Assignment 13B
Linux Scripting

Table of Contents

Introduction

The purpose of this assignment is to build a Bowling program which will display 2 teams.

Bedrock Lanes

In this exercise, you have an opportunity to create a bowling data entry program for 2 teams.

Exercise Overview

You will be creating a file named bowl.

You will be provided with the following files:

- bowlers – contains a list of all the bowlers for 12 teams with 5 bowlers per team.
  The file is organized as follows:
  - team number
  - bowler name
  - handicap
  - total pins accumulated
  - total games bowled
  - current average (total pins/total games bowled)

  Each field is separated by the "|" symbol

- teams – contains a list of team names
  The file is organized as follows:
  - team number
  - team name

  Each field is separated by the "|" symbol

- testscores – contains a list of scores which can be used for testing by issuing the following command:

```
bowl < testscores
```

Creating the Script Files

All the files we will be creating will be in the directory located under $HOME/ab_Assignment_13B (where a is the students first letter of their first name and b is the first letter of their last name.)

Your program will prompt for the left lane team number, right lane team number, 15 game scores for the left lane and 15 game scores for the right lane (5 bowlers * 3 games)/

Here are some helpful hints in creating this program:

- The program should prompt for the team on lane 1 and lane 2.
  Here is a sample of one of the prompts I used:

  ```
  echo -e "Enter the team number on Lane 1: \c"
  read teamleft
  ```

- The program should select all the bowlers on the left lane and the right lane. Here is a snippet of my program:

  ```
  grep "^${teamleft}|"  bowlers > leftbowlers
  ```

- The program should determine the team name. Here is a snippet of my program:

  ```
  teamleftname=`grep "^${teamleft}|" teams | cut -f2 -d"|"`
  ```

  The program will need to loop through each of the bowlers on the left lane. Here is a snippet of my program:

  ```
  i=1
  while read -r line
  do
          lname=`echo "$line" | cut -f2 -d"|"`
          lhcp=`echo "$line" | cut -f3 -d"|"`
          ltpins=`echo "$line" | cut -f4 -d"|"`

          case $i in
                  1) lname1="$lname"
                          lhcp1="$lhcp"
                          ltpins1="$ltpins" ;;

                  <<< Add code here >>>
          esac

          i=`expr $i +
  done < leftbowlers
  ```

- The program should prompt for the scores of left lane and right lane.
  Here is a snippet of my program for the left lane:

```
i=1
while [ $i -le 5 ]
do
      case $i in
            1) echo -e "\nEnter scores for ${lname1}" ;;
            2) echo -e "\nEnter scores for ${lname2}" ;;
            3) echo -e "\nEnter scores for ${lname3}" ;;
            4) echo -e "\nEnter scores for ${lname4}" ;;
            5) echo -e "\nEnter scores for ${lname5}" ;;
      esac

      echo -e "\tEnter Game 1 score: \c"
      read g1

      echo -e "\tEnter Game 2 score: \c"
      read g2

      echo -e "\tEnter Game 3 score: \c"
        read g3

      total=`expr $g1 + $g2 + $g3`

      case $i in
            1) lp1g1=$g1 ; lp1g2=$g2 ; lp1g3=$g3 ; lp1tot=$total ;;
```

- To display the heading, I have included the following snippet:

```
today=`date +%D`
echo -e "\n\n         BEDROCK MONDAY NIGHT MEN'S LEAGUE                         ${today}"
printf "TEAM:  %-16s (%2d)   LANE:   1    TEAM:  %-16s (%2d)   LANE  2\n" ${teamleftname}
${teamleft} ${teamrightname} ${teamright}
echo "-----------------------------------------------------------------------------------"
echo "HCP      NAME         G1   G2   G3  TOT     HCP      NAME         G1   G2   G3  TOT"
echo "-----------------------------------------------------------------------------------"
```

