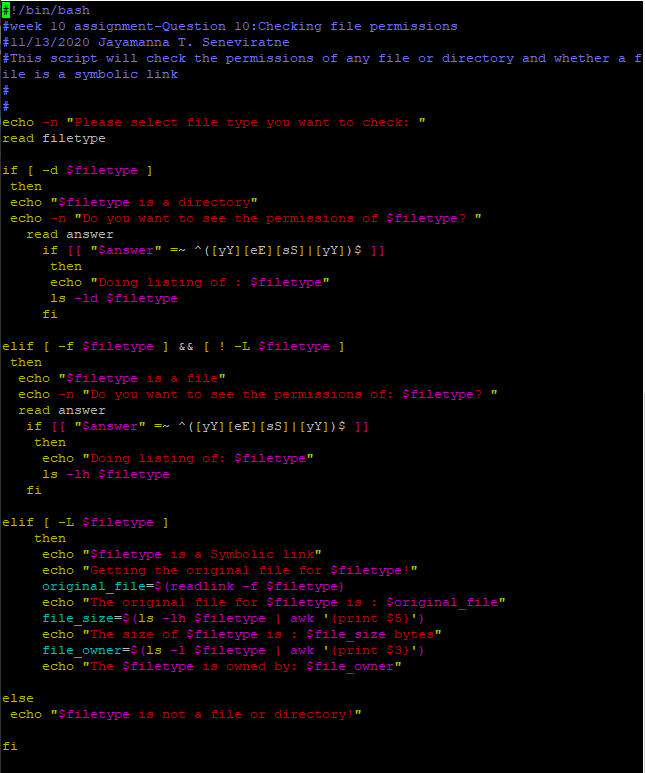
**Week 10 Assignment**

**1. Finish the lab script.**

Add another elif statement to check for symbolic link. If file is a symbolic link, then show me the original file that it points to.

 Example output example:

    calfile-link is a symbolic link. The original file for calfile-link is : calfile The size of calfile is : 453 bytes. The calfile is owned by: <owner name>  Make sure it runs without errors.



2. Comment every line in the scrip bellow and tell what each line is doing. Look up command in the man pages if you don't understand it. The script name is 'myargs'. Copy the script and run it on your system and give a detail comment of what each line is doing.

-----------------------------------------------------

#!/bin/bash 🡪 This statement is required for all bash scripts

# Script name: myargs 🡪 This is a comment about the name of the script

# 🡪 This is a blank comment

# Date: 🡪 A comment about the date of the script

# Name: 🡪 A comment about the name of the owner of the script

# Script to test command line arguments 🡪 This is a brief comment to describe the purpose of the script

if [ $# == 0 ] ; then 🡪 This means if the current set number of variables in the shell equals zero

   echo "Usage:  $(basename $0)  arg1 arg2 ... argn" \ 🡪 Then echo this command about the usage of the script.

   1>&2 🡪 The script STDOUT follows STDERROR

   exit 1 🡪 exit the script/program

fi 🡪 end if ‘if’ statement

echo "The name of this script is $0 ." 🡪 $0 gives the name of the script (The name of this script is ./myargs .)

echo "The arguments are $\* " 🡪 this give all the set command line arguments

echo "The first argument is $1" 🡪 this gives the first set command line argument

echo "The second argument is $2" 🡪 this gives the second set command line argument

echo "The number of arguments is $#" 🡪 This gives the count of the set command line arguments

previous\_args=$\* 🡪 this set up a variable called previous\_args that equals the output of all the set arguments

set niel khail nobo 🡪 this sets three new arguments (niel khail nobo)

echo "All the positional parameters are $\*" 🡪 this lists all three newly set command line arguments

echo "The number of positional parameters is $#" 🡪This counts the number of set command line arguments

echo $previous\_args 🡪 This outputs the previous\_args variable that was set in the current shell

set -- 🡪 This clears all the arguments in the shell

echo "Current args after -- are :  $\*" 🡪There are no set command line arguments to display

set $previous\_args 🡪 This sets the command line arguments back to previous\_args

echo $\* 🡪 This output the most recent set of arguments in the shell set as per “$previous\_args”

-----------------------------------------------

3. Copy the script and run it on your system and give a detail comment of what each line is doing.

 The comments about each line is given in red

------------------------------------------------

#!/bin/bash 🡪 This statement is required for all bash scripts

# Script name: tellage 🡪 This is a comment about the name of the script

# 🡪 This is a blank comment

read -p "How old are you? " 🡪 this statement prompts to read the answer to the question

age=$REPLY 🡪 creating a variable age that equals to the reply of the previous read statement

if (( age < 0 || age > 120 )) 🡪 this is the beginning of the if-then statement, Within double parenthesis the conditions are stated as age less than zero or (||) age greater than 120

then

      echo "You are not a real person! " 🡪 this statement will be echoed if the above conditions are met

      exit 1 🡪 exit set to a non-zero value so that the current if-then statements exists for sure

fi 🡪 this marks the end of the if-then statements

if (( age >= 0 && age < 13 )) 🡪 a new if-then statement begins here. The conditions to be met are stated within double parenthesis. The condition states age greater than or equal to zero and (&&) age less than 13.

then

      echo "You still have some of the best years of your life ahead." 🡪 If the above statement is true, echo this statement within double quotes.

elif (( age > 12 && age < 20 )) 🡪 If the previous condition is not met check for this condition within double parenthesis. The condition says age greater than 12 and (&&) age less than 20.

then

      echo "Important years to learn" 🡪 If this immediate statement is true, echo the comment within double quotes.

elif (( age >= 20 && age < 30 )) 🡪 If this above condition is also not true check for the condition given here. This condition says age greater than or equal to 20 and (&&) age less than 30.

then

      echo "Time to find a potential mate!!" 🡪 If this immediate elif condition is true, echo the statement within double brackets.

elif (( age >= 30 && age < 40 )) 🡪 If none of the conditions above are true check for this condition: age greater than or equal to 30 and (&&) age less than 40.

then

      echo "You are probably changing diapers" 🡪 If this most immediate condition is true, echo the statement within double quotes.

else 🡪 if none of the conditions mentioned above are met, echo the statement below.

      echo "Sorry I asked"

fi 🡪 This is the end of the if-then-elif statements