

UNIX510 - Unix BASH Shell Scripting

Assignment 2: Specifications

Objective: To gain experience with elementary scripting techniques.

Value: 20% of final course mark

TASK#1: 10%

Specifications: You are to write a Bash shell script(*store it in ~/Asn2_yourname/pathDisplay.bash*), on the Matrix server, which will show the permissions of a directory and all the directories in its path. Write a utility called **pathdisplay** that satisfies the following requirements:

1. **Usage: pathdisplay [dir-name]**
2. **pathdisplay** displays the permissions of a directory and all the directories in the directory's path. **pathdisplay** will also show other information about the directories, as the cursor is moved up and down the list of directory names.
3. **pathdisplay** will accept one argument, the name of a directory, including the path if required.
4. If there is no argument specified, then the current directory will be the default.
5. If there is more than one argument specified, there will be an error message and the script terminated (exit status 1).
6. If the specified directory is not a valid, existing directory, there will be an error message and the script terminated (exit status 1).
7. **Please see the examples for the exact display format, and match this format as closely as possible.** Note that the screen is cleared before the output is displayed. There are 3 spaces between the permission groups, and between the permission groups and the directory name. There is one space between the permissions within a permission group.
8. Ignore the situation of a very deep directory structure, which would require a multi-screen display.
9. Ignore the situation of a very long directory name, which would wrap to the next line.
10. On initial display, the cursor will be on the first character of the specified (or default) directory name. Information for this directory name will also be displayed.
11. There is a line showing the valid keys accepted during execution of the pathdisplay command, on the second line from the bottom of the terminal window, regardless of the terminal window size used.
12. After the initial display, the valid keys available are the following. Note that these are single-key options with no "enter" key required after any of them, and that they are lower case:
 - o 'u' (up) will move the cursor to the first character of the directory name above the current line. Also, information for the new directory level will be displayed,

instead of the previous directory level. This key will have no affect if the cursor is already on the top-most filename, which will be the root.

- 'd' (down) will move the cursor to the first character of the directory name below the current line. Also, information for the new directory level will be displayed, instead of the previous directory level. This key will have no affect if the cursor is already on the bottom-most directory name, which will be the specified (or default) directory.
 - "q" will terminate the **pathdisplay** script. Note that the screen is not cleared, and the command prompt will appear on the bottom line of the terminal window.
13. Any keys entered, other than the valid keys above, will be ignored.
 14. "Flashing" of the screen, during cursor movement, is OK. For example, when the cursor is moved up or down, you may redraw the entire screen. If you wish to eliminate the flashing, making the program more pleasant to use, you may do so, but it will slightly increase the complexity of your program.
 15. If **pathdisplay** produced full-screen output, it must end with an exit status of 0 (zero). If there was an error, then there is no full-screen output, and the exit status should be 1 (one).
 16. Ensure that your script is self-contained in one script file, called **pathdisplay**. Limit "sed" and "awk" to single-line commands with no separate script files required.
 17. Don't do any argument verification or any processing in **pathdisplay** other than that specified above.

NOTE:

1. Ensure that your output format matches the following **exactly** as the screenshots in the next page
2. **Important: The files in the sample outputs are available, please use them for testing. Your output should duplicate the outputs EXACTLY as shown (though some links, sizes, and date-time stamps might change), or marks will be liberally redirected to /dev/null**

Assignment 2/TASK-1: pathdisplay Sample Screen Shots

- Here are samples of the error messages (directory dir1 exists, dir2 does not exist). The screen size is 80 columns by 24 rows:

```
. ==> ./pathdisplay dir2
. dir2 is not a valid directory name
. ==> ./pathdisplay dir1 dir2
. Usage: pathdisplay [ dir-name ]
. ==>
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
```

- Entering `./pathdisplay ~unx510/sample.dir1/testdir1` produced the following. Note that the cursor is over the "t" of "testdir1":

```
. Owner      Group      Other      Filename
. -----
.
.
.  r w x    r - x    r - x    /
.
.  r w x    r - x    r - x    home
.
.  r w x    - - x    - - x    unx510
.
.
```

```

. r w x    r - x    r - x    sample.dir1
.
. r w x    r - x    r - x    testdir1
.   Links: 4  Owner: unx510  Group: users  Size: 229
Modified: Feb 22 2015
.
.
.
.
.
.
.
.
.
.
Valid commands:  u(p)  d(own)  q(uit)
.

