COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Group ID:

NAME OF MEMBER #1 CS-23025

NAME OF MEMBER #2 CS-23146

BATCH: 2023

Department of Computer and Information Systems Engineering

NED University of Engg. &amp; Tech. Karachi-

75270

CONTENTS

PROBLEM DESCRIPTION

With the goal interact with a free API, retrieve real-time environmental data (such as temperature and humidity), process, store, and report this data, the project involves constructing an integrated environmental monitoring system in C. The system will automate processes with shell scripts, optimize memory consumption via pointers and dynamic memory allocation, and apply system calls for real-time notifications.

Requirements:

- **API Integration**: Retrieve real-time environmental data via API (e.g., temperature, humidity).
- **Data Retrieval and Processing**: Process and filter the raw data for reporting and storage.
- **Data Storage**: Store both raw and processed data in files (JSON or text).
- **Automation via Shell Scripts**: Automate data retrieval and processing tasks using shell scripts.
- **Memory Optimization**: Use pointers and dynamic memory allocation to handle large datasets efficiently.
- **Real-Time Alerts**: Implement alerts for critical environmental conditions using system calls.
- **Modular Programming with Header Files**: Organize the code into modules using header files for clarity and maintainability.

Expected Outcomes:

- Practical experience with API interactions, system calls, shell scripting, and memory management in C.
- Skills in developing efficient, scalable real-time systems for environmental monitoring.

METHODOLOGY

The Integrated Environmental Monitoring System will be developed using a methodical and transparent process that is broken down into multiple crucial stages:

1-Requirement Analysis:

- Identify necessary environmental parameters (e.g., temperature, humidity).
- Choose a free API for real-time data.
- Set critical thresholds for alerts (e.g., extreme temperature).

2- System Design:

- Break the system into modules (e.g., data retrieval, processing, storage, alerting).
- Use dynamic memory allocation and pointers to manage data efficiently.

3-API Integration:

- Fetch data from the chosen API using HTTP requests.
- Use **libcurl** to handle communication
- Parse the JSON response with **jq**, using its command-line tool or **libjq**
- Schedule the task using **cron**, by adding jobs with `crontab -e`

4- Data Processing:

- Process raw data to extract useful information (temperature, humidity).
- Store processed data in appropriate structures or variables.

5-Data Storage:

- Store raw and processed data in separate files (e.g., CSV, JSON). Implement file I/O with error handling.

6-Automation with Shell Scripts:

- Create shell scripts to automate data retrieval and processing. Use cron jobs to schedule periodic tasks.

7-Real-Time Alerts:

- Implement alerts using system calls when data exceeds thresholds.
- Alerts can be terminal messages, notifications, or emails.

8-Testing and Debugging:

- Perform unit testing for individual modules.
- Test the full system for correct data handling and alerts.

9-Documentation:

- Document the code with comments and explanations. Prepare a user guide for system operation.

10-Optimization:

