OS-Linux-commands-Shell-scripting

Operating systems Lab exercise

Linux commands-Shell scripting

Linux commands-Shell scripting

AIM:

To practice Linux Commands and Shell Scripting

DESIGN STEPS:

Step 1:

Navigate to any Linux environment installed on the system or installed inside a virtual environment like virtual box/vmware or online linux JSLinux (https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192) or docker.

Step 2:

Execute the following commands

Step 3:

Testing the commands for the desired output.

COMMANDS:

Create the following files file1, file2 as follows:

cat > file1

chanchal singhvi c.k. shukla s.n. dasgupta sumit chakrobarty ^d

cat > file2

anil aggarwal barun sengupta c.k. shukla lalit chowdury s.n. dasgupta ^d

Display the content of the files

cat < file1

OUTPUT

chanchal singhvi c.k. shukla s.n. dasgupta sumit chakrobarty

cat < file2

OUTPUT

anil aggarwal barun sengupta c.k. shukla lalit chowdury s.n. dasgupta

Comparing Files

cmp file1 file2

OUTPUT

file1 file2 differ: char 1, line 1

comm file1 file2

OUTPUT

anil aggarwal barun sengupta c.k. shukla chanchal singhvi c.k. shukla lalit chowdury s.n. dasgupta sumit chakrobarty
diff file1 file2

OUTPUT

--- file1 +++ file2 @@ -1,4 +1,5 @@ -chanchal singhvi +anil aggarwal +barun sengupta c.k. shukla +lalit chowdury s.n. dasgupta -sumit chakrobarty
#Filters

Create the following files file11, file22 as follows:

cat > file11
Hello world This is my world ^d
cat > file22
1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer ^d
cut -c1-3 file11

OUTPUT

Hel Thi
cut -d "|" -f 1 file22

OUTPUT

1001 1002 1003
cut -d "|" -f 2 file22

OUTPUT

Ram tom Joe
cat > newfile
Hello world hello world ` cat > newfile Hello world hello world
grep Hello newfile

OUTPUT

Hello world
grep hello newfile

OUTPUT

hello world
grep -v hello newfile

OUTPUT

Hello world
cat newfile | grep -i "hello"

OUTPUT

Hello world hello world
cat newfile | grep -i -c "hello"

OUTPUT

2
grep -R ubuntu /etc

OUTPUT

grep -w -n world newfile

OUTPUT

1:Hello world 2:hello world
cat > newfile
Hello world hello world Linux is world number 1 Unix is predecessor Linux is best in this World
cat > newfile

egrep -w 'Hello|hello' newfile

OUTPUT

Hello world hello world
egrep -w '(H|h)ello' newfile

OUTPUT

Hello world hello world
egrep -w '(H|h)ell[a-z]' newfile

OUTPUT

Hello world hello world
egrep '(^hello)' newfile

OUTPUT

hello world
egrep '(world$)' newfile

OUTPUT

Hello world hello world
egrep '(World$)' newfile

OUTPUT

Linux is best in this World
egrep '((W|w)orld$)' newfile

OUTPUT

Hello world hello world Linux is best in this World
egrep '[1-9]' newfile

OUTPUT

Linux is world number 1
egrep 'Linux.*world' newfile

OUTPUT

Linux is world number 1
egrep 'Linux.*World' newfile

OUTPUT

Linux is best in this World
egrep l{2} newfile

OUTPUT

```
Hello world hello world
egrep 's{1,2}' newfile
```

OUTPUT

```
Linux is world number 1 Unix is predecessor Linux is best in this World
cat > file23
1001 | Ram | 10000 | HR 1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 |
Sam | 5000 | HR 1004 | Sit | 7000 | Dev 1003 | Joe | 7000 | Developer 1001 | Ram | 10000 | HR ^d
sed -n -e '3p' file23
```

OUTPUT

```
1002 | tom | 5000 | Admin
sed -n -e '$p' file23
```

OUTPUT

```
1001 | Ram | 10000 | HR
sed -e 's/Ram/Sita/' file23
```

OUTPUT

```
1001 | Sita | 10000 | HR 1001 | Sita | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 |
Sam | 5000 | HR 1004 | Sit | 7000 | Dev 1003 | Joe | 7000 | Developer 1001 | Sita | 10000 | HR
sed -e '2s/Ram/Sita/' file23
```

OUTPUT

```
1001 | Ram | 10000 | HR 1001 | Sita | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 |
Sam | 5000 | HR 1004 | Sit | 7000 | Dev 1003 | Joe | 7000 | Developer 1001 | Ram | 10000 | HR
sed '/tom/s/5000/6000/' file23
```

