gagchat

0.1

Generated by Doxygen 1.8.17

Chapter 1

gagchat

A simple implementation of a common chat program in C.

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

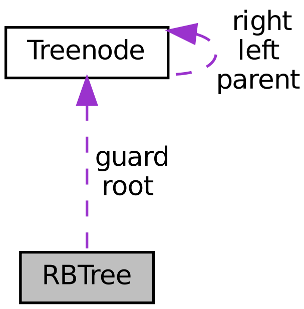
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 RBTree Struct Reference

Collaboration diagram for RBTree:



Public Attributes

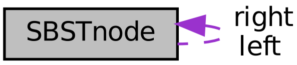
- Treenode ∗ **root**
- Treenode ∗ **guard**
- int(∗ **comparator**)(const void ∗, const void ∗)

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

- library/include/redblackTree.h

4.2 SBSTnode Struct Reference

Collaboration diagram for SBSTnode:



Public Attributes

- TInfoBST **info**
- struct SBSTnode ∗ **left**
- struct SBSTnode ∗ **right**

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

- library/include/TBST.h

4.3 SBucket Struct Reference

Struct of Hashtable.

```
#include <hashtable.h>
```

Public Attributes

- TKeyHashtable **key**
- TValueHashtable **value**

4.3.1 Detailed Description

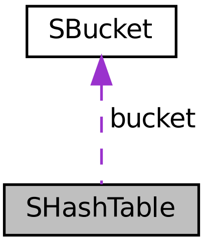
Struct of Hashtable.

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

- library/include/hashtable.h

4.4 SHashTable Struct Reference

Collaboration diagram for SHashTable:



Public Attributes

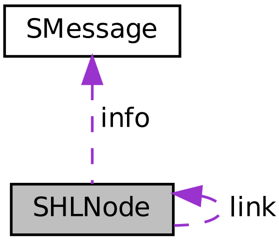
- TBucket ∗ **bucket**
- int ∗ **used**
- int **n_bucket**
- int **n_used**

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

- library/include/hashtable.h

4.5 SHLNode Struct Reference

Collaboration diagram for SHLNode:



Public Attributes

- TInfoList **info**
- THLNode ∗ **link**

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

- library/include/list.h

4.6 SMessage Struct Reference

Public Attributes

- char **user** [USR_MAXLEN]
- char **text** [TXT_MAXLEN]
- int **code**

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

- library/include/infoList.h
- library/include/message.h

4.7 STCPSocket Struct Reference

Public Attributes

- int **sockfd**
- struct sockaddr_in **servaddr**

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

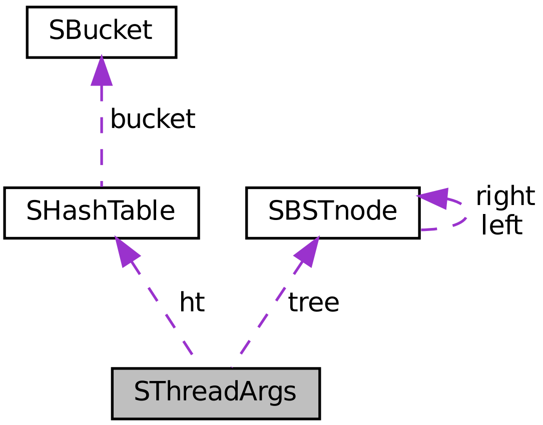
- library/include/tcp_socket.h

4.8 SThreadArgs Struct Reference

Struct exchanged by Threads.

`#include <server.h>`

Collaboration diagram for SThreadArgs:



Public Attributes

- THashTable ∗ **ht**
- char ∗ **selectedUser**
- pthread_mutex_t ∗ **lock**
- int **connection_fd**
- TBST ∗ **tree**

4.8.1 Detailed Description

Struct exchanged by Threads.

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

- include/client.h
- include/server.h

4.9 TInfoHashtable Struct Reference

definition of HashTable

`#include <infoHashtable.h>`

Public Attributes

- TKeyHashtable **key**
- TValueHashtable **value**

4.9.1 Detailed Description

definition of HashTable

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

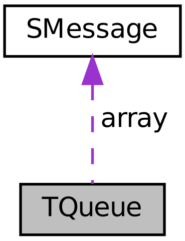
- library/include/infoHashtable.h

4.10 TQueue Struct Reference

Struct of the queue.

```
#include <queue.h>
```

Collaboration diagram for TQueue:



Public Attributes

- int **n_element**
- int **front**
- int **back**
- int **size**
- Message ** **array**

4.10.1 Detailed Description

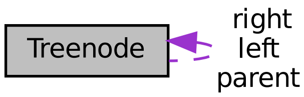
Struct of the queue.

