LINUX MODULE 5 ASSESSMENT SOLUTIONS

-BY SAKTHI KUMAR S

Overview:

This assignment involves creating a Bash script (file_analyzer.sh) that demonstrates key scripting concepts such as recursion, error handling, special parameters, regular expressions, and command-line arguments.

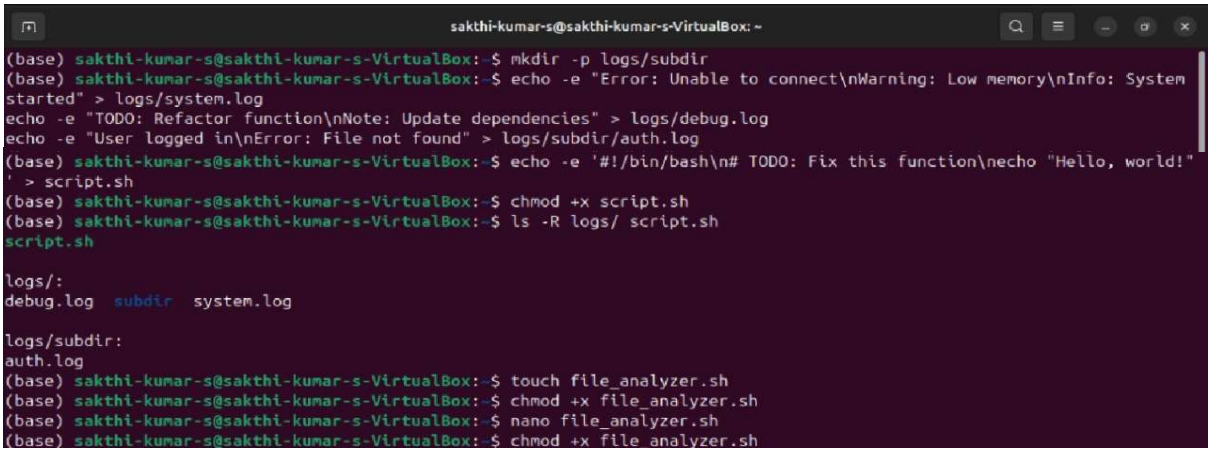
Tasks:

1. Recursive Function – Searches for a keyword in a directory and its subdirectories.
2. Error Handling & Logging – Logs errors to errors.log and displays them in the terminal.
3. Here Document & Here String – Displays a help menu and searches within files.
4. Special Parameters – Uses $0, $#, $?, $@ for meaningful feedback.
5. Regular Expressions – Validates inputs (checks if the file exists, keyword is valid).
6. Command-Line Arguments (getopts) – Handles -d <dir>, -k <keyword>, -f <file>, and --help.

Files Involved:

1. file_analyzer.sh – Bash script implementing all required functionalities.
2. Log Files (logs/system.log, logs/debug.log, logs/subdir/auth.log) – Sample files for testing.
3. Test Script (script.sh) – Contains sample content with TODO comments.
4. errors.log – Stores error messages.
5. README.md – Documentation for the project.

Outputs:



Steps:

- Recursively search a directory for a keyword, Search for a keyword in a file, Display the help menu.
- Here Document & Here String: Here Document (<<EOF) - Help Menu. When --help is passed, a here-document displays the help menu.
- Here String (<<<) - Keyword Search in File: Uses a here string (<<<) to pass keyword input to grep.
- Command-Line Arguments (getopts).

Output:

```
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -d logs -k error
Searching recursively for 'error' in directory: logs
logs/subdir/auth.log:2:Error: File not found
logs/system.log:1:Error: Unable to connect
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -f script.sh -k TODO
Searching for 'TODO' in file: script.sh
# TODO: Fix this function
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh --help
Usage: ./file_analyzer.sh [options]
Options:
  -d <directory>   Search for a keyword recursively in a directory
  -f <file>        Search for a keyword in a specific file
  -k <keyword>     Keyword to search
  --help           Display this help menu
```

```
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ls
 Desktop            logs         my-jupyter-env   script.sh  'vscode '
 Documents          miniconda3   Pictures         snap
 Downloads          Music        Public           Templates
 file_analyzer.sh   myenv        Sakthikumar      Videos
```

Steps:

- Redirection and Error Handling: Logs errors like **invalid directory**, **missing file**, or **empty keyword** to errors.log and displays them in the terminal.

Output:

```
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -d nonexistent_folder -k error
Error: Directory 'nonexistent_folder' not found!
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -f nonexistent_file -k error
Error: File 'nonexistent_file' not found!
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -d logs -k ""
Error: Keyword cannot be empty!
```

Steps: Special Parameters

- **$0** → Script name ($0) used in the help menu.
- **$#** → Checks the number of arguments (ensures keyword is provided).
- **$?** → Captures the exit status.
- **$@** → Used in getopts for argument parsing.
- Regular Expressions for Validation

Output:

```
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh -d logs -k error
Searching recursively for 'error' in directory: logs
logs/subdir/auth.log:2:Error: File not found
logs/system.log:1:Error: Unable to connect
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ echo $?
0
(base) sakthi-kumar-s@sakthi-kumar-s-VirtualBox:~$ ./file_analyzer.sh
Error: Keyword cannot be empty!
```