Lab 5
History and wc commands.
Types of quatations.
If-then-else in Bash.

Read carefully the instructions before performing!!!
Submission in pairs : 16/12/24 for the groups of Monday
19/12/24 for the groups of Thursday

Linux Commands

1. History

- The **history** command displays the shell history.
- HISTSIZE is a shell variable which contains number of commands defined in history.
- !N = Repeat command number N.
- !! = Repeat last command line.
- !string – Repeat command starting with the string.
- **export HISTSIZE=200** – change number of commands in history to be 200

2. wc - counts the lines, words, and bytes in a file

Flags: **-l** (counts the lines); **-w** (counts the words); **-c** (counts the bytes)

Bash

1. Types of quotation marks

“ ” = substitute the variable inside
‘ ’ = prints the text As Is
` ` = command substitution = $(command)

Example 1. File QuatTypes.sh

```
#!/bin/bash
VAR1='boo ls'
VAR2="foo $VAR1"
VAR3='foo $VAR1'

echo VAR3 = $VAR3
echo VAR2=$VAR2
echo "$VAR2"
echo "${VAR2}22"
echo '$VAR2'

echo `cat QuatTypes.sh`
```

2. IF-THEN-ELSE

if [condition-is-true] = Put attention to the spaces **(after if, after[, before])**

```
    then
        command 1
        command 2
        ......
        command N

    elif [ condition-is-true ]
        then
            command 1
            command 2
            ......
            command N
    else
            command 1
            command 2
            ......
            command N
fi
```

String operators:
-z STRING (True, if STRING is empty)
-n STRING (True, if STRING is not empty)
STR1 = STR2 (True, if the strings are equal. **Put attention to the spaces before and after =**)
STR1 != STR2 (True, if the strings are not equal)

Numerical operators:
num1 -eq num2 (True, if num1= num2)
num1 -ne num2 (True if num1 !=num2)
num1 -lt num2 (True if num1 < num2)
num1 -le num2 (True if num1 <= num2)
num1 -gt num2 (True if num1 > num2)
num1 -ge num2 (True if num1 >= num2)

Notes:
1. Also possible to use ((num1 op num2)), in statement for the numerical operations. For example: if ((num1+num2 == 5)) or if ((num1%num2 == 0)) or if ((num1<num2))
2. Additional option is using [[ ]] instead of [ ] in if statement. For example: if [["$MY_SHELL" == "bash"]]. There are differences between [] and [[]] which will be discussed later.

File operators:
-d FILE (True, if FILE is directory)
-f FILE (True, if FILE is file)
-r FILE (True, if FILE is readable)
-w FILE (True, if FILE is writable)
-x FILE (True, if FILE is executable)
-e FILE (True, if FILE exists)
-s FILE (True, if FILE exists and not empty)

Logical operators AND and OR:
-a for AND
-o for OR

Example 2. File IfElse.sh

```
#!/bin/bash

#PART1: Compare STRINGS
MY_SHELL="bash"

#echo -n "Enter your Shell:"
#read  MY_SHELL


if [ "$MY_SHELL" = "bash" ]
        then
                echo "You run bash"
elif [ $MY_SHELL = "CShell" ]
        then
                echo "You run CShell"
else
        echo "You run Unknown Shell"
fi

#########################################################

#PART2: Compare NUMBERS

read -p "Give me two numbers": NUM1 NUM2

if [ $NUM1 -eq $NUM2 ]
then
        echo "$NUM1 is equal to $NUM2 "
elif [ $NUM1 -lt $NUM2 ]
then
        echo "$NUM1 is smaller than $NUM2"
elif [ $NUM1 -gt $NUM2 ]
then
        echo "$NUM1 is greater than $NUM2"
fi

#Another Version of comparing numbers

if (($NUM1>$NUM2))
then
       echo "$NUM1 is greater than $NUM2 "
else
       echo "$NUM2 is greater than $NUM1$"
fi
```

```
##########################################
#PART3: Check files or directories
read -p "Give me a name of file/directory/somethingElse:" NAME

if [ -d $NAME  ]
then
        echo  -e "$NAME is a directory\n"
        cd $NAME
        echo -e "This directory contains the files: $(ls)\n"
        echo "The numbers of files in directory $NAME is equal to `ls|wc -w`."
elif [ -f $NAME ]
then
        echo -e "$NAME is a file\n"
        cat $NAME
        if [ -r $NAME ]
        then
                echo -e "$NAME is readable\n"
        fi
else
        echo "There is no file or directory named $NAME, but I will create one"
        touch $NAME
        echo -e "The result list of the files is \n $(ls) and number of files is"
fi

if [ -d $NAME -o -f $NAME ]; then
       echo $NAME is file or dir
fi
```

Notes: Above we used new flags for **echo** and **ls** as follows:
- **ls -1** = Prints the list of the files/directories in one column
- **echo -n** = by default, the echo command adds a newline at the end of the output. You can suppress this with the -n option

Example:	Output:
echo "Hello"	Hello
echo Miri	Miri
echo -n "Hello"	HelloMiri
echo Miri	

-**echo -e** = this option instructs the command to interpret backslash escape sequences in the specified text. By utilizing the '**\n**' sequence, we can add a new line and print the subsequent text on a new line below the previous one.
Example:
echo **-e** “The first line **\n** The second line”
Output:
The first line
The second line

Procedures

1. Check the value of HISTSIZE variable.

2. Change the value of HISTSIZE to be 500.

3. Run again the last command which changes the value of HISTSIZE (use !e).

4. Run the command history and then run one of the commands from the list of history, using !n, where n is the number of the command.

5. Repeat the last command using !!

6. Repeat the last touch command, using !to.

6. Create Lab5 directory.

7. In Lab5 write bash script named **IsPositiveNumber.sh** which is doing the following:

- Asks the user to choose a number and puts it in the variable.

- Checks if the number is positive, negative or zero and prints a corresponding message.

Run the program and check that it works properly.

8. In Lab5 write bash script named **IsInRange.sh** which is doing the following:

- Asks the user to choose a number and puts it in the variable.

- Uses only one **if** command to check if the number is between 10 and 100 and prints a corresponding message.

Run the program and check that it works properly.

9. In Lab5 write bash script named **numOfFiles.sh** which is doing the following:

- Asks for a number and puts it in the variable.

- Checks if the number of the files and directories of the present directory together is equal to the given number and prints a corresponding message.

Note: you do not need to check the number of files in the subDirectories of the present directory.

10. In Lab5 write bash script named **MoveFile.sh** which is doing the following:

- Asks a user to choose a string and puts it in the variable NAME.

- Checks if in Lab5 there is a file with the name defined in the variable NAME.

- If no, creates such a file.

- Checks if in Lab5 there is a directory with the name **NAMEdir**, where NAME was defined above.

- If there is no directory with such a name, makes it.

- Moves the file defined by NAME into the directory NAMEdir.

For example, if the user enters NAME=**foo** and there is file foo in Lab5, it will check if there is a directory foodir. If such directory does not exist, it will create it. Then the file foo will be moved into foodir.

To run and check the program, create few files in advance in Lab5. Then run the program and check that it works properly for various cases.

Submission:

Create one compressed file named Linux_Lab5_id1_id2.tar,
where id1 and id2 must be changed to the id numbers of the two partners.

This file will include the files **IsPositiveNumber.sh, IsInRange.sh, numOfFiles.sh and MoveFile.sh**.

To create such file in Lunix, use tar command.

Submit only the file Linux_Lab5_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.