

Shell Scripting

Set 2

Rwithik Manoj

March 27, 2019

1. Write a shell script that displays a special listing showing the permissions, size filename and last modification time of filename supplied as arguments. Provide suitable headers using the printf command.

Algorithm:

- (a) Start.
- (b) Check if the file exists.
- (c) If it does, print its details with `ls -l`.
- (d) Stop.

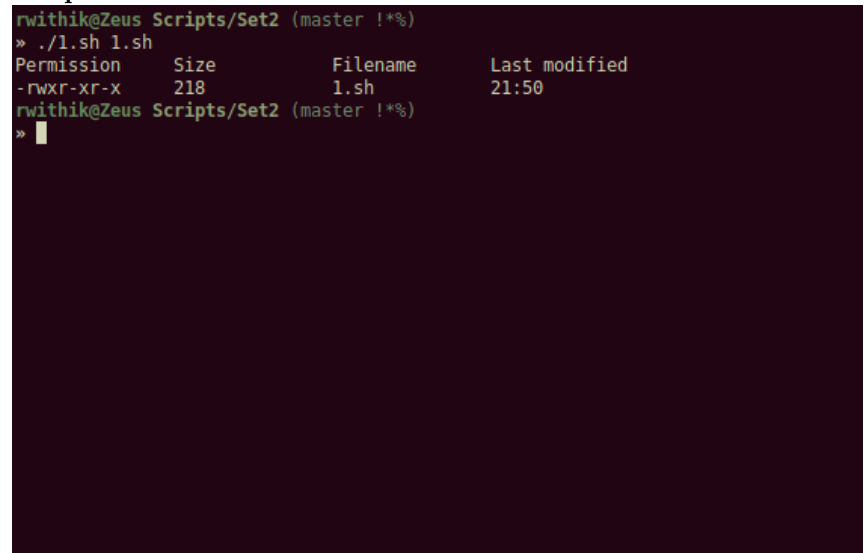
Script:

```
#!/bin/bash

if [[ ! -f $1 ]]
then
    printf "File does not exist"
    exit
fi

printf "%-15s" "Permission" "Size" "Filename" "Last modified"
printf "\n"
ls -l $1 | awk '{printf "%-15s%-15s%-15s%-15s\n", $1, $5, $9, $8}'
```

Output:



```
rwthik@Zeus Scripts/Set2 (master !*)
» ./1.sh 1.sh
Permission      Size      Filename      Last modified
-rwxr-xr-x      218       1.sh          21:50
rwthik@Zeus Scripts/Set2 (master !*)
» █
```

The screenshot shows a terminal window with a dark background. The prompt is 'rwthik@Zeus Scripts/Set2 (master !*)'. The user enters '» ./1.sh 1.sh'. The output is a table with four columns: 'Permission', 'Size', 'Filename', and 'Last modified'. The first row of data shows '-rwxr-xr-x', '218', '1.sh', and '21:50'. The prompt returns to 'rwthik@Zeus Scripts/Set2 (master !*)' and the user enters '»' followed by a cursor.

2. Write a script that compares two directories dir1 and dir2(supplied as arguments) and copies to dir1 from dir2 every file that is not present in dir1.

Algorithm:

- (a) Start.
- (b) Check if both the directories exist.
- (c) If it does, copy the contents using **cp -n** flag.
- (d) Stop.