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

- library/include/queue.h

4.11 Treenode Struct Reference

Collaboration diagram for Treenode:



Public Attributes

- void ∗ **data**
- char **color**
- struct Treenode ∗ **left**
- struct Treenode ∗ **right**
- struct Treenode ∗ **parent**

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

- library/include/treeNode.h

Chapter 5

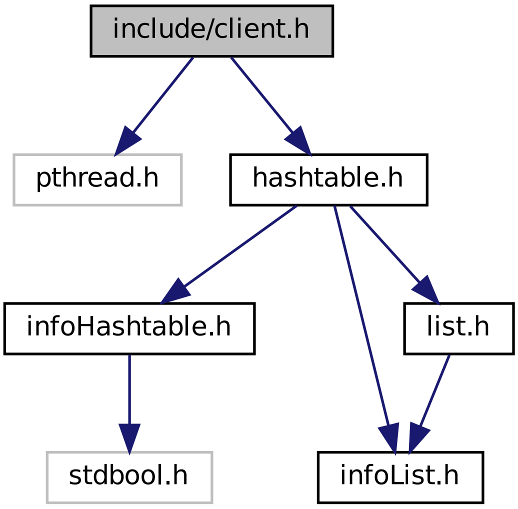
File Documentation

5.1 include/client.h File Reference

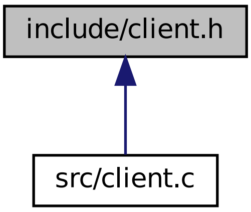
Client header file.

```
#include <pthread.h>
#include "hashtable.h"
```

Include dependency graph for client.h:



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



Classes

- struct SThreadArgs

  Struct exchanged by Threads.

Macros

- #define **QUEUE_SIZE** 32

Typedefs

- typedef struct SThreadArgs **ThreadArgs**

Functions

- void ∗ client_message_receiver (void ∗arg)

  thread for message receiving.

5.1.1 Detailed Description

Client header file.

Author

Giuseppe Gambardella (`g.gambardella23@studenti.unisa.it`)

Version

0.1

Date

2022-10-28

Copyright

5.1.2 Function Documentation

5.1.2.1 client_message_receiver()

```
void* client_message_receiver (
             void * arg )
```

thread for message receiving.

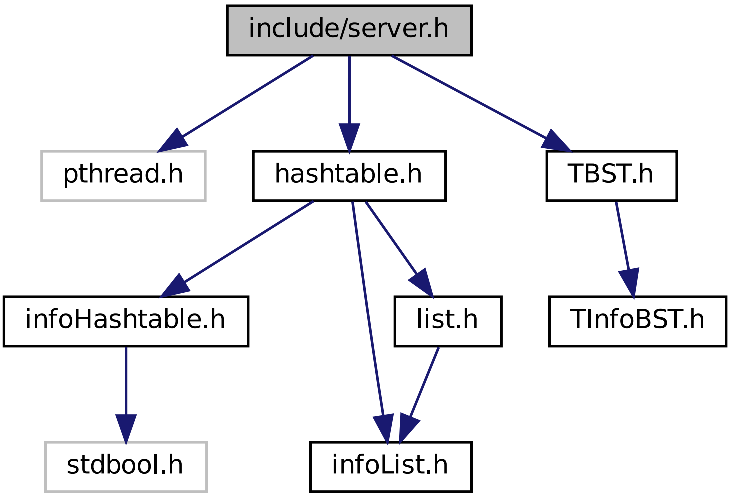
5.2 include/server.h File Reference

Server header file.

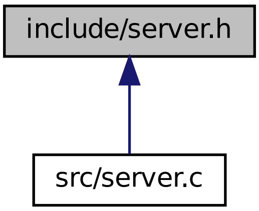
```
#include <pthread.h>
#include "hashtable.h"
```

```
#include "TBST.h"
```

Include dependency graph for server.h:



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



Classes

- struct SThreadArgs

    Struct exchanged by Threads.

Typedefs

- typedef struct SThreadArgs ThreadArgs

    Struct exchanged by Threads.

Functions

- void server_sendToAll (THashTable ∗ht, TBST tree, Message ∗message)

  Sends a message to all clients.
- void server_sendUsersTo (int connection_fd, TBST users)

  Sends the users list to specified client.
- void ∗ server_manage_client (void ∗arg)

  Thread for managing a generic client.

5.2.1 Detailed Description

Server header file.

Author

Giuseppe Gambardella (`g.gambardella23@studenti.unisa.it`)

Version

0.1

Date

2022-10-31

Copyright

Copyright (c) 2022 - All Rights Reserved

5.2.2 Typedef Documentation

5.2.2.1 ThreadArgs

```
typedef struct SThreadArgs ThreadArgs
```

Struct exchanged by Threads.

5.2.3 Function Documentation

5.2.3.1 server_manage_client()

```
void* server_manage_client (
             void * arg )
```

Thread for managing a generic client.

5.2.3.2 server_sendToAll()

```
void server_sendToAll (
             THashTable * ht,
             TBST tree,
             Message * message )
```

Sends a message to all clients.

Parameters

ht	the hashtable containing connections.
tree	tree list of users.
message	the message to send.

5.2.3.3 server_sendUsersTo()

```
void server_sendUsersTo (
             int connection_fd,
             TBST users )
```

Sends the users list to specified client.

