Operating Systems Project

File System (FS) – Part 1

Overview

In this project you will develop a simple File System.

The FS should operate on the simulated persistent memory with following properties:

- 20 Sectors, 64KB each
- Erase operation can be performed either on the single sector or on the whole simulated memory. Erase operation sets all **bits** of the sector(s) to value 1.
- Write operation can be performed only in WORDs and on a WORD boundary (WORD is defined as 2 bytes). Write operation can set **bits** to 0, but **cannot** set bits to 1, i.e., it is equal to the AND operation between the WORD on the simulated memory and the WORD to be written.
- Addressing is "flat" beginning with the 0 byte of the first sector

Part 1

Implement following "driver" operations (functions) operating on a simulated memory device:

Function	Description
EraseAllSectors ()	Sets all **bits** in all sectors in simulated memory to value 1. If necessary, create the file simulating the medium.
EraseSector (nSectorNr)	Sets all **bits** in the specified sector to 1. Sector numbers are 0-19. If file simulating simulated memory is not present, EraseAllSectors () is executed first
ReadWord (nAddress)	Reads a WORD (2 bytes) from the specified address. • nAddress – address of the WORD to be read. Address is the offset from the beginning of the simulated memory (i.e., byte 0 in Sector 0) and **not** from the particular sector. Address should be on the WORD boundary, i.e., addresses 0, 2, …, 2n are valid, addresses 1, 3, …, 2n-1 are not. If specified address is not WORD aligned or is outside the size of the simulated medium, this operation should fail.
WriteWord (nAddress, nWord)	Writes a WORD (2 bytes) to the specified address. • nAddress – address in which the WORD should be written • nWord – WORD to be written in the specified address If specified address is not WORD aligned or is outside the size of the simulated memory, this operation should fail, i.e., no information should be written in the simulated memory. If address is valid, the value written in the specified address should be equal to `nWord AND ReadWord (nAddress)` Whether this function returns value or not and what is the meaning of the returned value is a decision you should make.

Return values: It is your responsibility to decide whether these functions are void or return value, and which one in which case.

All these functions should be implemented using existing file operations provided by C/C++ and/or Windows API. The memory should be simulated using a **binary** file.

Note: if the file that simulates “physical medium” does not exist, you should create it whenever any of the “driver” functions is called.