

Challenge: Log Analyzer and Report Generator

Scenario

You are a system administrator responsible for managing a network of servers. Every day, a log file is generated on each server containing important system events and error messages. As part of your daily tasks, you need to analyze these log files, identify specific events, and generate a summary report.

Task

Write a Bash script that automates the process of analyzing log files and generating a daily summary report. The script should perform the following steps:

1. **Input:** The script should take the path to the log file as a command-line argument.

```
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ bash log_analyzer.sh
Please enter a log file name
Usage: ./log_analyzer.sh log_file.txt
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ bash log_analyzer.sh log
This file doesn't exist.
Please enter a proper file name
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ ./logfile.sh log.txt 10
Log file created at: log.txt with 10 lines.
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ bash log_analyzer.sh log.txt
Log has been analyzed and the report is generated
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$
```

2. **Error Count:** Analyze the log file and count the number of error messages. An error message can be identified by a specific keyword (e.g., "ERROR" or "Failed"). Print the total error count.

```
Total error count: 21
```

3. **Critical Events:** Search for lines containing the keyword "CRITICAL" and print those lines along with the line number.

```
List of critical events with line numbers:
Line No. 7 : 2023-08-04 21:01:50 [CRITICAL] - 1766
Line No. 11 : 2023-08-04 21:01:50 [CRITICAL] - 11674
Line No. 13 : 2023-08-04 21:01:50 [CRITICAL] - 23117
Line No. 16 : 2023-08-04 21:01:50 [CRITICAL] - 15256
Line No. 24 : 2023-08-04 21:01:50 [CRITICAL] - 27423
Line No. 27 : 2023-08-04 21:01:50 [CRITICAL] - 21693
Line No. 36 : 2023-08-04 21:01:50 [CRITICAL] - 7130
Line No. 40 : 2023-08-04 21:01:50 [CRITICAL] - 3756
Line No. 48 : 2023-08-04 21:01:50 [CRITICAL] - 11533
Line No. 56 : 2023-08-04 21:01:50 [CRITICAL] - 4344
Line No. 64 : 2023-08-04 21:01:50 [CRITICAL] - 9403
Line No. 72 : 2023-08-04 21:01:50 [CRITICAL] - 4871
Line No. 76 : 2023-08-04 21:01:50 [CRITICAL] - 28076
Line No. 84 : 2023-08-04 21:01:50 [CRITICAL] - 26040
Line No. 88 : 2023-08-04 21:01:50 [CRITICAL] - 28114
|
```

4. **Top Error Messages:** Identify the top 5 most common error messages and display them along with their occurrence count.

```
Top 5 error messages with their occurrence count:
7 Invalid input
6 Failed to connect
4 Disk full
3 Segmentation fault
1 Out of memory
```

5. **Summary Report:** Generate a summary report in a separate text file. The report should include:

- Date of analysis
- Log file name
- Total lines processed
- Total error count
- Top 5 error messages with their occurrence count

- List of critical events with line numbers

```
report.txt - Notepad
File Edit Format View Help
Date of analysis: 2023-08-04_21-02-04
Log file name: logs.txt
Total lines processed: 100
Total error count: 21
Top 5 error messages with their occurrence count:
    7 Invalid input
    6 Failed to connect
    4 Disk full
    3 Segmentation fault
    1 Out of memory
List of critical events with line numbers:
Line No. 7 : 2023-08-04 21:01:50 [CRITICAL] - 1766
Line No. 11 : 2023-08-04 21:01:50 [CRITICAL] - 11674
Line No. 13 : 2023-08-04 21:01:50 [CRITICAL] - 23117
Line No. 16 : 2023-08-04 21:01:50 [CRITICAL] - 15256
Line No. 24 : 2023-08-04 21:01:50 [CRITICAL] - 27423
Line No. 27 : 2023-08-04 21:01:50 [CRITICAL] - 21693
Line No. 36 : 2023-08-04 21:01:50 [CRITICAL] - 7130
Line No. 40 : 2023-08-04 21:01:50 [CRITICAL] - 3756
Line No. 48 : 2023-08-04 21:01:50 [CRITICAL] - 11533
Line No. 56 : 2023-08-04 21:01:50 [CRITICAL] - 4344
Line No. 64 : 2023-08-04 21:01:50 [CRITICAL] - 9403
Line No. 72 : 2023-08-04 21:01:50 [CRITICAL] - 4871
Line No. 76 : 2023-08-04 21:01:50 [CRITICAL] - 28076
Line No. 84 : 2023-08-04 21:01:50 [CRITICAL] - 26040
Line No. 88 : 2023-08-04 21:01:50 [CRITICAL] - 28114
|
```

