

PotatOS

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	BOOTSECTORSTRUCT Struct Reference . . . . .	5
3.2	command Struct Reference . . . . .	5
3.2.1	Detailed Description . . . . .	5
3.3	ENTRY Struct Reference . . . . .	6
3.3.1	Detailed Description . . . . .	6
3.4	PREVDIR Struct Reference . . . . .	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	command_utils.h File Reference . . . . .	9
4.1.1	Detailed Description . . . . .	10
4.1.2	Typedef Documentation . . . . .	10
4.1.2.1	ENTRY . . . . .	10
4.1.3	Function Documentation . . . . .	11
4.1.3.1	memcpyUpper(char *, const char *, int) . . . . .	11
4.1.3.2	split_args(char *command, char **argv) . . . . .	12
4.1.3.3	starsearch(const char *starryboi, const char *fileboi) . . . . .	12
4.1.3.4	trim_whitespace(char *str) . . . . .	12
4.2	commands.h File Reference . . . . .	12

4.2.1	Detailed Description	14
4.2.2	Function Documentation	14
4.2.2.1	cd_command(int argc, char **argv)	14
4.2.2.2	exit_command(int argc, char **argv)	14
4.2.2.3	help_command(int argc, char **argv)	15
4.2.2.4	ls_command(int argc, char **argv)	15
4.2.2.5	move_command(int argc, char **argv)	15
4.2.2.6	pbsi_command(int argc, char **argv)	16
4.2.2.7	prd_command(int argc, char **argv)	16
4.2.2.8	rename_command(int argc, char **argv)	16
4.2.2.9	root_command(int argc, char **argv)	17
4.2.2.10	search_commands(char *command)	17
4.2.2.11	type_command(int argc, char **argv)	18
4.3	file_wrangler.h File Reference	18
4.3.1	Detailed Description	20
4.3.2	Function Documentation	20
4.3.2.1	getBootSectorIn()	20
4.3.2.2	getCWD()	20
4.3.2.3	getCwdPath()	20
4.3.2.4	getDiabetes1()	20
4.3.2.5	getDiabetes2()	21
4.3.2.6	getImagePath()	21
4.3.2.7	getRoot()	21
4.3.2.8	getSystem()	21
4.3.2.9	getSystemSize()	21
4.3.2.10	loadCWD(BYTE *whole, uint32_t startingSec)	21
4.3.2.11	loadDir(BYTE **dirp, BYTE *sys, uint32_t startingSec)	22
4.3.2.12	loadEntireSystem(char *filename)	22
4.3.2.13	loadFAT(BYTE *fpln, uint32_t startingSector)	22
4.3.2.14	loadROOT(BYTE *sys)	22
4.3.2.15	setCwdPath(const char *jerry)	23
4.4	module6.c File Reference	23
4.4.1	Detailed Description	24
4.4.2	Function Documentation	24
4.4.2.1	file_recurse(BYTE *dir, BYTE *whole, const char *key)	24

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">BOOTSECTORSTRUCT</a>	5
<a href="#">command</a>	
A struct to hold command information	5
<a href="#">ENTRY</a>	
A struct to hold an directory entry	6
<a href="#">PREVDIR</a>	7



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all documented files with brief descriptions:

<a href="#">command_utils.h</a>	Holds standard typedefs and utility function headers for certain commands . . . . .	9
<a href="#">commands.h</a>	Holds all of the command prototypes . . . . .	12
<a href="#">file_wrangler.h</a>	Holds all of the prototypes for file_wrangler.c and has BOOTSECTOR struct . . . . .	18
<a href="#">module6.c</a>	Entry point for file system . . . . .	23





