DPS918 Assignment 2

```
#!/bin/bash
oldsettings=$(stty -g)
lines=$(tput lines)
columns=$(tput cols)
trap "stty $oldsettings; tput cup $lines 0; exit 0" SIGINT
if [ $# != 1 ]; then
   if [ $# = 0 ]; then
      set .
   elif [[ $# > 1 ]]; then
      echo "Usage: showpath [ filename ]" >&2
      exit 1
   fi
else
   if [ ! -e $1 ]; then
      echo "$1 is not a valid filename" >&2
      exit 1
   fi
fi
dirNames[0]="/"
count=$(readlink -f $1 | tr "/" " " | wc -w)
curLevel=0
curCol=9
col=([0]=2 [1]=4 [2]=6 [3]=10 [4]=12 [5]=14 [6]=18 [7]=20 [8]=22 [9]=26)
numLevels=$(( count ))
for (( i = 1; i <= count; i++ ))
do
   dirNames[i]=$(readlink -f $1 | tr "/" " " | awk '{print $'$i'}')
done
curLevel=$numLevels
rowsTaken=$(( ( ( numLevels + 1 ) * 2 ) + 2 ))
threshold=$(( lines - 4 ))
if [[ $rowsTaken -gt $threshold ]]; then
   exit 1
fi
stty -icanon min 1 time 10
stty -echo
clear
tput cup 0 2
echo "OWNER   GROUP   OTHER   FILENAME"
tput cup 1 2
echo "-----   -----   -----   --------"
tput cup 3 0
pathNames[0]=${dirNames[0]}
permString=$(ls -ld ${pathNames[0]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^ ] [^ ]/&
/g')
permString="$permString${dirNames[0]}"
echo "$permString"
for ((i = 1; i <= count; i++))
do
   if [ ! -d "${pathNames[i-1]}${dirNames[i]}/" ]; then
      pathNames[i]="${pathNames[i-1]}${dirNames[i]}"
   else
      pathNames[i]="${pathNames[i-1]}${dirNames[i]}/"
   fi
   permString=$(ls -ld ${pathNames[i]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^ ] [^ ]/&
/g')
   permString="$permString${dirNames[i]}"
   tput cup $(( ( ( i + 1 ) * 2 ) + 1 )) 0
   echo "$permString" | cut -c1-$columns
done
tput cup $(( lines - 4 )) 0
echo -n "Valid keys: k (up),  j (down): move between filenames"
tput cup $(( lines - 3 )) 12
echo -n "h (left), l (right): move between permissions"
tput cup $(( lines - 2 )) 12
echo -n "r, w, x, -: change permissions;  q: quit"
tput cup $(( ( ( curLevel + 1 ) * 2 ) + 2 )) 2
ls -ld ${pathNames[curLevel]} | awk '{ print "Links: " $2 "  Owner: " $3 "  Group: " $4 "  Size: " $5 "  Modified: " $6 " " $7 " " $8}'
key=
while [ "$key" != "q" ]
do
   if [ $curLevel = 0 ]; then
      tput cup 3 ${col[$curCol]}
   else
      tput cup $(( ( ( curLevel + 1 ) * 2 ) + 1 )) ${col[$curCol]}
   fi
   tput cnorm
   key=$(dd bs=1 count=1 2> /dev/null)
   if [ "$key" = "k" ]; then
      if [[ $curLevel != 0 ]]; then
         tput civis
         tput cup $(( ( ( curLevel + 1 ) * 2 ) + 2 )) 0
         tput el
         curLevels=$(( curLevel-- ))
         if [ $curLevel = 0 ]; then
            tput cup 4 2
         else
            tput cup $(( ( ( curLevel + 1 ) * 2 ) + 2 )) 2
         fi
         ls -ld ${pathNames[curLevel]} | awk '{ print "Links: " $2 "  Owner: " $3 "  Group: " $4 "  Size: " $5 "  Modified: " $6 " " $7 " " $8 }'
      fi
   fi
   if [ "$key" = "j" ]; then
      if [[ $curLevel < $numLevels ]]; then
         tput civis
         tput cup $(( ( ( curLevel + 1 ) * 2 ) + 2 )) 0
         tput el
