**Simulating File Management System Using GPU**

**Project Introduction**

File management systems are one of the most crucial tasks undertaken by operating systems (OS). This mechanism is required so that the OS may simply write to and read from computer storage in relation to user files. Maintaining file metadata, offering APIs for reading, writing, and deleting files, and managing storage are some of the duties handled by file management systems. By challenging students to replicate a file management system utilizing a graphics processing unit (GPU), this assignment intends to improve students' comprehension of OS jobs in maintaining files in a computer.

**Program Design**

The simulation is capable of doing exactly six file management tasks: opening a file, writing to a file, reading from a file, listing all available files in order of when they were last edited, listing all accessible files in order of file size, and finally deleting a file. In addition to those six activities, we have developed nine more functions that will help the first six tasks go smoothly. We must create an FCB design that will serve as the basis for our implementation if we want all functionalities to operate as intended. FCB is the location where we keep all of our file metadata. We finally arrive at the following design after taking numerous FCB designs into consideration. There will be 32 bytes of data in each FCB entry. A total of 20 bytes are allotted for the file name, therefore it can only include a maximum of 19 characters plus the character "0" that denotes the end of the string. The file location and file size are represented by two bytes each in the following four bytes. The following 4 bytes are used to create a doubly linked list, which will be used to store the file's rank if it is sorted by modification time. If a file is sorted by file size, the last 4 bytes contain the file's "rank." Additionally, we allocated 4KB of memory to serve as an indicator of how much memory is used and how much is free. We can get a total of 1MB of RAM from that 4KB. We will therefore have a total of 1060KB allocated RAM (which includes the indication, the FCBs, and the content).

**Function Open file:**

The open function simply searches for block, find if there is any available file in the FCB based on the file name. If the file exists, we will return the hex of the FCB file to the users. However, is it is not found we will just return a 0x0000FFFF hex and create a file new file if wanted.

Program Compile Successfully with Open Function

