**Università Degli Studi Niccolò Cusano**

**FACULTY OF ENGINEERING**

**Sistemi Operativi**

**PROJECT REPORT**



**Program Development Project in**

**C Language**

**Kaan Bilgili**

**ERASMUS 307**

**01.07.2025**

**1. Objective**

The objective of this project is to develop a C program that monitors a specified folder on a Windows system in real-time. The program detects and logs file system events such as ,

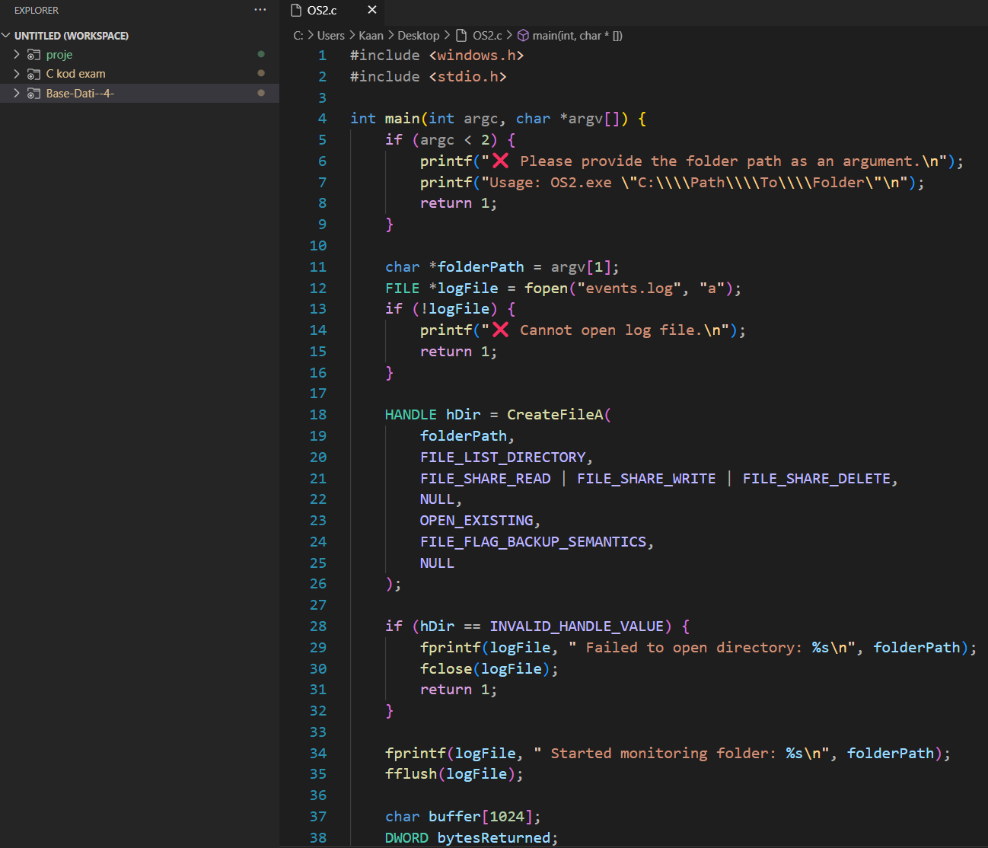
File creation

File modification (write operations) ,

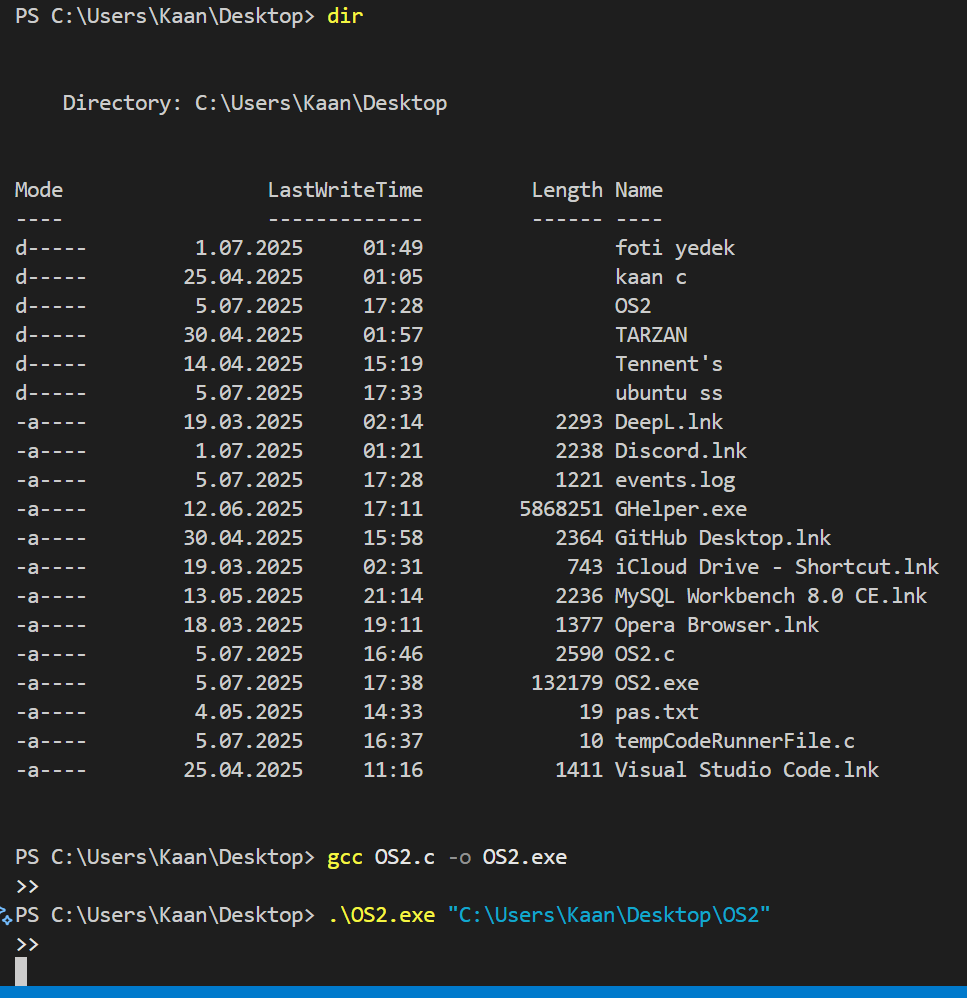
File deletion.

The program accepts the folder path as input and writes all detected events to a log file (events.log) with timestamps, enabling structured and continuous tracking of folder activity.

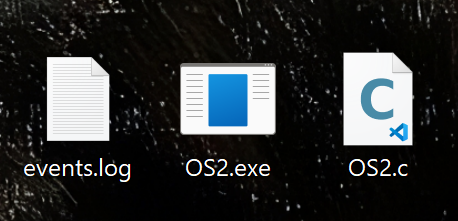
**2. Process**

****

This screenshot displays the first part of the C source code (OS2.c) written in Visual Studio Code. It shows the input validation for the folder path, log file creation (events.log), and the use of CreateFileA to open the specified directory for monitoring with Windows API constants.



This screenshot shows the compilation of OS2.c into OS2.exe using the command gcc OS2.c -o OS2.exe in the Windows PowerShell terminal. The compiled program is then executed with the target folder path (C:\Users\Kaan\Desktop\OS2) as input.

