

Space Mission Telemetry Analyzer

Aim

To design and implement a Linux shell script that processes multiple space mission telemetry log files to compute the average temperature and speed using the commands **grep**, **sed**, **awk**, and file globbing.

Abstract

The *Space Mission Telemetry Analyzer* script automates the extraction and processing of telemetry data from multiple mission logs. It utilizes text-processing tools available in Linux:

- **grep** to filter lines containing specific telemetry parameters.
- **sed** to clean and format extracted data.
- **awk** to perform arithmetic operations like computing averages.
- Globbing (***.log**) to automatically process multiple files.

This application demonstrates efficient data analysis using command-line scripting and highlights the importance of automation in scientific data processing.

Problem Statement

Space missions generate large volumes of telemetry data, including parameters like temperature, speed, and orbit details, across multiple log files. Manually analyzing these logs is time-consuming and prone to errors. The problem is to automate the extraction, cleaning, and analysis of telemetry data from multiple log files using Linux command-line tools so that meaningful summaries, such as average temperature and speed, can be obtained efficiently.

Objectives

1. Automate data extraction using **grep** and globbing to process multiple mission log files.
2. Clean and format data using **sed** to remove unnecessary characters, units, and spaces.
3. Perform numeric analysis using **awk** to calculate average temperature and speed.
4. Summarize mission performance with clear, readable output.
5. Demonstrate the practical use of Linux scripting commands to handle real-world data processing tasks.

Uses / Applications

- Space mission telemetry and performance analysis.
- Automated report generation for research data.
- IoT sensor log summarization.
- Educational demonstration of Linux text-processing.
- Error detection and early warning for abnormal readings.
- Template for large-scale log file analysis in scientific research.

Program Code and explanation

Listing 1: Space Mission Telemetry Analyzer Script

```
#!/bin/bash

echo "=== Space Mission Telemetry Analyzer ==="
echo "Processing all mission logs..."
echo
# echo prints text to the terminal.
# First line prints title
# Second line indicates start of processing
# Third line prints a blank line for readability

for file in mission_logs/*.log
do
    echo "Analyzing $file..."
    grep -E "Temperature|Speed" "$file" >> all_data.txt
done
# for file in mission_logs/*.log -> selects all .log files in mission_logs
# do ... done -> loop structure
# echo -> prints current file being processed
# grep -E -> searches for Temperature or Speed
# >> all_data.txt -> appends matched lines to all_data.txt

grep "Temperature" all_data.txt | sed 's/Temperature: //;s/C//' > temp_values.txt
# grep "Temperature" -> selects only temperature lines
# sed 's/Temperature: //;s/C//' -> removes label and unit
# > temp_values.txt -> saves cleaned numeric temperature values

grep "Speed" all_data.txt | sed 's/Speed: //;s/km\s//' > speed_values.txt
# Same as above, but for speed
# Removes label and unit, saves to speed_values.txt

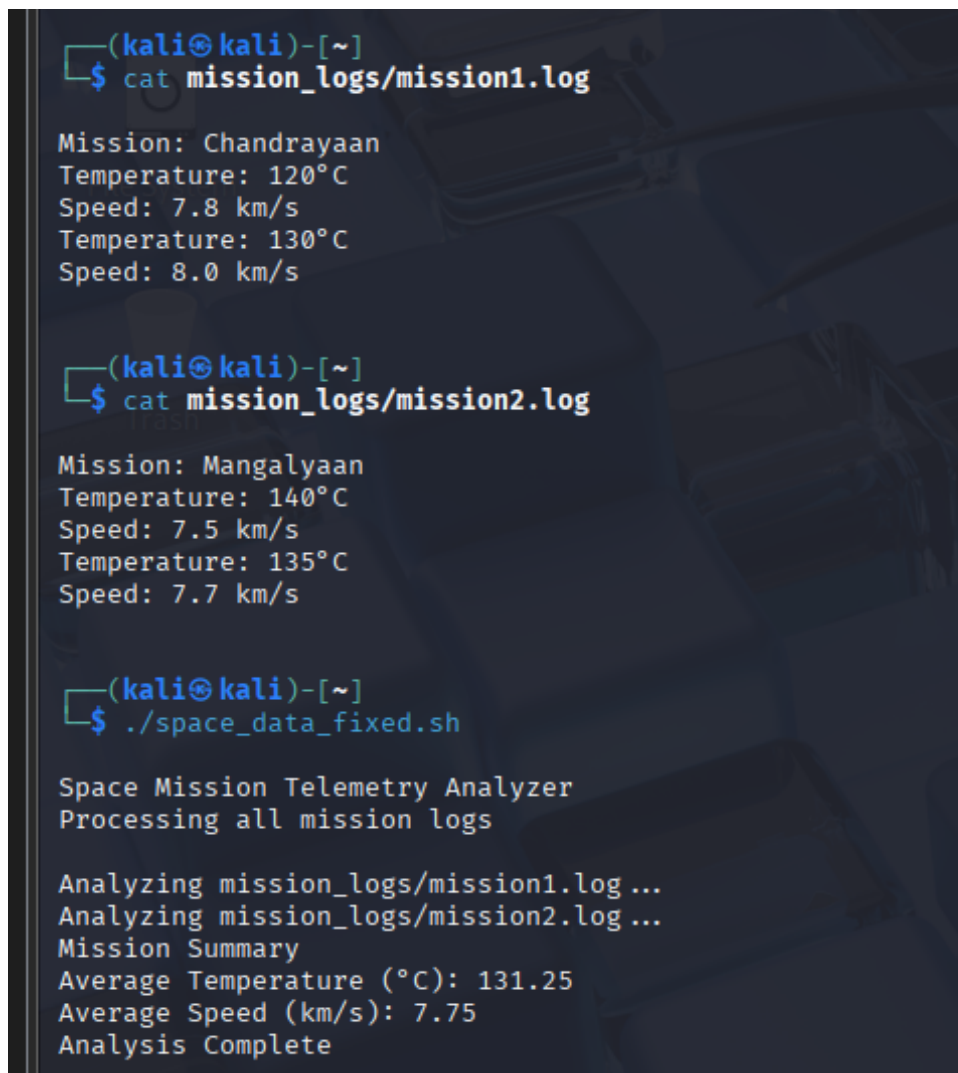
avg_temp=$(awk '{sum+=$1} END {if (NR>0) print sum/NR; else print "No data"}'
temp_values.txt)
# awk calculates average temperature
# {sum+=$1} -> sums each line's value
# END {if (NR>0) print sum/NR; else print "No data"} -> computes average if file has
data
# Stores result in avg_temp
```

```
avg_speed=$(awk '{sum+=$1} END {if (NR>0) print sum/NR; else print "No data"}'
speed_values.txt)
# Same as above, but for speed
# Stores result in avg_speed

echo "=== Mission Summary ==="
echo "Average Temperature (C): $avg_temp"
echo "Average Speed (km/s): $avg_speed"
echo "Analysis Complete!"
# Prints final summary
# Displays average temperature and speed
# Completion message
```