```

Hitting **d** and **x** had no effect. Hitting **u** (with no **Enter**) produced the following:

```

. Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
. r w x    r - x    r - x    /
.
. r w x    r - x    r - x    home
.
. r w x    - - x    - - x    unx510
.
. r w x    r - x    r - x    sample.dir1
.   Links: 4  Owner: unx510  Group: users  Size: 137
Modified: Apr 7 2019
. r w x    r - x    r - x    testdir1
.
.
.
.
.
.
.
.
.
.

```

```
.
. Valid commands:  u(p)  d(own)  q(uit)
.
```

```
. Hitting u again produced the following:
```

```
. Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
.  r w x      r - x      r - x      /
.
.  r w x      r - x      r - x      home
.
.  r w x      - - x      - - x      unx510
.  Links: 24  Owner: unx510  Group: users  Size: 4096
Modified: Jan 4 00:56
.  r w x      r - x      r - x      sample.dir1
.
.  r w x      r - x      r - x      testdir1
.
.
.
.
.
.
.
.
.
.
. Valid commands:  u(p)  d(own)  q(uit)
.
```

```
. Hitting u again produced the following:
```

```
. Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
.  r w x      r - x      r - x      /
.
.  r w x      r - x      r - x      home
.  Links: 11322  Owner: root  Group: root  Size:
286720  Modified: Feb 27 09:11
.  r w x      - - x      - - x      unx510
```



```
.
Valid commands:  u(p)  d(own)  q(uit)
.
```

• Hitting **u** again had no effect. Hitting **d** four (or more) times produced the following:

```
. Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
.  r w x      r - x      r - x      /
.
.  r w x      r - x      r - x      home
.
.  r w x      - - x      - - x      unx510
.
.  r w x      r - x      r - x      sample.dir1
.
.  r w x      r - x      r - x      testdir1
.  Links: 4   Owner: unx510   Group: users   Size: 229
Modified: Feb 22 2015
.
.
.
.
.
.
.
.
.
Valid commands:  u(p)  d(own)  q(uit)
.
```

• Hitting **q** produced the following. Note that the prompt is displayed after the bottom line of the **pathdisplay** display:

```
. Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
.  r w x      r - x      r - x      /
.
.  r w x      r - x      r - x      home
.
.  r w x      - - x      - - x      unx510
```

```

.
.  r w x    r - x    r - x    sample.dir1
.
.  r w x    r - x    r - x    testdir1
.  Links: 4  Owner: unx510  Group: users  Size: 229
Modified: Feb 22 2015
.
.
.
.
.
.
.
.
.
.
.  Valid commands:  u(p)  d(own)  q(uit)
.  ==>

```

- Increasing the screen size 32 rows by 96 columns, then entering `cd ~unx510/sample.dir1/testdir1` followed by `./pathdisplay` produced the following:

```

.  Owner      Group      Other      Filename
.  - - - - -  - - - - -  - - - - -  - - - - -
.
.  r w x    r - x    r - x    /
.
.  r w x    r - x    r - x    home
.
.  r w x    - - x    - - x    unx510
.
.  r w x    r - x    r - x    sample.dir1
.
.  r w x    r - x    r - x    testdir1
.  Links: 4  Owner: unx510  Group: users  Size: 229
Modified: Feb 22 2015
.
.
.
.
.
.
.
.