OUTPUT

```
1001 | Ram | 10000 | HR 1001 | Ram | 10000 | HR 1002 | tom | 6000 | Admin 1003 | Joe | 7000 | Developer 1005 |
Sam | 5000 | HR 1004 | Sit | 7000 | Dev 1003 | Joe | 7000 | Developer 1001 | Ram | 10000 | HR
sed -n -e '1,5p' file23
```

OUTPUT

```
1001 | Ram | 10000 | HR 1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 |
Sam | 5000 | HR
sed -n -e '2,/Joe/p' file23
```

OUTPUT

```
1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer
sed -n -e '/tom/,/Joe/p' file23
```

OUTPUT

```
1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer
seq 10
```

OUTPUT

```
1 2 3 4 5 6 7 8 9 10
seq 10 | sed -n '4,6p'
```

OUTPUT

4 5 6
seq 10 | sed -n '2,~4p'

OUTPUT

sed: no address after comma
seq 3 | sed '2a hello'

OUTPUT

1 2 hello 3
seq 2 | sed '2i hello'

OUTPUT

1 hello 2
seq 10 | sed '2,9c hello'

OUTPUT

1 hello 10
sed -n '2,4{s/^/$/;p}' file23

OUTPUT

$1001 | Ram | 10000 | HR $1002 | tom | 5000 | Admin $1003 | Joe | 7000 | Developer
sed -n '2,4{s/$/*/;p}' file23
1001 | Ram | 10000 | HR* 1002 | tom | 5000 | Admin* 1003 | Joe | 7000 | Developer*
#Sorting File content cat > file21
1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 | Sam | 5000 | HR 1004 | Sit | 7000 | Dev
sort file21

OUTPUT

1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1004 | Sit | 7000 | Dev 1005 | Sam | 5000 | HR
cat > file22
1001 | Ram | 10000 | HR 1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 | Sam | 5000 | HR 1004 | Sit | 7000 | Dev
uniq file22

OUTPUT

1001 | Ram | 10000 | HR 1002 | tom | 5000 | Admin 1003 | Joe | 7000 | Developer 1005 | Sam | 5000 | HR 1004 | Sit | 7000 | Dev
#Using tr command
cat file23 | tr [:lower:] [:upper:]

OUTPUT

1001 | RAM | 10000 | HR 1001 | RAM | 10000 | HR 1002 | TOM | 5000 | ADMIN 1003 | JOE | 7000 | DEVELOPER 1005 | SAM | 5000 | HR 1004 | SIT | 7000 | DEV 1003 | JOE | 7000 | DEVELOPER 1001 | RAM | 10000 | HR
cat < urllist.txt
www. yahoo. com www. google. com www. mrcet.... com ^d
cat > urllist.txt

```
www. yahoo. com www. google. com www. mrcet.... com
cat urllist.txt | tr -d ' '
```

OUTPUT

```
www.yahoo.com www.google.com www.mrcet....com
cat urllist.txt | tr -d ' ' | tr -s '.'
```

OUTPUT

```
www.yahoo.com www.google.com www.mrcet.com
#Backup commands tar -cvf backup.tar *
```

OUTPUT

```
bench.py file21 file22 file23 hello.c hello.js newfile readme.txt urllist.txt
mkdir backupdir
mv backup.tar backupdir
tar -tvf backup.tar
```

OUTPUT

```
-rw-r--r-- root/root 114 2020-07-05 23:17:07 bench.py -rw-r--r-- root/root 131 2024-02-24 09:25:54 file21 -rw-r-
-r-- root/root 155 2024-02-24 09:29:21 file22 -rw-r--r-- root/root 210 2024-02-24 09:14:03 file23 -rw-r--r--
root/root 76 2020-07-03 14:45:56 hello.c -rw-r--r-- root/root 22 2020-06-26 14:57:33 hello.js -rw-r--r-- root/root
96 2024-02-24 09:12:14 newfile -rw-r--r-- root/root 151 2020-07-05 23:19:13 readme.txt -rw-r--r-- root/root 52
2024-02-24 09:33:40 urllist.txt
tar -xvf backup.tar
```

OUTPUT

```
bench.py file21 file22 file23 hello.c hello.js newfile readme.txt urllist.txt
gzip backup.tar
ls .gz
```

OUTPUT

```
gunzip backup.tar.gz
```

OUTPUT

Shell Script

```
echo '#!/bin/sh' > my-script.sh echo 'echo Hello World'; exit 0 >> my-script.sh
chmod 755 my-script.sh ./my-script.sh
```