## Chapter 3

# Data Structure Documentation

### 3.1 BOOTSECTORSTRUCT Struct Reference

#### Data Fields

- uint16\_t **bytesPerSector**
- uint16\_t **sectorsPerCluster**
- uint16\_t **numReservedSectors**
- uint16\_t **numFATCopies**
- uint16\_t **maxNumRoot**
- uint16\_t **numOfSectors**
- uint16\_t **numSECTORSPEFAT**
- uint16\_t **sectorsPerTrack**
- uint16\_t **numberHeads**
- uint16\_t **sectorCountFAT32**
- uint16\_t **bootSig**
- uint32\_t **volumeID**
- char \* **volumeLabel**
- char \* **fileSystemType**

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

- [file\\_wrangler.h](#)

### 3.2 command Struct Reference

A struct to hold command information.

```
#include <command_utils.h>
```

#### Data Fields

- char \* **command\_name**
- CommandFunction **function**
- char \* **command\_usage**
- char \* **command\_description**
- char \* **command\_examples**

#### 3.2.1 Detailed Description

A struct to hold command information.

**Parameters**

<i>command_name</i>	The command's name
<i>function</i>	A pointer to the command's function
<i>command_usage</i>	Syntax for executing the command
<i>command_description</i>	A description of the command
<i>command_examples</i>	Example uses of the command

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

- [command\\_utils.h](#)

**3.3 ENTRY Struct Reference**

A struct to hold an directory entry.

```
#include <command_utils.h>
```

**Data Fields**

- **BYTE empty**
- **char fileName [9]**
- **char extension [4]**
- **BYTE attributes**
- **uint16\_t reserved**
- **uint16\_t creationTime**
- **uint8\_t creationHour**
- **uint8\_t creationMin**
- **uint8\_t creationSec**
- **uint16\_t creationDate**
- **uint16\_t creationYear**
- **uint8\_t creationMonth**
- **uint8\_t creationDay**
- **uint16\_t lastAccessDate**
- **uint16\_t lastAccessYear**
- **uint8\_t lastAccessMonth**
- **uint8\_t lastAccessDay**
- **uint16\_t lastWriteTime**
- **uint8\_t lastWriteHour**
- **uint8\_t lastWriteMin**
- **uint8\_t lastWriteSec**
- **uint16\_t lastWriteDate**
- **uint16\_t lastWriteYear**
- **uint8\_t lastWriteMonth**
- **uint8\_t lastWriteDay**
- **uint16\_t firstLogicalCluster**
- **uint32\_t fileSize**

**3.3.1 Detailed Description**

A struct to hold an directory entry.

## Parameters

<i>empty</i>	If the entry is empty
<i>fileName</i>	The name of the entry
<i>extension</i>	The extension, type, of the entry
<i>attributes</i>	The attributes of the entry
<i>reserved</i>	If the entry is reserved
<i>creationTime</i>	Entry creation time
<i>creationHour</i>	Entry creation hour
<i>creationMin</i>	Entry creation min
<i>creationSec</i>	Entry creation sec
<i>creationDate</i>	Entry creation date
<i>creationYear</i>	Entry creation year
<i>creationMonth</i>	Entry creation month
<i>creationDay</i>	Entry creation day
<i>lastAccessDate</i>	Entry access date
<i>lastAccessYear</i>	Entry access year
<i>lastAccessMonth</i>	Entry access month
<i>lastAccessDay</i>	Entry access day
<i>lastWriteTime</i>	Entry last write time
<i>lastWriteHour</i>	Entry last write hour
<i>lastWriteMin</i>	Entry last write min
<i>lastWriteSec</i>	Entry last write sec
<i>lastWriteDate</i>	Entry last write date
<i>lastWriteYear</i>	Entry last write year
<i>lastWriteMonth</i>	Entry last write month
<i>lastWriteDay</i>	Entry last write day
<i>firstLogicalCluster</i>	First logical cluster
<i>fileSize</i>	Entry filesize

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

- [command\\_utils.h](#)

## 3.4 PREVDIR Struct Reference

## Data Fields

- char \* **dirName**
- uint32\_t **sectorStart**

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

- [command\\_utils.h](#)