```


Valid commands: u(p) d(own) q(uit)

- Hitting **q** produced the following. Note that the prompt is still displayed after the bottom line of the **pathdisplay** display:

Owner	Group	Other	Filename
r w x	r - x	r - x	/
r w x	r - x	r - x	home
r w x	- - x	- - x	unx510
r w x	r - x	r - x	sample.dir1
r w x	r - x	r - x	testdir1

Links: 4 Owner: unx510 Group: users Size: 229
Modified: Feb 22 2015

Valid commands: u(p) d(own) q(uit)

 \Rightarrow

TASK#2: 10%

Write a utility called **showDir** that satisfies the following requirements:

1. **Usage: showDir [dir-name]**
2. **showDir** displays the permission of a directory and all the directories in the directory's absolute path. **showDir** will also show other information about the directories, as the cursor is moved up and down the list of directory names. **showDir** will also facilitate changing permission of the directories, assuming that you are the owner of the appropriate directory or the superuser.
3. **showDir** will accept one argument, the name of a directory, including the path if required.
4. If there is no argument specified, then the current directory will be the default.
5. If there is more than one argument specified, there will be an error message and the script terminated (exit status 1) with no full-screen display.
6. If the specified Dir_Name is not a valid, existing directory, there will be an error message and the script terminated (exit status 1) with no full-screen display.
7. **Please see the examples for the exact display format, and match this format as closely as possible.** Note that the screen is cleared before full-screen output is displayed. There are 3 spaces between the permission groups, and between the permission groups and the Dir_Name. There is one space between the permission within a permission group.
8. Ignore the situation of a very deep directory structure, which would require a multi-screen display.
9. Do not let output lines wrap to the next line, ensure that all lines are truncated to the width of the screen size.
10. Once your program is running, you do NOT have to take into account changes in screen size. You also do NOT have to take into

account changes performed from another process, such as changes in permission of the displayed directories or changes in the displayed directory structure.

11. On initial display, the cursor will be on the first character of the specified (or default) Dir_Name. Information for this directory will also be displayed.
12. There are three lines showing the valid keys accepted during execution of the **showDir** command, on the fourth, third, and second lines from the bottom of the terminal window, regardless of the terminal window size used.
13. After the initial display, the valid keys available are the following. Note that there should be no "enter" key required after any of them, and that they are lower case:
 - "k" will move the cursor to the line containing the Dir_Name above the current line. Also, information for the new directory level will be displayed, instead of the previous directory level. This key will have no affect if the cursor is already on the top-most Dir_Name, which will be the root.
 - "j" will move the cursor to the line containing the Dir_Name below the current line. Also, information for the new directory level will be displayed, instead of the previous directory level. This key will have no affect if the cursor is already on the bottom-most Dir_Name, which will be the specified (or default) directory.
 - "h" will move the cursor to the field immediately to the left of the current field. This key will have no affect if the cursor is already on the "r" permission position for "owner". The fields are the nine permission fields, and the Dir_Name.
 - "l" will move the cursor to the field immediately to the right of the current field. This key will have no effect if the cursor is already on the right-most field, which will be the Dir_Name.
 - "r" will set read permission to the current Dir_Name, for owner, group, or other, depending on which field the cursor is on. If the cursor is on any field other than one of the three "r" fields, then the "r" key will be ignored.
 - "w" will set write permission to the current Dir_Name, for owner, group, or other, depending on which field the cursor is on. If the cursor is on any field other than one of the three "w" fields, then the "w" key will be ignored.
 - "x" will set execute permission to the current Dir_Name, for owner, group, or other, depending on which field the cursor is on. If the

cursor is on any field other than one of the three "x" fields, then the "x" key will be ignored.