OUTPUT

```
cat << stop > herecheck.txt
hello in this world i cant stop for this non stop movement stop
cat herecheck.txt
```

OUTPUT

cat < scriptest.sh bash #!/bin/sh echo “File name is $0 ” echo "File name is " `basename $0` echo “First arg. is ” $1 echo “Second arg. is ” $2 echo “Third arg. is ” $3 echo “Fourth arg. is ” $4echo'The$@ is ' $@echo'The$# is '

You can't use 'macro parameter character #' in math mode

```
$1#
echo 'The $
```
$ is ' $$ ps ^d

cat scriptest.sh bash #!/bin/sh echo “File name is $0 ” echo "File name is " `basename $0` echo “First arg. is ” $1 echo “Second arg. is ” $2 echo “Third arg. is ” $3 echo “Fourth arg. is ” $4echo'The$@ is ' $@echo'The$# is '

You can't use 'macro parameter character #' in math mode

```
$#
echo 'The $
```
$ is ' $$ ps

chmod 777 scriptest.sh
./scriptest.sh 1 2 3

OUTPUT

File name is ./scriptest.sh File name is scriptest.sh First arg. is 1 Second arg. is 2 Third arg. is 3 Fourth arg. is The $@ is 123The$# is

You can't use 'macro parameter character #' in math mode

```
$#
The $
```
$ is 124

ls file1

OUTPUT

file1
echo $?

OUTPUT

./one bash: ./one: Permission denied
echo $?

OUTPUT

0
abcd
echo $?

OUTPUT

1

mis-using string comparisons

cat < strcomp.sh bash #!/bin/bash val1=baseball val2=hockey if [$val1 > $val2] then echo "$val1 is greater than $val2" else echo "$val1 is less than $val2" fi ^d
cat strcomp.sh bash #!/bin/bash val1=baseball val2=hockey if [$val1 > $val2] then echo "$val1 is greater than $val2" else echo "$val1 is less than $val2" fi
##OUTPUT
val1=baseball val2=hockey if [$val1 > $val2] then echo "$val1 is greater than $val2" else echo "$val1 is less than $val2" fi

```
chmod 755 strcomp.sh
./strcomp.sh
```

OUTPUT

```
baseball is less than hockey
```

check file ownership

```
cat < psswdperm.sh bash #!/bin/bash if [ -O /etc/passwd ] then echo “You are the owner of the /etc/passwd file” else echo “Sorry, you are not the owner of the /etc/passwd file” fi ^d
cat psswdperm.sh bash /#!/bin/bash if [ -O /etc/passwd ] then echo “You are the owner of the /etc/passwd file” else echo “Sorry, you are not the owner of the /etc/passwd file” fi
./psswdperm.sh
```

OUTPUT

```
You are the owner of the /etc/passwd file
```

check if with file location

```
cat>ifnested.sh bash #!/bin/bash if [ -e $HOME ] then echo “$HOME The object exists, is it a file?” if [ -f $HOME ] then echo “Yes,$HOME it is a file!” else echo “No,$HOME it is not a file!” if [ -f $HOME/.bash_history ] then echo “But $HOME/.bash_history is a file!” fi fi else echo “Sorry, the object does not exist” fi ^d
cat ifnested.sh
#!/bin/bash if [ -e $HOME ] then echo “$HOME The object exists, is it a file?” if [ -f $HOME ] then echo “Yes,$HOME it is a file!” else echo “No,$HOME it is not a file!” if [ -f $HOME/.bash_history ] then echo “But $HOME/.bash_history is a file!” fi fi else echo “Sorry, the object does not exist” fi
./ifnested.sh
```

OUTPUT

```
/root The object exists, is it a file? No,/root it is not a file!
```

using numeric test comparisons

```
cat > iftest.sh bash #!/bin/bash val1=10 val2=11 if [ $val1 -gt 5 ] then echo “The test value $val1 is greater than 5” fi if [ $val1 -eq $val2 ] then echo “The values are equal” else echo “The values are different” fi ^d
cat iftest.sh bash #!/bin/bash val1=10 val2=11 if [ $val1 -gt 5 ] then echo “The test value $val1 is greater than 5” fi if [ $val1 -eq $val2 ] then echo “The values are equal” else echo “The values are different” fi
$ chmod 755 iftest.sh
$ ./iftest.sh ##OUTPUT
“The test value 10 is greater than 5” “The values are different”
```

check if a file

```
cat > ifnested.sh bash #!/bin/bash if [ -e $HOME ] then echo “$HOME The object exists, is it a file?” if [ -f $HOME ] then echo “Yes,$HOME it is a file!” else echo “No,$HOME it is not a file!” if [ -f $HOME/.bash_history ] then echo “But $HOME/.bash_history is a file!” fi fi else echo “Sorry, the object does not exist” fi ^d
cat ifnested.sh bash #!/bin/bash if [ -e $HOME ] then echo “$HOME The object exists, is it a file?” if [ -f $HOME ] then echo “Yes,$HOME it is a file!” else echo “No,$HOME it is not a file!” if [ -f $HOME/.bash_history ] then echo “But $HOME/.bash_history is a file!” fi fi else echo “Sorry, the object does not exist” fi
$ chmod 755 ifnested.sh
$ ./ifnested.sh ##OUTPUT
```

