

# File System Manager

Generated by Doxygen 1.9.6



<b>1 Data Structure Index</b>	<b>1</b>
1.1 Data Structures	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 Data Structure Documentation</b>	<b>5</b>
3.1 _MBR Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 bootcode	5
3.1.2.2 part	5
3.1.2.3 signature	6
3.2 _partition Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Field Documentation	6
3.2.2.1 bootable	6
3.2.2.2 first_chs	6
3.2.2.3 last_chs	7
3.2.2.4 lba_offset	7
3.2.2.5 sector_count	7
3.2.2.6 type	7
3.3 BootSector Struct Reference	7
3.3.1 Detailed Description	8
3.3.2 Field Documentation	8
3.3.2.1 bytesPerSector	8
3.3.2.2 executableCode	8
3.3.2.3 executableMarker	8
3.3.2.4 extendedSignature	8
3.3.2.5 fatName	8
3.3.2.6 jumpCode	8
3.3.2.7 logicalDriveNumber	9
3.3.2.8 maxRootDirEntries	9
3.3.2.9 mediaDescriptor	9
3.3.2.10 numCopiesOfFAT	9
3.3.2.11 numHeads	9
3.3.2.12 numHiddenSectors	9
3.3.2.13 numSectorsLarge	9
3.3.2.14 numSectorsSmall	9
3.3.2.15 oemName	10
3.3.2.16 reservedSectors	10
3.3.2.17 sectorsPerCluster	10
3.3.2.18 sectorsPerFAT	10

3.3.2.19 sectorsPerTrack . . . . .	10
3.3.2.20 serialNumber . . . . .	10
3.3.2.21 volumeName . . . . .	10
3.4 FAT16Table Struct Reference . . . . .	11
3.4.1 Detailed Description . . . . .	11
3.4.2 Field Documentation . . . . .	11
3.4.2.1 entry . . . . .	11
3.5 RootDirectory Struct Reference . . . . .	11
3.5.1 Field Documentation . . . . .	12
3.5.1.1 count . . . . .	12
3.5.1.2 entries . . . . .	12
3.6 RootDirectoryEntry Struct Reference . . . . .	12
3.6.1 Detailed Description . . . . .	12
3.6.2 Field Documentation . . . . .	13
3.6.2.1 attributes . . . . .	13
3.6.2.2 creationDate . . . . .	13
3.6.2.3 creationTime . . . . .	13
3.6.2.4 ext . . . . .	13
3.6.2.5 filename . . . . .	13
3.6.2.6 fileSize . . . . .	13
3.6.2.7 lastAccessDate . . . . .	13
3.6.2.8 modifiedDate . . . . .	14
3.6.2.9 modifiedTime . . . . .	14
3.6.2.10 reserved1 . . . . .	14
3.6.2.11 reserved2 . . . . .	14
3.6.2.12 startingCluster . . . . .	14
<b>4 File Documentation</b>	<b>15</b>
4.1 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/FileManagerLibrary.h File Reference	15
4.1.1 Typedef Documentation . . . . .	17
4.1.1.1 ROOTDIRECTORY . . . . .	17
4.1.2 Function Documentation . . . . .	17
4.1.2.1 __attribute__() . . . . .	17
4.1.2.2 backToRootDir() . . . . .	18
4.1.2.3 ChangeDirectory() . . . . .	18
4.1.2.4 dumpBootSector() . . . . .	18
4.1.2.5 dumpDir() . . . . .	18
4.1.2.6 dumpFAT16Table() . . . . .	18
4.1.2.7 dumpMBR() . . . . .	18
4.1.2.8 ParseUSB() . . . . .	18
4.1.2.9 printData() . . . . .	19
4.1.2.10 ReadDir() . . . . .	19

---

4.1.2.11 readFatTables()	19
4.1.2.12 readFile()	20
4.1.2.13 readLBA()	20
4.1.2.14 ReadMBR()	20
4.1.2.15 readPartitions()	21
4.1.3 Variable Documentation	21
4.1.3.1 attributes	21
4.1.3.2 bootable	21
4.1.3.3 bootcode	21
4.1.3.4 bytesPerSector	21
4.1.3.5 creationDate	21
4.1.3.6 creationTime	21
4.1.3.7 entry	22
4.1.3.8 executableCode	22
4.1.3.9 executableMarker	22
4.1.3.10 ext	22
4.1.3.11 extendedSignature	22
4.1.3.12 fatName	22
4.1.3.13 filename	22
4.1.3.14 fileSize	22
4.1.3.15 first_chs	23
4.1.3.16 jumpCode	23
4.1.3.17 last_chs	23
4.1.3.18 lastAccessDate	23
4.1.3.19 lba_offset	23
4.1.3.20 logicalDriveNumber	23
4.1.3.21 maxRootDirEntries	23
4.1.3.22 mediaDescriptor	23
4.1.3.23 modifiedDate	24
4.1.3.24 modifiedTime	24
4.1.3.25 numCopiesOfFAT	24
4.1.3.26 numHeads	24
4.1.3.27 numHiddenSectors	24
4.1.3.28 numSectorsLarge	24
4.1.3.29 numSectorsSmall	24
4.1.3.30 oemName	24
4.1.3.31 part	25
4.1.3.32 reserved1	25
4.1.3.33 reserved2	25
4.1.3.34 reservedSectors	25
4.1.3.35 sector_count	25
4.1.3.36 sectorsPerCluster	25

4.1.3.37 sectorsPerFAT . . . . .	25
4.1.3.38 sectorsPerTrack . . . . .	25
4.1.3.39 serialNumber . . . . .	26
4.1.3.40 signature . . . . .	26
4.1.3.41 startingCluster . . . . .	26
4.1.3.42 type . . . . .	26
4.1.3.43 volumeName . . . . .	26
4.2 FileManagerLibrary.h . . . . .	26
4.3 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/utility.h File Reference . . . . .	28
4.3.1 Macro Definition Documentation . . . . .	29
4.3.1.1 FILE_ATTRIBUTE_ARCHIVE . . . . .	29
4.3.1.2 FILE_ATTRIBUTE_DIRECTORY . . . . .	29
4.3.1.3 FILE_ATTRIBUTE_HIDDEN . . . . .	29
4.3.1.4 FILE_ATTRIBUTE_LFN . . . . .	29
4.3.1.5 FILE_ATTRIBUTE_READONLY . . . . .	29
4.3.1.6 FILE_ATTRIBUTE_SYSTEM . . . . .	30
4.3.1.7 FILE_ATTRIBUTE_VOLUME . . . . .	30
4.3.2 Function Documentation . . . . .	30
4.3.2.1 EightDotThreeString() . . . . .	30
4.3.2.2 FileAttributeString() . . . . .	30
4.3.2.3 GetPathSeparator() . . . . .	30
4.3.2.4 HumanNumberString() . . . . .	30
4.3.2.5 MediaTypeString() . . . . .	31
4.3.2.6 PartitionTypeString() . . . . .	31
4.3.2.7 PrintHexDump() . . . . .	31
4.4 utility.h . . . . .	31
4.5 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/FileManagerLibrary.c File Reference . . . . .	31
4.5.1 Macro Definition Documentation . . . . .	32
4.5.1.1 DIRECTORY_ATTRIBUTE_SUBDIRECTORY . . . . .	32
4.5.2 Function Documentation . . . . .	32
4.5.2.1 backToRootDir() . . . . .	33
4.5.2.2 changeDirectory() . . . . .	33
4.5.2.3 dumpBootSector() . . . . .	33
4.5.2.4 dumpDir() . . . . .	33
4.5.2.5 dumpFAT16Table() . . . . .	33
4.5.2.6 dumpMBR() . . . . .	33
4.5.2.7 getOffset() . . . . .	33
4.5.2.8 ParseUSB() . . . . .	33
4.5.2.9 printData() . . . . .	34
4.5.2.10 ReadDir() . . . . .	34
4.5.2.11 readFatTables() . . . . .	34
4.5.2.12 readFile() . . . . .	35

