Emori, Noboru  
CIS 045, Fall 2020

G08125504  
Lab Week 11

2020/11/12

## 1. Finish the lab script

[cs45aa17@mc-redhat01 lab-11]$ ll

total 12

-rwxrwxr-x. 1 cs45aa17 cs45aa17 1537 Nov 13 00:34 checkfiletype

-rwxrw-r--. 1 cs45aa17 cs45aa17 385 Nov 7 14:09 etcdir

lrwxrwxrwx. 1 cs45aa17 cs45aa17 5 Nov 7 15:16 etc-link -> /etc/

lrwxrwxrwx. 1 cs45aa17 cs45aa17 11 Nov 7 15:46 passwd-link -> /etc/passwd

-rwxrw-r--. 1 cs45aa17 cs45aa17 121 Nov 7 13:44 test14b

[cs45aa17@mc-redhat01 lab-11]$ ./checkfiletype

Please select the file you want to check the type of: /etc/passwd

/etc/passwd is a file

Do you want to see the permissions of /etc/passwd? yes

Doing listing of /etc/passwd

-rw-r--r--. 1 root root 5.4K Oct 10 16:06 /etc/passwd

[cs45aa17@mc-redhat01 lab-11]$ ./checkfiletype

Please select the file you want to check the type of: /etc

/etc is a directory

Do you want to see the permissions of /etc? yes

Doing listing of: /etc

drwxrwxr-x. 2 cs45aa17 cs45aa17 91 Nov 13 00:40 .

[cs45aa17@mc-redhat01 lab-11]$ ./checkfiletype

Please select the file you want to check the type of: etc-link

etc-link is a symbolic link

The original folder for etc-link is: /etc/

The size of /etc/ is: 12288 bytes

The /etc/ is owned by: root

[cs45aa17@mc-redhat01 lab-11]$ ./checkfiletype

Please select the file you want to check the type of: passwd-link

passwd-link is a symbolic link

The original file for passwd-link is: /etc/passwd

The size of /etc/passwd is: 5467 bytes

The /etc/passwd is owned by: root

### Output when used:

### Source code

#!/bin/bash

# Script Name: checkfiletype

#

# Late Edit: 11/12/2020

# Noboru Emori (within Husayn Campbell's Class)

echo -n "Please select the file you want to check the type of: "

read filetype

# If directory and not a symlink

if [ -d $filetype ] && [ ! -L $filetype ]

then echo "$filetype is a directory"

echo -n "Do you want to see the permissions of $filetype? "

read answer

if [ $answer = "yes" ]

then

echo "Doing listing of: $filetype"

ls -ld $fileytpe

fi

# If regular file and not a symlink

elif [ -f $filetype ] && [ ! -L $filetype ]

then

echo "$filetype is a file"

echo -n "Do you want to see the permissions of $filetype? "

read answer

if [ $answer = "yes" ]

then

echo "Doing listing of $filetype"

ls -lh $filetype

fi

# If a symlink

elif [ -L $filetype ]

then

echo "$filetype is a symbolic link"

fullpath=$(ls -l $filetype | awk '{print $NF}')

# if a file and a symlink

if [ -f $fullpath ]

then

echo "The original file for $filetype is: $fullpath"

echo "The size of $fullpath is: $(ls -l $fullpath | awk '{print $5}') bytes"

echo "The $fullpath is owned by: $(ls -l $fullpath | awk '{print $3}')"

# if a directory and symlink

else

echo "The original folder for $filetype is: $fullpath"

echo "The size of $fullpath is: $(ls -ld $fullpath | awk '{print $5}') bytes"

echo "The $fullpath is owned by: $(ls -ld $fullpath | awk '{print $3}')"

fi

else

# If not any of the above

echo "$filetype is not a file or directory or symbolic link"

fi

## Commenting on myargs

### Commented Code:

[cs45aa17@mc-redhat01 lab-11]$ cat myargs

#!/bin/bash

# Script name: myargs

#

# Date: 11/12/2020

# Name: Code: Husayn Campbell; Comments: Noboru Emori

# Script to test command line arguments

if [ $# == 0 ] ; then # If there are no positional parameters (arguments), go inside this statement

echo "Usage: $(basename $0) arg1 arg2 ... argn" \ # Print how to use the script; $(basename $0) prints myargs

1>&2 # Send any stdout into stderr

exit 1 # Exit with code 1, general unknown error. Can be used with echo $?

fi

echo "The name of this script is $0 ." # Prints the name of the file, which is $0

echo "The arguments are $\* " # Prints string with all of the positional parameters supplied when running command

echo "The first argument is $1" # Prints the first positional parameter

echo "The second argument is $2" # Prints the second positional parameters

echo "The number of arguments is $#" # Prints the number of positional parameters

previous\_args=$\* # Puts the list of positional parameters into a variable called previous\_args

set niel khail nobo # Sets a new list of positional parameters. $0 is still script, but $1 is niel, $2 is khail, $3 is nobo

echo "All the positional parameters are $\*" # Prints string with all of new positional parameters

echo "The number of positional parameters is $#" # Prints the number of new positional paramaters (3, in this case)

echo $previous\_args # Prints the list of previously saved positional parameters

set -- # Unsets all the positional parameters, clearing neil, kahil, and nobo in this case

echo "Current args after -- are : $\*" # Prints current positional parameters; Should show nothing

set $previous\_args # sets positional parameters to the first list

echo $\* # Prints the positional parameters, which is the first list

### Sample output

[cs45aa17@mc-redhat01 lab-11]$ ./myargs

Usage: myargs arg1 arg2 ... argn

[cs45aa17@mc-redhat01 lab-11]$ ./myargs hello I am Noboru

The name of this script is ./myargs .

The arguments are hello I am Noboru

The first argument is hello

The second argument is I

The number of arguments is 4

All the positional parameters are niel khail nobo

The number of positional parameters is 3

hello I am Noboru

Current args after -- are :

hello I am Noboru

# tellage Script

### Commented Code

[cs45aa17@mc-redhat01 lab-11]$ cat tellage

#!/bin/bash

# Scriptname: tellage

#

read -p"How old are you? " # Prompts a user input on same line while printing "How old are you"

age=$REPLY # Sets the user input as the variable age

# Note for all below, age needs to be an integer to work

if (( age < 0 || age > 120 )) # If age is negative or greater than 120, go into here

then

echo "You are not a real person! " # Print this statement

exit 1 # Stop the program with code 1, unknown error

fi

# If <0 or >120 did not trigger, program keeps going

if (( age >= 0 && age < 13 )) # If age is between 0 inclusive and 13 exclusive or [0,13)

then

echo "You still have some of the best years of your life ahead." # Hope this is true!

elif (( age > 12 && age < 20 )) # Else if age is (12,20), i.e. teenager, go here

then

echo "Important years to learn" # Learn in the teenage years

elif (( age >= 20 && age < 30 )) # Else if age is [20,30), i.e. 20s, go here

then

echo "Time to find a potential mate!!" # Probability wise probably true

elif (( age >= 30 && age < 40 )) # Else if age is [30,39), i.e. 30s, go here

then

echo "You are probably changing diapers" # Sounds fun

else # If none of the ranges above apply, so age [40,120]

echo "Sorry I asked" # Oh no!

fi

### Sample Output

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? -1

You are not a real person!

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 125

You are not a real person!

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 10

You still have some of the best years of your life ahead.

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 13

Important years to learn

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 29

Time to find a potential mate!!

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 30

You are probably changing diapers

[cs45aa17@mc-redhat01 lab-11]$ ./tellage

How old are you? 62

Sorry I asked