# Assignment 3 – File System – GROUP PROJECT

Welcome to your third homework assignment, this is a GROUP assignment.

Again, code should be neat, well documented. Variables should have meaningful names and be in a consistent format (I do not care if you use camelCase or under_scores in variables but be consistent. In addition, each file should have a standard header as defined below.

```
/**************************************************************
* Class:  CSC-415-0# Spring 2020
* Group Name:
* Name:
* Student ID:
* Name:
* Student ID:
* Name:
* Student ID:
*
* Project: <project name, like Assignment 3 – File System
*
* File: <name of this file>
*
* Description:
*
**************************************************************/
```

This is a GROUP assignment written in C. Only one person on the team needs to submit the project.

Over the past month you and your team have been designing components of a file system. You have defined the goals and designed the directory entry structure, the volume structure and the free space. Now it is time to implement your file system.

To help I have written the low level LBA based read and write. The routines are in fsLow.c, the necessary header for you to include file is fsLow.h. You do NOT need to understand the code in fsLow, but you do need to understand the header file and the functions. There are 4 key functions:

```
int startPartitionSystem (char * filename, uint64_t * volSize, uint64_t * blockSize);
```

startPartitionSystem – you specify a file name that is the "volume" of your hard drive. The volume size is rounded to even block size and is only used when creating the volume file. The block size must be a power of 2 (nominally 512 bytes).

On Return, the function will return 0 if success or a negative number if an error occurs. The values pointed to by volSize and blockSize are filled in with the values of the existing volume file.

```
int closePartitionSystem ();
```

closePartitionSystem – closes down the volume file and must be called prior to terminating the process, but no read or writes can happen after this call.

```
uint64_t LBAwrite (void * buffer, uint64_t lbaCount, uint64_t lbaPosition);

uint64_t LBAread (void * buffer, uint64_t lbaCount, uint64_t lbaPosition);
```

LBAread and LBAwrite take a buffer, a count of LBA blocks and the starting LBA block number (0 based). The buffer must be large enough for the number of blocks * the block size.

On return, these function returns the number of blocks read or written.

# Assignment 3 – File System – GROUP PROJECT

The source file fsLowDriver.c is a simple driver for the driver system and should not be part of your program.

In addition, I have written a hexdump utility that will allow you to analyze your volume file.

Your assignment is to write a file system and a test "main" that will be the driver to test you file system. The driver should be interactive (with all built in commands) to list directories, create directories, add and remove files, copy files, move files, setting meta data, and two "special commands" one to copy from the normal filesystem to your filesystem and the other from your filesystem to the normal filesystem.

This is deliberately vague, as it is dependent on your filesystem design. And this is about all you will get initially for a real-world assignment, so if you have questions, please ask.

We will discuss some of this in class.

For our purposes use 10,000,000 or less (minimum 500,000) bytes for the volume size and 512 bytes per sector. These are the values to pass into *`startPartitionSystem`*.

What needs to be submitted:

- All source files (.c and .h)
- Driver program (must be a program that just utilizes the header file for your file system).
- The Driver program must be named: **fsdriver3.c**
- A make file (named "**makefile**") to build your entire program
  - Simple Tutorials on Make:
    - http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/
    - https://www.cs.swarthmore.edu/~newhall/unixhelp/howto_makefiles.html
    - https://makefiletutorial.com/

- A PDF writeup on project that should include:
  - A description of your file system
  - Issues you had
  - Detail of how your driver program works
  - Screen shots showing each of the commands listed above
- Your volume file (limit 10MB)