“/root The object exists, is it a file?” “No,/root it is not a file!”

looking for a possible value using elif

```
cat elifcheck.sh bash #!/bin/bash if [ $USER = Ram ] then echo "Welcome $USER" echo "Please enjoy your visit"
elif [ $USER = Rahim ] then echo "Welcome $USER" echo "Please enjoy your visit" elif [ $USER = Robert ] then
echo "Special testing account" elif [ $USER = gganesh ] then echo "$USER, Do not forget to logout when you're
done" else echo "Sorry, you are not allowed here" fi
$ chmod 755 elifcheck.sh
$ ./elifcheck.sh
```

OUTPUT

“/root The object exists, is it a file?” “No,/root it is not a file!”

testing compound comparisons

```
cat> ifcompound.sh bash #!/bin/bash if [ -d $HOME ] && [ -w $HOME ] then echo "The file exists and you can
write to it" else echo "I cannot write to the file" fi
$ chmod 755 ifcompound.sh $ ./ifcompound.sh
```

OUTPUT

Welcome Ram Please enjoy your visit Welcome Rahim Please enjoy your visit Special testing account gganesh, Do not forget to logout when you're done Sorry, you are not allowed here

using the case command

```
cat >casecheck.sh bash case $USER in Ram | Robert) echo "Welcome, $USER" echo "Please enjoy your visit";;
Rahim) echo "Special testing account";; gganesh) echo "$USER, Do not forget to log off when you're done";; *)
echo "Sorry, you are not allowed here";; esac
$ chmod 755 casecheck.sh
$ ./casecheck.sh
cat > whiletest bash #!/bin/bash #while command test var1=10 while [ $var1 -gt 0 ] do echo var1var1 = [ $var1
- 1 ] done
$ chmod 755 whiletest.sh
$ ./whiletest.sh
cat untiltest.sh bash #using the until command var1=100 until [ $var1 -eq 0 ] do echo var1var1 = [ $var1 - 25 ]
done
$ chmod 755 untiltest.sh
cat forin1.sh bash #!/bin/bash #basic for command for test in Alabama Alaska Arizona Arkansas California
Colorado do echo The next state is $test done
$ chmod 755 forin1.sh
cat forin2.sh bash #!/bin/bash # another example of how not to use the for command for test in I don't know if
this'll work do echo “word:$test” done
$ chmod 755 forin2.sh
cat forin2.sh bash #!/bin/bash # another example of how not to use the for command for test in I don't know if
this'll work do echo “word:$test” done
$ chmod 755 forin2.sh
$ ./forin2.sh
cat forin3.sh bash #!/bin/bash # another example of how not to use the for command for test in I don't know if
"this'll" work do echo "word:$test" done
```

$ ./forin3.sh
cat forin1.sh bash #!/bin/bash

basic for command

for test in Alabama Alaska Arizona Arkansas California Colorado do echo The next state is $test done
$ chmod 755 forin1.sh

OUTPUT

word:I word:dont know if thisll word:work
cat forinfile.sh bash #!/bin/bash

reading values from a file

file="cities" for state in `cat $file` do echo "Visit beautiful $file” done
$ chmod 777 forinfile.sh $ cat cities Hyderabad Alampur Basara Warangal Adilabad Bhadrachalam Khammam

OUTPUT

Visit beautiful Hyderabad Visit beautiful Alampur Visit beautiful Basara Visit beautiful Warangal Visit beautiful Adilabad Visit beautiful Bhadrachalam Visit beautiful Khammam
cat forctype.sh bash #!/bin/bash

testing the C-style for loop

for ((i=1; i <= 5; i++)) do echo "The value of i is $i"done‘$ chmod 755 forctype.sh $ ./forctype.sh

OUTPUT

The value of i is 1 The value of i is 2 The value of i is 3 The value of i is 4 The value of i is 5
cat forctype1.sh bash #!/bin/bash

multiple variables

for ((a=1, b=5; a <= 5; a++, b--)) do echo "$a - $b" done
$ chmod 755 forctype.sh $ ./forctype1.sh

