# Author, 2019

#!/bin/bash

# Used paths

OPPATH=~/operations

LOGPATH=$OPPATH/logs

LOG=$LOGPATH/access.log

# Reserved words

DIRS='double data'

COMMANDS='initialize deny allow increase print'

# Script parameters

i=""

n=0

TARGETPATH=""

# Dynamic variable

TARGETFILE=""

CTS=''

COMMAND=""

# Messages

pUSAGE="\nUsage: ./solution <initialize|deny|allow|increase|print> [-d data|double [-i <0-9>]]\n"

pTMA="Error: too many command line arguments"

pTLA="Error: too little command line arguments"

pWAC="Error: wrong argument count"

pUCN="Error: unknown command '"

pUDN="Error: unknown directory name '"

pMRA="Error: required argument '-d' is not specified"

pODM="Error: operation directory '$OPPATH' doesn't exist"

pTDM="Error: target directory $TARGETPATH doesn't exist"

pTFM="Error: target file $TARGETFILE doesn't exist"

pLPM="Error: log directory $LOGPATH doesn't exist"

pLFM="Error: log file $LOG doesn't exist"

pWNA="Error: wrong numeric argument '-i': "

function msg2cli () {

case $1 in

0)

echo -e $pUSAGE

;;

1)

echo $pTMA

msg2cli 0

;;

2)

echo $pWAC

msg2cli 0

;;

3)

echo $pUCN$2"'"

msg2cli 0

;;

4)

echo $pTLA

msg2cli 0

;;

5) echo $pUDN$2"'"

msg2cli 0

;;

6)

echo $pMRA

msg2cli 0

;;

7)

echo $pODM

msg2cli

;;

8)

echo $pTDM

msg2cli

;;

9)

echo $pTFM

msg2cli

;;

10)

echo $pLPM

;;

11)

echo $pLFM

;;

12)

echo $pWNA $2

;;

\*)

;;

esac

}

function write2log () {

if [[ ! -d $LOGPATH ]]; then

msg2cli 10

return 1

elif [[ ! -f $LOG ]]; then

msg2cli 11

return 1

else

CTS=$(date +"[%Y.%m.%d %H:%M:%S]: ")

echo $CTS $1 1>> $LOG

return 0

fi

}

function print () {

write2log "Printing the file '$TARGETFILE'"

if [[ -s $TARGETFILE ]]; then

echo "Printing the file '$TARGETFILE':"

cat $TARGETFILE

else

echo "File '$TARGETFILE' is empty"

fi

}

function increase () {

write2log "Increasing value in file '$TARGETFILE'"

if [[ -w $TARGETFILE ]]; then

if [[ -s $TARGETFILE ]]; then

read n<$TARGETFILE

n=$(($n+1))

echo $n>$TARGETFILE

else

echo $n>$TARGETFILE

fi

else

echo "File '$TARGETFILE' is write restricted"

exit 0

fi

}

function allow () {

write2log "Enabling write grants to file '$TARGETFILE'"

sudo chattr =e $TARGETFILE

sudo chattr -i $TARGETFILE

}

function deny () {

write2log "Disabling write grants from file '$TARGETFILE'"

sudo chattr =e $TARGETFILE

sudo chattr +i $TARGETFILE

}

function initialize () {

if [[ -d $OPPATH ]]; then

find $OPPATH -exec sudo chattr =e {} \;

rm -R $OPPATH

fi

mkdir $OPPATH

cd $OPPATH

for diritem in $DIRS

do

mkdir $diritem

for (( i=0; i<10; i++ ))

do

touch $diritem/File$i.txt

done

touch $diritem/default.txt

done

mkdir $LOGPATH

touch $LOG

sudo chattr +a $LOG

write2log "Directory structure was initialized"

}

# main

if [[ $# -eq 0 ]]; then

msg2cli 0

exit 0

elif [ $# -gt 5 ]; then

msg2cli 1

exit 0

elif [ $# -eq 2 -o $# -eq 4 ]; then

msg2cli 2

exit 0

else

COMMAND=$1

case $COMMAND in

initialize)

if [ $# -gt 1 ]; then

msg2cli 1

else

initialize

fi

;;

allow|deny|increase|print)

shift

if [ $# -eq 0 ]; then

msg2cli 4

exit 0

else

while [[ $# -gt 0 ]]

do

case $1 in

-i)

if [[ -n $2 ]]; then

if [ $2 -ge 0 ] && [ $2 -le 9 ]; then

i=$2

else

msg2cli 12 $2

exit 0

fi

else

msg2cli 12 $2

exit 0

fi

shift

shift

;;

-d)

if [ "$2" == "data" -o "$2" == "double" ]; then

TARGETPATH="$OPPATH/$2"

else

msg2cli 5 $2

exit 0

fi

shift

shift

;;

\*)

;;

esac

done

if [[ -z "$TARGETPATH" ]]; then

msg2cli 6

exit 0

elif [[ -z $i ]]; then

TARGETFILE="$TARGETPATH/default.txt"

else

TARGETFILE="$TARGETPATH/File$i.txt"

fi

fi

if [[ ! -d "$OPPATH" ]]; then

msg2cli 7

exit 0

elif [[ ! -d "$TARGETPATH" ]]; then

msg2cli 8

exit 0

elif [[ ! -f "$TARGETFILE" ]]; then

msg2cli 9

exit 0

else

$COMMAND

fi

;;

\*)

msg2cli 3 $1

exit 0

;;

esac

fi

# Operation "Initialize"

# echo -e "Usage: ./solution <command> [parameters]"

# echo -e "<command> may be one of next words: initialize deny allow increase print"

# echo -e "[parameters] must have next format: '-d <PATH> [-i <NUMBER>]', where"

# echo -e "<PATH> must be one of two words: data or double"

# echo -e "<NUMBER> must be 0-9"