

Module – 5

Assignment Submission – embedUR Linux Course

05.02.2026

Baalamurugan S P

baalamurugan.sp@gmail.com

Question – 1

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 getopt

- 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.

Example usage:

```
# Recursively search a directory for a keyword
./file_analyzer.sh -d logs -k error
```

```
# Search for a keyword in a file
./file_analyzer.sh -f script.sh -k TODO
```

```
# Display the help menu
./file_analyzer.sh --help
```

```
baala@ItsB-Laptop:~/assignments/module5$ nano file_analyzer.sh
baala@ItsB-Laptop:~/assignments/module5$ chmod +x file_analyzer.sh
baala@ItsB-Laptop:~/assignments/module5$ nano fake_system.log
baala@ItsB-Laptop:~/assignments/module5$ cat fake_system.log
INFO System boot initiated
INFO Kernel loaded successfully
INFO Network interface eth0 up
WARNING CPU temperature high
INFO User login: admin
ERROR Failed to mount /dev/sdal
INFO Retry mount operation
ERROR Disk read timeout
WARNING Low available memory
INFO Memory cleanup started
INFO Cleanup completed
ERROR Network connection lost
INFO Attempting reconnection
INFO Reconnection successful
INFO System running normally
baala@ItsB-Laptop:~/assignments/module5$ ./file_analyzer.sh -f fake_system.log -k ERROR
Searching in: fake_system.log
ERROR Failed to mount /dev/sdal
ERROR Disk read timeout
ERROR Network connection lost
Done!
baala@ItsB-Laptop:~/assignments/module5$ ./file_analyzer.sh -f fake_system.log -k INFO
Searching in: fake_system.log
INFO System boot initiated
INFO Kernel loaded successfully
INFO Network interface eth0 up
INFO User login: admin
INFO Retry mount operation
INFO Memory cleanup started
INFO Cleanup completed
INFO Attempting reconnection
INFO Reconnection successful
INFO System running normally
Done!
```

```
baala@ItsB-Laptop:~/assignments/module5$ ./file_analyzer.sh --help
Usage: ./file_analyzer.sh [OPTIONS]
```

Options:

- d <directory> Search directory
- k <keyword> Keyword to find
- f <file> Search single file
- help Show help

Examples:

- ./file_analyzer.sh -d logs -k ERROR
- ./file_analyzer.sh -f fake_system.log -k WARNING

```

baala@ItsB-Laptop:~/assignments/module5$ mkdir -p logs/app
baala@ItsB-Laptop:~/assignments/module5$ mkdir -p logs/system
baala@ItsB-Laptop:~/assignments/module5$ mkdir -p logs/network
baala@ItsB-Laptop:~/assignments/module5$ cat <<EOF > logs/app.log
INFO Application started
ERROR Database connection failed
WARNING Slow query detected
EOF
baala@ItsB-Laptop:~/assignments/module5$ cat <<EOF > logs/app/debug.log
INFO Debug mode enabled
ERROR Memory allocation failed
INFO Processing request
EOF
baala@ItsB-Laptop:~/assignments/module5$ cat <<EOF > logs/system/kernel.log
INFO Kernel loaded
ERROR Disk I/O error
WARNING High CPU usage
EOF
baala@ItsB-Laptop:~/assignments/module5$ cat <<EOF > logs/network/wifi.log
INFO WiFi connected
ERROR Connection timeout
INFO Reconnecting
EOF

```

```

baala@ItsB-Laptop:~/assignments/module5$ ./file_analyzer.sh -d logs -k ERROR
Searching in: logs
logs/app/debug.log
Found in: logs/app/debug.log
logs/app.log
Found in: logs/app.log
logs/network/wifi.log
Found in: logs/network/wifi.log
logs/system/kernel.log
Found in: logs/system/kernel.log
Done! Total args: 4
baala@ItsB-Laptop:~/assignments/module5$ |

```

Code:

```

#!/bin/bash

# Recursive function to search files
search_directory() {
    local dir=$1
    local keyword=$2
    for item in "$dir"/*; do
        if [ -f "$item" ]; then
            grep -l "$keyword" "$item" 2>/dev/null && echo "Found in: $item"
        elif [ -d "$item" ]; then
            search_directory "$item" "$keyword"
        fi
    done
}

```

```

        done
    }

# Help menu using here document

show_help() {
    cat << EOF
Usage: $0 [OPTIONS]
Options:
    -d <directory>    Search directory
    -k <keyword>       Keyword to find
    -f <file>          Search single file
    --help              Show help

Examples:
    $0 -d logs -k ERROR
    $0 -f fake_system.log -k WARNING
EOF
}

# Show help if requested

if [ "$1" == "--help" ]; then
    show_help
    exit 0
fi

# Check arguments

if [ $# -eq 0 ]; then
    echo "Error: No arguments" 2>&1 | tee -a errors.log
    exit 1
fi

# Parse options with getopt

while getopt "d:k:f:" opt; do
    case $opt in
        d) directory=$OPTARG ;;
        k) keyword=$OPTARG ;;
        f) file=$OPTARG ;;
        *) echo "Invalid option" 2>&1 | tee -a errors.log; exit 1 ;;
    esac
done

# Validate keyword with regex

```

```

if [[ ! "$keyword" =~ ^[a-zA-Z0-9_]+$ ]]; then
    echo "Error: Invalid keyword" 2>&1 | tee -a errors.log
    exit 1
fi

# Search in file using here string
if [ -n "$file" ]; then
    if [ ! -f "$file" ]; then
        echo "Error: File not found" 2>&1 | tee -a errors.log
        exit 1
    fi
    echo "Searching in: $file"
    grep "$keyword" <<< "$(cat "$file")"
    if [ $? -eq 0 ]; then
        echo "Done!"
    else
        echo "Not found"
    fi
    exit 0
fi

# Search in directory recursively
if [ -n "$directory" ]; then
    if [ ! -d "$directory" ]; then
        echo "Error: Directory not found" 2>&1 | tee -a errors.log
        exit 1
    fi

    echo "Searching in: $directory"
    search_directory "$directory" "$keyword"
    echo "Done! Total args: $#"
    exit 0
fi

echo "Error: Use -d or -f" 2>&1 | tee -a errors.log
exit 1

```