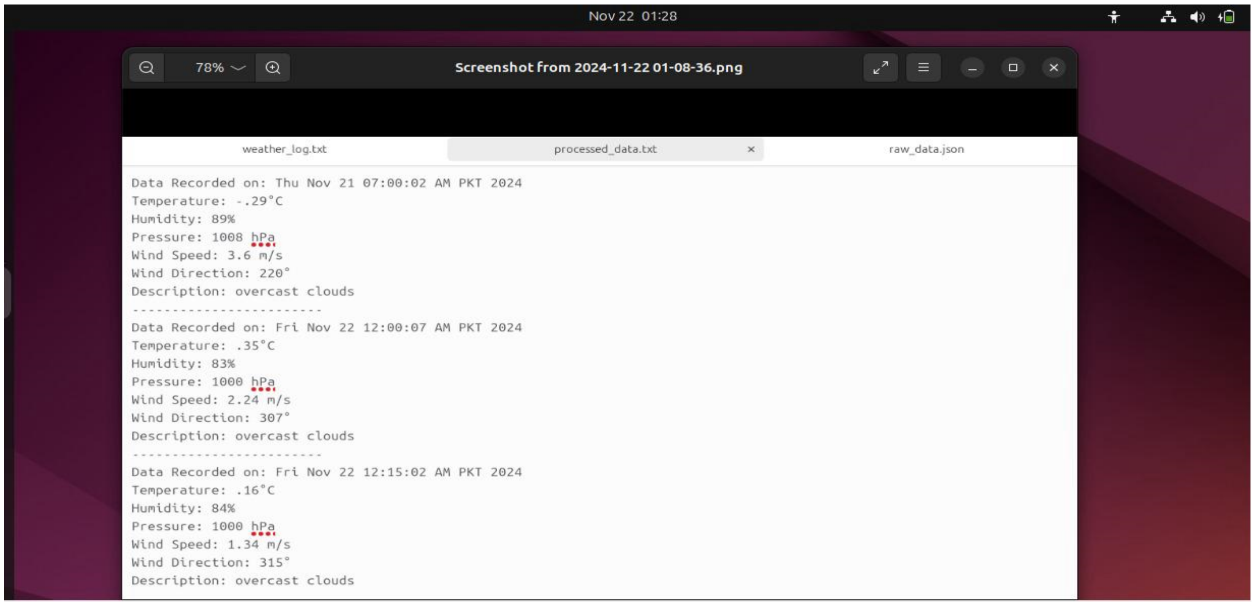
- Optimize memory usage and processing speed for efficiency.

11-Deployment:

- Deploy the system on a Linux environment for real-time testing. Ensure background operation and proper alerting/logging.

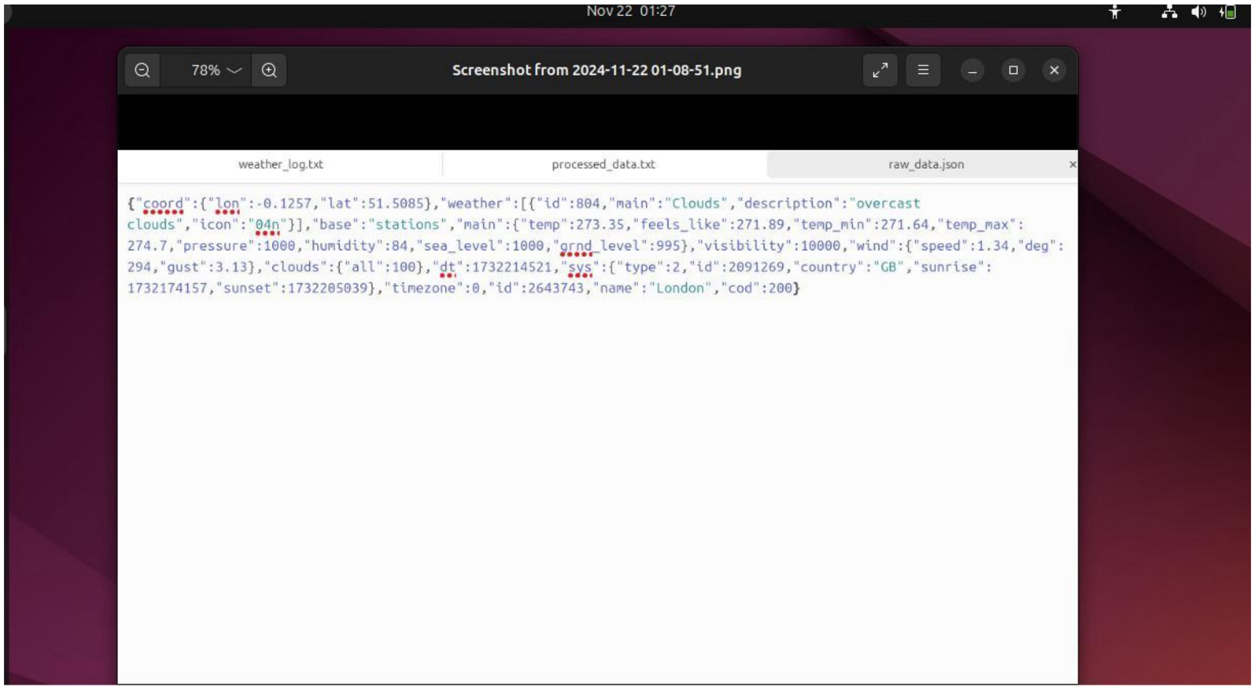
This methodology ensures an efficient, well-organized system for real-time environmental monitoring.

