

Atypon assignment report
Advanced Shell Scripting and Optimization Assignment
By Rayyan al Hourani

A-How to use the script

- 1- Usage of script: `./script.sh [directory_path] [extensions] [options]`
- 2- The first argument is the directory you want to search for files inside.
- 3- The second argument is used to choose the file extensions. Multiple extensions can be specified, separated by commas. Examples: 'txt' or 'txt,doc' (without quotation marks).
- 4- Optional arguments:
 - '-s': Generates a summary of files at the top of the report, including total files, total owners, and total file size.
 - '-f': Filters files based on minimum and maximum file size, minimum and maximum modified date, and file permissions.
 - For the '-f' option:
 - The script will prompt the user to enter the minimum and maximum file sizes.
 - Next, the user will be asked to enter the minimum and maximum dates in 'yyyy-mm-dd' format.
 - Finally, the user can enter the file permissions separated by commas. Examples: 'rw-r--r--' or 'rw-r--r--,drwxr-xr-x' (without quotation marks).
- 5- If no arguments are provided or if '-h' or '--help' is used, the script will display a help message explaining how to use it.
- 6- In case the user provides incorrect or missing arguments, the script will display an error message and guide the user on how to access the help message for proper usage.

B- how the script works

1- Argument Validation (line 5-64):

- Function (line 5-14): Displays the script's usage instructions.
- Help Message (line 17-20): Shows the help message if no arguments are provided or if '-h' or '--help' is used.
- Directory Validation (line 23-27): Checks if the directory is valid and displays an error message if it is not.
- Extension Validation (line 30-35): Checks if the user entered the extensions correctly and shows an error message if they are not in the right format.
- Argument Count (line 37-42): Checks if the user entered more than four arguments and displays an error message if so.
- File Analysis (line 44-58): Ensures that the file_analysis.txt file doesn't exist to avoid counting it during the file search, and to regenerate it if it previously existed.

2- File Filtering by Extensions (line 68-85):

- Extensions Separation (line 70): Separates extensions and converts them into an array by removing commas.
- Directory Files (line 73-76): Retrieves all files within the directory and converts them into an array of file paths.
- Extension Filtering (line 79-85): Filters files by extension, traverses through all files, and adds the matching files to an array.

3- File Filtering based on Size, Date, and Permissions (line 89-147):

- Size Filtering (line 91-110): Asks the user to enter the minimum and maximum file sizes and ensures valid input.
- Date Filtering (line 113-120): Asks the user to enter the minimum and maximum dates in 'yyyy-mm-dd' format and ensures valid input.
- Permission Filtering (line 136-147): Asks the user to enter permissions separated by commas.

4- Storing File Details in Arrays and Associative Arrays (line 152-195):

- Data Storage (line 152-167): Uses associative arrays to store file details, owner names with their corresponding file paths, and total sizes of files for each owner.
- File Data Extraction (line 172-177): Extracts the data of each file for filtering and storage.
- Filtering and Storing (line 179-191): Filters files based on user-selected criteria and stores the file details. If no filtering is chosen, all files are stored.

- Final File Filtering (line 193-195): Copies filtered files to a separate array for later use, if filtering was selected.

5- Sorting Owners by Total Size (line 200-215):

- Owner Sorting (line 200-215): Uses selection sort to sort the owners by their total file size.

6- Generating the Report (line 219-246):

- Summary Generation (line 220-232): Adds a summary of the files to the report if the user chooses to include it.
- Owner and File Details (line 236-246): Adds details of each owner and their corresponding files to the report.

C - Optimization

To optimize my script, I employ several techniques:

- 1- I begin by filtering the files based on their extensions. This helps reduce the number of operations performed on a large number of files.
- 2- While traversing through the files, I save all the necessary details to avoid extracting them multiple times.
- 3- I utilize associative arrays to directly access data. This enables me to quickly retrieve the size based on the owner's name, access file details using the file path, or retrieve all files belonging to a specific owner by their owner name.

By implementing these strategies, I aim to enhance the efficiency and performance of my script.

D - Advance features

- 1- The script allows multiple extensions to be specified as arguments, separated by commas. For example: "txt,doc".
- 2- An optional argument, '-s', enables the generation of a summary of files on top of the report.
- 3- Another optional argument, '-f', provides the ability for the user to enter the minimum and maximum file size, modified date, and desired permissions. Permissions can be specified by separating them with commas. For example: 'rw-r--r--' or 'rw-r--r--,drwxr-xr-x'.

E- User-friendliness aspects

Consider the following user-friendliness aspects of the script:

- 1- If no arguments are provided or if the user includes '-h' or '--help' as an argument, a help message is displayed, explaining how to use the script.
- 2- In case the user provides an incorrect directory, misses one of the required arguments, enters an invalid argument, or doesn't enter the extensions in the correct format, an error message is shown, indicating the specific issue.
- 3- When the user includes a filter argument, the script prompts the user to enter a valid input. If an invalid input is provided, the script requests the user to enter the value again, ensuring that valid input is provided for proper execution.

These user-friendly features aim to assist users in understanding and using the script effectively, while providing clear instructions and error handling to resolve any potential issues

F- The reflection section

1- Learning Experience:

- I expanded my knowledge about Linux commands and their usage.
- I gained proficiency in writing and utilizing Bash scripts, along with incorporating Linux commands within them.
- My searching skills improved as I conducted extensive research to gather information on Bash scripting and problem-solving techniques.

2- Challenges Faced:

- Initially, I found Bash scripting to be somewhat challenging, but with consistent practice, it became easier to grasp.
- I encountered difficulties in finding relevant resources that could assist me in learning and solving the assignment effectively.

3- Future Improvements:

- To enhance the script's performance, I intend to explore and implement parallel programming techniques. This will enable simultaneous operations on multiple files, thereby reducing execution time.
- By reflecting on my learning experience, overcoming challenges, and identifying areas for future improvement, I can continue to enhance my skills in Bash scripting and optimize the script further.