## Chapter 4

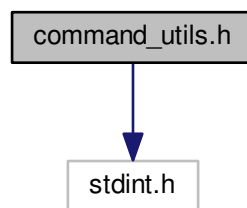
# File Documentation

### 4.1 command\_utils.h File Reference

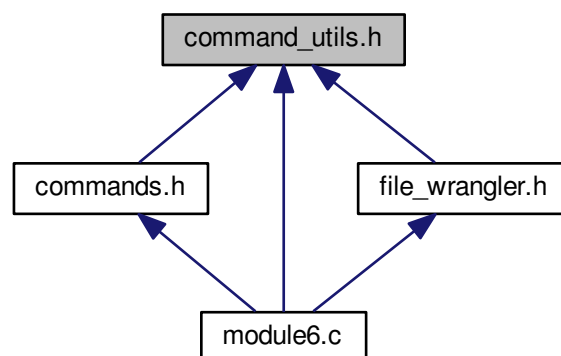
Holds standard typedefs and utility function headers for certain commands.

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

Include dependency graph for command\_utils.h:



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



## Data Structures

- struct [command](#)  
*A struct to hold command information.*
- struct [ENTRY](#)  
*A struct to hold an directory entry.*
- struct [PREVDIR](#)

## Macros

- #define **SECTORSIZE** 512
- #define **SECTORSPERFAT** 9
- #define **FATTABLESIZE** ((SECTORSPERFAT \* SECTORSIZE \* 8) / 12)
- #define **MAXDIRENTRY** 16
- #define **MAXENTRIESPERDIR** 16
- #define **DEBLOC**() printf("\033[33;1m%s(%d)\033[0m\n", \_\_FILE\_\_, \_\_LINE\_\_)

## Typedefs

- typedef unsigned char **BYTE**
- typedef int(\* **CommandFunction**) (int, char \*\*)
- typedef struct [ENTRY](#) **ENTRY**  
*A struct to hold an directory entry.*
- typedef struct [PREVDIR](#) **PREVDIR**
- typedef struct [command](#) **command\_t**

## Functions

- char \* [trim\\_whitespace](#) (char \*str)  
*A function to trim strings.*
- uint32\_t [split\\_args](#) (char \*[command](#), char \*\*argv)  
*A function to split up a input string into arguments for command functions.*
- void [memcpyUpper](#) (char \*, const char \*, int)  
*A memcpy function that takes all characters to their uppercase equivalent.*
- int **strcimp** (char const \*a, char const \*b)
- uint32\_t [starsearch](#) (const char \*starryboi, const char \*fileboi)  
*Star search for ls.*
- uint16\_t **findDotPosition** (const char \*)

### 4.1.1 Detailed Description

Holds standard typedefs and utility function headers for certain commands.

### 4.1.2 Typedef Documentation

#### 4.1.2.1 typedef struct [ENTRY](#) **ENTRY**

A struct to hold an directory entry.

## Parameters

<i>empty</i>	If the entry is empty
<i>fileName</i>	The name of the entry
<i>extension</i>	The extension, type, of the entry
<i>attributes</i>	The attributes of the entry
<i>reserved</i>	If the entry is reserved
<i>creationTime</i>	Entry creation time
<i>creationHour</i>	Entry creation hour
<i>creationMin</i>	Entry creation min
<i>creationSec</i>	Entry creation sec
<i>creationDate</i>	Entry creation date
<i>creationYear</i>	Entry creation year
<i>creationMonth</i>	Entry creation month
<i>creationDay</i>	Entry creation day
<i>lastAccessDate</i>	Entry access date
<i>lastAccessYear</i>	Entry access year
<i>lastAccessMonth</i>	Entry access month
<i>lastAccessDay</i>	Entry access day
<i>lastWriteTime</i>	Entry last write time
<i>lastWriteHour</i>	Entry last write hour
<i>lastWriteMin</i>	Entry last write min
<i>lastWriteSec</i>	Entry last write sec
<i>lastWriteDate</i>	Entry last write date
<i>lastWriteYear</i>	Entry last write year
<i>lastWriteMonth</i>	Entry last write month
<i>lastWriteDay</i>	Entry last write day
<i>firstLogicalCluster</i>	First logical cluster
<i>fileSize</i>	Entry filesize

