My Project

Generated by Doxygen 1.12.0

Chapter 1

File Index

1.1 File List

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

Chapter 2

File Documentation

2.1 main.c File Reference

Handle the main functions of the 'Secured' project.

```
#include "hashtable.h"
```

Functions

- hashtable_t ∗ new_hashtable (int(∗hash)(char ∗, int), int len)

  This is a function that adds two integers.
- int ht_delete (hashtable_t ∗ht, char ∗key)

  This is a function that deletes a node from the whole hashtable.
- int hash (char ∗key, int len)

  This is a hashes a given key.
- char ∗ ht_search (hashtable_t ∗ht, char ∗key)

  This is a function that search a key in a hashtable.
- void ht_dump (hashtable_t ∗ht)

  This is a function that displays the whole hashtable.

2.1.1 Detailed Description

Handle the main functions of the 'Secured' project.

Includes new_hashtable | ht_insert | ht_delete | ht_searh

2.1.2 Function Documentation

2.1.2.1 hash()

```
int hash (
              char * key,
              int len)
```

This is a hashes a given key.

Parameters

∗*key*	The key to hash
len	The size of the key

Returns

int A simple status code

2.1.2.2 ht_delete()

```
int ht_delete (
             hashtable_t * ht,
             char * key)
```

This is a function that deletes a node from the whole hashtable.

Parameters

∗*ht*	The structure which contains some params and the hashstable
∗*key*	The key to find and delete in the hashtable

Returns

int A simple status code

2.1.2.3 ht_dump()

```
void ht_dump (
             hashtable_t * ht)
```

This is a function that displays the whole hashtable.

Parameters

∗*ht*	The structure that contains the hashtable to display

Returns

void

2.1.2.4 ht_search()

```
char * ht_search (
             hashtable_t * ht,
             char * key)
```

This is a function that search a key in a hashtable.

Parameters

∗*ht*	The structure which contains the hashtable
∗*key*	The key to find

Returns

char∗ The value linked to the key or NULL if key not found

2.1.2.5 new_hashtable()

```
hashtable_t * new_hashtable (
              int(* hash )(char *, int),
              int len)
```

This is a function that adds two integers.

Parameters

a	The first integer
b	The second integer

Returns

int The sum of a and b

This is a function that create and initialize a new hashtable inside a structure

Parameters

hash	A pointer to a function that hashes the keys
len	The size of the new hashtable to create

Returns

hashtable_t∗ A pointer to the new stucture which contains the hashtable.

Index