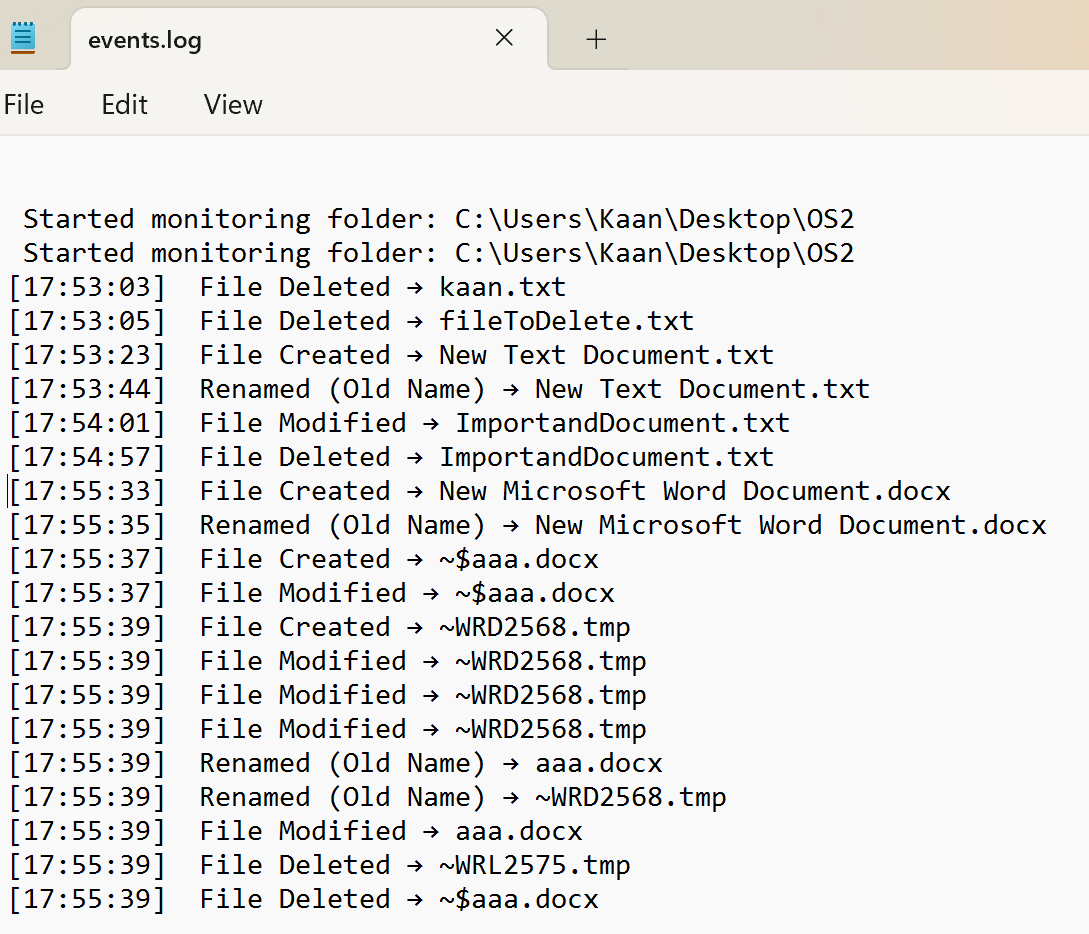


This image shows the three required output files for the project:

OS2.c: the C source code,

OS2.exe: the compiled executable,

events.log: the log file generated during monitoring.



This screenshot shows part of the events.log file created by the program. It clearly shows that the program is working — it detects and records when files are created, deleted, renamed, or changed. Each action is saved with the time and file name.

**3. Pseudocode- Structure**

The program begins by checking if the user has provided a folder path as a command-line argument. If not, it prints a usage message and exits. Once the input is validated, the program uses the Windows API to open the specified folder for monitoring and creates (or appends to) a log file called events.log. After this initial setup, the program enters an infinite loop that listens for file system changes using the ReadDirectoryChangesW function. Whenever a change is detected, the program determines the type of event (such as file creation, deletion, or modification), retrieves the current system time, and writes a descriptive log entry to events.log. This continues in real-time until the program is manually terminated.

**4. Conclusion**

The program effectively monitors a selected folder and logs file system events such as creation, modification, and deletion. It provides a simple but useful real-time tracking solution for folder activity on Windows.