4.5.2.13 readLBA()	35
4.5.2.14 ReadMBR()	35
4.5.2.15 readPartitions()	36
4.6 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/main.c File Reference	36
4.6.1 Function Documentation	36
4.6.1.1 main()	36
4.6.1.2 print_help()	37
4.7 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/utility.c File Reference	37
4.7.1 Function Documentation	37
4.7.1.1 EightDotThreeString()	37
4.7.1.2 FileAttributeString()	38
4.7.1.3 GetPathSeparator()	38
4.7.1.4 HumanNumberString()	38
4.7.1.5 MediaTypeString()	38
4.7.1.6 PartitionTypeString()	38
4.7.1.7 PrintHexDump()	38
<b>Index</b>	<b>39</b>





# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">_MBR</a>	5
<a href="#">_partition</a>	6
<a href="#">BootSector</a>	7
<a href="#">FAT16Table</a>	11
<a href="#">RootDirectory</a>	11
<a href="#">RootDirectoryEntry</a>	12



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/ <a href="#">FileManagerLibrary.h</a> . . . . .	15
C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/ <a href="#">utility.h</a> . . . . .	28
C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/ <a href="#">FileManagerLibrary.c</a> . . . . .	31
C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/ <a href="#">main.c</a> . . . . .	36
C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/ <a href="#">utility.c</a> . . . . .	37



## Chapter 3

# Data Structure Documentation

### 3.1 `_MBR` Struct Reference

```
#include <FileManagerLibrary.h>
```

#### Data Fields

- `uint8_t` [bootcode](#) [446]
- `partition` [part](#) [4]
- `uint16_t` [signature](#)

#### 3.1.1 Detailed Description

Defines the structure of the MBR.

#### 3.1.2 Field Documentation

##### 3.1.2.1 `bootcode`

```
uint8_t bootcode[446]
```

##### 3.1.2.2 `part`

```
partition part[4]
```

### 3.1.2.3 signature

```
uint16_t signature
```

The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

## 3.2 \_partition Struct Reference

```
#include <FileManagerLibrary.h>
```

### Data Fields

- uint8\_t [bootable](#)
- uint8\_t [first\\_chs](#) [3]
- uint8\_t [type](#)
- uint8\_t [last\\_chs](#) [3]
- uint32\_t [lba\\_offset](#)
- uint32\_t [sector\\_count](#)

### 3.2.1 Detailed Description

Defines the structure of a partition in the MBR.

### 3.2.2 Field Documentation

#### 3.2.2.1 bootable

```
uint8_t bootable
```

#### 3.2.2.2 first\_chs

```
uint8_t first_chs[3]
```

### 3.2.2.3 last\_chs

```
uint8_t last_chs[3]
```

### 3.2.2.4 lba\_offset

```
uint32_t lba_offset
```

### 3.2.2.5 sector\_count

```
uint32_t sector_count
```

### 3.2.2.6 type

```
uint8_t type
```

The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

## 3.3 BootSector Struct Reference

```
#include <FileManagerLibrary.h>
```

### Data Fields

- uint8\_t [jumpCode](#) [3]
- uint8\_t [oemName](#) [8]
- uint16\_t [bytesPerSector](#)
- uint8\_t [sectorsPerCluster](#)
- uint16\_t [reservedSectors](#)
- uint8\_t [numCopiesOfFAT](#)
- uint16\_t [maxRootDirEntries](#)
- uint16\_t [numSectorsSmall](#)
- uint8\_t [mediaDescriptor](#)
- uint16\_t [sectorsPerFAT](#)
- uint16\_t [sectorsPerTrack](#)
- uint16\_t [numHeads](#)
- uint32\_t [numHiddenSectors](#)
- uint32\_t [numSectorsLarge](#)
- uint16\_t [logicalDriveNumber](#)
- uint8\_t [extendedSignature](#)
- uint32\_t [serialNumber](#)
- uint8\_t [volumeName](#) [11]
- uint8\_t [fatName](#) [8]
- uint8\_t [executableCode](#) [448]
- uint16\_t [executableMarker](#)

### 3.3.1 Detailed Description

Defines the structure of a boot sector.

### 3.3.2 Field Documentation

#### 3.3.2.1 bytesPerSector

```
uint16_t bytesPerSector
```

#### 3.3.2.2 executableCode

```
uint8_t executableCode[448]
```

#### 3.3.2.3 executableMarker

```
uint16_t executableMarker
```

#### 3.3.2.4 extendedSignature

```
uint8_t extendedSignature
```

#### 3.3.2.5 fatName

```
uint8_t fatName[8]
```

#### 3.3.2.6 jumpCode

```
uint8_t jumpCode[3]
```



### 3.3.2.7 logicalDriveNumber

uint16\_t logicalDriveNumber

### 3.3.2.8 maxRootDirEntries

uint16\_t maxRootDirEntries

### 3.3.2.9 mediaDescriptor

uint8\_t mediaDescriptor

### 3.3.2.10 numCopiesOfFAT

uint8\_t numCopiesOfFAT

### 3.3.2.11 numHeads

uint16\_t numHeads

### 3.3.2.12 numHiddenSectors

uint32\_t numHiddenSectors

### 3.3.2.13 numSectorsLarge

uint32\_t numSectorsLarge

### 3.3.2.14 numSectorsSmall

uint16\_t numSectorsSmall

**3.3.2.15 oemName**

```
uint8_t oemName[8]
```

**3.3.2.16 reservedSectors**

```
uint16_t reservedSectors
```

**3.3.2.17 sectorsPerCluster**

```
uint8_t sectorsPerCluster
```

**3.3.2.18 sectorsPerFAT**

```
uint16_t sectorsPerFAT
```

**3.3.2.19 sectorsPerTrack**

```
uint16_t sectorsPerTrack
```

**3.3.2.20 serialNumber**

```
uint32_t serialNumber
```

**3.3.2.21 volumeName**

```
uint8_t volumeName[11]
```

The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

## 3.4 FAT16Table Struct Reference

```
#include <FileManagerLibrary.h>
```

### Data Fields

- uint16\_t [entry](#) [32768]

### 3.4.1 Detailed Description

Defines the structure of a FAT16 table.