- "-" will remove a permission to the current Dir_Name, for owner, group, or other, depending on which field the cursor is on. If the cursor is on any field other than one of the nine permission fields, then the "-" key will be ignored.
 - "q" will terminate the **showDir** script. Note that the screen is not cleared, and the command prompt will appear on the bottom line of the terminal window.
 - <Ctrl>-c will have exactly the same effect as "q"
14. Any keys entered, other than the valid keys above, will be ignored.
 15. If the user doesn't have authority to change a permission, then the "r", "w", "x", and "-" keystrokes will be ignored. If the permission could be changed, then the output display will reflect the new permission.
 16. When moving down or up, the cursor will be on the same field on the new line as on the current one.
 17. If **showDir** produced full-screen output, it must end with an exit status of zero. If there was an error, then there is no full-screen output, and the exit status should be one.
 18. Ensure that your script is self-contained in one script file, called **showDir**
 19. Don't do any argument verification or any processing in **showDir** other than that specified above. If you're interested in adding more capabilities, please do so in a different script.
 20. "Flashing" of the screen, during cursor movement, is OK. For example, when the cursor is moved up or down, you may redraw the entire screen.

(BONUS mark) you can eliminate the flashing, and making the program more pleasant to use.

NOTE:

1. Ensure that your output format matches the following **exactly** as the screenshots in the next page
2. **Important: The files in the sample outputs are available, please use them for testing. Your output should duplicate the outputs EXACTLY as shown (though some links, sizes, and date-time stamps might change), or marks will be liberally redirected to /dev/null**

Assignment 2/TASK-2: showDir Sample Screen Shots

- After changing to the sample directory, there is a display of the current path, a listing of the current directory, and samples of the required error messages (file file1 and directory dir1 exist, file2 and dir2 do not exist). No other error messages should be produced, even if they "roll off" the screen. The screen size is 80 columns by 24 rows:

```
. ==> cd ~unx510/sample.dir1/testdir1/this*
. ==> pwd
. /home/unx510/sample.dir1/testdir1/this.is.an
.   .incredibly.long.directory.name.to.s
. ee.what.will.happen.with.incredibly.long.dir
.   ectory.names
. ==> ls -l
. total 4
. -rw----- 1 unx510 users 445 Mar 13 02:24
.   cars
. drwxr-xr-x 2 unx510 users 15 Mar 13 02:33
.   dir1
. -rw----- 1 unx510 users 0 Mar 14 2019
.   file1
. ==> showDir file2
. file2 is not a valid directory name
. ==> showDir file1 file2
. Usage: showDir [ dir-name ]
. ==> showDir dir2
. dir2 is not a valid directory name
. ==> showDir dir1 dir2
. Usage: showDir [ dir-name ]
. ==>
.
.
.
.
.
.
.
```

- Entering **showDir dir1** produced the following. Note that the cursor is over the "d" of "dir1":

```

• Owner      Group      Other      Dir_Name
• -----
•
• r - x      r - x      r - x      /
•   w x      r - x      r - x      home
• r
•   w x      - - x      - - x      unx510
• r
•   w x      r - x      r - x      sample.dir1
• r
•   w x      r - x      r - x      testdir1
• r
•
• r w x      r - x      r - x
  this.is.an.incredibly.long.directory.name.to
  .see.what.wi
•
• r w x      r - x      r - x  ddir1
•   Links: 2  Owner: unx510  Group: users
  Size: 15   Modified: Mar 13 02:33
•
•
•
• Valid keys: k (up), j (down): move between
  Dir_Names
•               h (left), l (right): move
  between permission
•               r, w, x, -: change permission;
  q: quit
•

```

- Hitting **j** and **x** had no effect. Hitting **k** twice (with no **<Enter>**) produced the following:

```

• Owner      Group      Other      Dir_Name
• -----

```

```

.
.  r - x    r - x    r - x    /
.    w x    r - x    r - x    home
.  r
.    w x    - - x    - - x    unx510
.  r
.    w x    r - x    r - x    sample.dir1
.  r
.
.  r w x    r - x    r - x    testdir1
.    Links: 4  Owner: unx510  Group: users
.    Size: 229  Modified: Feb 22 2015
.  r w x    r - x    r - x
.    this.is.an.incredibly.long.directory.name.to
.    .see.what.wi
.
.  r w x    r - x    r - x    dir1
.
.
.
.
.
.
.  Valid keys: k (up), j (down): move between
.    Dir_Names
.                h (left), l (right): move
.    between permission
.                r, w, x, -: change permission;
.    q: quit
.

