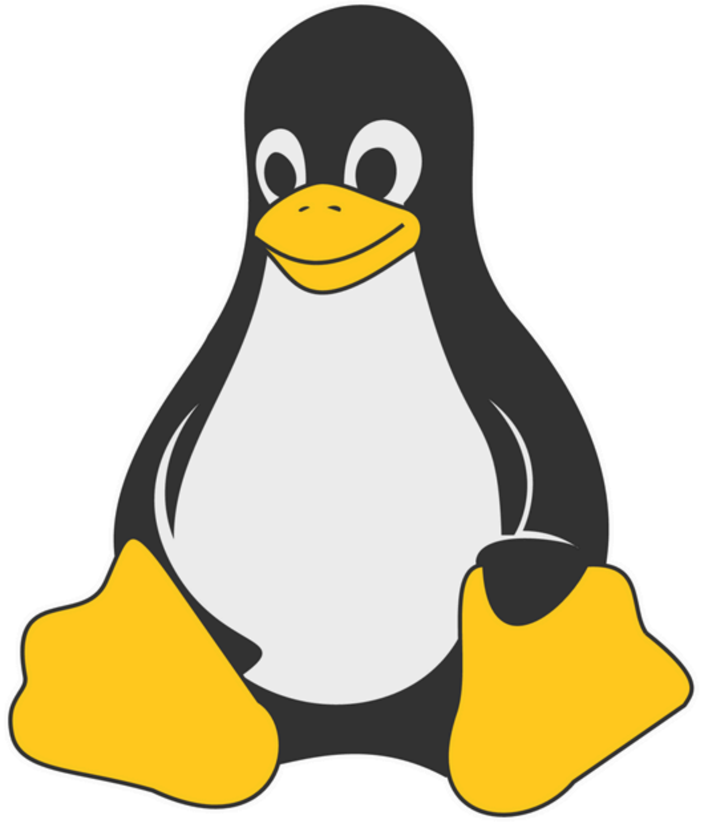

Ex - 04

1300

Bash Scripting; Attach all .sh files

Ref: Bash Scripting and Shell Programming (Linux Command Line)-udemy-Jason Cannon

Exercise 1

100

Write a shell script that prints "Shell Scripting is Fun!" to the screen.

Hint 1: Remember to make the shell script executable with the “chmod” command.

Hint 2: Remember to start your script with a shebang!

Exercise 2

Modify the shell script from exercise 1 to include a variable. The variable will hold the contents of the message "Shell Scripting is Fun!".

Exercise 3

Write a shell script to check to see if the file "/etc/shadow**" exists. If it does exist, display** "Shadow passwords are enableed."

Next, check to see if you can write to the file. If you can, display "You have permission to edit /etc/shadow.**" If you cannot, display "**You do NOT have permission to edit /etc/shadow**."**

Exercise 4

Write a shell script that displays "man", "bear", "pig", "dog", "cat", and sheep **on the screen with each appearing on a separate line. Try to do this in as few lines as possible.**

Hint: Loops can be used to perform repetitive tasks.

Exercise 5

Write a shell script that prompts the user for the name of a file or directory and reports if it is a regular file, a directory, or other type of file.

Also perform an ls command against the file or directory with the long listing option.

Exercise 6

Modify the previous script so that it accepts the file or directory name as an argument instead of prompting the user to enter it.

Exercise 7

Modify the previous script to accept an unlimited number of files and directories as arguments.

Hint: You'll want to use a special variable.

Exercise 8

Write a shell script that displays "This script will exit with a 0 exit status." **Be sure that the script does indeed exit with a 0 exit status.**

Exercise 9

Write a shell script with a function that displays the number of files in the present working directory. Name this function "file_count" and call it in your script. If you use a variable in your function, remember to make it a local variable.

Hint: The wc utility is used to count the number of lines, words, and bytes.

Exercise 10

Write a shell script that loops through the /etc/passwd file one line at a time. Prepend each line with a line number followed by a colon and then a space.

Example output:

```
1: root:x:0:0:root:/root:/bin/bash
2: daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3: bin:x:2:2:bin:/bin:/usr/sbin/nologin
4: sys:x:3:3:sys:/dev:/usr/sbin/nologin
```

Exercise 11

Write a shell script that renames all files in the current directory that end in ".jpg" to begin with today's date in the following format: YYYYMMDD. For example, if a picture of my cat was in the current directory and today was October 31, 2016 it would change name from "mycat.jpg" to "20161031mycat.jpg".

Hint: Look at the format options to the date command.
For "extra credit" make sure to gracefully handle instances where there are no ".jpg" files in the current directory. (Hint: Man bash and read the section on the nullglob option.)