**Programmer’s Manual: Module 6**

Oreo Team

**Class List:**

Here are the classes, structs, unions, and interfaces:

* directoryEntry
* bootSector

**directoryEntry:**

#include <stdio.h>

**Public Attributes:**

char fileName

char ext

int attribute

int fileSize

int firstCluster

uint16\_t reserved

uint16\_t creationTime

uint16\_t creationDate

uint16\_t lastAccessDate

uint16\_t ignore

uint16\_t lastWriteTime

uint16\_t lastWriteDate

**bootSector:**

#include <stdio.h>

**Public Attributes:**

uint8\_t ignore

uint16\_t bytesPerSector

uint8\_t sectorsPerCluster

uint16\_t numOfReservedSectors

unint8\_t numOfFatCopies

uint16\_t maxNumberOfRootDirect

uint16\_t totalNumOfSectors

uint8\_t ignore2

uint16\_t numOfSectorsPerFat

uint16\_t sectorsPerTrack

uint16\_t numOfHeads

uint32\_t numOfHiddenSectors

uint32\_t totalSectorCount

uint8\_t ignore3

uint32\_t totalSCforF32

uint8\_t ignore4

uint8\_t bootSignature

uint32\_t volumeID

unsigned char volumeLabel

unsigned char fileSysType

uint8\_t ignore5

**File List:**

* r6.c
* r6.h

**r6.c File Reference:**

#include <stdint.h>

#include <stdio.h>

#include <stdlib.h>

#include “r6.h”

#include <string.h>

int FAT[4086]

int directory

char current\_directory[]

int quit

char current\_path

**Functions:**

int getInt(int sizeOfbytes);

param: sizeOfByte – bytes you want to convert

return: int – the converted byte to int type

This function converts the bytes to int type.

void moveToSector(int sector);

param: sector – sector you want to move to

return: void

This function moves to the sector given.

void printRoot(void);

param: void

return: void

This function calls the print root directory function.

void printDir(directoryEntry\* p, int num);

param: p – pointer to entry

param: num – number of entries in directory

return: void

This function prints the files and subdirectory in current directory.

void printEntry(directoryEntry entry);

param: entry – the entry to the directory

return: void

This function prints an entry’s attributes.

void Type(char\* name, char\* ext);

param: name – name of file

param: ext – extention of file

return: void

This function prints a file when given the name and extention.

void init();

param: void

return: void

This function initializes the boot and root directory structures.

int numberOfSectors(int startingSector);

param: startingSector – the beginning sector int

return: int – number of sectors

This function gets the number of sectors from the starting sector.

void locateDirectory(directoryEntry\* directory,int numberOfentries,int firstSector);

param: directory – current directory pointer

param: numberOfEntries – number of entries to cycle through

param: firstSector – the first sector int

return: void

This function searches for a directory by moving through sectors.

void interface(FILE\*);

param: file – the file pointer to the disk image

return: void

This function is the main interface, calls the command line for user input.

void command\_line(FILE\*);

param: file – the file pointer to the disk image

return: void

This function is the command line for the user input.

void help();

param: void

return: void

This function presents a list of commands and their use to the user.

void printBootSector();

param: void

return: void

This function prints the boot sector.

int renameFile(File \*, const char \*old , const char \* new);

param: file – the file pointer to the disk image

param: old – the original name of the file

param: new – the new name of the file

This function renames a file.

int Equals(const char \*str1, const char \* str2);

param: str1 – filename string

param: str2 – filename string

return: int – if matches or not

This function compares two filenames.

void change\_directory(char\* name);

param: name – directory name

return: void

This function changes the current directory.

void list\_directory();

param: void

return: void

This function lists everything in the current directory.

**r6.h File Reference:**

FILE\* fpointer;

int fatTale;

directoryEntry root[224];

directoryEntry\* curr;

int sizeOfCurr;

int startOfCurr;

#define color\_red "\x1b[31m"

#define color\_green "\x1b[32m"

#define color\_blue "\x1b[34m"

#define color\_yellow "\x1b[33m"

#define color\_clear "\x1b[0m"

**Structs & Typedefs:**

bootSector

directoryEntry