### 3.4.2 Field Documentation

#### 3.4.2.1 entry

```
uint16_t entry[32768]
```

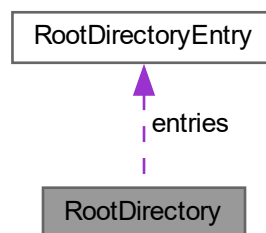
The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

## 3.5 RootDirectory Struct Reference

```
#include <FileManagerLibrary.h>
```

Collaboration diagram for RootDirectory:



## Data Fields

- [RootDirectoryEntry entries](#) [512]
- [int count](#)

### 3.5.1 Field Documentation

#### 3.5.1.1 count

```
int count
```

#### 3.5.1.2 entries

```
RootDirectoryEntry entries[512]
```

The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

## 3.6 RootDirectoryEntry Struct Reference

```
#include <FileManagerLibrary.h>
```

## Data Fields

- [uint8\\_t filename](#) [8]
- [uint8\\_t ext](#) [3]
- [uint8\\_t attributes](#)
- [uint16\\_t reserved1](#)
- [uint16\\_t creationTime](#)
- [uint16\\_t creationDate](#)
- [uint16\\_t lastAccessDate](#)
- [uint16\\_t reserved2](#)
- [uint16\\_t modifiedTime](#)
- [uint16\\_t modifiedDate](#)
- [uint16\\_t startingCluster](#)
- [uint32\\_t fileSize](#)

### 3.6.1 Detailed Description

Defines the structure of a root directory entry.

## 3.6.2 Field Documentation

### 3.6.2.1 attributes

uint8\_t attributes

### 3.6.2.2 creationDate

uint16\_t creationDate

### 3.6.2.3 creationTime

uint16\_t creationTime

### 3.6.2.4 ext

uint8\_t ext[3]

### 3.6.2.5 filename

uint8\_t filename[8]

### 3.6.2.6 fileSize

uint32\_t fileSize

### 3.6.2.7 lastAccessDate

uint16\_t lastAccessDate

### 3.6.2.8 modifiedDate

```
uint16_t modifiedDate
```

### 3.6.2.9 modifiedTime

```
uint16_t modifiedTime
```

### 3.6.2.10 reserved1

```
uint16_t reserved1
```

### 3.6.2.11 reserved2

```
uint16_t reserved2
```

### 3.6.2.12 startingCluster

```
uint16_t startingCluster
```

The documentation for this struct was generated from the following file:

- C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/[FileManagerLibrary.h](#)

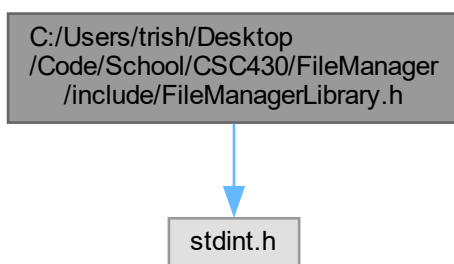
## Chapter 4

# File Documentation

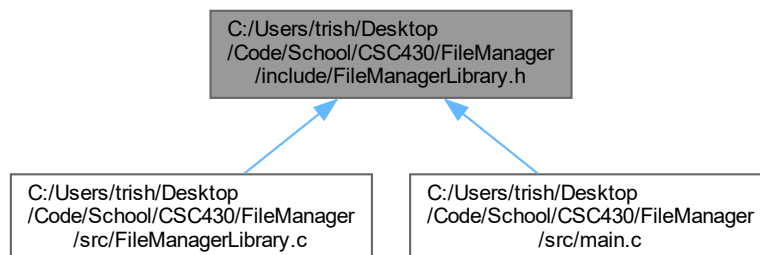
### 4.1 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/↔ FileManagerLibrary.h File Reference

```
#include <stdint.h>
```

Include dependency graph for FileManagerLibrary.h:



This graph shows which files directly or indirectly include this file:



## Data Structures

- struct [\\_partition](#)
- struct [BootSector](#)
- struct [\\_MBR](#)
- struct [FAT16Table](#)
- struct [RootDirectoryEntry](#)
- struct [RootDirectory](#)

## Typedefs

- typedef struct [RootDirectory](#) [ROOTDIRECTORY](#)

## Functions

- struct [\\_partition](#) [\\_\\_attribute\\_\\_\(\(packed\)\)](#) partition
- int [ReadMBR](#) (const char \*filename)
- int [ParseUSB](#) (const char \*filename)
- void [dumpMBR](#) ()
- void [dumpBootSector](#) ()
- void [printData](#) ()
- void [readPartitions](#) ()
- int [readLBA](#) (uint32\_t offset)
- int [readFatTables](#) (uint32\_t offset)
- int [ReadDir](#) (uint32\_t offset)
- int [ChangeDirectory](#) (const char \*dirName)
- void [backToRootDir](#) ()
- void [dumpDir](#) ()
- void [readFile](#) (char \*filename)
- void [dumpFAT16Table](#) ()

## Variables

- uint8\_t [bootable](#)
- uint8\_t [first\\_chs](#) [3]
- uint8\_t [type](#)
- uint8\_t [last\\_chs](#) [3]
- uint32\_t [lba\\_offset](#)
- uint32\_t [sector\\_count](#)
- uint8\_t [jumpCode](#) [3]
- uint8\_t [oemName](#) [8]
- uint16\_t [bytesPerSector](#)
- uint8\_t [sectorsPerCluster](#)
- uint16\_t [reservedSectors](#)
- uint8\_t [numCopiesOfFAT](#)
- uint16\_t [maxRootDirEntries](#)
- uint16\_t [numSectorsSmall](#)
- uint8\_t [mediaDescriptor](#)
- uint16\_t [sectorsPerFAT](#)
- uint16\_t [sectorsPerTrack](#)
- uint16\_t [numHeads](#)
- uint32\_t [numHiddenSectors](#)



- uint32\_t numSectorsLarge
- uint16\_t logicalDriveNumber
- uint8\_t extendedSignature
- uint32\_t serialNumber
- uint8\_t volumeName [11]
- uint8\_t fatName [8]
- uint8\_t executableCode [448]
- uint16\_t executableMarker
- uint8\_t bootcode [446]
- partition part [4]
- uint16\_t signature
- uint16\_t entry [32768]
- uint8\_t filename [8]
- uint8\_t ext [3]
- uint8\_t attributes
- uint16\_t reserved1
- uint16\_t creationTime
- uint16\_t creationDate
- uint16\_t lastAccessDate
- uint16\_t reserved2
- uint16\_t modifiedTime
- uint16\_t modifiedDate
- uint16\_t startingCluster
- uint32\_t fileSize

### 4.1.1 Typedef Documentation

#### 4.1.1.1 ROOTDIRECTORY

```
typedef struct RootDirectory ROOTDIRECTORY
```

### 4.1.2 Function Documentation

#### 4.1.2.1 \_\_attribute\_\_((packed))

```
struct RootDirectoryEntry __attribute__((packed)) {
```

Defines the structure of a partition in the MBR.

Defines the structure of a boot sector.

Defines the structure of the MBR.

Defines the structure of a FAT16 table.

Defines the structure of a root directory entry.

#### 4.1.2.2 backToRootDir()

```
void backToRootDir ( )
```

Changes the current working directory back to the root directory.

#### 4.1.2.3 ChangeDirectory()

```
int ChangeDirectory (
    const char * dirName )
```

Changes the current working directory to the specified directory.

##### Parameters

<i>dirName</i>	the name of the directory to change to.
----------------	-----------------------------------------

##### Returns

0 if the directory was successfully changed, otherwise returns a negative integer.

#### 4.1.2.4 dumpBootSector()

```
void dumpBootSector ( )
```

Prints the contents of the boot sector to the console.

#### 4.1.2.5 dumpDir()

```
void dumpDir ( )
```

Prints the contents of the current working directory to the console.

#### 4.1.2.6 dumpFAT16Table()

```
void dumpFAT16Table ( )
```

Prints the contents of the File Allocation Table (FAT) to the console.

#### 4.1.2.7 dumpMBR()

```
void dumpMBR ( )
```

Prints the contents of the MBR to the console.