## 4.1.3 Function Documentation

## 4.1.3.1 void memcpyUpper ( char \*, const char \*, int )

A memcpy function that takes all characters to their uppercase equivalent.

## Parameters

<i>dest</i>	The destination char pointer
<i>source</i>	The source char pointer
<i>nchars</i>	The number of chars to copy, and toUpper, from source to dest

## Returns

void method, copies in place

#### 4.1.3.2 `uint32_t split_args ( char * command, char ** argv )`

A function to split up a input string into arguments for command functions.

##### Parameters

<i>The</i>	command input to split up
<i>A</i>	pointer to the destination of the argument slices, argv

##### Returns

The amount of arguments split up, argc

#### 4.1.3.3 `uint32_t starsearch ( const char * starryboi, const char * fileboi )`

Star search for ls.

##### Parameters

<i>starryboi</i>	The char pointer that contains a star search
------------------	--

#### 4.1.3.4 `char* trim_whitespace ( char * str )`

A function to trim strings.

##### Parameters

<i>str</i>	The string to trim
------------	--------------------

##### Returns

A pointer to the beginning of the trimmed string

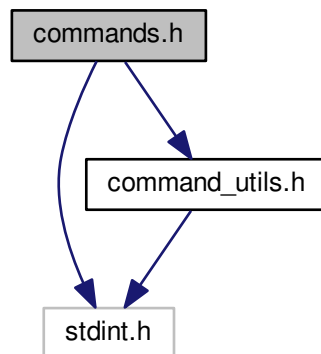
## 4.2 `commands.h` File Reference

Holds all of the command prototypes.

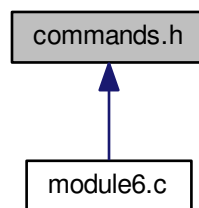
```
#include <stdint.h>
#include "command_utils.h"
```



Include dependency graph for commands.h:



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



## Functions

- `command_t * search_commands (char *command)`  
*Search commands for a given command.*
- `int help_command (int argc, char **argv)`  
*Command to get help about other commands.*
- `int pbsi_command (int argc, char **argv)`  
*print boot sector information*
- `int prd_command (int argc, char **argv)`  
*print root directory*
- `int root_command (int argc, char **argv)`  
*change directory to root directory*
- `int cd_command (int argc, char **argv)`  
*change directory to requested directory*
- `int ls_command (int argc, char **argv)`

- list directory*
- int `type_command` (int argc, char \*\*argv)  
*prints out contents of file*
- int `rename_command` (int argc, char \*\*argv)  
*renames a file or directory*
- int `move_command` (int argc, char \*\*argv)  
*move a file to an adjacent directory*
- int `exit_command` (int argc, char \*\*argv)  
*exit the m6 environnement*

### 4.2.1 Detailed Description

Holds all of the command prototypes.

### 4.2.2 Function Documentation

#### 4.2.2.1 int cd\_command ( int argc, char \*\* argv )

change directory to requested directory

##### Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

##### Returns

0

#### 4.2.2.2 int exit\_command ( int argc, char \*\* argv )

exit the m6 environnement

##### Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

##### Returns

0

exit the m6 environnement

Confirms user wants to exit, and saves the state of the image.

## Parameters

<i>argc</i>	Number of arguments
<i>argv</i>	List of arguments. (None are applicable).

## Returns

int Success condition.

4.2.2.3 int help\_command ( int *argc*, char \*\* *argv* )

Command to get help about other commands.