         curLevels=$(( curLevel++ ))
         tput cup $(( ( ( curLevel + 1 ) * 2 ) + 2 )) 2
         ls -ld ${pathNames[curLevel]} | awk '{ print "Links: " $2 "  Owner: " $3 "  Group: " $4 "  Size: " $5 "  Modified: " $6 " " $7 " " $8 }'
      fi
   fi
   if [[ "$key" = "h" ]]; then
      if (( "$curCol" > 0 )); then
         curCols=$(( curCol-- ))
      fi
   fi
   if [ "$key" = "l" ]; then
      if (( "$curCol" < 9 )); then
         curCols=$(( curCol++ ))
      fi
   fi
   if [ "$key" = "r" ]; then
```

```
      case $curCol in
         0) chmod u+r ${pathNames[curLevel]} 2> /dev/null;;
         3) chmod g+r ${pathNames[curLevel]} 2> /dev/null;;
         6) chmod o+r ${pathNames[curLevel]} 2> /dev/null;;
      esac
      permString=$(ls -ld ${pathNames[$curLevel]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^
] [^ ]/&  /g')
      permString="$permString${dirNames[$curLevel]}"
      tput cup $(( ( ( curLevel +1 ) * 2 ) + 1 )) 0
      echo "$permString" | cut -c1-$columns
   fi
   if [ "$key" = "w" ]; then
      case $curCol in
         1) chmod u+w ${pathNames[curLevel]} 2> /dev/null;;
         4) chmod g+w ${pathNames[curLevel]} 2> /dev/null;;
         7) chmod o+w ${pathNames[curLevel]} 2> /dev/null;;
      esac
      permString=$(ls -ld ${pathNames[$curLevel]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^
] [^ ]/&  /g')
      permString="$permString${dirNames[$curLevel]}"
      tput cup $(( ( ( curLevel + 1 ) * 2 ) + 1 )) 0
      echo "$permString" | cut -c1-$columns
   fi
   if [ "$key" = "x" ]; then
      case $curCol in
         2) chmod u+x ${pathNames[curLevel]} 2> /dev/null;;
         5) chmod g+x ${pathNames[curLevel]} 2> /dev/null;;
         8) chmod o+x ${pathNames[curLevel]} 2> /dev/null;;
      esac
      permString=$(ls -ld ${pathNames[$curLevel]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^
] [^ ]/&  /g')
      permString="$permString${dirNames[$curLevel]}"
      tput cup $(( ( ( curLevel + 1 ) * 2 ) + 1 )) 0
      echo "$permString" | cut -c1-$columns
   fi

   if [ "$key" = "-" ]; then
      case $curCol in
         0) chmod u-r ${pathNames[curLevel]} 2> /dev/null;;
         1) chmod u-w ${pathNames[curLevel]} 2> /dev/null;;
         2) chmod u-x ${pathNames[curLevel]} 2> /dev/null;;
         3) chmod g-r ${pathNames[curLevel]} 2> /dev/null;;
         4) chmod g-w ${pathNames[curLevel]} 2> /dev/null;;
         5) chmod g-x ${pathNames[curLevel]} 2> /dev/null;;
         6) chmod o-r ${pathNames[curLevel]} 2> /dev/null;;
         7) chmod o-w ${pathNames[curLevel]} 2> /dev/null;;
         8) chmod o-x ${pathNames[curLevel]} 2> /dev/null;;
      esac
      permString=$(ls -ld ${pathNames[$curLevel]} | cut -c1-10 | sed 's/./& /g' | sed 's/[^ ]* *[^ ] [^
] [^ ]/&  /g')
      permString="$permString${dirNames[$curLevel]}"
      tput cup $(( ( ( curLevel + 1 ) * 2 ) + 1 )) 0
      echo "$permString" | cut -c1-$columns
   fi
done
tput cnorm
stty $oldsettings
tput cup $lines 0
exit 0
```