#### 4.1.2.8 ParseUSB()

```
int ParseUSB (
    const char * filename )
```

Parses the USB device specified by the filename and populates the necessary data structures.

### Parameters

<i>filename</i>	the name of the USB device to parse.
-----------------	--------------------------------------

### Returns

0 if the USB device was successfully parsed, otherwise returns a negative integer.

#### 4.1.2.9 printData()

```
void printData ( )
```

Prints the parsed data from the USB device to the console.

#### 4.1.2.10 ReadDir()

```
int ReadDir (
    uint32_t offset )
```

Parses a directory given the offset of the directory

### Parameters

<i>offset</i>	location of the Directory
---------------	---------------------------

### Returns

EXIT\_SUCCESS OR EXIT\_FAILURE

#### 4.1.2.11 readFatTables()

```
int readFatTables (
    uint32_t offset )
```

Parses the fat tables via offset recieved from the LBA

### Parameters

<i>offset</i>	location of the Fat Tables
---------------	----------------------------

### Returns

EXIT\_SUCCESS OR EXIT\_FAILURE

#### 4.1.2.12 readFile()

```
void readFile (
    char * filename )
```

Reads and prints the contents of the specified file to the console.

##### Parameters

<i>filename</i>	the name of the file to read and print.
-----------------	-----------------------------------------

#### 4.1.2.13 readLBA()

```
int readLBA (
    uint32_t offset )
```

Reads the LBA recieved from the Partitions.

##### Parameters

<i>offset</i>	location of the LBA
---------------	---------------------

##### Returns

EXIT\_SUCCESS OR EXIT\_FAILURE

#### 4.1.2.14 ReadMBR()

```
int ReadMBR (
    const char * filename )
```

Reads the Master Boot Record (MBR) from the specified file.

##### Parameters

<i>filename</i>	the name of the file to read the MBR from.
-----------------	--------------------------------------------

##### Returns

0 if the MBR was successfully read, otherwise returns a negative integer.