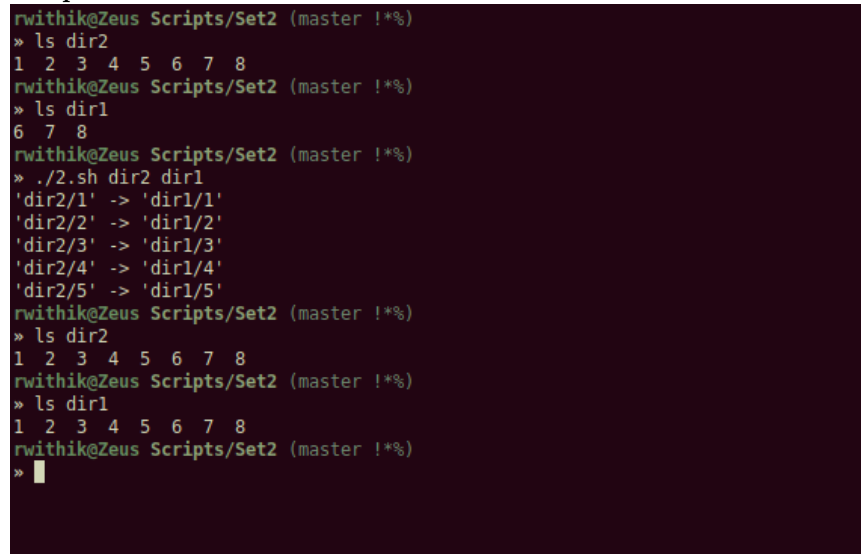
Script:

```
#!/bin/bash

if [[ ! -d $1 ]] || [[ ! -d $2 ]]
then
    printf "Invalid Arguments\n"
    exit
fi

cp -n -r -v ${1}/* ${2}
```

Output:



```
rwthik@Zeus Scripts/Set2 (master !*)
» ls dir2
1 2 3 4 5 6 7 8
rwthik@Zeus Scripts/Set2 (master !*)
» ls dir1
6 7 8
rwthik@Zeus Scripts/Set2 (master !*)
» ./2.sh dir2 dir1
'dir2/1' -> 'dir1/1'
'dir2/2' -> 'dir1/2'
'dir2/3' -> 'dir1/3'
'dir2/4' -> 'dir1/4'
'dir2/5' -> 'dir1/5'
rwthik@Zeus Scripts/Set2 (master !*)
» ls dir2
1 2 3 4 5 6 7 8
rwthik@Zeus Scripts/Set2 (master !*)
» ls dir1
1 2 3 4 5 6 7 8
rwthik@Zeus Scripts/Set2 (master !*)
» █
```

3. Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file name followed by its permissions.

Algorithm:

- (a) Start.
- (b) Exit the script if the files don't exist.
- (c) Get the permissions of the files with **awk** and **ls -l**.
- (d) If the permissions are same, print that they are equal, else print the permissions.
- (e) Stop.

Script:

```
#!/bin/bash

if [[ ! -f $1 ]] || [[ ! -f $2 ]]
then
    printf "Invalid Arguments\n"
    exit
fi

PERM1=`ls -l $1 | awk '{print $1}'`
PERM2=`ls -l $2 | awk '{print $1}'`

if [[ $PERM1 == $PERM2 ]]
then
    printf "Equal\n"
else
    ls -l $1 | awk '{print $1 " " " $9}'
    ls -l $2 | awk '{print $1 " " " $9}'
fi
```

Output:

```
rwthik@Zeus Scripts/Set2 (master !*%)
» ./3.sh 1.sh 2.sh
-rwxr-xr-x 1.sh
-rwxrwxrwx 2.sh
rwthik@Zeus Scripts/Set2 (master !*%)
» ./3.sh 1.sh 1.sh
Equal
rwthik@Zeus Scripts/Set2 (master !*%)
» █
```

4. Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

Algorithm:

- (a) Start.
- (b) Exit the script if the files don't exist.
- (c) Check if the files are the same with **diff**.
- (d) If the contents are the same, delete the second file.
- (e) Stop.

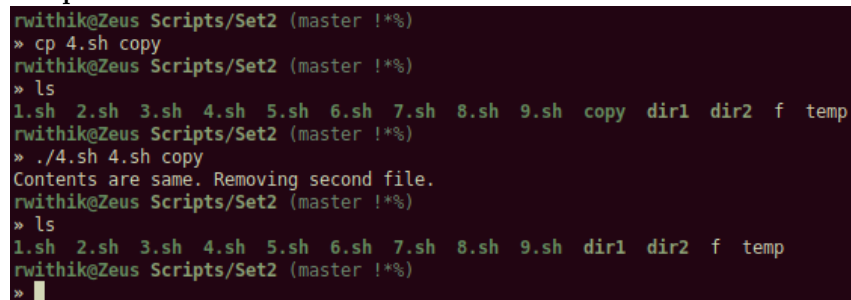
Script:

```
#!/bin/bash

if [[ ! -f $1 ]] || [[ ! -f $2 ]]
then
    printf "Invalid Arguments\n"
    exit
fi

if [[ `diff $1 $2 | wc -l` == 0 ]]
then
    printf "Contents are same. Removing second file.\n"
    rm $2
fi
```

