

# Folder Synchronization Program Report

## **Objective:**

The primary goal of this program is to synchronize the contents of two folders: a source folder and a replica folder. After synchronization, the contents of the replica folder should match the source folder exactly.

## **Overview:**

The program is designed for one-way synchronization, meaning that the source folder acts as the authoritative copy. Any discrepancies between the two folders are resolved in favor of the source folder. The synchronization process is repeated periodically based on a specified time interval. Moreover, any file operations such as creation, copying, or removal are logged both to the console and a designated log file.

## **Key Features:**

**One-Way Synchronization:** Changes in the source folder are reflected in the replica folder, but not vice versa. This ensures that the replica always mirrors the source without any back-propagation of changes.

**Periodic Synchronization:** The program continuously syncs the folders at regular intervals, ensuring up-to-date replication.

**Comprehensive Logging:** All synchronization operations are logged, providing transparency and a history of changes.

## **Design and Implementation**

### **MD5 Checksums for File Comparison:**

To determine if a file has changed, the program calculates the MD5 checksum of files in both the source and replica folders.

If the checksums differ, it indicates that the file contents are different, prompting the program to overwrite the file in the replica folder.

Using MD5 allows efficient comparison of file contents without reading the entire files line by line.

### **Recursive Folder Traversal:**

The program uses a recursive approach to handle nested folders.

If a sub-directory is encountered in the source, the function calls itself to synchronize the contents of that sub-directory.

### **Cleanup Mechanism:**

The program ensures that any extra files or directories present in the replica folder (which do not exist in the source) are removed. This is essential for achieving a true mirror of the source.

### **Logging Mechanism:**

A custom logging function is implemented to handle both console and file logging.

The function attempts to write to the provided log file path and also outputs to the console. Exception handling ensures that file access issues, like permission errors, are gracefully handled.

### **Command Line Interface:**

The program utilizes argparse to accept command-line arguments. This provides flexibility in specifying the source folder, replica folder, synchronization interval, and log file path.

## Decision-making

### Choice of MD5 for File Comparison:

MD5 was chosen due to its efficiency and widespread use. While there are concerns about MD5 in cryptographic contexts, for file comparison in synchronization tasks, it's apt and efficient.

### Infinite Loop for Periodic Sync:

An infinite while loop combined with `time.sleep()` is used to achieve periodic synchronization. This keeps the program running continuously, checking for changes at the specified intervals.

### Error Handling in Logging:

It's crucial to ensure that the synchronization process doesn't halt due to logging issues. Hence, exceptions related to file access are caught and displayed to the user without terminating the program.

```
C:\Users\Roque\.spyder-py3\My_py_programs>python folder_sync.py C:\Users\Roque\.spyder-py3\My_py_programs\source_folder C:\Users\Roque\.spyder-py3\My_py_programs\replica_folder 10 C:\Users\Roque\.spyder-py3\My_py_programs\sync_log.txt
File copied/overwritten: C:\Users\Roque\.spyder-py3\My_py_programs\source_folder\fil2.txt -> C:\Users\Roque\.spyder-py3\My_py_programs\replica_folder\fil2.txt
Traceback (most recent call last):
  File "C:\Users\Roque\.spyder-py3\My_py_programs\folder_sync.py", line 96, in <module>
    synchronize(src_folder, replica_folder)
  File "C:\Users\Roque\.spyder-py3\My_py_programs\folder_sync.py", line 43, in synchronize
    shutil.copy2(src_path, replica_path)
  File "C:\Users\Roque\AppData\Local\Programs\Python\Python311\Lib\shutil.py", line 436, in copy2
    copyfile(src, dst, follow_symlinks=follow_symlinks)
  File "C:\Users\Roque\AppData\Local\Programs\Python\Python311\Lib\shutil.py", line 256, in copyfile
    with open(src, 'rb') as fsrc:
        ^^^^^^^^^^^^^^^^^
PermissionError: [Errno 13] Permission denied: 'C:\\Users\\Roque\\.spyder-py3\\My_py_programs\\source_folder\\file1.txt'

C:\Users\Roque\.spyder-py3\My_py_programs>
```