#### 4.1.2.15 readPartitions()

```
void readPartitions ( )
```

Reads the partitions in the MBR

### 4.1.3 Variable Documentation

#### 4.1.3.1 attributes

```
uint8_t attributes
```

#### 4.1.3.2 bootable

```
uint8_t bootable
```

#### 4.1.3.3 bootcode

```
uint8_t bootcode[446]
```

#### 4.1.3.4 bytesPerSector

```
uint16_t bytesPerSector
```

#### 4.1.3.5 creationDate

```
uint16_t creationDate
```

#### 4.1.3.6 creationTime

```
uint16_t creationTime
```

#### 4.1.3.7 entry

```
uint16_t entry[32768]
```

#### 4.1.3.8 executableCode

```
uint8_t executableCode[448]
```

#### 4.1.3.9 executableMarker

```
uint16_t executableMarker
```

#### 4.1.3.10 ext

```
uint8_t ext[3]
```

#### 4.1.3.11 extendedSignature

```
uint8_t extendedSignature
```

#### 4.1.3.12 fatName

```
uint8_t fatName[8]
```

#### 4.1.3.13 filename

```
uint8_t filename[8]
```

#### 4.1.3.14 fileSize

```
uint32_t fileSize
```

#### 4.1.3.15 first\_chs

```
uint8_t first_chs[3]
```

#### 4.1.3.16 jumpCode

```
uint8_t jumpCode[3]
```

#### 4.1.3.17 last\_chs

```
uint8_t last_chs[3]
```

#### 4.1.3.18 lastAccessDate

```
uint16_t lastAccessDate
```

#### 4.1.3.19 lba\_offset

```
uint32_t lba_offset
```

#### 4.1.3.20 logicalDriveNumber

```
uint16_t logicalDriveNumber
```

#### 4.1.3.21 maxRootDirEntries

```
uint16_t maxRootDirEntries
```

#### 4.1.3.22 mediaDescriptor

```
uint8_t mediaDescriptor
```

**4.1.3.23 modifiedDate**

```
uint16_t modifiedDate
```

**4.1.3.24 modifiedTime**

```
uint16_t modifiedTime
```

**4.1.3.25 numCopiesOfFAT**

```
uint8_t numCopiesOfFAT
```

**4.1.3.26 numHeads**

```
uint16_t numHeads
```

**4.1.3.27 numHiddenSectors**

```
uint32_t numHiddenSectors
```

**4.1.3.28 numSectorsLarge**

```
uint32_t numSectorsLarge
```

**4.1.3.29 numSectorsSmall**

```
uint16_t numSectorsSmall
```

**4.1.3.30 oemName**

```
uint8_t oemName[8]
```



#### 4.1.3.31 part

```
partition part[4]
```

#### 4.1.3.32 reserved1

```
uint16_t reserved1
```

#### 4.1.3.33 reserved2

```
uint16_t reserved2
```

#### 4.1.3.34 reservedSectors

```
uint16_t reservedSectors
```

#### 4.1.3.35 sector\_count

```
uint32_t sector_count
```

#### 4.1.3.36 sectorsPerCluster

```
uint8_t sectorsPerCluster
```

#### 4.1.3.37 sectorsPerFAT

```
uint16_t sectorsPerFAT
```

#### 4.1.3.38 sectorsPerTrack

```
uint16_t sectorsPerTrack
```

#### 4.1.3.39 serialNumber

```
uint32_t serialNumber
```

#### 4.1.3.40 signature

```
uint16_t signature
```

#### 4.1.3.41 startingCluster

```
uint16_t startingCluster
```

#### 4.1.3.42 type

```
uint8_t type
```

#### 4.1.3.43 volumeName

```
uint8_t volumeName[11]
```

## 4.2 FileManagerLibrary.h

[Go to the documentation of this file.](#)

```
00001 #ifndef FILESYSTEMMANAGER_FILEMANAGERLIBRARY_H
00002 #define FILESYSTEMMANAGER_FILEMANAGERLIBRARY_H
00003
00004 #include <stdint.h>
00005
00009 typedef struct _partition {
00010     uint8_t bootable;
00011     uint8_t first_chs[3];
00012     uint8_t type;
00013     uint8_t last_chs[3];
00014     uint32_t lba_offset;
00015     uint32_t sector_count;
00016 } __attribute__((packed)) partition;
00017
00021 typedef struct BootSector {
00022     uint8_t jumpCode[3];
00023     uint8_t oemName[8];
00024     uint16_t bytesPerSector;
00025     uint8_t sectorsPerCluster;
00026     uint16_t reservedSectors;
00027     uint8_t numCopiesOfFAT;
00028     uint16_t maxRootDirEntries;
00029     uint16_t numSectorsSmall;
00030     uint8_t mediaDescriptor;
00031     uint16_t sectorsPerFAT;
00032     uint16_t sectorsPerTrack;
```

```

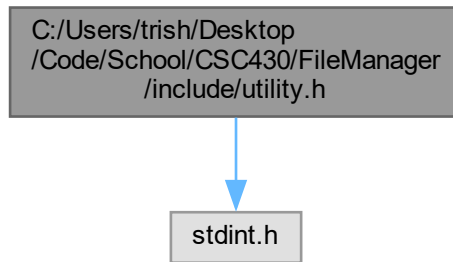
00033     uint16_t numHeads;
00034     uint32_t numHiddenSectors;
00035     uint32_t numSectorsLarge;
00036     uint16_t logicalDriveNumber;
00037     uint8_t extendedSignature;
00038     uint32_t serialNumber;
00039     uint8_t volumeName[11];
00040     uint8_t fatName[8];
00041     uint8_t executableCode[448];
00042     uint16_t executableMarker;
00043 } __attribute__((packed)) BootSector;
00044
00048 typedef struct _MBR {
00049     uint8_t bootcode[446];
00050     partition part[4];
00051     uint16_t signature;
00052 } __attribute__((packed)) MBR;
00053
00057 typedef struct FAT16Table {
00058     uint16_t entry[32768];
00059 } __attribute__((packed)) FAT16Table;
00060
00064 typedef struct RootDirectoryEntry {
00065     uint8_t filename[8];
00066     uint8_t ext[3];
00067     uint8_t attributes;
00068     uint16_t reserved1;
00069     uint16_t creationTime;
00070     uint16_t creationDate;
00071     uint16_t lastAccessDate;
00072     uint16_t reserved2;
00073     uint16_t modifiedTime;
00074     uint16_t modifiedDate;
00075     uint16_t startingCluster;
00076     uint32_t fileSize;
00077 } __attribute__((packed)) RootDirectoryEntry;
00078
00079 typedef struct RootDirectory {
00080     RootDirectoryEntry entries[512];
00081     int count;
00082 } ROOTDIRECTORY;
00083
00090 int ReadMBR(const char* filename);
00091
00098 int ParseUSB(const char* filename);
00099
00103 void dumpMBR();
00104
00108 void dumpBootSector();
00109
00113 void printData();
00114
00118 void readPartitions();
00119
00125 int readLBA(uint32_t offset);
00126
00132 int readFatTables(uint32_t offset);
00133
00139 int ReadDir(uint32_t offset);
00140
00147 int ChangeDirectory(const char* dirName);
00148
00152 void backToRootDir();
00153
00157 void dumpDir();
00158
00164 void readFile(char *filename);
00165
00169 void dumpFAT16Table();
00170
00171 #endif
00172

```

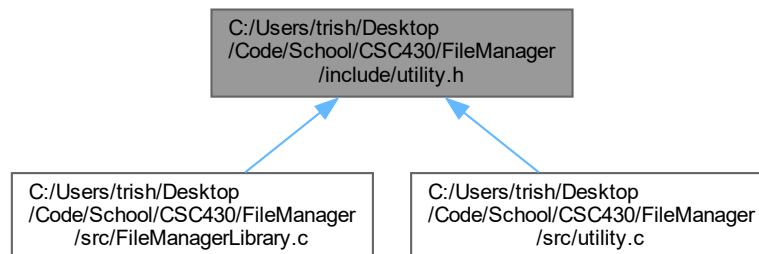
### 4.3 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/utility.h File Reference

```
#include <stdint.h>
```

Include dependency graph for utility.h:



This graph shows which files directly or indirectly include this file:



### Macros

- #define `FILE_ATTRIBUTE_READONLY` 0x01
- #define `FILE_ATTRIBUTE_HIDDEN` 0x02
- #define `FILE_ATTRIBUTE_SYSTEM` 0x04
- #define `FILE_ATTRIBUTE_VOLUME` 0x08
- #define `FILE_ATTRIBUTE_LFN`
- #define `FILE_ATTRIBUTE_DIRECTORY` 0x10
- #define `FILE_ATTRIBUTE_ARCHIVE` 0x20

## Functions

- const char \* [PartitionTypeString](#) (uint8\_t type\_code)
- const char \* [FileAttributeString](#) (uint8\_t attribute)
- const char \* [MediaTypeString](#) (int media\_type)
- const char \* [HumanNumberString](#) (uint64\_t size)
- const char \* [EightDotThreeString](#) (const uint8\_t name[8], const uint8\_t ext[3])
- const char \* [GetPathSeparator](#) (const char \*path)
- void [PrintHexDump](#) (const char \*buffer, int count)

## 4.3.1 Macro Definition Documentation

### 4.3.1.1 FILE\_ATTRIBUTE\_ARCHIVE

```
#define FILE_ATTRIBUTE_ARCHIVE 0x20
```

### 4.3.1.2 FILE\_ATTRIBUTE\_DIRECTORY

```
#define FILE_ATTRIBUTE_DIRECTORY 0x10
```

### 4.3.1.3 FILE\_ATTRIBUTE\_HIDDEN

```
#define FILE_ATTRIBUTE_HIDDEN 0x02
```

### 4.3.1.4 FILE\_ATTRIBUTE\_LFN

```
#define FILE_ATTRIBUTE_LFN
```

#### Value:

```
(FILE_ATTRIBUTE_READONLY | FILE_ATTRIBUTE_HIDDEN | \  
FILE_ATTRIBUTE_SYSTEM | FILE_ATTRIBUTE_VOLUME)
```

### 4.3.1.5 FILE\_ATTRIBUTE\_READONLY

```
#define FILE_ATTRIBUTE_READONLY 0x01
```

#### 4.3.1.6 FILE\_ATTRIBUTE\_SYSTEM

```
#define FILE_ATTRIBUTE_SYSTEM 0x04
```

#### 4.3.1.7 FILE\_ATTRIBUTE\_VOLUME

```
#define FILE_ATTRIBUTE_VOLUME 0x08
```

### 4.3.2 Function Documentation

#### 4.3.2.1 EightDotThreeString()

```
const char * EightDotThreeString (  
    const uint8_t name[8],  
    const uint8_t ext[3] )
```

#### 4.3.2.2 FileAttributeString()

```
const char * FileAttributeString (  
    uint8_t attribute )
```

#### 4.3.2.3 GetPathSeparator()

```
const char * GetPathSeparator (  
    const char * path )
```

#### 4.3.2.4 HumanNumberString()

```
const char * HumanNumberString (  
    uint64_t size )
```

#### 4.3.2.5 MediaTypeString()

```
const char * MediaTypeString (
    int media_type )
```

#### 4.3.2.6 PartitionTypeString()

```
const char * PartitionTypeString (
    uint8_t type_code )
```

#### 4.3.2.7 PrintHexDump()

```
void PrintHexDump (
    const char * buffer,
    int count )
```

## 4.4 utility.h

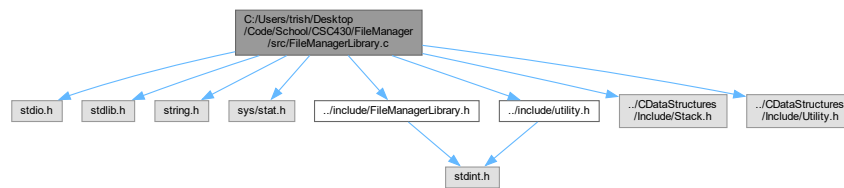
[Go to the documentation of this file.](#)

```
00001 #pragma once
00002
00003 #include <stdint.h>
00004
00005
00006 #define FILE_ATTRIBUTE_READONLY 0x01
00007 #define FILE_ATTRIBUTE_HIDDEN 0x02
00008 #define FILE_ATTRIBUTE_SYSTEM 0x04
00009 #define FILE_ATTRIBUTE_VOLUME 0x08
00010 #define FILE_ATTRIBUTE_LFN \
00011     (FILE_ATTRIBUTE_READONLY | FILE_ATTRIBUTE_HIDDEN | \
00012      FILE_ATTRIBUTE_SYSTEM | FILE_ATTRIBUTE_VOLUME)
00013 #define FILE_ATTRIBUTE_DIRECTORY 0x10
00014 #define FILE_ATTRIBUTE_ARCHIVE 0x20
00015
00016 const char* PartitionTypeString(uint8_t type_code);
00017 const char* FileAttributeString(uint8_t attribute);
00018 const char* MediaTypeString(int media_type);
00019 const char* HumanNumberString(uint64_t size);
00020 const char* EightDotThreeString(const uint8_t name[8], const uint8_t ext[3]);
00021 const char* GetPathSeparator(const char* path);
00022
00023 void PrintHexDump(const char* buffer, int count);
```

