#!/bin/bash

#

#

#

# Script Name : Filetype.sh

# Author: Ashley Garam

# Date: 20/06/2014

#

# Exitting Function For Script

# (This Function allocates a process ID number To The Script which is Launched, to the variable $TOP\_PID. So when the "closeloop4" Function is activied, within the script, it then kills that process ID. # which is this script, which in turn shut down or terminates the current script running.)

#

#

**trap** "exit 1" **TERM**

**export TOP\_PID=$$**

**function closeloop4**()

{

**echo ""**

**kill -s TERM $TOP\_PID**

}

#

#

#Actual script for Filetype.sh shown below.

#

#

#

#Colour Variables

#

**SET** Colour Wh='\e[0;37'm # White Default Colour

#

**SET** Colour BR='\e[1;31'm # Red options

**SET** Colour BG='\e[1;32'm # Green Selection

**SET** Colour BP='\e[1;35'm # Purple Display titles

**SET** Colour BB='\e[1;34'm # Blue exit

**SET** Colour BY='\e[1;33'm # Yellow mistakes

#

#

# Variables which are used throughout out the Script

#

#Dirfind="` ls -Q -I -l -F $userinput 2>/dev/null | cut -f1-30 -d" "| wc -l `" ( Is a Variable used Is To See If the directory exists within the current filing system.)

#

# FileLocate="`ls -Q -I -l -f -1 $userinput 2>/dev/null | cut -f1-30 -d" " | wc -l`" (Is a Variable used To See If The Current File ( Variable $Userinput ) exists within the current directory of the file system.)

#

# FileShow="`ls -pQ $userinput 2>/dev/null | grep -v /`" (Is a Variable used To Show the current files within the users input Directory.)

#

# DirShow="`ls -Q -I -l -F $userinput 2>/dev/null | cut -f1-30 -d" " | grep /`" ( Is a Variable used To Show the current Directories matching the users input Data)

#

# Exit Loop Variable Below

#

**SET** Loop exloop4=5

#

# Functions For Case Statements

#

# Function 1 (Directory Finder Result) below (An If Statement to see weather the directory exists within the current filing system)

#

**FUNCTION** Dirfindresult(){

**IF** **Result** [ "Dirfind" > "0" ]

**THEN**

**DISPLAY “”**

**DISPLAY “”**

**DISPLAY -e** "Found That ${BR}$userinput${Wh} Exist! Listing Content Below"

**DISPLAY**

**DISPLAY -e** "Total contents within ${BR}$userinput${Wh} = ${BR}$Dirfind${Wh}"

**DISPLAY**

**DISPLAY**

**DISPLAY -e** "Showing the ${BG}Sub Directories${Who} Below"

**DISPLAY**

**DISPLAY -e** ${BG}"$Disown"${Who} | more

**DISPLAY**

**DISPLAY -e** "Showing The ${BG}Files${Who} below within ${BG}$user put${Who}"

**DISPLAY**

**DISPLAY -e** ${BG}"$File Show"${Who} | more

**DISPLAY**

**DISPLAY**

**DISPLAY -e** "Please Press ${BR}Enter${Who} To Continue"

**READ** enter

**DISPLAY**

**DISPLAY -e** " ${BB}Exiting This Program and Returning back to Sub Main Menu Now${Who}"

**WAIT 3**

**EXECUTE**

./filetype\_Msub.sh

**ELIF Result** [ "Dirfind" = "0" ]

**THEN**

**DISPLAY “”**

**DISPLAY “”**

**DISPLAY -e** " This ${BR}Directory${Who} Doesn't Exist!! Please Retype your Section when Menu Returns!"

**DISPLAY**

**DISPLAY -e** "Please Press ${BR} Enter ${Wh} To Continue"

**READ** enter

**DISPLAY**

**DISPLAY -e** " ${BB}Returning back to Filetype.sh Menu${Who}"

**WAIT 4**

**EXECUTE**

./filetype.sh

**ENDIF**

}

#

#

# Function 2 (File Finder Results) Below (An If Statement to see weather the file exists within the current filing system)

#

**FUNCTION** FileFinderResult(){

**IF Result** [ "FileLocate" = "1" ]

**THEN**

**DISPLAY “”**

**DISPLAY “”**

**DISPLAY -e** "Found That ${BR}$userinput${Wh} Exist in ${BR}`pwd`${Wh}! Listing Content Below"

**DISPLAY**

**DISPLAY**

**DISPLAY -e** "Please Press ${BR}Enter${Who} To Continue"

**READ** enter1

**DISPLAY**

**DISPLAY**

**DISPLAY To Screen** (cat) $userinput | more

**DISPLAY**

**DISPLAY -e** "Please Press ${BR}Enter${Wh} To Continue"

**READ** enter2

**DISPLAY**

**DISPLAY -e** " ${BB}Exiting This Program and Returning back to Sub Main Menu Now${Wh}"

**WAIT 3**

**EXECUTE**

./filetype\_Msub.sh

**ELIF Result** [ "FileLocate" = "0" ]

**THEN**

**DISPLAY “”**

**DISPLAY “”**

**DISPLAY -e** " This ${BG}File${Wh} Doesn't Exist within ${BR}`pwd`${Wh} !! Please Retype your Section when Menu Returns!"