```

• Hitting **l** had no effect. Hitting **h** five times produced the following:

```

.  Owner      Group      Other      Dir_Name
.  -----
.
.  r - x      r - x      r - x      /
.
.  r w x      r - x      r - x      home

```



```

.
.  r w x    - - x    - - x    unx510
.    w x    r - x    r - x    sample.dir1
.  r
.    w x    r - x    r - x    testdir1
.  r
.    Links: 4  Owner: unx510  Group: users
.    Size: 229  Modified: Feb 22 2015
.  r w x    r - x    r - x
.    this.is.an.incredibly.long.directory.name.to
.    .see.what.wi
.
.  r w x    r - x
.
.
.
.
.
.
.  Valid keys: k
.    Dir_Names
.
.    h
.    between permission
.
.    r, w, x, -: change permission;
.    q: quit
.

```

. Hitting x had no effect. Hitting w produced the following:

```

.  Owner      Group      Other      Dir_Name
.  -----
.
.  r - x      r - x      r - x      /
.    w x      r - x      r - x      home
.  r
.    w x      - - x      - - x      unx510
.  r
.    w x      r - x      r - x      sample.dir1
.  r

```

```

.
.  r w x   r w x   r - x   testdir1
.    Links: 4   Owner: unx510   Group: users
.    Size: 229   Modified: Feb 22 2015
.  r w x   r - x   r - x
.    this.is.an.incredibly.long.directory.name.to
.    .see.what.wi
.
.  r w x   r - x   r - x   dir1
.
.
.
.
.
.
.  Valid keys: k (up), j (down): move between
.    Dir_Names
.
.    h (left), l (right): move
.    between permission
.
.    r, w, x, -: change permission;
.    q: quit
.

```

• Hitting **k** four times produced the following:

```

.  Owner      Group      Other      Dir_Name
.  -----
.
.  r - x   r - x   r - x   /
.    Links: 18   Owner: root   Group: root   Size:
.    4096   Modified: Aug 21 2019
.  r w x   r - x   r - x   home
.    w x   - - x   - - x   unx510
.  r
.    w x   r - x   r - x   sample.dir1
.  r
.    w x   r w x   r - x   testdir1
.  r
.

```

```

. r w x   r - x   r - x
  this.is.an.incredibly.long.directory.name.to
  .see.what.wi
.
. r w x   r -
.
.
.
.
.
. Valid keys:
  Dir_Names
.
  between permission
.           r, w, x, -: change permission;
q: quit
.

```

- Hitting **w** and **k** had no effect. Hitting **l** twice, then hitting **j** five times, then hitting **-** produced the following:

```

. Owner      Group      Other      Dir_Name
. -----
.
. r - x      r - x      r - x      /
.   w x      r - x      r - x      home
. r
.   w x      - - x      - - x      unx510
. r
.   w x      r - x      r - x      sample.dir1
. r
.   w x      r w x      r - x      testdir1
. r
.   w x      r - x      - - x
. r
  this.is.an.incredibly.long.directory.name.to
  .see.what.wi

```

```

•   Links: 3   Owner: unx510   Group: users
    Size: 43   Modified: Mar 22 2015
•   r w x     r -
•
•
•
•
•
•   Valid keys:
    Dir_Names
•
    between permission
•           r, w, x, -: change permission;
    q: quit
•

```

• Hitting **h** six times produced the following:

```

•   Owner      Group      Other      Dir_Name
•   -----
•
•   r - x      r - x      r - x      /
•   w x        r - x      r - x      home
•   r
•   w x        - - x      - - x      unx510
•   r
•   w x        r - x      r - x      sample.dir1
•   r
•   w x        r w x      r - x      testdir1
•   r
•
•   r w x      r - x      - - x
    this.is.an.incredibly.long.directory.name.to
    .see.what.wi
•   Links: 3   Owner: unx510   Group: users
    Size: 43   Modified: Mar 22 2015
•   r w x      r - x      r - x      dir1
•

```

```

.
.
.
. Valid keys: k (up), j (down): move between
  Dir_Names
.           h (left), l (right): move
  between permission
.           r, w, x, -: change permission;
  q: quit
.