## 4.5 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/FileManagerLibrary.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/stat.h>
#include "../include/FileManagerLibrary.h"
#include "../include/utility.h"
```

```
#include "../CDataStructures/Include/Stack.h"
#include "../CDataStructures/Include/Utility.h"
Include dependency graph for FileManagerLibrary.c:
```



## Macros

- `#define DIRECTORY_ATTRIBUTE_SUBDIRECTORY 0x10`

## Functions

- void `printData` ()
- int `ParseUSB` (const char \*filename)
- int `ReadMBR` (const char \*filename)
- void `readPartitions` ()
- int `readLBA` (uint32\_t offset)
- int `readFatTables` (uint32\_t offset)
- int `ReadDir` (uint32\_t offset)
- uint32\_t `getOffset` (char \*subDirectoryName)
- void `changeDirectory` (char \*subDirectoryName)
- void `backToRootDir` ()
- void `dumpDir` ()
- void `dumpMBR` ()
- void `dumpBootSector` ()
- void `dumpFAT16Table` ()
- void `readFile` (char \*filename)

## 4.5.1 Macro Definition Documentation

### 4.5.1.1 DIRECTORY\_ATTRIBUTE\_SUBDIRECTORY

```
#define DIRECTORY_ATTRIBUTE_SUBDIRECTORY 0x10
```

## 4.5.2 Function Documentation



#### 4.5.2.1 backToRootDir()

```
void backToRootDir ( )
```

Changes the current working directory back to the root directory.

#### 4.5.2.2 changeDirectory()

```
void changeDirectory (
    char * subDirectoryName )
```

#### 4.5.2.3 dumpBootSector()

```
void dumpBootSector ( )
```

Prints the contents of the boot sector to the console.

#### 4.5.2.4 dumpDir()

```
void dumpDir ( )
```

Prints the contents of the current working directory to the console.

#### 4.5.2.5 dumpFAT16Table()

```
void dumpFAT16Table ( )
```

Prints the contents of the File Allocation Table (FAT) to the console.

#### 4.5.2.6 dumpMBR()

```
void dumpMBR ( )
```

Prints the contents of the MBR to the console.

#### 4.5.2.7 getOffset()

```
uint32_t getOffset (
    char * subDirectoryName )
```

#### 4.5.2.8 ParseUSB()

```
int ParseUSB (
    const char * filename )
```

Parses the USB device specified by the filename and populates the necessary data structures.

**Parameters**

<i>filename</i>	the name of the USB device to parse.
-----------------	--------------------------------------

**Returns**

0 if the USB device was successfully parsed, otherwise returns a negative integer.

**4.5.2.9 printData()**

```
void printData ( )
```

Prints the parsed data from the USB device to the console.

**4.5.2.10 ReadDir()**

```
int ReadDir (
    uint32_t offset )
```

Parses a directory given the offset of the directory

**Parameters**

<i>offset</i>	location of the Directory
---------------	---------------------------

**Returns**

EXIT\_SUCCESS OR EXIT\_FAILURE

**4.5.2.11 readFatTables()**

```
int readFatTables (
    uint32_t offset )
```

Parses the fat tables via offset recieved from the LBA

**Parameters**

<i>offset</i>	location of the Fat Tables
---------------	----------------------------

**Returns**

EXIT\_SUCCESS OR EXIT\_FAILURE

#### 4.5.2.12 readFile()

```
void readFile (
    char * filename )
```

Reads and prints the contents of the specified file to the console.

##### Parameters

<i>filename</i>	the name of the file to read and print.
-----------------	-----------------------------------------

#### 4.5.2.13 readLBA()

```
int readLBA (
    uint32_t offset )
```

Reads the LBA recieved from the Partitions.

##### Parameters

<i>offset</i>	location of the LBA
---------------	---------------------

##### Returns

EXIT\_SUCCESS OR EXIT\_FAILURE

#### 4.5.2.14 ReadMBR()

```
int ReadMBR (
    const char * filename )
```

Reads the Master Boot Record (MBR) from the specified file.

##### Parameters

<i>filename</i>	the name of the file to read the MBR from.
-----------------	--------------------------------------------

##### Returns

0 if the MBR was successfully read, otherwise returns a negative integer.

#### 4.5.2.15 readPartitions()

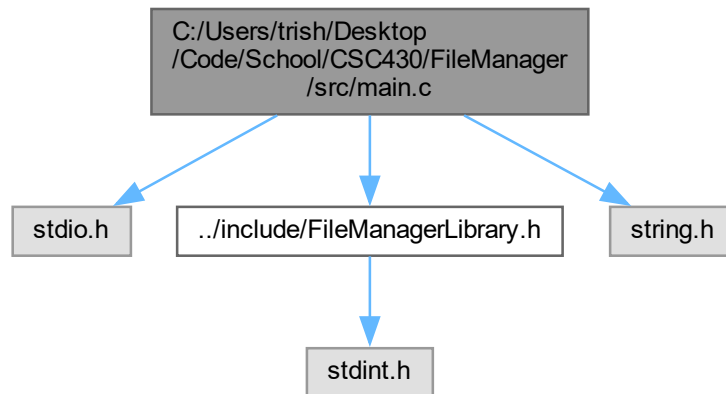
```
void readPartitions ( )
```

Reads the partitions in the MBR

## 4.6 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/main.c File Reference

```
#include <stdio.h>
#include "../include/FileManagerLibrary.h"
#include <string.h>
```

Include dependency graph for main.c:



### Functions

- void [print\\_help](#) ()
- int [main](#) ()

#### 4.6.1 Function Documentation

##### 4.6.1.1 main()

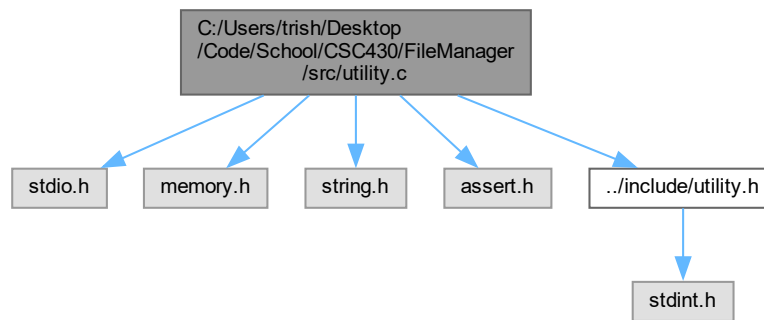
```
int main ( )
```

#### 4.6.1.2 print\_help()

```
void print_help ( )
```

### 4.7 C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/utility.c File Reference

