021

EXPERIMENT-14**LINUX SHELL SCRIPTING PROBLEMS****1. Write a shell script to get the current date, time, username and current working?**

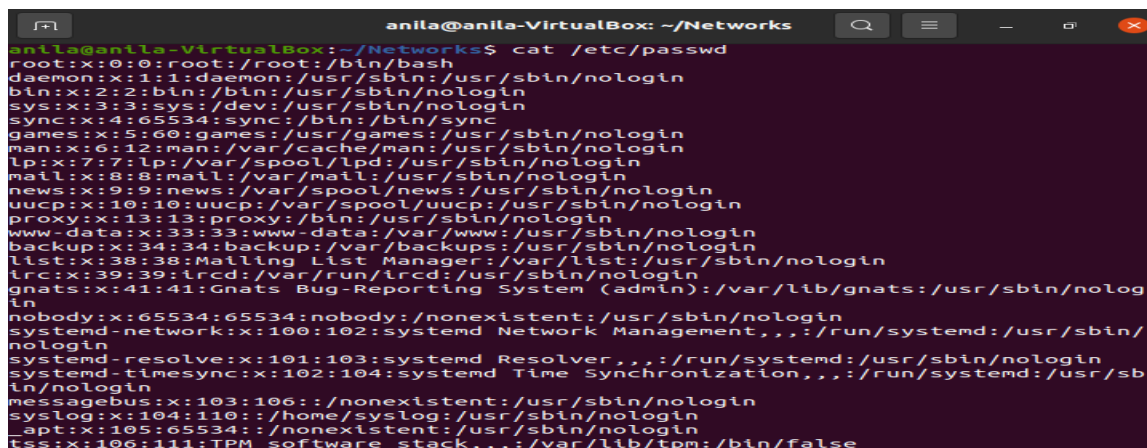
```
#!/bin/bash
echo "Hello, $LOGNAME"
echo "Current date is $(date)"
echo "User is $(whoami)"
echo "Current directory $(pwd)"
```



```
anila@anila-VirtualBox: ~/Networks$ bash prblm1.sh
Hello, anila
Current date is Tuesday 21 September 2021 01:23:12 AM IST
User is anila
Current directory /home/anila/Networks
anila@anila-VirtualBox: ~/Networks$
```

2. How to print the login names of all users on a system?

```
cat etc/passwd
```



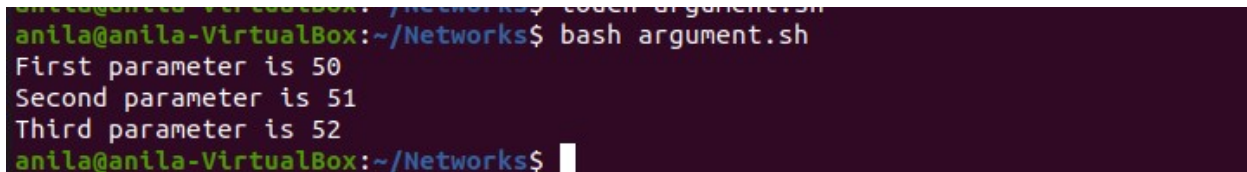
```
anila@anila-VirtualBox: ~/Networks$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:/home/syslog:/usr/sbin/nologin
apt:x:105:65534:/:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
```

3. How can we pass arguments to a script in Linux? And how to access these arguments from within the script?

We can write a bash script that can accept arguments from the command line in the following manner.

```
$ ./scriptName "arg1"

"arg2"... "argn" #!/bin/bash
# sh parameters .sh 50,51,52
echo "First parameter is 50"
echo "Second parameter is 51"
echo "Third parameter is 52"
exit 0
```



```
anila@anila-VirtualBox: ~/Networks$ bash argument.sh
First parameter is 50
Second parameter is 51
Third parameter is 52
anila@anila-VirtualBox: ~/Networks$
```

4. How to set an array in Linux?

1. Indirect Declaration

In Indirect declaration, We assigned a value in a particular index of Array Variable. No need to first declare.

ARRAY NAME[INDEXNR]=value

2. Explicit Declaration

In Explicit Declaration, First We declare an array then assign the values.

declare -a ARRAYNAME

3. Compound Assignment

In Compound Assignment, We declare an array with a bunch of values. We can add other values later too.