OUTPUT

cat fornested1.sh bash #!/bin/bash

nesting for loops

for ((a = 1; a <= 3; a++)) do echo "Starting loop $a:" for ((b = 1; b <= 3; b++)) do echo " Inside loop: $b" done done
$ chmod 755 fornested1.sh
$ ./fornested1.sh

OUTPUT

1 - 5 2 - 4 3 - 3 4 - 2 5 - 1
cat forbreak.sh bash #!/bin/bash

breaking out of a for loop

for var1 in 1 2 3 4 5 do if [$var1 -eq 3] then break fi echo "Iteration number: $var1" done echo "The for loop is completed“

OUTPUT

Iteration number: 1 Iteration number: 2 The for loop is completed
$ chmod 755 forbreak.sh
$ ./forbreak.sh
cat forbreak.sh bash #!/bin/bash

breaking out of a for loop

for var1 in 1 2 3 4 5 do if [$var1 -eq 3] then continue fi echo "Iteration number: $var1" done echo "The for loop is completed"
$ chmod 755 forcontinue.sh
$ ./forcontinue.sh

OUTPUT

Iteration number: 1 Iteration number: 2 Iteration number: 4 Iteration number: 5 The for loop is completed
cat exread.sh bash #!/bin/bash

testing the read command

echo -n "Enter your name: " read name echo "Hello $name, welcome to my program. "
$ chmod 755 exread.sh
$ ./exread.sh

OUTPUT

Enter your name: John Hello John, welcome to my program.
cat exread1.sh bash #!/bin/bash

testing the read command

read -p "Enter your name: " name echo "Hello $name, welcome to my program. "
$ chmod 755 exread1.sh

OUTPUT

Enter your name: sanju Hello sanju, welcome to my program.
$ ./exread1.sh
cat funcex.sh bash #!/bin/bash

trying to access script parameters inside a function

function func { echo $[$1 *

Extra close brace or missing open brace

```
$2 ]
}
if [ $
```

-eq 2] then value= `func $1 $2` echo "The result is $value" else echo "Usage: badtest1 a b" fi

OUTPUT

./funcex.sh ./funcex.sh 1 2 $ bash script.sh 1 2 The result is 2
cat argshift.sh

~~~
```bash
#!/bin/bash
 while (( "$#" )); do
  echo $1
  shift
done

 ./funcex.sh


 ./funcex.sh 1 2


cat argshift.sh
bash
#!/bin/bash
 while (( "$#" )); do
  echo $1
  shift
done

$ chmod 777 argshift.sh

## OUTPUT
$ ./argshift.sh 1 2 3

 cat argshift1.sh
bash
 #/bin/bash
 # store arguments in a special array
args=("$@")
# get number of elements
ELEMENTS=${#args[@]}
 # echo each element in array
# for loop
for (( i=0;i<$ELEMENTS;i++)); do
    echo ${args[${i}]}
done

$ chmod 777 argshift.sh
## OUTPUT

$ ./argshift.sh 1 2 3
1
2
3

$ ./argshift.sh 1 2 3

cat argshift.sh
bash
~~~

```
#!/bin/bash
set -x
while (( "$#" )); do
  echo $1
  shift
done
set +x
```

OUTPUT

```
 ./argshift.sh 1 2 3
 + (( 0 ))
 + set +x
```

```
cat > nc.awk
bash
BEGIN{}
{
print len=length($0),"\t",$0
wordcount+=NF
chrcnt+=len
}
END {
print "total characters",chrcnt
print "Number of Lines are",NR
print "No of Words count:",wordcount
}
```

```
cat>data.dat
bash
bcdfghj
abcdfghj
bcdfghj
ebcdfghj
bcdfghj
ibcdfghj
bcdfghj
obcdfghj
bcdfghj
ubcdfghj
```

```
awk -f nc.awk data.dat
```

OUTPUT

```
cat > palindrome.sh
bash
#num=545
echo "Enter the number"
read num
```

```
s=0
rev=""
temp=$num
while [ $num -gt 0 ]
do
        # Get Remainder
        s=$(( $num % 10 ))
        # Get next digit
        num=$(( $num / 10 ))
        # Store previous number and
        # current digit in reverse
        rev=$( echo ${rev}${s} )
done
if [ $temp -eq $rev ];
then
        echo "Number is palindrome"
else
        echo "Number is NOT palindrome"
fi

## OUTPUT
Enter the number
121
Number is palindrome
Enter the number
69
Number is NOT palindrome


# RESULT:
The Commands are executed successfully.
```