DPS918 Assignmen 1

#!/bin/bash

# Assignment 1

# Course: DPS918

# Family Name: Tran

# Given Name: Elvis

# Student Number: 065-743-114

# Login name: etran12

# Professor: Les Czegel

# Due Date: March 17, 2017

#

# I declare that the attached assignment is my own work in accordance with

# Seneca Academic Policy. No part of this assignment has been copied manually

# or electronically from any other source (including web sites) or distributed

# to other students.

#Testing filename

ls -ald $1 2> /dev/null 1> /dev/null

#Filename Exit Status for ls -ald

filenameExit=$?

#Error Messages

validInput=0

if [ $# -gt 1 ]; then

echo "Usage: showpath [ filename ] "

exit 1

elif [ $filenameExit != 0 ]; then

echo "$1 is not a valid filename"

exit 1

fi

case $# in

0) dir=. ;;

1) dir=$1 ;;

\*) echo "cdir: only one argument is allowed" >&2

exit 1 ;;

esac

#Setting absolute path name

if [ $validInput == 0 ]; then

#Check to see if argument was a relative path

ls -ald $(pwd)/$dir 2> /dev/null 1> /dev/null

#Checks to see if arugment is an absolute path name, if not, then add the pwd to make it an absolute directory

#Otherwise the arugment was already an absolute path name

if [ $? == 0 ]; then

pathName=$(pwd)/$dir

else

pathName=$dir

fi

#Breaking down Path Name into individual layers by new lines

pNameFile=$(mktemp)

pNameFile2=$(mktemp)

echo $pathName | sed 's/\//\n/g' > ${pNameFile2}

#Add root layer to the path name file first

echo "/" >> ${pNameFile}

#Remove first line which is a \n, end result is all the layers of the absolute path delimited by a new line

cat ${pNameFile2} | sed 1d >> ${pNameFile}

#Creating the Permission File

permissionFile=$(mktemp)

while read line

do

if [ $line == '/' ] ; then

currentLevel='/'

else

currentLevel=$currentLevel/$line

fi

ls -ald $currentLevel | cut -c1-10 >> ${permissionFile}

done < ${pNameFile}

#Creating Details File

#Links: 1 Owner: dps918 Group: users Size: 445 Modified: Mar 20 2014

detailFile=$(mktemp)

while read line

do

if [ $line == '/' ]; then

currentLevel='/'

else

currentLevel=$currentLevel/$line

fi

newLine="Links: $(ls -ald $currentLevel | awk '{print $2}') "

newLine="$newLine Owner: $(ls -ald $currentLevel | awk '{print $3}') "

newLine="$newLine Group: $(ls -ald $currentLevel | awk '{print $3}') "

newLine="$newLine Size: $(ls -ald $currentLevel | awk '{print $5}') "

newLine="$newLine Modified: $(ls -ald $currentLevel | awk '{print $6,$7,$8}') "

echo $newLine >> ${detailFile}

done < ${pNameFile}

#Counting Number of Layers

#Declare that numberOfLayers is an integer not a string, also from here on currentLevel variable is used for display grid

declare -i maxLayers

declare -i currentLevel

#Number of Levels in the File Path

maxLayers=$(wc -l ${pNameFile} | awk '{print $1}')

#Sets the Current Level to be the selected file/directory

currentLevel=$maxLayers

#Creating the Display file for View

displayFile=$(mktemp)

mainPermissions=$(mktemp)

firstCharacterPermission=$(mktemp)

editedPermissions=$(mktemp)

while read line

do

echo -n $line | cut -c2-10 | sed 's/.../& /g' | sed 's/./& /g' >> ${mainPermissions}

done < ${permissionFile}

while read line

do

echo $line | cut -c1 >> ${firstCharacterPermission}

done < ${permissionFile}

paste -d' ' ${firstCharacterPermission} ${mainPermissions} ${pNameFile} > ${editedPermissions}

while read line

do

echo $line | sed 's/\n/&\n/g' >> ${displayFile}

echo "" >> ${displayFile}

done < ${editedPermissions}

#Output Algorithm

clear

#echo $currentLevel

echo " Owner Group Other Filename"

echo " ----- ----- ----- --------"

echo " "

cat ${displayFile} | awk 'BEGIN{f="([^ ]+ )"} {print $1" " $2" " $3" " $4" " $5" " $6" " $7" " $8" " $9" " $10" " $11" ";}'

startCursor=$((((maxLayers \* 2))+1))

#First character of Filename

column=26

row=$startCursor

lastRow=$(tput lines)

detailedLine=0

stty -icanon min 1 time 0 -echo

#Prints the Initial Details Line

tput cup $((row+1)) 0

cat ${detailFile} | awk 'NR == '$currentLevel''

tput cup $((lastRow - 2)) 0

printf "Valid commands: u(p) d(own) q(uit) \n"

line=$currentLevel #Keeps track of which line to read in the Detail File

while true

do

# Places the cursor at the correct coordinates and waits for input

tput cup $row $column

command=$(dd bs=3 count=1 2> /dev/null)

case $command in

u) if [ "$row" -gt 3 ]

then

# Clears the details line then prints details line for one directory level up

tput cup $((row+1)) 0

tput el

row=$((row - 2))

detailsLine=$((row + 1))

tput cup $detailsLine 0

line=$((line-1))

cat ${detailFile} | awk 'NR == '$line''

fi;;

d) if [ "$row" -lt $startCursor ]

then

# Clears the details line then prints details line for one directory level down

tput cup $((row+1)) 0

tput el

row=$((row + 2))

detailsLine=$((row + 1))

tput cup $detailsLine 0

line=$((line+1))

cat ${detailFile} | awk 'NR == '$line''

fi;;

q) rm /tmp/showpath.temp.$$ 2> /dev/null

stty icanon echo

tput cup $lastRow 0

exit 0;;

esac

done

fi