```
#include <stdio.h>
#include <memory.h>
#include <string.h>
#include <assert.h>
#include "../include/utility.h"
Include dependency graph for utility.c:
```



## Functions

- `const char * PartitionTypeString (uint8_t type_code)`
- `const char * MediaTypeString (int media_type)`
- `const char * HumanNumberString (uint64_t size)`
- `const char * FileAttributeString (uint8_t attribute)`
- `const char * EightDotThreeString (const uint8_t name[8], const uint8_t ext[3])`
- `const char * GetPathSeparator (const char *path)`
- `void PrintHexDump (const char *buffer, int count)`

### 4.7.1 Function Documentation

#### 4.7.1.1 EightDotThreeString()

```
const char * EightDotThreeString (
    const uint8_t name[8],
    const uint8_t ext[3] )
```

#### 4.7.1.2 FileAttributeString()

```
const char * FileAttributeString (
    uint8_t attribute )
```

#### 4.7.1.3 GetPathSeparator()

```
const char * GetPathSeparator (
    const char * path )
```

#### 4.7.1.4 HumanNumberString()

```
const char * HumanNumberString (
    uint64_t size )
```

#### 4.7.1.5 MediaTypeString()

```
const char * MediaTypeString (
    int media_type )
```

#### 4.7.1.6 PartitionTypeString()

```
const char * PartitionTypeString (
    uint8_t type_code )
```

#### 4.7.1.7 PrintHexDump()

