File shredder

Generated by Doxygen 1.9.1

Chapter 1

File shredder documentation

Version

1.0

Date

December, 2021

Destination

This project was designed for shredding files in disks.

Using

The utility fills with zeros the memory where the original file was, so the data that was there before cannot be restored. Specify the path to the file to destroy as the only command line argument. For example: ***$ shredd file.txt***

Building

You need to use "make" command to duild this projet.

Chapter 2

File Index

2.1 File List

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

Chapter 3

File Documentation

3.1 filecleaner.h File Reference

File with functions for file shredding.

```
#include <assert.h>
#include <stdio.h>
#include <stdint.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
```

Enumerations

- enum CODES {
  SUCCESS = 0 , ERROR_NULLPTR = 1 , ERROR_FOPEN = 2 , ERROR_NULLSIZE = 3 ,
  ERROR_BUFFER = 4 , ERROR_ALLOC = 5 , ERROR_FWRITE = 6 , ERROR_RMFILE = 7 ,
  ERROR_NOFILE = 8 }

  Enum for return codes of functions in this file.

Functions

- uint8_t destroy_file (const char ∗file_name)

  Function for destroying a file.
- uint64_t get_file_size (const char ∗file_name)

  Function for getting a file size.
- uint8_t fill_in_file (FILE ∗file, const uint64_t file_size)

  Function for filling file by zeros.
- const char ∗ status_description (uint8_t status_code)

  Function for getting text description of CODES elements.

3.1.1 Detailed Description

File with functions for file shredding.

Author

Maxmosk

Version

1.0

Date

December, 2021

3.1.2 Enumeration Type Documentation

3.1.2.1 CODES

```
enum CODES
```

Enum for return codes of functions in this file.

Note

Used only in functions with uint8_t type

Enumerator

SUCCESS	Successful performance of the function.
ERROR_NULLPTR	A null pointer was passed as a parameter.
ERROR_FOPEN	Error in function 'fopen'.
ERROR_NULLSIZE	File has zero size.
ERROR_BUFFER	Error in setting the buffer size for writing to a file.
ERROR_ALLOC	Error in memory allocation function.
ERROR_FWRITE	Error in function 'fwrite'.
ERROR_RMFILE	Error in the removing file function.
ERROR_NOFILE	File with the specified name does not exist.

3.1.3 Function Documentation

3.1.3.1 destroy_file()

```
uint8_t destroy_file (
             const char * file_name )
```

Function for destroying a file.

Parameters

in	*file_name*	Name of file for destroying

Returns

Code from enum CODES

3.1.3.2 fill_in_file()

```
uint8_t fill_in_file (
             FILE * file,
             const uint64_t file_size )
```

Function for filling file by zeros.

Parameters

in	*file*	Pointer to FILE for filling
in	*fize_size*	Size of data for filling

Returns

Code from enum CODES

3.1.3.3 get_file_size()

```
uint64_t get_file_size (
             const char * file_name )
```

Function for getting a file size.

Parameters

in	*file_name*	Name of file for getting the size

Returns

Size of file from stat or zero in case of null pointer error or stat error

3.1.3.4 status_description()

```
const char* status_description (
            uint8_t status_code )
```

Function for getting text description of CODES elements.

Parameters

in	*status_code*	Element of CODES to get it's description

Returns

Pointer to string with description of element of CODES

3.2 main.c File Reference

File with main program for file shredding.

```
#include <assert.h>
#include <stdio.h>
#include "filecleaner.h"
```

Functions

- int main (int argc, char ∗argv[ ])
  Function for shredding files from command line.

3.2.1 Detailed Description

File with main program for file shredding.

Author

Maxmosk

Version

1.0

Date

December, 2021

Index