## Parameters

<i>argc</i>	Number of arguments
<i>argv</i>	Pointer to arguments

## Returns

exit status code

4.2.2.4 int ls\_command ( int *argc*, char \*\* *argv* )

list directory

## Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

## Returns

0

4.2.2.5 int move\_command ( int *argc*, char \*\* *argv* )

move a file to an adjacent directory

## Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

**Returns**

0

move a file to an adjacent directory

Takes the name of a file and directory within the same directory, and moves the file entry into the specified directory.

**Parameters**

<i>argc</i>	Number of arguments.
<i>argv</i>	List of arguments <i>argv</i> [1] is the file to move, <i>argv</i> [2] is the destination directory.

**Returns**

int Success condition.

**4.2.2.6 int pbsi\_command ( int *argc*, char \*\* *argv* )**

print boot sector information

**Parameters**

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

**Returns**

0

**4.2.2.7 int prd\_command ( int *argc*, char \*\* *argv* )**

print root directory

**Parameters**

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

**Returns**

0

**4.2.2.8 int rename\_command ( int *argc*, char \*\* *argv* )**

renames a file or directory

## Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

## Returns

0

4.2.2.9 int root\_command ( int *argc*, char \*\* *argv* )

change directory to root directory

## Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

## Returns

0

change directory to root directory

Sends user's current working directory to the root.

## Parameters

<i>argc</i>	Number of arguments
<i>argv</i>	List of string arguments.

## Returns

int Success condition.

4.2.2.10 command\_t\* search\_commands ( char \* *command* )

Search commands for a given command.

## Parameters

<i>command</i>	A command name to search for
----------------	------------------------------

## Returns

Pointer to command if found, else NULL

#### 4.2.2.11 int type\_command ( int argc, char \*\* argv )

prints out contents of file

##### Parameters

<i>argc</i>	count of arguments passed in
<i>argv</i>	list of strings separated by space characters passed in

##### Returns

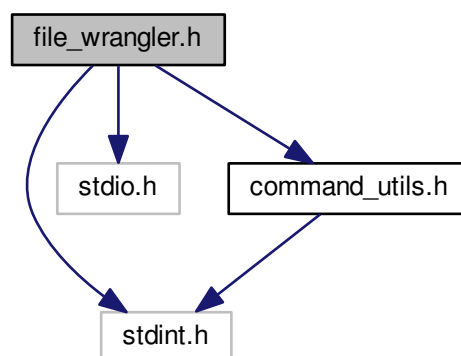
0

## 4.3 file\_wrangler.h File Reference

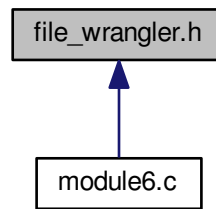
Holds all of the prototypes for file\_wrangler.c and has BOOTSECTOR struct.

```
#include <stdint.h>
#include <stdio.h>
#include "command_utils.h"
```

Include dependency graph for file\_wrangler.h:



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



## Data Structures

- struct [BOOTSECTORSTRUCT](#)

## Typedefs

- typedef struct [BOOTSECTORSTRUCT](#) **BOOTSECTORSTRUCT**

## Functions

- void **loadBootSector** (FILE \*fpIn)
- uint16\_t \* **loadFAT** (BYTE \*fpIn, uint32\_t startingSector)  
*loads the FAT*
- void **loadDir** (BYTE \*\*dirp, BYTE \*sys, uint32\_t startingSec)  
*loads a directory*
- void **loadCWD** (BYTE \*whole, uint32\_t startingSec)  
*loads the current working directory into global BYTE\* cwd*
- void **loadROOT** (BYTE \*sys)  
*loads the root into global BYTE\* root*
- void **loadEntireSystem** (char \*filename)  
*loads the entire system*
- [BOOTSECTORSTRUCT](#) \* **getBootSectorIn** ()  
*gets boot sector in*
- BYTE \* **getSystem** ()  
*gets entire system pointer*
- uint32\_t **getSystemSize** ()  
*gets entire system size*
- uint16\_t \* **getDiabetes1** ()  
*gets fat table 1*
- uint16\_t \* **getDiabetes2** ()  
*gets fat table 2*
- BYTE \* **getCWD** ()  
*gets current working directory BYTE\**
- BYTE \* **getRoot** ()