```

- Hitting **h** had no effect. Hitting **q** produced the following (Note that the prompt is displayed after the bottom line of the **showDir** display:

```

. Owner      Group      Other      Dir_Name
. -----
.
. r - x      r - x      r - x      /
.   w x      r - x      r - x      home
. r
.
. r w x      - - x      - - x      unx510
.   w x      r - x      r - x      sample.dir1
. r
.   w x      r w x      r - x      testdir1
. r
.   w x      r - x      - - x
. r
. this.is.an.incredibly.long.directory.name.to
. .see.what.wi
.   Links: 3  Owner: unx510  Group: users
.   Size: 43  Modified: Mar 22 2015
. r w x      r - x      r - x      dir1
.
.
.
.
.

```

```

. Valid keys: k (up), j (down): move between
  Dir_Names
.             h (left), l (right): move
  between permission
.             r, w, x, -: change permission;
  q: quit
. ==>

```

- Increasing the screen size to 32 rows by 96 columns, then entering **showDir** produced the following:

```

. Owner      Group      Other      Dir_Name
. -----
.
. r - x      r - x      r - x      /
.
. r w x      r - x      r - x      home
.
. r w x      - - x      - - x      unx510
.
. r w x      r - x      r - x      sample.dir1
.
. r w x      r w x      r - x      testdir1
.
. r w x      r - x      - - x
. this.is.an.incredibly.long.directory.name.to
. .see.what.will.happen.with.i
.   Links: 3  Owner: unx510  Group: users
.   Size: 43  Modified: Mar 22 2015
.
.
.
.
.
.
.
.
.

```

```

.
.
.
.
.
. Valid keys: k (up), j (down): move between
  Dir_Names
.           h (left), l (right): move
  between permission
.           r, w, x, -: change permission;
  q: quit
.

```

• Hitting the keys **hhhrkhh-** produced the following:

```

. Owner      Group      Other      Dir_Name
. -----
.
. r - x      r - x      r - x      /
.
. r w x      r - x      r - x      home
.
. r w x      - - x      - - x      unx510
.
. r w x      r - x      r - x      sample.dir1
.
. r w x      r - x      r - x      testdir1
.   Links: 4  Owner: unx510  Group: users
.   Size: 229  Modified: Feb 22 2015
. r w x      r - x      r - x
. this.is.an.incredibly.long.directory.name.to
. .see.what.will.happen.with.i
.
.
.
.
.
.

```

```

.
.
.
.
.
.
.
.
.
.
.
Valid keys: k (up), j (down): move between
Dir_Names
.
          h (left), l (right): move
between permission
.
          r, w, x, -: change permission;
q: quit
.

```

- Hitting **q** produced the following. Note that the prompt is still displayed after the bottom line of the **showDir** display:

```

. Owner      Group      Other      Dir_Name
. -----
.
. r - x      r - x      r - x      /
.   w x      r - x      r - x      home
. r
.   w x      - - x      - - x      unx510
. r
.   w x      r - x      r - x      sample.dir1
. r
.   w x      r - x      r - x      testdir1
. r
.   Links: 4  Owner: unx510  Group: users
Size: 229  Modified: Feb 22 2015
. r w x      r - x      r - x
this.is.an.incredibly.long.directory.name.to
.see.what.will.happen.with.i