ARRAYNAME=(value1 value2.....valueN)

5. How to check if a directory exists?

```
if [ -d "/path/to/dir" ]
then
```

```

echo "Directory /path/to/dir exists."
else
echo "Error: Directory /path/to/dir does not exists."
fi

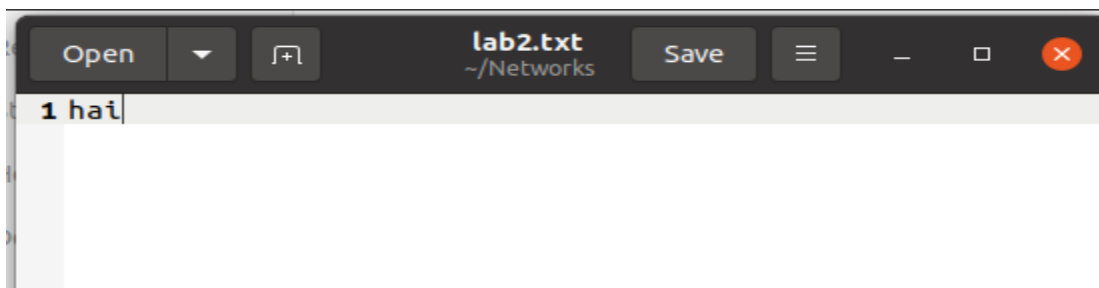
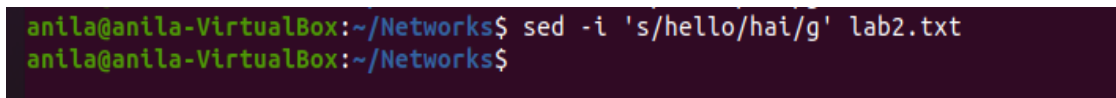
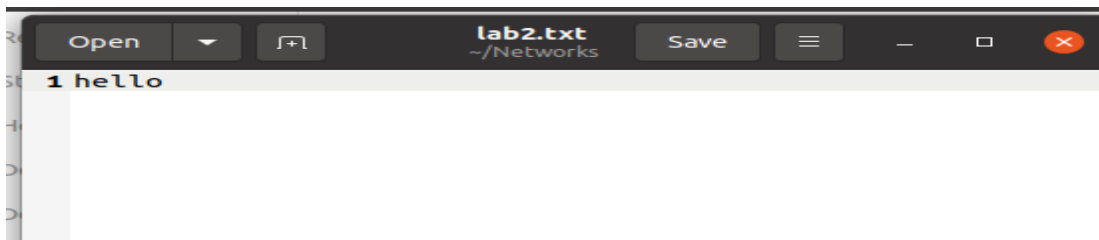
```

6. What is the difference between \$* and \$@?

Ans: \$@ treats each quoted arguments as separate arguments but \$* considers the entire set of positional parameters as a single string.

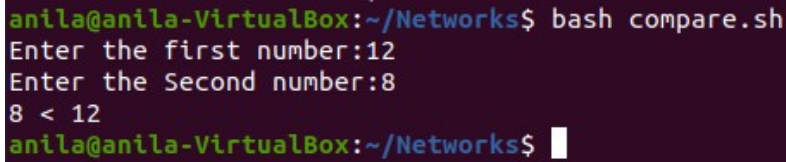
7. Use the sed command to replace the content of the file?

Ans: `sed -i 's/old-text/new-text/g' input.txt`



8. Write a script to compare numbers?

```
#!/bin/bash
# Script to do numeric comparisons
var1=10
var2=20
if [ $var2 -gt $var1 ]
then
    echo "$var2 is greater than $var1"
fi
# Second comparison
If [ $var1 -gt 30]then
    echo "$var is greater than 30"
else
    echo "$var1 is less than 30"
fi
```



```
anila@anila-VirtualBox:~/Networks$ bash compare.sh
Enter the first number:12
Enter the Second number:8
8 < 12
anila@anila-VirtualBox:~/Networks$
```

9. Write a shell script to check to see if the file “file_path” exists.

If it does exist, display “file_path passwords are enabled.” Next, check to see if you can write to the file. If you can, display “You have permissions to edit “file_path.””If you cannot, display “You do NOT have permissions to edit“file_path””?

```
#!/bin/bash
FILE=""
if [ -e "$FILE" ]
then
    echo "$FILE passwords are enabled"
fi

if [ -x "$FILE" ]
then
    echo "You have permission to execute $FILE" else
    echo "You do Not have permissions to execute
$FILE"
fi
```

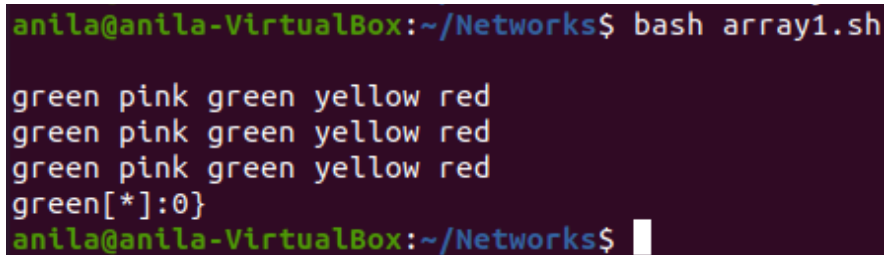
10. How to print all array indexes?

```
echo ${ARRAYNAME[*]}
#!/bin/bash
# To declare static Array

arr=(green pink green yellow red)

# To print all elements of array

echo ${arr[@]}
echo ${arr[*]}
echo ${arr[@]:0}
echo ${arr[*]:0}
```



```
anila@anila-VirtualBox:~/Networks$ bash array1.sh
green pink green yellow red
green pink green yellow red
green pink green yellow red
green[*]:0
anila@anila-VirtualBox:~/Networks$
```