- gets the root directory*
- char \* [getCwdPath](#) ()  
*gets current working directory path*
- char \* [getImagePath](#) ()  
*gets root directory*
- void [setCwdPath](#) (const char \*jerry)  
*sets cwd path*

### 4.3.1 Detailed Description

Holds all of the prototypes for file\_wrangler.c and has BOOTSECTOR struct.

### 4.3.2 Function Documentation

#### 4.3.2.1 BOOTSECTORSTRUCT\* getBootSectorIn ( )

gets boot sector in

##### Returns

pointer to boot sector in

#### 4.3.2.2 BYTE\* getCWD ( )

gets current working directory BYTE\*

##### Returns

pointer to current working directory

#### 4.3.2.3 char\* getCwdPath ( )

gets current working directory path

##### Returns

char pointer that contains current working directory path

#### 4.3.2.4 uint16\_t\* getDiabetes1 ( )

gets fat table 1

##### Returns

pointer to fat table 1



#### 4.3.2.5 uint16\_t\* getDiabetes2 ( )

gets fat table 2

##### Returns

pointer to fat table 2

#### 4.3.2.6 char\* getImagePath ( )

gets root directory

##### Returns

pointer to root directory

#### 4.3.2.7 BYTE\* getRoot ( )

gets the root directory

##### Returns

pointer to the root directory

#### 4.3.2.8 BYTE\* getSystem ( )

gets entire system pointer

##### Returns

pointer to entire system

#### 4.3.2.9 uint32\_t getSystemSize ( )

gets entire system size

##### Returns

u32 int that is system size

#### 4.3.2.10 void loadCWD ( BYTE \* *whole*, uint32\_t *startingSec* )

loads the current working directory into global BYTE\* cwd

## Parameters

<i>sys</i>	raw byte array of entire file system
<i>startingSector</i>	number of the sector to start loading the cwd from

4.3.2.11 void loadDir ( BYTE \*\* *dirp*, BYTE \* *sys*, uint32\_t *startingSec* )

loads a directory

## Parameters

<i>sys</i>	raw byte array of entire file system
<i>startingSector</i>	number of the sector to start loading the fat from

## Returns

pointer to FAT

4.3.2.12 void loadEntireSystem ( char \* *filename* )

loads the entire system

## Parameters

<i>filename</i>	filename of the correct FAT12 Image
-----------------	-------------------------------------

4.3.2.13 uint16\_t\* loadFAT ( BYTE \* *fpln*, uint32\_t *startingSector* )

loads the FAT

## Parameters

<i>fpln</i>	raw byte array of entire file system
<i>startingSector</i>	number of the sector to start loading the fat from

## Returns

pointer to FAT

4.3.2.14 void loadROOT ( BYTE \* *sys* )

loads the root into global BYTE\* root

## Parameters

<code>sys</code>	raw byte array of entire file system
------------------	--------------------------------------

4.3.2.15 void setCwdPath ( const char \* *jerry* )