Output:



```
rwthik@Zeus Scripts/Set2 (master !*)
» cp 4.sh copy
rwthik@Zeus Scripts/Set2 (master !*)
» ls
1.sh 2.sh 3.sh 4.sh 5.sh 6.sh 7.sh 8.sh 9.sh copy dir1 dir2 f temp
rwthik@Zeus Scripts/Set2 (master !*)
» ./4.sh 4.sh copy
Contents are same. Removing second file.
rwthik@Zeus Scripts/Set2 (master !*)
» ls
1.sh 2.sh 3.sh 4.sh 5.sh 6.sh 7.sh 8.sh 9.sh dir1 dir2 f temp
rwthik@Zeus Scripts/Set2 (master !*)
» █
```

5. Write a shell script that, given a file name as the argument will count vowels, blank spaces, characters, number of line and symbols.

Algorithm:

- (a) Start.
- (b) Exit the script if the file does not exist.
- (c) Count the number of vowels using `cat`, `grep` and `wc`.
- (d) Count the number of characters using `wc -c`.
- (e) Count the number of blank lines by searching for the pattern “`^$`”.
^ matches the beginning of a line, and \$ matches the end of a line.
So the pattern matches the lines in which the beginning and end are adjacent, ie blank lines.
- (f) Count the number of lines using `wc -l`.
- (g) Stop.

Script:

```
#!/bin/bash

if [[ ! -f $1 ]]
then
    printf "File does not exist\n"
    exit
fi

echo "Vowels : `cat $1 | tr '[a-z]' '[A-Z]' | grep -o "A\\|E\\|I\\|O\\|U" | wc -l`"
echo "Characters : $(cat $1 | wc -c)"
echo "Blank lines : $(grep -c '^$' $1)"
echo "Lines : $(cat $1 | wc -l )"
```

Output:

```
rwithik@Zeus Scripts/Set2 (master !*)
» ./5.sh ../../Testfiles/Story.txt
Vowels : 167
Characters : 446
Blank lines : 0
Lines : 1
rwithik@Zeus Scripts/Set2 (master !*)
» █
```


6. Write a shell script that will take an input file and remove identical lines.

Algorithm:

- (a) Start.
- (b) Exit the script if the file does not exist.
- (c) Use `awk '!seen[$0]++'` .
- (d) This makes a dictionary, named seen. When a new line is encountered, it sets the value of the key as one. It prints the line only if the value of seen[\$0] is zero. So the next time the same line is encountered, it is not printed.
- (e) Stop.

Script:

```
#!/bin/bash

if [[ ! -f $1 ]]
then
    printf "File does not exist"
fi

awk '!seen[$0]++' $1
```

7. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.

Algorithm:

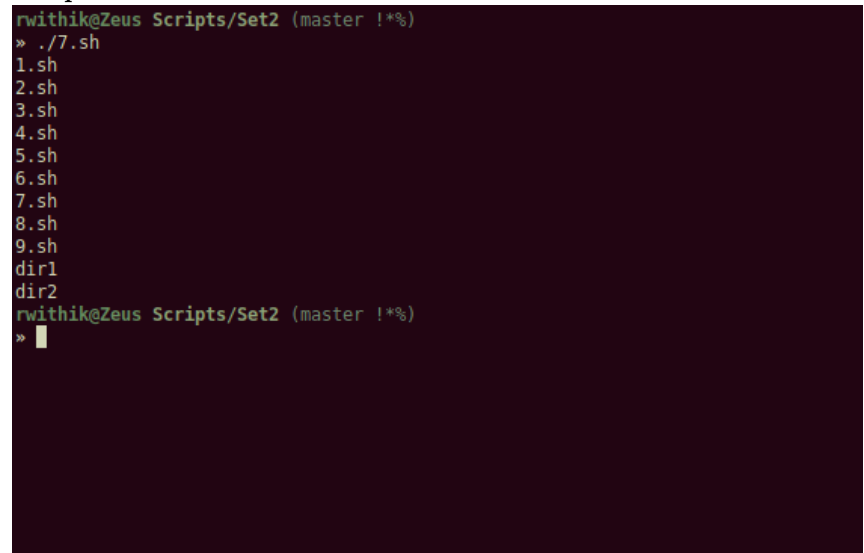
- (a) Start.
- (b) Write the contents of the current directory to a file, f.
- (c) Redirect the input stream to the file.
- (d) Check the permissions of the file with the `-r`, `-w` and `-x` flags of `test`.
- (e) Stop.