Optional Enhancement: Add a feature to automatically archive or move processed log files to a designated directory after analysis.

```
Log has been analyzed and the report is generated
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ cd logs_archive/
ubuntu@ubuntuvm:~/workspace/New folder/bash-5/logs_archive$ ls
2023-08-04_20-56-29_log.txt  2023-08-04_21-02-04_log.txt
2023-08-04_20-57-44_log.txt
ubuntu@ubuntuvm:~/workspace/New folder/bash-5/logs_archive$
```


Usage of the script:

1. Create a log file using the logfile.sh-

./logfile.sh <log_filename> <no_of_logs_in_file>

```
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ ./logfile.sh logs.txt 100
Log file created at: logs.txt with 100 lines.
```

2. Use the log_analyzer.sh to create the report.

./log_analyzer.sh <log_filename>

```
Log has been analyzed and the report is generated
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ ./logfile.sh logs.txt 100
Log file created at: logs.txt with 100 lines.
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ bash log_analyzer.sh logs.txt
Log has been analyzed and the report is generated
```

You will find the report in report.txt and the log files will be renamed according to the date and time of analysis in the logs_archive folder.

```
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ bash log_analyzer.sh logs.txt
Log has been analyzed and the report is generated
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ ls
Log_Analyzer.md  logfile.sh      report.txt
log_analyzer.sh  logs_archive   Sample_Log_Data.md
ubuntu@ubuntuvm:~/workspace/New folder/bash-5$ cd logs_archive/
ubuntu@ubuntuvm:~/workspace/New folder/bash-5/logs_archive$ ls -l
total 13
-rwxrwxrwx 1 root root 410 Aug  4 20:56 2023-08-04_20-56-29_log.txt
-rwxrwxrwx 1 root root 394 Aug  4 20:57 2023-08-04_20-57-44_log.txt
-rwxrwxrwx 1 root root 4050 Aug  4 21:01 2023-08-04_21-02-04_log.txt
-rwxrwxrwx 1 root root 4001 Aug  4 21:09 2023-08-04_21-09-31_log.txt
-rwxrwxrwx 1 root root 4094 Aug  4 21:09 2023-08-04_21-09-57_log.txt
ubuntu@ubuntuvm:~/workspace/New folder/bash-5/logs_archive$
```

The code -

```
1  #!/bin/bash
2  if [ $# -eq 0 ]
3  then
4      echo "Please enter a log file name"
5      echo "Usage: ./log_analyzer.sh log_file.txt"
6  else
7      file=$1
8      if [ -f "$file" ];
9      then
10         i=1
11         no_of_error=0
12         date_of_analysis=$(date +%Y-%m-%d_%H-%M-%S)
13         lines=$(cat $file | wc -l)
14         no_of_error=$(cat $file | grep -E "ERROR|FAILED" | wc -l)
15         top_error=$(cat $file | awk -F'|' '{print $2}' | sort | uniq -c | sort -nr | head -n 6 | tail -n 5)
16
17         echo "Date of analysis: $date_of_analysis" > report.txt
18         echo "Log file name: $file" >> report.txt
19         echo "Total lines processed: $lines" >> report.txt
20         echo "Total error count: $no_of_error" >> report.txt
21         echo "Top 5 error messages with their occurrence count: " >> report.txt
22         echo "$top_error" >> report.txt
23
24         while read line; do
25             if [[ $line == *"CRITICAL"* ]]; then
26                 echo "Line No. $i : $line" >> report.txt
27                 fi
28                 i=$((i+1))
29             done < $file
30             dest_path="logs_archive"
31             mv $file "$date_of_analysis"_log.txt
32             file="$date_of_analysis"_log.txt
33             mkdir -p $dest_path
34             mv $file $dest_path
35             echo "Log has been analyzed and the report is generated"
36         else
37             echo "This file doesn't exist."
38             echo "Please enter a proper file name"
39         fi
40     fi
41
```

Thanks!