sets cwd path

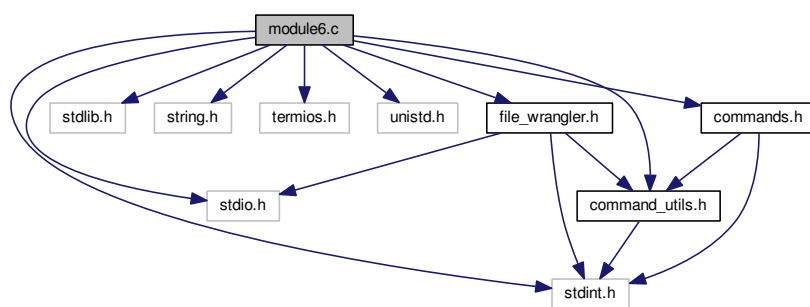
## Parameters

<i>jerry</i>	string to make cwdPath
--------------	------------------------

## 4.4 module6.c File Reference

Entry point for file system.

```
#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <termios.h>
#include <unistd.h>
#include "command_utils.h"
#include "commands.h"
#include "file_wrangler.h"
Include dependency graph for module6.c:
```



## Functions

- **ENTRY** \* `file_recurse` (BYTE \*dir, BYTE \*whole, const char \*key)  
Searches through an entire file system for a file.
- int **main** (int argc, char \*\*argv)

#### 4.4.1 Detailed Description

Entry point for file system.

#### 4.4.2 Function Documentation

##### 4.4.2.1 ENTRY \* file\_recurse ( BYTE \* *dir*, BYTE \* *whole*, const char \* *key* )

Searches through an entire file system for a file.

Using a recursive search, this function will search through an entire image to find a specified file. This then returns an entry for this file.

##### Parameters

<i>dir</i>	Current searching directory
<i>whole</i>	Pointer to the system.
<i>key</i>	Name of the file to search for, including extension.

##### Returns

ENTRY\* Entry of the file that was being searched for. NULL if the file is not found.

# Index

BOOTSECTORSTRUCT, 5

cd\_command

commands.h, 14

command, 5

command\_utils.h, 9

ENTRY, 10

memcpyUpper, 11

split\_args, 11

starsearch, 12

trim\_whitespace, 12

commands.h, 12

cd\_command, 14

exit\_command, 14

help\_command, 15

ls\_command, 15

move\_command, 15

pbsi\_command, 16

prd\_command, 16

rename\_command, 16

root\_command, 17

search\_commands, 17

type\_command, 17

ENTRY, 6

command\_utils.h, 10

exit\_command

commands.h, 14

file\_recurse

module6.c, 24

file\_wrangler.h, 18

getBootSectorIn, 20

getCWD, 20

getCwdPath, 20

getDiabetes1, 20

getDiabetes2, 20

getImagePath, 21

getRoot, 21

getSystem, 21

getSystemSize, 21

loadCWD, 21

loadDir, 22

loadEntireSystem, 22

loadFAT, 22

loadROOT, 22

setCwdPath, 23

getBootSectorIn

file\_wrangler.h, 20

getCWD

file\_wrangler.h, 20

getCwdPath

file\_wrangler.h, 20

getDiabetes1

file\_wrangler.h, 20

getDiabetes2

file\_wrangler.h, 20

getImagePath

file\_wrangler.h, 21

getRoot

file\_wrangler.h, 21

getSystem

file\_wrangler.h, 21

getSystemSize

file\_wrangler.h, 21

help\_command

commands.h, 15

loadCWD

file\_wrangler.h, 21

loadDir

file\_wrangler.h, 22

loadEntireSystem

file\_wrangler.h, 22

loadFAT

file\_wrangler.h, 22

loadROOT

file\_wrangler.h, 22

ls\_command

commands.h, 15

memcpyUpper

command\_utils.h, 11

module6.c, 23

file\_recurse, 24

move\_command

commands.h, 15

PREVDIR, 7

pbsi\_command

commands.h, 16

prd\_command

commands.h, 16

rename\_command

commands.h, 16

root\_command

commands.h, 17

- search\_commands
  - commands.h, [17](#)
- setCwdPath
  - file\_wrangler.h, [23](#)
- split\_args
  - command\_utils.h, [11](#)
- starsearch
  - command\_utils.h, [12](#)
- trim\_whitespace
  - command\_utils.h, [12](#)
- type\_command
  - commands.h, [17](#)