**SUMMARY OF CODE IF NEEDED**

**Overview:**

This code is designed to monitor and log system resource usage information (such as CPU, memory, disk usage, processes, network activity, and swap usage) periodically, and upload the log file to Dropbox. The program performs the following tasks:

1. Logs detailed system usage.
2. Uploads the log file to Dropbox every minute.
3. Runs in an infinite loop, continuously logging and uploading the system resource usage.

**Detailed Explanation:**

**Included Libraries:**

1. **stdio.h**: Provides standard input/output functions for file operations and printing to the console.
2. **stdlib.h**: Used for memory management and other utility functions.
3. **string.h**: Provides string manipulation functions like strstr() for searching substrings.
4. **time.h**: Used for time-related functions, such as obtaining the current system time.
5. **unistd.h**: Provides access to POSIX functions, such as sleep(), used to pause the program.
6. **sys/statvfs.h**: Provides functions for retrieving file system statistics (for disk usage).
7. **curl/curl.h**: The cURL library is used for making HTTP requests to upload files to Dropbox.

**Key Definitions:**

* **LOG\_FILE**: The name of the log file where system resource data will be written (system\_usage.log).
* **DROPBOX\_TOKEN**: A placeholder for the Dropbox API token used for authentication when uploading the file.

**Functions:**

**1. log\_system\_usage()**

This function is responsible for collecting system resource information and writing it to the log file.

**Steps inside log\_system\_usage():**

* **Opening Log File**: The log file (system\_usage.log) is opened in append mode (a), so new data is added to the file without overwriting it.
* **Logging Time**: The current time is retrieved using time() and formatted with ctime(). The formatted time is written to the log file.
* **Logging CPU Usage**:
  + The function reads the /proc/stat file to retrieve CPU statistics.
  + The first line contains information like user, system, idle time, etc., which is written to the log file.
* **Logging Memory Usage**:
  + The /proc/meminfo file is read to get memory information.
  + The first two lines, MemTotal and MemFree, are written to the log file.
* **Logging Disk Usage**:
  + The statvfs() function is used to retrieve file system statistics about the root directory (/).
  + The total, used, and free disk space in MB is calculated and logged.
* **Logging Processes**:
  + The /proc/loadavg file is read, which contains load averages and the number of running processes. This data is logged.
* **Logging Network Usage**:
  + The /proc/net/dev file is checked for information related to network interfaces (eth0, wlan0).
  + The network statistics for these interfaces (like bytes sent and received) are logged.
* **Logging Swap Usage**:
  + The /proc/meminfo file is read again for SwapTotal and SwapFree values, which are logged to track swap space usage.
* **Closing the Log File**: After logging all the information, the file is closed.

**2. upload\_to\_dropbox(const char \*file\_path)**

This function uploads the log file (system\_usage.log) to Dropbox using the Dropbox API.

**Steps inside upload\_to\_dropbox():**

* **Initialize cURL**: The curl\_global\_init() function initializes the cURL library. curl\_easy\_init() creates a new cURL session.
* **Open Log File**: The log file is opened in binary read mode (rb).
* **Read File Content**: The entire file is read into a buffer so it can be uploaded.
* **Set Headers**: cURL headers are set for the HTTP request:
  + Authorization: Bearer <DROPBOX\_TOKEN> for Dropbox authentication.
  + Dropbox-API-Arg: {"path": "/system\_usage.log", "mode": "overwrite"} specifies the file path in Dropbox and the mode (overwrite existing file if it already exists).
  + Content-Type: application/octet-stream for the file data.
* **Perform Upload**: The cURL options are set, including the URL (https://content.dropboxapi.com/2/files/upload), HTTP method (POST), and file data. curl\_easy\_perform() sends the file to Dropbox.
* **Cleanup**: After the upload, curl\_easy\_cleanup() is called to free resources, and the log file is closed.

**3. main()**

The main() function runs an infinite loop that:

* Calls log\_system\_usage() to log system usage every minute.
* Calls upload\_to\_dropbox(LOG\_FILE) to upload the log file to Dropbox every minute.
* Pauses the execution for 60 seconds using sleep(60) before repeating the process.

**How the Program Works:**

1. **Infinite Loop**: The program enters an infinite loop in main() where it collects system resource data and uploads it to Dropbox every 60 seconds.
2. **Logging Data**: Every minute, the system usage is logged (CPU, memory, disk, network, etc.) and appended to the log file (system\_usage.log).
3. **Uploading to Dropbox**: After logging the data, the log file is uploaded to Dropbox using the Dropbox API.
4. **Syncing**: The log file is continuously uploaded, and with Dropbox syncing enabled on the mobile device, the log file can be accessed from both the desktop and mobile Dropbox apps.

**Requirements and Assumptions:**

1. **Dropbox Token**: You need a valid Dropbox API token for the upload function to work.
2. **Linux System**: This code relies on Linux-specific files (/proc/stat, /proc/meminfo, etc.) to get system information.
3. **cURL Library**: You must have the cURL library installed and linked during compilation for the Dropbox upload to function.
4. **Mobile Access**: The Dropbox mobile app must be installed and synchronized to allow users to access the log file on their phones.

**Summary of Key Features:**

* Logs detailed system resource usage (CPU, memory, disk, processes, network, swap).
* Uploads the log file to Dropbox every minute.
* Runs as a background service that continuously monitors and uploads data.