Script:

```
#!/bin/bash

ls -l > f
exec < f
while read file
do
    if [[ -r $file && -w $file && -x $file ]]
    then
        printf "$file\n"
    fi
done
```

Output:



```
rwithik@Zeus Scripts/Set2 (master !*%)
» ./7.sh
1.sh
2.sh
3.sh
4.sh
5.sh
6.sh
7.sh
8.sh
9.sh
dir1
dir2
rwithik@Zeus Scripts/Set2 (master !*%)
» █
```

8. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th. A \ is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be through a text file created by the user.

Algorithm:

- (a) Start.
- (b) Exit the script if the file doesn't exist.
- (c) Store the number of characters in a variable, n.
- (d) Loop through the file and cut 40 characters in each iteration.
- (e) Stop.

Script:

```
#!/bin/bash

if [[ ! -f $1 ]]
then
    printf "File does not exist"
    exit
fi

n=`wc -l $1 | cut -d " " -f 1`
i=1

while [ $i -le $n ]
do
    LINE=`sed -n "$i p" $1`
    CHARCOUNT=`echo $LINE | wc -c | cut -d " " -f 1`
    while [ $CHARCOUNT -ge 40 ]
    do
        EXT=`echo $LINE | cut -c 41-`
        LINE=`echo $LINE | cut -c 1-40`
        echo "$LINE \\"
        LINE=$EXT
        CHARCOUNT=`echo $EXT | wc -c | cut -d " " -f 1`
    done
    echo "$LINE"
    i=`expr $i + 1`
done
```

Output:

```
rwthik@Zeus Scripts/Set2 (master !*)
» ./8.sh ../../Testfiles/Story.txt
Lorem ipsum dolor sit amet, consectetur \
adipiscing elit, sed do eiusmod tempor i \
ncididunt ut labore et dolore magna aliq \
ua. Ut enim ad minim veniam, quis nostru \
d exercitation ullamco laboris nisi ut a \
liquip ex ea commodo consequat. Duis aut \
e irure dolor in reprehenderit in volupt \
ate velit esse cillum dolore eu fugiat n \
ulla pariatur. Excepteur sint occaecat c \
upidatat non proident, sunt in culpa qui \
officia deserunt mollit anim id est labo \
rum.
rwthik@Zeus Scripts/Set2 (master !*)
» █
```

9. Write a shell script to delete all lines containing a specific word in one or more file supplied as argument to it.

Algorithm:

- (a) Start.
- (b) Read the word.
- (c) Loop through the files.
- (d) Check if the file exists.
- (e) Delete the lines containing the word, with **sed**.
- (f) Stop.


Script:

```
#!/bin/bash

printf "Enter the word: "
read WORD

for file in $@
do
    if [[ -f $file ]]
    then
        printf "File: $file\n"
        cat $file | sed "/$WORD/d"
        printf "\n\n\n"
    else
        printf "File does not exist"
    fi
done
```

Output:



```
rwithik@Zeus Scripts/Set2 (master !*)
» ./9.sh temp
Enter the word: Lorem
File: temp
adipiscing elit, sed do eiusmod tempor i \
ncididunt ut labore et dolore magna aliq \
ua. Ut enim ad minim veniam, quis nostru \
d exercitation ullamco laboris nisi ut a \
liquip ex ea commodo consequat. Duis aut \
e irure dolor in reprehenderit in volupt \
ate velit esse cillum dolore eu fugiat n \
ulla pariatur. Excepteur sint occaecat c \
upidatat non proident, sunt in culpa qui \
officia deserunt mollit anim id est labo \
rum.

rwithik@Zeus Scripts/Set2 (master !*)
» █
```