Campus Hires 2026
LINUX Assessment Module 4

S Vimal

Q.) Create a Bash script 'file_analyzer.sh', to demonstrate the following concepts:

1. Recursive functions

- Write a recursive function to search for files in a directory and its subdirectories containing a specific keyword.

2. Redirection and error handling

- Log errors (e.g., invalid arguments, missing files) to 'errors.log' and display them in the terminal.

3. Here document and here string

- Use a here document to display a help menu when the '--help' option is passed.

- Search for a keyword in a specified file using a here string

4. Special parameters

- Use parameters like '$0', '$#', '$?' and '$@' to provide meaningful feedback.

5. Regular expressions

- Validate inputs with regular expressions (Check if the file exists and the keyword is not empty and valid)

6. Command-line arguments using getopts

- Use 'getopts' to handle:

'-d <directory>': Directory to search.

'-k <keyword>': Keyword to search.

'-f <file>': File to search directly.

'--help': Display the help menu

Output and Solution:

The code is tested with many test cases for checking the proper functionality:

1. **The script is test with no arguments provided**
2. **Using the –help argument for the manual**

```
vimal@vimal-VirtualBox: ~/Desktop
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh
ERROR: No arguments provided. Use --help
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh --help
Usage: ./file_analyzer.sh [-d directory] [-f file] [-k keyword]
-d : Recursive search in directory
-f : Search in a specific file
-k : The keyword to find
--help : Show help
Script name: ./file_analyzer.sh
```

3. **Case where the given file contains the keyword provided**
4. **Case where the text file does not have the keyword**

```
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -f sample.txt -k society
All arguments: -f sample.txt -k society
Argument count: 4
Keyword 'society' found in sample.txt
Completed Successfully - script: ./file_analyzer.sh
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -f sample.txt -k linux
All arguments: -f sample.txt -k linux
Argument count: 4
Keyword not found
Completed Successfully - script: ./file_analyzer.sh
```

5. **To test the recursion ability of the code, the keyword Linux is entered in 2 text files across different depths of a directory.**
6. **Also it is tested when the given file doesn't exist**
7. **And the error log is shown for reference**

```
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -d testdir -k Linux
All arguments: -d testdir -k Linux
Argument count: 4
Recursive search started..
Found in: testdir/a.txt
Found in: testdir/sub1/b.txt
Completed Successfully - script: ./file_analyzer.sh
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -f nullfile -k Linux
All arguments: -f nullfile -k Linux
Argument count: 4
ERROR: File not found
vimal@vimal-VirtualBox:~/Desktop$ cat errors.log
ERROR: No arguments provided. Use --help
ERROR: No arguments provided. Use --help
ERROR: Keyword invalid
ERROR: File not found
ERROR: File not found
ERROR: File not found
vimal@vimal-VirtualBox:~/Desktop$
```

8. **Checking a directory that doesn't exist**
9. **Checking with an invalid keyword**
10. **Error log**

```
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -d nulldirectory -k Linux
All arguments: -d nulldirectory -k Linux
Argument count: 4
ERROR: Directory not found
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -f sample.txt -k "hello@67"All arguments:
-f sample.txt -k hello@67
Argument count: 4
ERROR: Keyword invalid
vimal@vimal-VirtualBox:~/Desktop$ cat errors.log
ERROR: No arguments provided. Use --help
ERROR: No arguments provided. Use --help
ERROR: Keyword invalid
ERROR: File not found
ERROR: File not found
ERROR: File not found
ERROR: File not found
ERROR: Directory not found
ERROR: Keyword invalid
vimal@vimal-VirtualBox:~/Desktop$
```

11. **Invalid option**

```
vimal@vimal-VirtualBox:~/Desktop$ ./file_analyzer.sh -x test
All arguments: -x test
Argument count: 2
ERROR: Invalid option: -?
```

file_analyzer.sh

```bash
#!/bin/bash

LOG_ERROR="errors.log"

log_error(){
        local info="$1"
        echo "ERROR: $info" >> "$LOG_ERROR"
        echo "ERROR: $info" >&2

}

show_help() {
cat << EOF
Usage: $0 [-d directory] [-f file] [-k keyword]
-d : Recursive search in directory
-f : Search in a specific file
-k : The keyword to find
--help : Show help
EOF
}
```

```bash
#recursive search
recursive_search(){
        local target_dir="$1"
        local keyword="$2"

        for item in "$target_dir"/*; do
                [[ -e "$item" ]] || continue

                if [[ -d "$item" ]]; then
                        recursive_search "$item" "$keyword"
                elif [[ -f "$item" ]]; then
                        if grep -q "$keyword" "$item" 2>>"$LOG_ERROR"; then
                                echo "Found in: $item"
                        fi
                fi
        done
}

# --help check
if [[ $1 == "--help" ]]; then
        show_help
        echo "Script name: $0"
        exit 0
fi

#argument count check
if [[ $# -eq 0 ]]; then
        log_error "No arguments provided. Use --help"
        exit 1
fi
```

```bash
#getopts
while getopts ":d:f:k:" args; do
        case $args in
        d) search_dir="$OPTARG" ;;
        k) keyword="$OPTARG" ;;
        f) target_file="$OPTARG" ;;
        :) log_error "Option -$OPTARG requires value"; exit 1 ;;
        *) log_error "Invalid option: -$args"; exit 1;;
        esac
done

#keyword regex validation
if [[ -z "$keyword" || ! "$keyword" =~ ^[A-Za-z0-9_.-]+$ ]]; then
        log_error "Keyword invalid/empty"
        exit 1
fi

#file mode(using HERE String)

if [[ -n "$target_file" ]]; then
        [[ -f "$target_file" ]] || { log_error "File not found"; exit 1; }
        file_content=$(cat "$target_file")

        if grep -q "$keyword" <<< "$file_content"; then
                echo "Keyword '$keyword' found in $target_file"
        else
                echo "Keyword not found"
        fi
```

```bash
# directory mode (RECURSIVE)

elif [[ -n "$search_dir" ]]; then

        [[ -d "$search_dir" ]] || { log_error "Directory not found"; exit 1; }

        echo "Recursive search started.."
        recursive_search "$search_dir" "$keyword"
fi

# exit status demo
if [[ $? -eq 0 ]]; then
        echo "Completed Successfully - script: $0"
else
        echo "Execution failed - script: $0"
fi
```