Screenshot of Output

The following screenshot shows the script execution with sample mission log files and the calculated averages:



```
(kali㉿kali)-[~]
$ cat mission_logs/mission1.log

Mission: Chandrayaan
Temperature: 120°C
Speed: 7.8 km/s
Temperature: 130°C
Speed: 8.0 km/s

(kali㉿kali)-[~]
$ cat mission_logs/mission2.log

Mission: Mangalyaan
Temperature: 140°C
Speed: 7.5 km/s
Temperature: 135°C
Speed: 7.7 km/s

(kali㉿kali)-[~]
$ ./space_data_fixed.sh

Space Mission Telemetry Analyzer
Processing all mission logs

Analyzing mission_logs/mission1.log ...
Analyzing mission_logs/mission2.log ...
Mission Summary
Average Temperature (°C): 131.25
Average Speed (km/s): 7.75
Analysis Complete
```

Figure 1: Execution of Space Mission Telemetry Analyzer Script

Sample Input

mission1.log

```
Mission: Chandrayaan
Temperature: 120°C
Speed: 7.8 km/s
Temperature: 130°C
Speed: 8.0 km/s
```

mission2.log

```
Mission: Mangalyaan
Temperature: 140°C
Speed: 7.5 km/s
Temperature: 135°C
Speed: 7.7 km/s
```

Sample Output

```
=== Space Mission Telemetry Analyzer ===
Processing all mission logs...
```

```
Analyzing mission_logs/mission1.log...
Analyzing mission_logs/mission2.log...
```

```
=== Mission Summary ===
Average Temperature (°C): 131.25
Average Speed (km/s): 7.75
Analysis Complete!
```

Result

The script successfully analyzed multiple space mission telemetry logs and computed the average temperature and speed using Linux commands. It demonstrated efficient use of text-processing and arithmetic tools available in the shell.

Conclusion

This experiment shows how Linux commands can be integrated to perform data analytics on structured logs. The combination of `grep`, `sed`, and `awk` provides a powerful toolkit for automation and numeric computation, while globbing simplifies batch file processing. Such automation techniques are valuable in analyzing large-scale scientific or telemetry data.