**DISPLAY**

**DISPLAY -e** "Please Press ${BR} Enter ${Wh} To Continue"

**READ** enter

**DISPLAY**

**DISPLAY -e** "${BB} Returning back to Filetype.sh Menu${Wh}"

**WAIT 4**

**EXECUTE**

./filetype.sh

**ENDIF**

}

#

#

# Function 3. Case Statement for userinput variables

#

#2.

**FUNCTION** userinputmenu(){

**CASE** "userinput"

#

# Looks for a / (Forward Slash) in the variable $userinput, When Found it shows what’s below and sets the variable

#

**INPUT** /\*)

**CLEAR Screen**

**DISPLAY -e** "Searching For ${BR}$userinput${Wh} Exists below "

**DISPLAY**

**DISPLAY “”**

**DISPLAY “”**

**STORE** **Userinput To** Dirfind="`ls -Q -I -l -F $userinput 2>/dev/null | cut -f1-30 -d" "| wc -l`" # ( Is a Variable used Is To See If the directory exists within the current filing system.)

**DISPLAY**

**DISPLAY**

#

# Loads Function 1 (Directory Finder Result) Which is a the top of the script (An If Statement to see weather the directory exists within the current filing system)

#

**CALL FUNCTION** Dirfindresult

;;

#

# The Exit Statement below.

#

**INPUT** xxx)

**CLEAR Screen**

**DISPLAY**

**DISPLAY -e** "${BB} Exiting This Program and Returning back to Sub Main Menu Now${Wh}"

**DISPLAY**

**EXECUTE**

./filetype\_Msub.sh

;;

#

# Looks for a "." (dot. Which Means a file has been entered) in the variable $userinput, When Found it shows what’s below and sets the variable

#

**INPUT** \*.)

**DISPLAY**

**DISPLAY**

**DISPLAY -e** "Searching For File ${BR}$userinput${Wh}, within `pwd` To see if exists below "

**DISPLAY**

#

# Is a Variable used To See If The Current File ( Sets Variable $Userinput ) exists within the current directory of the file system.

#

**STORE Userinput To** FileLocate="`ls -Q -I -l –f $userinput 2>/dev/null | cut -f1-30 -d" " | wc -l`"

#

# Loads Function 2 (File Finder Results) Which is at the top of the script (An If Statement to see weather the file exists within the current filing system)

#

**CALL FUNCTION** FileFinderResult

**WAIT 2**

;;

#

# Looks for any letter within the usersinput variable. Because some files in Linux only have a filename.

#

**INPUT** [123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ]\*)

**CLEAR Screen**

**WAIT 3**

**DISPLAY -e** "Searching For File ${BR}$userinput${Wh}, within ${BR}`pwd`${Wh} To see if exists below "

#

# Loads Function 2 (File Finder Results) Which is at the top of the script (An If Statement to see weather the file exists within the current filing system)

#

**CALL FUNCTION** FileFinderResult

;;

#

# Looks for any letter within the usersinput variable. Because some files in Linux only have a filename.

#

**INPUT** [abcdefghijklmnopqrstuvwxyz]\*)

**CLEAR Screen**

**WAIT 3**

**DISPLAY “”**

**DISPLAY “”**

**DISPLAY -e** "Searching For File ${BR}$userinput${Wh}, within ${BR}`pwd`${Wh} To see if exists below1 "

#

# Loads Function 2 (File Finder Results) Which is at the top of the script (An If Statement to see weather the file exists within the current filing system)

#

**CALL FUNCTION** FileFinderResult

;;

**ENDCASE**

}

#

# The While loop Starts Below

#

**DOWHILE** [ exloop4 = 5 ]

#

# The Mini Menu for the FileType Script Below

#

**CLEAR Screen**

**DISPLAY**

**DISPLAY**

**DISPLAY**

**DISPLAY -e** "${BP}This Script Checks to See whether the users input is idler a directory or a file.${Wh}"

**DISPLAY**

**DISPLAY**

**DISPLAY**

**DISPLAY -e** " To Quit This Script, Type ${BR}(xxx)${Wh} "

**DISPLAY**

**DISPLAY**

**DISPLAY -e** " Type Idler a ${BG}Directory${Wh} ${BR}(/'YourFolder')${Wh} or a ${BG}File${Wh} ${BR}(script.sh)${Wh} to Search for below"

**DISPLAY**

**DISPLAY**

**DISPLAY**

**DISPLAY**

#

# The User Inputs they request

#

**READ** userinput

#

# All of the Variables below are set with the $usersinput variable. Which are used throughout the script

#

**CLEAR Screen**

**STORE Userinput to** Dirfind="`ls -Q -I -l -F $userinput 2>/dev/null | cut -f1-30 -d" "| wc -l`"

**STORE Userinput to** FileLocate="`ls -Q -I -l -f -1 $userinput 2>/dev/null | cut -f1-30 -d" " | wc -l`"

**STORE Userinput to** FileShow="`ls -pQ $userinput 2>/dev/null | grep -v /`"

**STORE Userinput to** DirShow="`ls -Q -I -l -F $userinput 2>/dev/null | cut -f1-30 -d" " | grep /`"

**CLEAR Screen**

#

# Load the function 3 Case Statement (userinputmenu) for the variable $usersinput, to analyse weather what the user has inputted is a file or a directory.

#

**CALL FUNCTION** userinputmenu

Closeloop4

**ENDO**