- Since arrays can be difficult in bash, I displayed my scores as follows (notice the curly braces around the variable and the usage of printf):

```
while [ $i -le 5 ]
do
   case $i in
      1) printf "%3d %-15s  " ${lhcp1} ${lname1}
         printf "%3d  %3d  %3d  %3d     " ${lp1g1} ${lp1g2} ${lp1g3}
${lp1tot}

         printf "%3d %-15s  " ${rhcp1} ${rname1}
         printf "%3d  %3d  %3d  %3d\n" ${rp1g1} ${rp1g2} ${rp1g3}
${rp1tot}

         lg1tot=`expr $lg1tot + $lp1g1`
         lg2tot=`expr $lg2tot + $lp1g2`
         lg3tot=`expr $lg3tot + $lp1g3`
         ltothcp=`expr $ltothcp + $lhcp1`
```

```
               rg1tot=`expr $rg1tot + $rp1g1`
               rg2tot=`expr $rg2tot + $rp1g2`
               rg3tot=`expr $rg3tot + $rp1g3`
               rtothcp=`expr $rtothcp + $rhcp1` ;;
```

- To add the scores and the handicap, my program did the following:

```
leftscore1=`expr $ltothcp + $lg1tot`
leftscore2=`expr $ltothcp + $lg2tot`
leftscore3=`expr $ltothcp + $lg3tot`
lefttotal=`expr $leftscore1 + $leftscore2 + $leftscore3`

rightscore1=`expr $rtothcp + $rg1tot`
rightscore2=`expr $rtothcp + $rg2tot`
rightscore3=`expr $rtothcp + $rg3tot`
righttotal=`expr $rightscore1 + $rightscore2 + $rightscore3`
```

- To display the final line determining which team won, my program did the following:

```
if [ $leftscore1 -ge $rightscore1 ]
then
        winleft1="X"
fi

if [ $rightscore1 -ge $leftscore1 ]
then
        winright1="X"
fi
```

<<< Add code for game 2, game 3 and the total >>>

```
echo " "
printf "          TOTAL     %4d %4d %4d %4d" $leftscore1 $leftscore2
$leftscore3 $lefttotal
printf "              TOTAL      %4d %4d %4d %4d\n" $rightscore1 $rightscore2
$rightscore3 $righttotal
echo -e "      GAMES WON      $winleft1    $winleft2    $winleft3
$winlefttot        \c"
echo -e "      GAMES WON        $winright1    $winright2    $winright3
$winrighttot"
~
```

Assignment 13B
Linux Scripting

Program Output

The expected program output is shown below. The Team Names and GAMES WON should not be hard coded.

```
         BEDROCK MONDAY NIGHT MEN'S LEAGUE                         05/03/22
TEAM:  Yahoos           ( 4)   LANE:   1    TEAM:  Snappers         ( 9)   LANE  2
-----------------------------------------------------------------------------------
HCP      NAME         G1   G2   G3  TOT     HCP      NAME         G1   G2   G3  TOT
-----------------------------------------------------------------------------------
 46 R.Sundararaman   169  144  168  481      22 J.Hickey         168  167  204  539
 43 R.Kreis          116  165  168  449      26 T.Miller         175  170  204  549
 34 P.Coogan         171  132  180  483      58 L.Pottebaum      140  161  152  453
  9 W.Beach          203  246  207  656      38 M.Malnik         148  160  164  472
 36 R.Javorek        178  150  200  528       2 G.Daly           234  214  220  668

          TOTAL     1005 1005 1091 3101              TOTAL      1011 1018 1090 3119
      GAMES WON                X                 GAMES WON        X    X         X
```

Sample Data

All the necessary input data is provided in the testscores file.

Program Notes

Students will need to create a file:

bowl

The Linux bowl file is required to have a programmer header containing the following:

```
#-------------------------------------------------------------------------------
# Assignment:    Assignment 13B
#
# Program Name:  fileName
#                where fileName is the name of the file
#
# Purpose:       The purpose of this program is
#                Fill in the purpose of this program here.
#
# Author:        Your Name
# Course:        231CIS109.950
#
# Created:       Today’s Date in the Month Day, Year (4 digit) format
#
#------------------------------------------------------------------------------
```

Submitting Instructions

- You will need to download your **ab_Assignment_13B** folder to your computer and zip the folder before submitting it on Canvas.

Note: a is the students first letter of their first name and b is the first letter of their last name.