```



```
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
.
Valid keys: k (up), j (down): move between
Dir_Names
.           h (left), l (right): move
between permission
.           r, w, x, -: change permission;
q: quit
```

$$\Rightarrow$$

Assignment 2: Hints and Tips

Here are some hints that may help you with this assignment:

1. There are several aspects to this assignment that may be new to you. It's probably easiest to approach this kind of assignment by using a step-by-step process. During each step, you can better understand a new scripting facility, and this might help you understand the processing required in the next step.
2. Here is one possible approach, although there are other approaches which are equally valid:
 - a. Take care of argument handling and error messages.
 - b. Calculate the number of directory levels to be displayed. This will be used to make sure we don't move down too far. Also, create another variable which keeps track of the current level for the additional information to be displayed.
 - c. Create the initial display, ignoring user input, and ignoring the extra information required for the "current" level. Include the "Valid commands" line near the bottom of the screen.
 - d. Set up a loop that does the following until a "q" is entered. Don't worry yet about "enter" being required for user input:
 - i. Inside the loop, display the screen of information using your code from the previous step. Keep track of the level being displayed so that you can use the current level variable to determine when to display the additional information.
 - ii. At the end of the loop, get user input. If "u" or "d" is entered, update the current level variable, but only if this wouldn't produce a level too high or too low for the number of levels being displayed.
 - e. Add the ability to read user input without "enter" being required.
3. Note that you have the appropriate permissions to access the test directory used for the examples. As part of your testing, I suggest that you use the same test directory, and see if your script will exactly duplicate the format of the screen shots. However, the number of links, sizes, and modified dates may have changed since the screen shots were taken.

Assignment 2: Marking Scheme

The marking scheme for Assignment 2 will be as follows:

1. If your script can't run the assignment examples correctly, with output **EXACTLY** duplicating the sample output format, then the assignment may be a resubmit.
2. Marks will be deducted for errors in the following areas:
 - a. Comments as specified: course, family name, etc.
 - b. Output format as specified:
 - heading and "Valid commands" lines
 - correct directories displayed
 - alignment as specified
 - cursor displayed and moved correctly
 - detailed info displayed and moved correctly
 - handles different window sizes correctly
 - c. Error messages as specified, and redirected properly.
 - d. Proper exit status, terminal characteristics reset to normal.
 - e. Handles pathnames at any directory level, including relative, relative-to-home, and absolute pathnames.
 - f. Handles "u", "d", and "q" properly, and ignores other "normal" keys. Ignore handling of "Ctrl-C" for now.
 - g. Handles attempts to move up or down too far.
 - h. Uses /tmp directory and \$\$ for temporary files, and temporary files deleted (if used).

Assignment Submission

1. Submitted assignments **MUST** contain the student's **OWN** solution to the assignment.
2. Ensure that your script is self-contained in one file, called **pathDisplay**
3. At the beginning of your script, include comments with the following information formatted neatly, and reflecting accurate information for you.

```
# Assignment 2
# Course:                UNX510
# Family Name:           Student
# Given Name:            John
# Student Number:        123-456-789
# Login name:             jstudent
# Due Date:              MMM DD, YYYY
#
# 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.
```

What to Submit:

1) Please submit **the script** that you developed during this Assignment.

2) Please submit a **recorded-video** which explains how you did the lab and demonstrate your solutions. This will give everyone the opportunity to present their solution. Please record a video (2~10 minutes) with the following contents:

- Introduce yourself
- Show/demonstrate how your assignment works.
- Explain your code (walkthrough), how you design it (a quick/detailed walk-through of the programming code , running the commands in Matrix)
- Speak about challenges that you faced during this Assingment.
- Evaluate yourself. Have you implemented all requirements of the lab? how do you evaluate yourself out of 10 for this Assingment?

NOTE: You can record the video using some screen-capture software (like OBS :

<https://obsproject.com/>) or your cellphone. To submit the video:

- You can upload the video on the youtube (you may make it unlisted) and submit the link here.
- You can also directly upload the video to the BB.