```
void PrintHexDump (
    const char * buffer,
    int count )
```

# Index

- [\\_MBR](#), [5](#)
  - [bootcode](#), [5](#)
  - [part](#), [5](#)
  - [signature](#), [5](#)
- [\\_\\_attribute\\_\\_](#)
  - [FileManagerLibrary.h](#), [17](#)
- [\\_partition](#), [6](#)
  - [bootable](#), [6](#)
  - [first\\_chs](#), [6](#)
  - [last\\_chs](#), [6](#)
  - [lba\\_offset](#), [7](#)
  - [sector\\_count](#), [7](#)
  - [type](#), [7](#)
- [attributes](#)
  - [FileManagerLibrary.h](#), [21](#)
  - [RootDirectoryEntry](#), [13](#)
- [backToRootDir](#)
  - [FileManagerLibrary.c](#), [32](#)
  - [FileManagerLibrary.h](#), [17](#)
- [bootable](#)
  - [\\_partition](#), [6](#)
  - [FileManagerLibrary.h](#), [21](#)
- [bootcode](#)
  - [\\_MBR](#), [5](#)
  - [FileManagerLibrary.h](#), [21](#)
- [BootSector](#), [7](#)
  - [bytesPerSector](#), [8](#)
  - [executableCode](#), [8](#)
  - [executableMarker](#), [8](#)
  - [extendedSignature](#), [8](#)
  - [fatName](#), [8](#)
  - [jumpCode](#), [8](#)
  - [logicalDriveNumber](#), [8](#)
  - [maxRootDirEntries](#), [9](#)
  - [mediaDescriptor](#), [9](#)
  - [numCopiesOfFAT](#), [9](#)
  - [numHeads](#), [9](#)
  - [numHiddenSectors](#), [9](#)
  - [numSectorsLarge](#), [9](#)
  - [numSectorsSmall](#), [9](#)
  - [oemName](#), [9](#)
  - [reservedSectors](#), [10](#)
  - [sectorsPerCluster](#), [10](#)
  - [sectorsPerFAT](#), [10](#)
  - [sectorsPerTrack](#), [10](#)
  - [serialNumber](#), [10](#)
  - [volumeName](#), [10](#)
- [bytesPerSector](#)

- [BootSector](#), [8](#)
- [FileManagerLibrary.h](#), [21](#)
- [C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/FileMa](#)
  - [15](#), [26](#)
- [C:/Users/trish/Desktop/Code/School/CSC430/FileManager/include/utility.h](#)
  - [28](#), [31](#)
- [C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/FileManag](#)
  - [31](#)
- [C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/main.c](#),
  - [36](#)
- [C:/Users/trish/Desktop/Code/School/CSC430/FileManager/src/utility.c](#),
  - [37](#)
- [ChangeDirectory](#)
  - [FileManagerLibrary.h](#), [18](#)
- [changeDirectory](#)
  - [FileManagerLibrary.c](#), [33](#)
- [count](#)
  - [RootDirectory](#), [12](#)
- [creationDate](#)
  - [FileManagerLibrary.h](#), [21](#)
  - [RootDirectoryEntry](#), [13](#)
- [creationTime](#)
  - [FileManagerLibrary.h](#), [21](#)
  - [RootDirectoryEntry](#), [13](#)
- [DIRECTORY\\_ATTRIBUTE\\_SUBDIRECTORY](#)
  - [FileManagerLibrary.c](#), [32](#)
- [dumpBootSector](#)
  - [FileManagerLibrary.c](#), [33](#)
  - [FileManagerLibrary.h](#), [18](#)
- [dumpDir](#)
  - [FileManagerLibrary.c](#), [33](#)
  - [FileManagerLibrary.h](#), [18](#)
- [dumpFAT16Table](#)
  - [FileManagerLibrary.c](#), [33](#)
  - [FileManagerLibrary.h](#), [18](#)
- [dumpMBR](#)
  - [FileManagerLibrary.c](#), [33](#)
  - [FileManagerLibrary.h](#), [18](#)
- [EightDotThreeString](#)
  - [utility.c](#), [37](#)
  - [utility.h](#), [30](#)
- [entries](#)
  - [RootDirectory](#), [12](#)
- [entry](#)
  - [FAT16Table](#), [11](#)
  - [FileManagerLibrary.h](#), [21](#)
- [executableCode](#)

- BootSector, [8](#)
  - FileManagerLibrary.h, [22](#)
- executableMarker
  - BootSector, [8](#)
  - FileManagerLibrary.h, [22](#)
- ext
  - FileManagerLibrary.h, [22](#)
  - RootDirectoryEntry, [13](#)
- extendedSignature
  - BootSector, [8](#)
  - FileManagerLibrary.h, [22](#)
- FAT16Table, [11](#)
  - entry, [11](#)
- fatName
  - BootSector, [8](#)
  - FileManagerLibrary.h, [22](#)
- FILE\_ATTRIBUTE\_ARCHIVE
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_DIRECTORY
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_HIDDEN
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_LFN
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_READONLY
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_SYSTEM
  - utility.h, [29](#)
- FILE\_ATTRIBUTE\_VOLUME
  - utility.h, [30](#)
- FileAttributeString
  - utility.c, [37](#)
  - utility.h, [30](#)
- FileManagerLibrary.c
  - backToRootDir, [32](#)
  - changeDirectory, [33](#)
  - DIRECTORY\_ATTRIBUTE\_SUBDIRECTORY, [32](#)
  - dumpBootSector, [33](#)
  - dumpDir, [33](#)
  - dumpFAT16Table, [33](#)
  - dumpMBR, [33](#)
  - getOffset, [33](#)
  - ParseUSB, [33](#)
  - printData, [34](#)
  - ReadDir, [34](#)
  - readFatTables, [34](#)
  - readFile, [35](#)
  - readLBA, [35](#)
  - ReadMBR, [35](#)
  - readPartitions, [35](#)
- FileManagerLibrary.h
  - \_\_attribute\_\_, [17](#)
  - attributes, [21](#)
  - backToRootDir, [17](#)
  - bootable, [21](#)
  - bootcode, [21](#)
  - bytesPerSector, [21](#)
  - ChangeDirectory, [18](#)
  - creationDate, [21](#)
  - creationTime, [21](#)
  - dumpBootSector, [18](#)
  - dumpDir, [18](#)
  - dumpFAT16Table, [18](#)
  - dumpMBR, [18](#)
  - entry, [21](#)
  - executableCode, [22](#)
  - executableMarker, [22](#)
  - ext, [22](#)
  - extendedSignature, [22](#)
  - fatName, [22](#)
  - filename, [22](#)
  - fileSize, [22](#)
  - first\_chs, [22](#)
  - jumpCode, [23](#)
  - last\_chs, [23](#)
  - lastAccessDate, [23](#)
  - lba\_offset, [23](#)
  - logicalDriveNumber, [23](#)
  - maxRootDirEntries, [23](#)
  - mediaDescriptor, [23](#)
  - modifiedDate, [23](#)
  - modifiedTime, [24](#)
  - numCopiesOfFAT, [24](#)
  - numHeads, [24](#)
  - numHiddenSectors, [24](#)
  - numSectorsLarge, [24](#)
  - numSectorsSmall, [24](#)
  - oemName, [24](#)
  - ParseUSB, [18](#)
  - part, [24](#)
  - printData, [19](#)
  - ReadDir, [19](#)
  - readFatTables, [19](#)
  - readFile, [20](#)
  - readLBA, [20](#)
  - ReadMBR, [20](#)
  - readPartitions, [20](#)
  - reserved1, [25](#)
  - reserved2, [25](#)
  - reservedSectors, [25](#)
  - ROOTDIRECTORY, [17](#)
  - sector\_count, [25](#)
  - sectorsPerCluster, [25](#)
  - sectorsPerFAT, [25](#)
  - sectorsPerTrack, [25](#)
  - serialNumber, [25](#)
  - signature, [26](#)
  - startingCluster, [26](#)
  - type, [26](#)
  - volumeName, [26](#)
- filename
  - FileManagerLibrary.h, [22](#)
  - RootDirectoryEntry, [13](#)
- fileSize
  - FileManagerLibrary.h, [22](#)
  - RootDirectoryEntry, [13](#)



- first\_chs
  - \_partition, 6
  - FileManagerLibrary.h, 22
- getOffset
  - FileManagerLibrary.c, 33
- GetPathSeparator
  - utility.c, 38
  - utility.h, 30
- HumanNumberString
  - utility.c, 38
  - utility.h, 30
- jumpCode
  - BootSector, 8
  - FileManagerLibrary.h, 23
- last\_chs
  - \_partition, 6
  - FileManagerLibrary.h, 23
- lastAccessDate
  - FileManagerLibrary.h, 23
  - RootDirectoryEntry, 13
- lba\_offset
  - \_partition, 7
  - FileManagerLibrary.h, 23
- logicalDriveNumber
  - BootSector, 8
  - FileManagerLibrary.h, 23
- main
  - main.c, 36
- main.c
  - main, 36
  - print\_help, 36
- maxRootDirEntries
  - BootSector, 9
  - FileManagerLibrary.h, 23
- mediaDescriptor
  - BootSector, 9
  - FileManagerLibrary.h, 23
- MediaTypeString
  - utility.c, 38
  - utility.h, 30
- modifiedDate
  - FileManagerLibrary.h, 23
  - RootDirectoryEntry, 13
- modifiedTime
  - FileManagerLibrary.h, 24
  - RootDirectoryEntry, 14
- numCopiesOfFAT
  - BootSector, 9
  - FileManagerLibrary.h, 24
- numHeads
  - BootSector, 9
  - FileManagerLibrary.h, 24
- numHiddenSectors
  - BootSector, 9
- FileManagerLibrary.h, 24
  - numSectorsLarge
    - BootSector, 9
    - FileManagerLibrary.h, 24
  - numSectorsSmall
    - BootSector, 9
    - FileManagerLibrary.h, 24
  - oemName
    - BootSector, 9
    - FileManagerLibrary.h, 24
  - ParseUSB
    - FileManagerLibrary.c, 33
    - FileManagerLibrary.h, 18
  - part
    - \_MBR, 5
    - FileManagerLibrary.h, 24
  - PartitionTypeString
    - utility.c, 38
    - utility.h, 31
  - print\_help
    - main.c, 36
  - printData
    - FileManagerLibrary.c, 34
    - FileManagerLibrary.h, 19
  - PrintHexDump
    - utility.c, 38
    - utility.h, 31
  - ReadDir
    - FileManagerLibrary.c, 34
    - FileManagerLibrary.h, 19
  - readFatTables
    - FileManagerLibrary.c, 34
    - FileManagerLibrary.h, 19
  - readFile
    - FileManagerLibrary.c, 35
    - FileManagerLibrary.h, 20
  - readLBA
    - FileManagerLibrary.c, 35
    - FileManagerLibrary.h, 20
  - ReadMBR
    - FileManagerLibrary.c, 35
    - FileManagerLibrary.h, 20
  - readPartitions
    - FileManagerLibrary.c, 35
    - FileManagerLibrary.h, 20
  - reserved1
    - FileManagerLibrary.h, 25
    - RootDirectoryEntry, 14
  - reserved2
    - FileManagerLibrary.h, 25
    - RootDirectoryEntry, 14
  - reservedSectors
    - BootSector, 10
    - FileManagerLibrary.h, 25
  - ROOTDIRECTORY
    - FileManagerLibrary.h, 17

- RootDirectory, 11
  - count, 12
  - entries, 12
- RootDirectoryEntry, 12
  - attributes, 13
  - creationDate, 13
  - creationTime, 13
  - ext, 13
  - filename, 13
  - fileSize, 13
  - lastAccessDate, 13
  - modifiedDate, 13
  - modifiedTime, 14
  - reserved1, 14
  - reserved2, 14
  - startingCluster, 14
- sector\_count
  - \_partition, 7
  - FileManagerLibrary.h, 25
- sectorsPerCluster
  - BootSector, 10
  - FileManagerLibrary.h, 25
- sectorsPerFAT
  - BootSector, 10
  - FileManagerLibrary.h, 25
- sectorsPerTrack
  - BootSector, 10
  - FileManagerLibrary.h, 25
- serialNumber
  - BootSector, 10
  - FileManagerLibrary.h, 25
- signature
  - \_MBR, 5
  - FileManagerLibrary.h, 26
- startingCluster
  - FileManagerLibrary.h, 26
  - RootDirectoryEntry, 14
- type
  - \_partition, 7
  - FileManagerLibrary.h, 26
- utility.c
  - EightDotThreeString, 37
  - FileAttributeString, 37
  - GetPathSeparator, 38
  - HumanNumberString, 38
  - MediaTypeString, 38
  - PartitionTypeString, 38
  - PrintHexDump, 38
- utility.h
  - EightDotThreeString, 30
  - FILE\_ATTRIBUTE\_ARCHIVE, 29
  - FILE\_ATTRIBUTE\_DIRECTORY, 29
  - FILE\_ATTRIBUTE\_HIDDEN, 29
  - FILE\_ATTRIBUTE\_LFN, 29
  - FILE\_ATTRIBUTE\_READONLY, 29
  - FILE\_ATTRIBUTE\_SYSTEM, 29
  - FILE\_ATTRIBUTE\_VOLUME, 30
  - FileAttributeString, 30
  - GetPathSeparator, 30
  - HumanNumberString, 30
  - MediaTypeString, 30
  - PartitionTypeString, 31
  - PrintHexDump, 31
- volumeName
  - BootSector, 10
  - FileManagerLibrary.h, 26