COMPUTER ENGINEERING WORKSHOP

S.E. (CIS) OEL REPORT

Project Group ID:

Ateeb Hussain Khan Sherwani CS-23069
Azan Ali Muhammad CS-23064
S.M Sajjad Baqar Abidi CS-23062

Batch 2023

Department of Computer and Information Systems Engineering

NED University of Engineering & Technology , Karachi-75270

1. Problem Description

Construct an integrated environmental monitoring system in C, covering a range of fundamental concepts and practical applications. The project involves interacting with a free API that provides real-time environmental data. The system's core functionalities include data retrieval, processing and reporting.
Requirements of the project include:

- Interact with a free API to retrieve real-time environmental data (e.g., temperature, humidity).
- Store raw and processed data in files.
- Create shell scripts to automate tasks such as data retrieval and processing.
- Utilize pointers and dynamic memory allocation in the C program to optimize data manipulation and enhance efficiency
- Implement real-time alerts using Linux system calls to notify relevant personnel of critical environmental readings.
- Use header files to modularize the C code and enhance code readability.

2. Methodology

Code Modularization

To enhance **readability**, **maintainability**, and **scalability**, the program was organized into multiple files:

- **Header Files (`.h`)**: Contain function declarations, macros, and shared data structures.
- **Source Files (`.c`)**: Implement specific functionalities such as:
  - API interaction
  - Data processing
  - Alert generation
- **Main Program**: Acts as the central control unit, coordinating the various modules.

API Interaction

The program retrieves real-time environmental data (e.g., temperature, humidity) using the **libcurl** library for HTTP requests.

- The API response, in **JSON format**, is parsed using **json-c**, a lightweight library chosen for its efficiency and simplicity.

Data Storage

The system stores both raw and processed data:

- **Raw Data**: Saved in `raw_data.json` to preserve the original API response.
- **Processed Data**: Key information such as city name, temperature, and humidity is saved in `output.txt` in a user-friendly format.

Automation

A shell script automates the data retrieval and processing:

- The program retrieves data from API once a day
- This ensures periodic updates without manual intervention.

Real-Time Alerts

- Alerts are generated if the temperature exceeds **30°C** or subceeds **5°C**.
- Notifications are displayed using the **notify-send** command on Linux, triggered via system calls from the program.

Memory Management

Dynamic memory allocation ensures efficient handling of API responses:

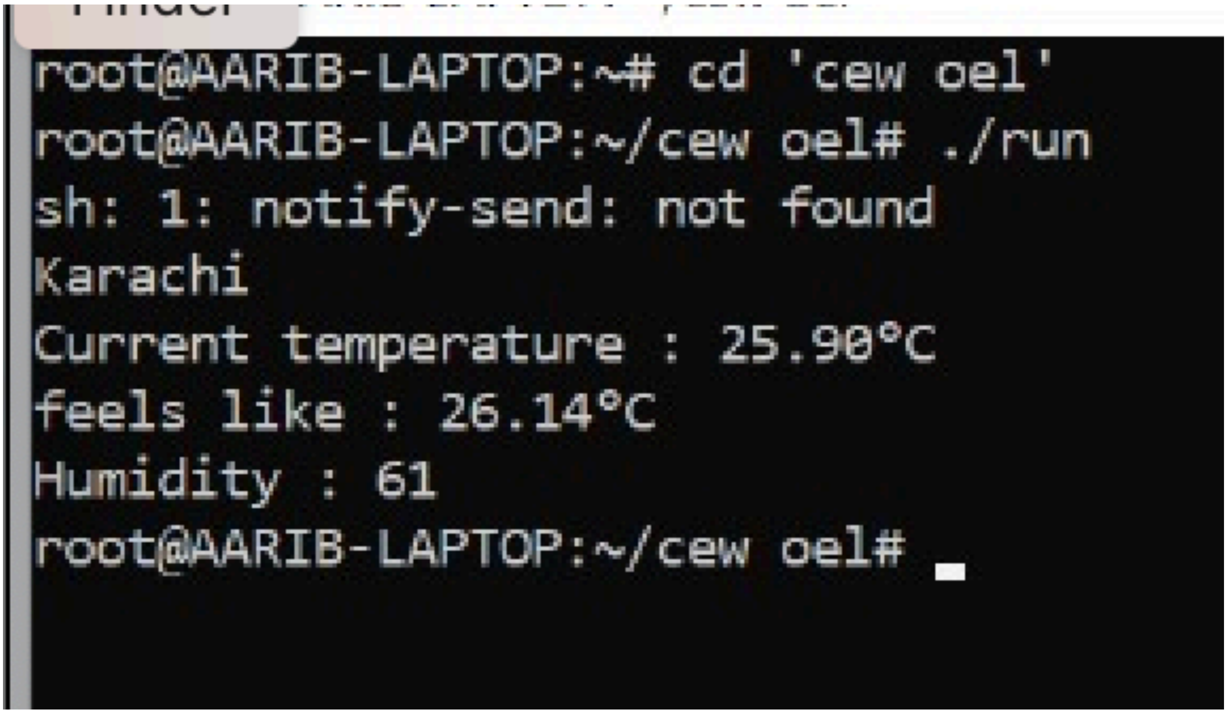
- A memory structure dynamically allocates space based on the size of the response.
- This avoids excessive or wasteful memory usage.

System Workflow

1. **Data Retrieval**: The program fetches data from the API once a day
2. **Data Processing**: The retrieved JSON data is parsed and stored in both raw and structured formats.
3. **Alerts**: Real-time notifications are displayed for critical temperature conditions.

By modularizing the code and leveraging libraries like **libcurl** and **json-c**, the system achieves robustness, efficiency, and maintainability.

3. Result



```
root@AARIB-LAPTOP:~# cd 'cew oel'
root@AARIB-LAPTOP:~/cew oel# ./run
sh: 1: notify-send: not found
Karachi
Current temperature : 25.90°C
feels like : 26.14°C
Humidity : 61
root@AARIB-LAPTOP:~/cew oel# 
```