RESULTS:



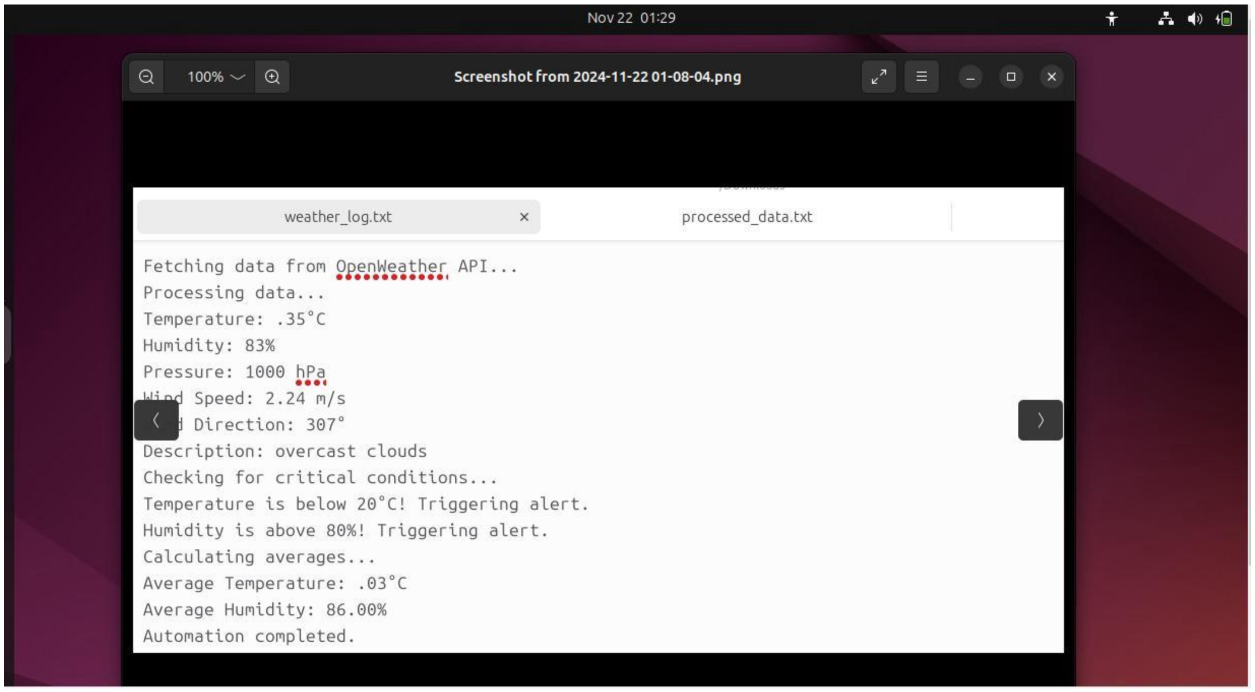
```
Data Recorded on: Thu Nov 21 07:00:02 AM PKT 2024
Temperature: -.29°C
Humidity: 89%
Pressure: 1008 hPa
Wind Speed: 3.6 m/s
Wind Direction: 220°
Description: overcast clouds
------------------------
Data Recorded on: Fri Nov 22 12:00:07 AM PKT 2024
Temperature: .35°C
Humidity: 83%
Pressure: 1000 hPa
Wind Speed: 2.24 m/s
Wind Direction: 307°
Description: overcast clouds
------------------------
Data Recorded on: Fri Nov 22 12:15:02 AM PKT 2024
Temperature: .16°C
Humidity: 84%
Pressure: 1000 hPa
Wind Speed: 1.34 m/s
Wind Direction: 315°
Description: overcast clouds
```

