Lab 6
File redirection.
Alias and sort commands.
For loop in Bash

Read carefully the instructions before performing!!!
Submission in pairs : 23/12/24 for the groups of Monday
26/12/24 for the groups of Thursday

File redirection

Redirection is a feature that permits you to change the standard input/output/error devices. The standard input (STDIN. FileDescriptor 0) device is the keyboard. The standard output (STDOUT. FilesDescriptor 1) device is the screen. The standard error device is the screen (STDERR. FileDescriptor 2).

> output redirection

>> appends output to an existing file

< input redirection

2> error redirection

Examples:

1. Output redirection

ls **>** LsFile

ls -a **>>** LsFile

2. Input redirection

wc**<**CountFile

3. Error redirection

./Myfile **2>** errorsMyfile

Linux Commands

1. **Alias =** is a user-defined shortcut for a command.

- **alias** shows all the shortcuts defined in the shell

Prof. Tamar Zemach. All rights reserved©

- **alias new_name=”old_name”** defines new shortcuts in the shell

- **alias new_name=”first_command && second_command”** defined new shortcut using two or more commands.

Note: Also possible **alias new_name=”first_command;second_command”**

- **unalias name** cancels the shortcut name

2. **Sort** = sorts lines of file in an ascending/descending order.

- **sort filename**

Options:

-r: sort the input in reverse order.

-n: sort the input numerically.

-k: sort the input based on a specific field or column.

-b: ignore the leading blanks.

-t: specify the field separator.

-u: remove duplicate lines from the output.

-o: specify the output file.

Notes:

1). sort is non case sensitive (according to the present definition in locale. You can check the definitions using the Linux command locale)

2). the lines are compared as strings. Thus 22 comes before 123.

3). sort does not change the existing file. It only shows the result of the sorting.

It is possible to save the sorting result using redirection.

3. Arrays

Defined by **ArrName=(val1 val2 ... valN)**

Example: NumArray=(1 4 6 8)

WordArray=(boo doo moo)

StrArr=(“I am here” “Hello” “Linux is fun”)

- To read/write the elements of the array: arr[**0**], arr[1],...arr[N-1]

- echo ${arr[0]} – print the value of arr[0] (**Put attention to the brackets { }**)

- arr[4]=”foo ”

Prof. Tamar Zemach. All rights reserved©

- echo ${arr[@]} – print all the values of arr

- echo ${#arr[@]}- print number of elements in array

Bash: For-loop

Syntax:

```
for VAR in val1 val2..valN
do
      command1 on $VAR
      command2
      commandN
done
```

```
for VAR in file1 file2 file3
do
  command1 on $VAR
  command2
  commandN
done
```

```
for VAR in $(command)
do
   command1
   command2
   commandN
done
```

```
for (( Initial_EXP; condition; step ))
do
          command1
          command2
          command3
done
```

Prof. Tamar Zemach. All rights reserved©

Example 1. File ForExample.sh

```
#!/bin/bash

##############################################
#WORKING WITH SET OF VALUES
for i in 9 22  4 45
do
	echo "the value of i is equal to $i"
done

echo -e "\n"
##############################################
WORKING WITH ARRAY

# Declare and array
ColorList=( Blue Green Pink White Red )
# Define loop to  iterate the array values
for color in ${ColorList[@]}
do
   # Check the value is pink or not
   if [ $color == 'Pink' ]
   then
        echo "My favorite color is $color"
   fi
done
##############################################
#WORKING WITH FILES

for FILE in foo1 foo2 foo3
do
	touch $FILE
done

##############################################
read -p "Enter number" NUM
for FILE in  $(ls)
do
	if ! [ -d ${FILE}dir${NUM} ]
	then
		mkdir ${FILE}dir${NUM}

	fi
done

##############################################
#C-TYPE loop
```

Prof. Tamar Zemach. All rights reserved©

```
sum=0
for (( i=1; i<=3; i++ ))
do
   echo "The value of i is  $i"
       sum=$(($sum+$i))
done
echo "The value of sum is $sum"
```

Procedures

1.Create directory Lab6.

2. Create a shortcut named myLs which will work exactly like the command ls.

3. Create a shortcut named LsL, which will execute ls -l.

4. Create a shortcut which will first go to the home directory, then go to directory Lab6 and then will show the contain of Lab6.

5. Using redirection, put the aliases which were created by you in 2-4, into file **MyAlias.sh** in Lab6 (see example in Note 1). Run the file and check that it works properly (see Note 2).

Notes:

1). The file MyAlias.sh should contain the definitions of the alias and their runs. Example of the contain of such a file:

MyAlias.sh

```
#!/bin/bash

alias myCD=”cd Lab5”

myCD

alias WCL=”wc -l”

WCL
```

2). To run MyAlias.sh, instead of the usual method ./MyAlias.sh, use

source MyAlias.sh.

Why? Because when we use the normal ./ method, the subshell is opened and the definitions of the new commands obtained from the alias are not saved in the global shell. So, in this case you won't see the alias results. However, when a source command is used, we continue to work in the global shell. So all the new settings are saved there.

Prof. Tamar Zemach. All rights reserved©

6. In Lab6 create file named WritersFile which contains the First names, Second names, years of birth, countries of birth of 5 famous writers. The fields will be separated by *

For example, one line of such a file can be:

Lewis*Carroll*1832*England

7. Sort the WritersFile

(a) by the first name and put the result into the file **SortedByFirstName**.

(b) by the country of birth and put the result into the file **SortedByCountry**.

(c) by the year of birth in increasing order and put the result into the file **SortedByYearInc**.

(d) by the year of birth in decreasing order and put the result into the file **SortedByYearDec**.

8. Define array named TestArray and put in its first field the first and the second name of one of the authors you like. In the second field its year of birth. And in the third field the country of his birth.

Print the contain of TestArray and its length. (Use @ and #)

Then redirect all the commands of TestArray, used above, into a file **TestArrayFile.sh** in Lab6. Run this file and check.

9. In Lab6 write bash script named **MathArray.sh** which is doing the following:

- Asks for the positive number and puts it into NUM

- Asks for NUM numbers and defines array which contains these numbers.

- Go through the array and sum the values. Then print the sum of the array.

- Go through the array, find and print the maximum of array.

10. In Lab6 write bash script named **Files.sh** which is looking for all the files which contain the letter **k** and then change their name so they finish with the running number, strating with 1. The resulting list of the files will be written in the increasing order of the symbol which comes after the letter **k** in file FileList.

Thus, if in Lab6 there are files

book

look

click

cake

Prof. Tamar Zemach. All rights reserved©

make

crocks

foo

then after running the Files.sh script, the names of the files will be:

book1

look2

click3

cake4

crocks5

make6

And the file FileList will be:

book1

look2

click3

cake4

make6

crocks5

Submission:

Create one compressed file named Linux_Lab6_id1_id2.tar,
where id1 and id2 must be changed to the id numbers of the two partners.

This file will include the files **MyAlias.sh, WritersFile, SortedByFirstName, SortedByCountry, SortedByYearInc, SortedByYearDec, TestArrayFile.sh, MathArray.sh, Files.sh.**

To create such a file in Lunix, use tar command.

Submit only the file Linux_Lab6_id1_id2.tar in Moodle and only by one of the partners.

Appeals:

Can be submitted by e-mail to Elad mail@eladhuttner.net within a week from the date of publication of the grades.

Prof. Tamar Zemach. All rights reserved©