HashMap

Generated by Doxygen 1.9.8

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

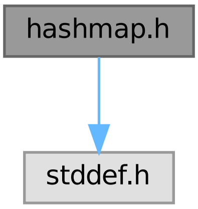
Chapter 2

File Documentation

2.1 hashmap.h File Reference

```
#include <stddef.h>
```

Include dependency graph for hashmap.h:



Typedefs

- typedef struct HashMap ∗ HashMap
- typedef size_t(∗ hash) (const void ∗key, size_t key_size)

Functions

- HashMap hm_create (size_t hm_capacity, size_t key_size, size_t value_size)

  Creates a new HashMap with a generic hash function.
- HashMap hm_create_ch (size_t hm_capacity, size_t key_size, size_t value_size, hash hash_func)

  Creates a new HashMap with a custom hash function.
- int hm_destroy (HashMap hm)

  Destroys the HashMap.
- void ∗ hm_get (HashMap hm, const void ∗key)

  Retrieves the value at the specified key.
- int hm_set (HashMap hm, const void ∗key, const void ∗value)

Updates the value at the specified key.

- int hm_put (HashMap hm, const void ∗key, const void ∗value)

  Adds a new key-value pair to the HashMap.
- size_t hm_size (HashMap hm)

  Returns the size of the HashMap.
- int hm_remove (HashMap hm, const void ∗key)

  Removes the key-value pair from the HashMap.

2.1.1 Typedef Documentation

2.1.1.1 hash

```
typedef size_t(* hash) (const void *key, size_t key_size)
```

2.1.1.2 HashMap

```
typedef struct HashMap* HashMap
```

2.1.2 Function Documentation

2.1.2.1 hm_create()

```
HashMap hm_create (
             size_t hm_capacity,
             size_t key_size,
             size_t value_size )
```

Creates a new HashMap with a generic hash function.

Parameters

hm_capacity	The initial capacity of the HashMap
key_size	The sizeof value of the key
value_size	The sizeof value of the value

Returns

HashMap

2.1.2.2 hm_create_ch()

```
HashMap hm_create_ch (
             size_t hm_capacity,
             size_t key_size,
             size_t value_size,
             hash hash_func )
```

Creates a new HashMap with a custom hash function.

Parameters

hm_capacity	The initial capacity of the HashMap
key_size	The sizeof value of the key
value_size	The sizeof value of the value
hash_func	The custom hash function

Returns

HashMap

2.1.2.3 hm_destroy()

```
int hm_destroy (
            HashMap hm )
```

Destroys the HashMap.

Parameters

hm	The HashMap

Returns

Success code

2.1.2.4 hm_get()

```
void * hm_get (
            HashMap hm,
            const void * key )
```

Retrieves the value at the specified key.

Parameters

hm	The HashMap
key	The key

Returns

The value at the key

2.1.2.5 hm_put()

```
int hm_put (
            HashMap hm,
            const void * key,
            const void * value )
```

Adds a new key-value pair to the HashMap.

Parameters

hm	The HashMap
key	The key
value	The value

Returns

Success code

2.1.2.6 hm_remove()

```
int hm_remove (
                HashMap hm,
                const void * key )
```

Removes the key-value pair from the HashMap.

Parameters

hm	The HashMap
key	The key

Returns

Success code

2.1.2.7 hm_set()

```
int hm_set (
                HashMap hm,
                const void * key,
                const void * value )
```

Updates the value at the specified key.

Parameters

hm	The HashMap
key	The key
value	The value

Returns

Success code

2.1.2.8 hm_size()

```
size_t hm_size (
            HashMap hm )
```

Returns the size of the HashMap.

Parameters

hm	The HashMap

Returns

The size

2.2 hashmap.h

Go to the documentation of this file.

```
//
// Created by Emanuel on 02.09.2024.
//

#ifndef HASHMAP_H
#define HASHMAP_H

#include <stddef.h>

typedef struct HashMap *HashMap;
typedef size_t (*hash)(const void *key, size_t key_size);

HashMap hm_create(size_t hm_capacity, size_t key_size, size_t value_size);

HashMap hm_create_ch(size_t hm_capacity, size_t key_size, size_t value_size, hash hash_func);

int hm_destroy(HashMap hm);

void *hm_get(HashMap hm, const void *key);

int hm_set(HashMap hm, const void *key, const void *value);

int hm_put(HashMap hm, const void *key, const void *value);

size_t hm_size(HashMap hm);

int hm_remove(HashMap hm, const void *key);

#endif //HASHMAP_H
```

Index