Figure 1 Process File



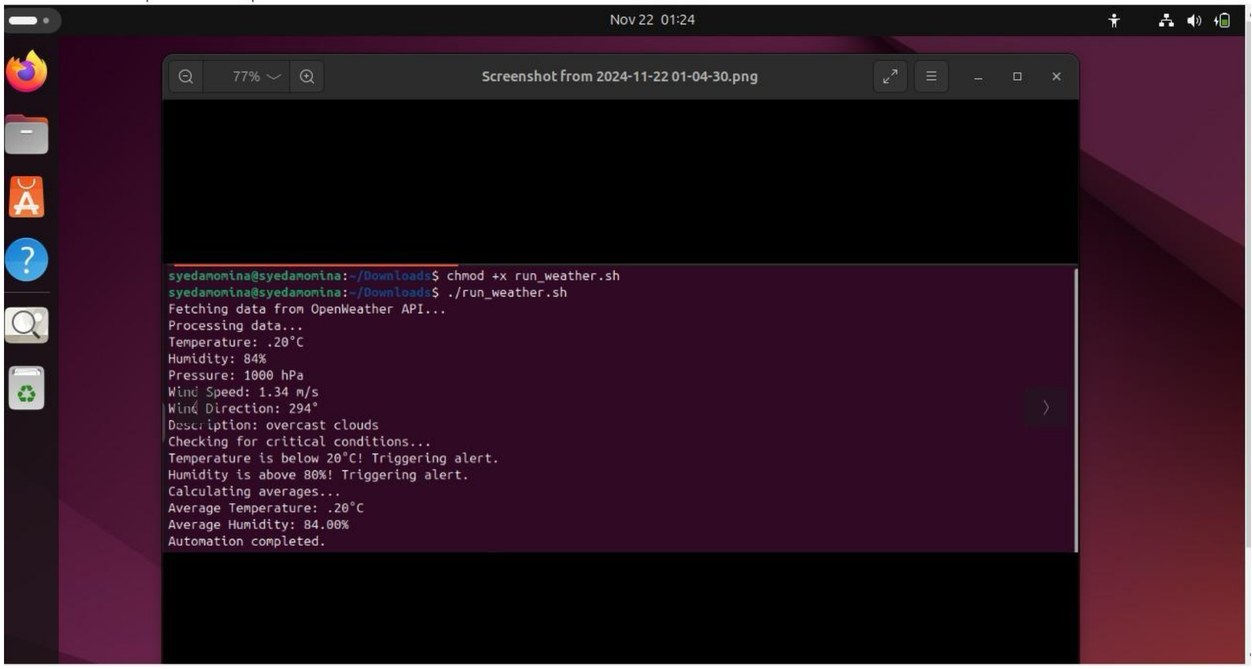
```
{"coord":{"lon":-0.1257,"lat":51.5085},"weather":[{"id":804,"main":"Clouds","description":"overcast
clouds","icon":"04n"}],"base":"stations","main":{"temp":273.35,"feels_like":271.89,"temp_min":271.64,"temp_max":
274.7,"pressure":1000,"humidity":84,"sea_level":1000,"grnd_level":995},"visibility":10000,"wind":{"speed":1.34,"deg":
294,"gust":3.13},"clouds":{"all":100},"dt":1732214521,"sys":{"type":2,"id":2091269,"country":"GB","sunrise":
1732174157,"sunset":1732205039},"timezone":0,"id":2643743,"name":"London","cod":200}
```

