**Bash Scripting**

I want to create a program run from the Terminal that automates commands entered at the Terminal. In this example, the script will perform a software update, **sudo apt update** and **sudo apt dist-upgrade**

The advantage of the BASH script is that all I need to enter at the terminal is the command **sudo up**

STEPS

Go to the Terminal, navigate to the **usr/local** directory and add a folder named **bin**. To get to **usr/local**, I typed the following commands at the Terminal

**pwd** (This tells me where I am in the folder hierarchy)

**cd /** (This takes me to root)

**ls -l** (This shows me the -subfolders in the **root** folder)

**cd usr/local** (This takes me to the **usr/local** folder)
**mkdir bin** (This creates a folder called **bin** inside the bin folder)

From within the **bin** folder, type **sudo** **nano up**. This will open the **Nano Editor** to a newly created file called **up.** If you just type **nano up**, you will not have permission to save the file.

In the Nano Editor, type the following, noting that all lines that begin with **#** are commented out. They are your notes-to-self, not programming instructions. The exception is the first line, **#!/bin/bash.** It identifies the file as a bash script.

#!/bin/bash

# Simple script to update Ubuntu

sudo apt update

#Replace **sudo apt update** with **sudo apt update -yy** to answer Yes to all prompts

sudo apt dist-upgrade

# Replace sudo apt dist-upgrade with sudo apt dist-upgrade -yy to answer Yes to all prompts

exit

# The **exit** command closes the terminal

-----

To save the file, hold down the **CTRL** key and press **o** (This is a letter, not a zero)

Enter the name that you choose for the file. I named mine **up** (but the file name could be anything as long as it didn’t contain spaces)

If you did not save the **up** file to the **bin** folder mentioned earlier, you can move/copy it there using the GUI file system or the **Terminal**.

Close the Nano Editor by holding down the **Ctrl** key and pressing **x**

You must tell the computer that the **up** file should be executed. Do this by navigating to the same folder where the **up** file resides and in the **Terminal**, not the **Nano Editor**, type

**sudo** **chmod +x up** (where **up** is the name of your file)

You will be prompted for your password.

Type the password and then press **Enter**.

Note that when you type your password, it will not appear on the screen. The **up** command will simply run.

If you mistyped your password, the **Terminal** will tell you that the password is incorrect,

To run this bash script, open the **Terminal** and type **up** (or whatever you named your bash script).

To run the BASH script in the future, type **sudo up.** You will be prompted to enter your password. Do that and the BASH script will run. If you just type **up**, it will not.

Below is another bash script. Simply follow the same procedure as before but replace the text in the up file with the new text and follow the remaining procedures. This script displays your System Memory, Disk Usage, Uptime and runs Neofetch, a System report,

#!/bin/bash

#Check My System

echo"Memory:"

free -h

echo "Disk Usage"

df -h

echo "Uptime"

uptime

neofetch

exit

If you wanted greater simplicity, you could append the second script to the first, making sure to remove **exit** at the end of the first script.

Note: Most of the information here came from Joe Collin’s YouTube video at [https://www.youtube.com/watch?v=\_n5ZegzieSQ](https://www.youtube.com/watch?v=_n5ZegzieSQ%20)