Exercises for Linux for Power Users
Presented by UWB ACM
May 3rd 2019 - Spring 2019

Step 0: Clone the Repo

Run these commands:

```
cd ~
git clone https://www.github.com/UWB-ACM/Linux_MysteryBox
```

Step 1: tree

1. Run tree on your home folder. Is there a lot of output?
2. Run tree on `~/Linux_MysteryBox/tree`. What do you see?
3. Run tree on `/etc`. What do you see? A whole lot of stuff?

Consider whether tree is always the best way to understand how directories are structured.

Step 2: find

1. Run `find ~/Linux_MysteryBox`. Is there a lot of output? Do you have a good way of counting how many files/directories you found?
2. Try running find on the Mystery Box and looking for C++ files. How many did you find?

Step 3: wildcards

1. Rerun find on the Mystery Box and look for C++ files again, but use a wildcard this time. Were you able to find more files?
2. What would be the difference between using ? and * as wildcards in this example?

Step 4: redirects

1. Run:

   ```
   cd Linux_MysteryBox/find/extra/deploy
   python linux.py > out.txt
   cat < out.txt
   ```

   What do you see?

Step 5: piping

1. Run `fortune`. Then, run `fortune | cowsay`. What's the difference between these two commands?

Step 6: wc

1. Let's take another look at Step 2, Q1. Run `find ~/Linux_MysteryBox`, but this time, get a count of the number of lines using a pipe and `wc`.
    a. How many lines of output were returned?
    b. What's the length of the longest line of output?

Step 7: grep

1. Is there an executable C++ program in the Mystery Box? How do you know?
2. How many Java PSVM methods are in the Mystery Box? Which file is the main method in?

Step 8: if

1. Run the command on slide #60 in the console. What is your output?
   `if [[ ~ == $HOME ]]; then cd ~; echo "hi Toto"; else echo "not in Kansas"; fi`
2. Try converting this one-liner command into a script that you type into a file. (Use nano if you aren't familiar with another text editor: `nano script.sh` to create the file, type your text in, then `Ctrl+X y y` to save the file.)
    a. Run the script (type `bash script.sh`). Do you get any syntax errors? (If so, ask for help)
    b. What is your output?

Step 9: scripting

1. Create a script file for the script shown on Slide #65. Use these steps:
    a. `cd ~/Linux_MysteryBox/grep/cpp`
    b. `nano run.sh`
    c. Type in content from the slide, save, and close
    d. Run the script by typing `bash run.sh`
    e. What is your output?

Step ∞: explore your filesystem

1. How many items do you have in your home directory?
2. How many files do you have in your home directory?
3. How many configuration files are listed under /etc? (Hint: most config files using the "conf" extension)
4. How many java files are in your home directory? C++? Python? Text files? How many of these are in the top 2 directory levels? What else do you want to know about these files?