Scripting Languages: Workshop 8

Pre-requisites:

- If you have not already done so, log into your Linux instance, start VS Code and navigate your way into the **ws8** folder.

- To complete these tasks you will need to place the following three (3) files into your current working directory (from *Canvas -> Week 8 -> Workshop Tasks -> ws_files.zip*):
    - **dirscan.sh**
    - **searchsort.sh**
    - **logfile.csv**
    - **process_log.sh**
    - **linux_log.csv**
    - **stockcalc_errors.sh**
    - **source.csv**

Write the Code

Task 1

1. Modify the **dirscan.sh** script provided so that it does two (2) things that it presently does not do:
    a. Allow the user to **optionally** nominate a specific type of file to search for, e.g. txt, pdf, png, rather than all file types as is its current default behaviour
    b. In addition to showing the appropriate formatted size of each matching file found, also display the total size of all matching files found combined, also appropriately formatted as Kb, Mb, etc
2. Before starting, make a copy of *dirscan.sh*, naming it **dirscan_b.sh**, and then make your modifications to this latter file
3. Also create a folder somewhere in your directory structure and populate it with a variety of files, e.g. shell files, text files, PDF files, images files, word docs, etc, and use this to test your script as you work on it
4. If you approach this task cleverly, you won't have to change much code at all, only needing to add a few lines of code in a couple of places to achieve the desired outcome; if you find yourself making major changes to the existing code – you're doing it wrong 😊

IMPORTANT NOTE: You may **not** copy and paste any of the existing code in *dirscan.sh* or any of its duplicates into any of your assessments. It use is strictly for this workshop only!

Comment the Code

Task 2

1. Download the files *searchsort.sh* and *logfile.csv* to your Linux development environment into the ws8 directory you created in Week 1. The *logfile.csv* file is the data source the *searchsort.sh* script will act upon.
2. Using only the lecture notes (Modules 1-8 inclusive) and what you have learned so far, fully comment the *searchsort.sh* script to explain:
    a. The purpose of the script
    b. Its inputs
    c. Its main processing logic
    d. Its outputs

*Do **not** run the script before you comment it. Complete the commenting in full and then run the script to see how much of your commenting was accurate.*

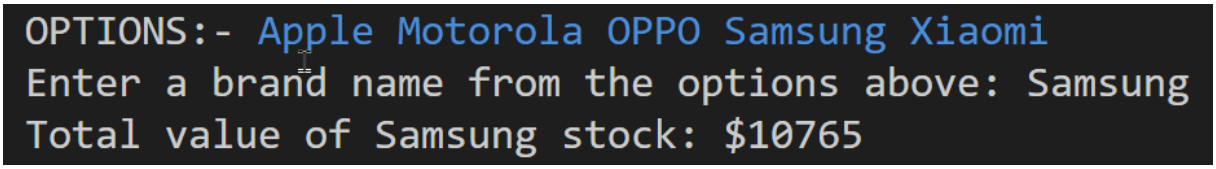
*Do **not** ask any AI tool to comment the script for you, otherwise you will learn nothing!!!*

Fix/Debug the Code

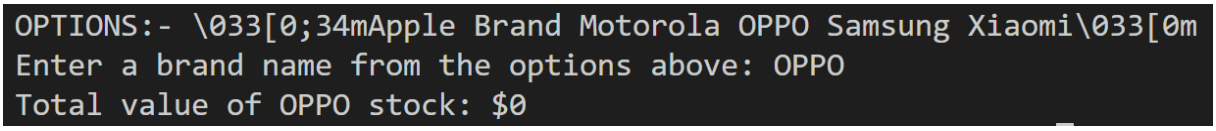
Task 3

1. Download the files *stockcalc_errors.sh* and *source.csv* to your Linux development environment into the ws8 directory you created in Week 1. The *source.csv* file is the data source the *stockcalc_errors.sh* script acts upon.
2. SCENARIO: A junior team member has come to you for help with a shell script they are writing. The *stockcalc_errors.sh* script they've written should be producing the results shown in the first image below, but rather, is producing the results in the second image below.

Results should be this…



However, results are this…



3. Examine the *stockcalc_errors.sh* script the junior team member has brought to you, and:
    a) Clearly identify the issues in the script that are causing the incorrect output
    b) Explain what they have done wrong and how to fix these issues
    c) Modify the script as required so that it produces the correct outputs as shown in the image on the left above; call this file *stockcalc_corrected.sh*

Use only the lecture notes (Modules 1-8 inclusive) and what you have learned so far to guide you in this process

Use comments to identify/document the issues within the script

*Do **not** ask any AI tool to tell you what the issues are or how to fix them, otherwise you will learn nothing!!!*

Critique the Code

Task 4

1. Download the files *process_log.sh* and *linux_log.csv* to your Linux development environment into the ws8 directory you created in Week 1. The *linux_log.csv* file is the data source the *process_log.sh* script acts upon.
2. SCENARIO: You asked a junior team member to write a shell script that reads through the records in standard report in.csv file, e.g. *linux_log.csv*, and displays records to the screen that meet the following criteria:
    a. The time is in the Time field is in the PM only
    b. The PID field is empty
    c. The Component field contains the string "kernel"

    The output to the terminal should be four columns as per the example below:

LineId	DateTime	Component	EventTemplate
1994	Jul-27@14:41	kernel	SELinux: Registering netfilter hooks.

3. The junior team member has now come to you with the script they've written (*process_log.sh*) and asked if you will approve it for production use. As the senior team member, would you approve this script for use in production? If not, record a short Panopto video explaining to the junior team member why you will not approve their script for production and outline what they need to do to make it acceptable for production use. Then send this video to your lecturer along with your version of the script (call it *process_log _better.sh*) to show the junior team member how you would have coded it as a learning opportunity for them.

*Do **not** ask any AI tool to critique the junior team members script for you or write a more efficient version, otherwise you will learn nothing!!!*

Task 5

1. **Copy** the ***.sh*** files you created in today's workshop to the *backups* directory using the same *_bu* name modification you used in last week's workshop
2. Navigate to the *backups* directory and make sure the copy procedure was successful

Conclude:

Close the *RDP connection* to your Azure VM (if you're using one) and then **power off** your VM in Azure.