Figure 2 Raw_data_jsaon file



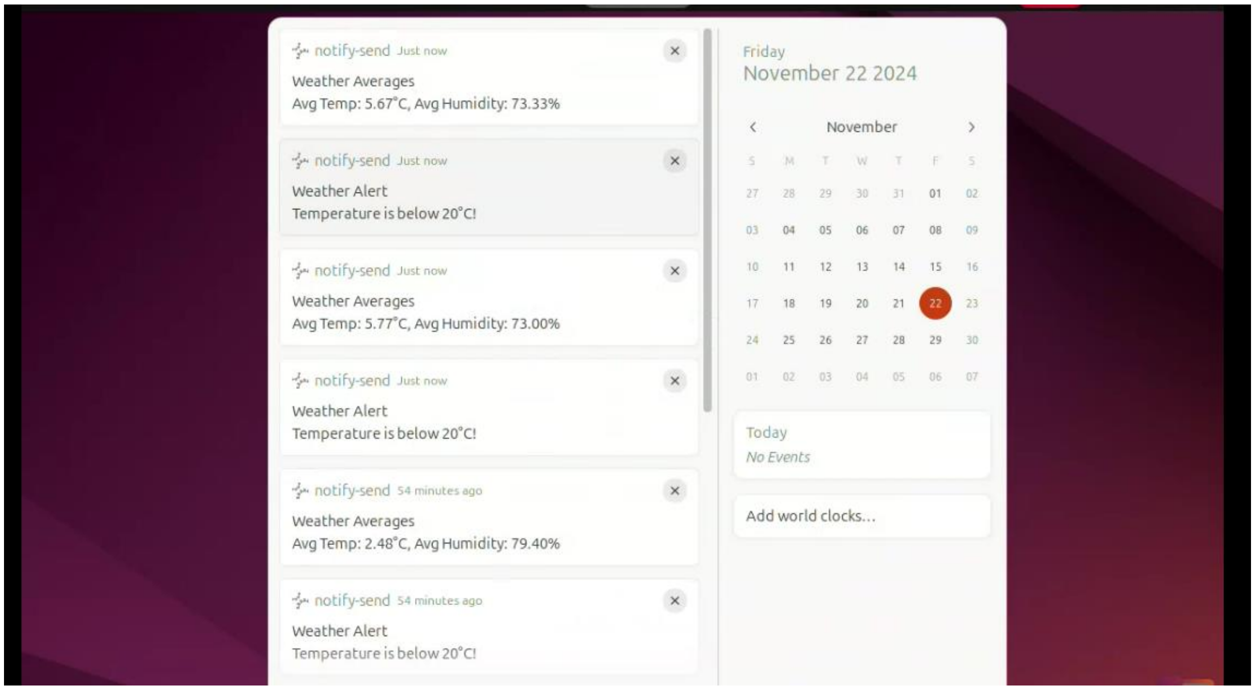
```
Fetching data from OpenWeather API...
Processing data...
Temperature: .35°C
Humidity: 83%
Pressure: 1000 hPa
Wind Speed: 2.24 m/s
 d Direction: 307°
Description: overcast clouds
Checking for critical conditions...
Temperature is below 20°C! Triggering alert.
Humidity is above 80%! Triggering alert.
Calculating averages...
Average Temperature: .03°C
Average Humidity: 86.00%
Automation completed.
```

Figure 3 weather_log_txt file



```
syedamomina@syedamomina:~/Downloads$ chmod +x run_weather.sh
syedamomina@syedamomina:~/Downloads$ ./run_weather.sh
Fetching data from OpenWeather API...
Processing data...
Temperature: .20°C
Humidity: 84%
Pressure: 1000 hPa
Wind Speed: 1.34 m/s
Wind Direction: 294°
Description: overcast clouds
Checking for critical conditions...
Temperature is below 20°C! Triggering alert.
Humidity is above 80%! Triggering alert.
Calculating averages...
Average Temperature: .20°C
Average Humidity: 84.00%
Automation completed.
```

Figure 4 Output



notify-send Just now
Weather Averages
Avg Temp: 5.67°C, Avg Humidity: 73.33%

notify-send Just now
Weather Alert
Temperature is below 20°C!

notify-send Just now
Weather Averages
Avg Temp: 5.77°C, Avg Humidity: 73.00%

notify-send Just now
Weather Alert
Temperature is below 20°C!

notify-send 54 minutes ago
Weather Averages
Avg Temp: 2.48°C, Avg Humidity: 79.40%

notify-send 54 minutes ago
Weather Alert
Temperature is below 20°C!

Figure 5 Notification Screen Shot