11. Write a shell script to display the last updated file or the newest file in a directory?

```
#!/bin/bash

ls -lrt | grep ^- | awk 'END{print $NF}'
```

12. Write a shell script that adds an extension “.new” to all the files in Directory?

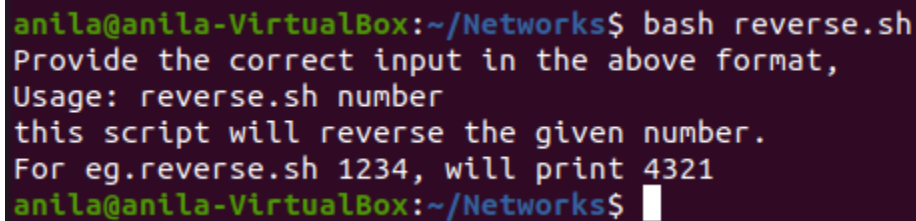
running the test script.

```
#!/bin/bash
dir=$1
for file in `ls $1/*`
do
    mv $file $file.new
done
```

13. Write a shell script to print a number in reverse order. It should support the following requirements.

```
#!/bin/bash
if [ $# -ne 1 ]
then
    echo "Provide the correct input in the above format."
    echo "Usage: $0 number"
    echo "    this script will reverse the given
number." echo " For eg. $0 1234, will print 4321"
    exit 1
fi

n=$1
rev=0
sd=0
while [ $n -gt 0 ]
do
    sd=`expr $n % 10`
    rev=`expr $rev \* 10 + $sd`
    n=`expr $n / 10`
done
echo "Reverse number is $rev"
```



```
anila@anila-VirtualBox:~/Networks$ bash reverse.sh
Provide the correct input in the above format,
Usage: reverse.sh number
this script will reverse the given number.
For eg. reverse.sh 1234, will print 4321
anila@anila-VirtualBox:~/Networks$
```

14. Write a shell script delete a file which has special characters in its file name?

In Linux or Unix-like systems, you may come across file names including the following special characters, white spaces, backslashes and more.

```
-
--
;
&
$
?
*
```

Bash shell considers most of the above special characters as commands. Thus, the “rm” command may not be able to delete such files. The simplest way to delete files having special characters in its name is by using the inode number

15. Write a shell script to find out the unique words in a file and also count the occurrence of each of these words. We can say that the file under consideration contains many lines, and each line has multiple words

```
$ cat animal.txt
tiger bear
elephant tiger bear
bear
```

Following test script/command will count the unique words.

```
$ awk '{for(i=1;i<=NF;i++)a[$i]++;}END{for(i in a){print i, a[i];}}' animal.txt
```

16. Write a script to print the first 10 elements of the Fibonacci series.

```
# Program for Fibonacci
# Series

# Static input fo N
N=6

# First Number of the
# Fibonacci Series
a=0

# Second Number of the
# Fibonacci Series
b=1

echo "The Fibonacci series is : "

for (( i=0; i<N; i++ ))
do
    echo -n "$a "
    fn=$((a + b))
    a=$b
    b=$fn
done
# End of for loop
```

```
anila@anila-VirtualBox:~/Networks$ bash fibonacci.sh
The fibonacci series is:
0
1
1
2
3
5
anila@anila-VirtualBox:~/Networks$
```

17. Write a shell script to get the total count of the word “Linux” in all the “.txt” files and also across files present in subdirectories.

The following is the test script/command which recursively searches for the “.txt” files and returns the total occurrences of a word <Linux>.

```
$ find . -name *.txt -exec grep -c Linux '{}' \; | awk '{x+=$0;}END{print x}'
```

18. Write a shell script to validate password strength. Here are a few assumptions for the password string.

- Length – minimum of 8 characters.
- Contain both the alphabet and number.
- Include both the small and capital case letters.

If the password doesn’t comply with any of the above conditions, then the script should report it as a <Weak Password>

Ans.

Password Validation Script

```
echo "Enter your password"
```

```
read password len="$"
```

```
{#password}"
```

```
if test $len -ge 8 ; then
```

```
    echo "$password" | grep -q [0-9]
```

```
    if test $? -eq 0 ; then
```

```
        echo "$password" | grep -q [A-Z]
```



```
if test $? -eq 0 ; then
    echo "$password" | grep -q [a-z]
    if test $? -eq 0 ; then
        echo "Strong Password"
    else
        echo "Weak Password -> Should include a lower case letter."
    fi
else
    echo "Weak Password -> Should include a capital case letter."
fi
else
    echo "Weak Password -> Should use numbers in your
password." fi
else
    echo "Weak Password -> Password length should have at least 8 characters."
fi
```