Parameters

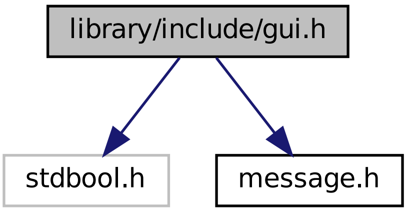
connection↵ _fd	connection file descriptor.
users	tree with all of users.

5.3 library/include/gui.h File Reference

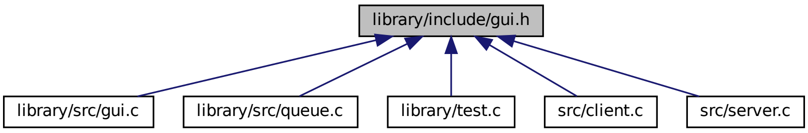
```
#include <stdbool.h>
```

```
#include "message.h"
```

Include dependency graph for gui.h:



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



Macros

- #define **WIDHT_SPACE** 15
- #define **COLUMNS** 70
- #define **ROWS** 20
- #define **Color_Off** "\033[0m"
- #define **Default_Color** "\033[0m"
- #define **Black** "\033[30m"
- #define **Red** "\033[31m"
- #define **Green** "\033[32m"
- #define **Yellow** "\033[33m"
- #define **Blue** "\033[34m"
- #define **Purple** "\033[35m"
- #define **Cyan** "\033[36m"
- #define **White** "\033[37m"
- #define **BBlack** "\033[1;30m"
- #define **BRed** "\033[1;31m"
- #define **BGreen** "\033[1;32m"
- #define **BYellow** "\033[1;33m"
- #define **BBlue** "\033[1;34m"
- #define **BPurple** "\033[1;35m"
- #define **BCyan** "\033[1;36m"
- #define **BWhite** "\033[1;37m"
- #define **UBlack** "\033[4;30m"

- #define **URed** "\033[4;31m"
- #define **UGreen** "\033[4;32m"
- #define **UYellow** "\033[4;33m"
- #define **UBlue** "\033[4;34m"
- #define **UPurple** "\033[4;35m"
- #define **UCyan** "\033[4;36m"
- #define **UWhite** "\033[4;37m"
- #define **On_Black** "\033[40m"
- #define **On_Red** "\033[41m"
- #define **On_Green** "\033[42m"
- #define **On_Yellow** "\033[43m"
- #define **On_Blue** "\033[44m"
- #define **On_Purple** "\033[45m"
- #define **On_Cyan** "\033[46m"
- #define **On_White** "\033[47m"
- #define **IBlack** "\033[90m"
- #define **IRed** "\033[91m"
- #define **IGreen** "\033[92m"
- #define **IYellow** "\033[93m"
- #define **IBlue** "\033[94m"
- #define **IPurple** "\033[95m"
- #define **ICyan** "\033[96m"
- #define **IWhite** "\033[97m"
- #define **BIBlack** "\033[1;90m"
- #define **BIRed** "\033[1;91m"
- #define **BIGreen** "\033[1;92m"
- #define **BIYellow** "\033[1;93m"
- #define **BIBlue** "\033[1;94m"
- #define **BIPurple** "\033[1;95m"
- #define **BICyan** "\033[1;96m"
- #define **BIWhite** "\033[1;97m"
- #define **On_IBlack** "\033[100m"
- #define **On_IRed** "\033[101m"
- #define **On_IGreen** "\033[102m"
- #define **On_IYellow** "\033[103m"
- #define **On_IBlue** "\033[104m"
- #define **On_IPurple** "\033[195m"
- #define **On_ICyan** "\033[106m"
- #define **On_IWhite** "\033[107m"

Functions

- void gui_set_color (char ∗color)

  set the color of text or background, apply a text style
- void gui_print_message (Message ∗msg, bool receved)
- void **gui_print_menu** (char ∗user)
- void **gui_print_list_users_header** ()
- void **gui_print_list_user** (char ∗user)
- void **gui_print_list_users_footer** ()
- void **gui_clear_Screen** ()

5.3.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-11-03

Copyright

5.3.2 Function Documentation

5.3.2.1 gui_print_message()

```
void gui_print_message (
            Message * msg,
            bool receved )
```

Parameters

msg	
receved	

5.3.2.2 gui_set_color()

```
void gui_set_color (
            char * color )
```

set the color of text or background, apply a text style

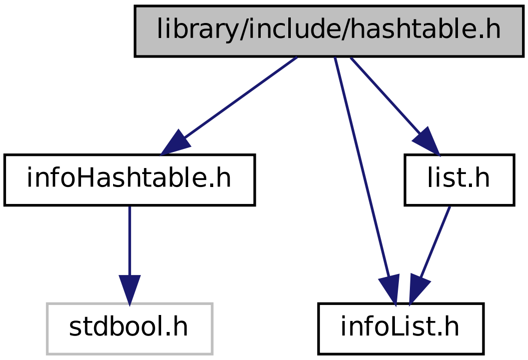
Parameters

color	

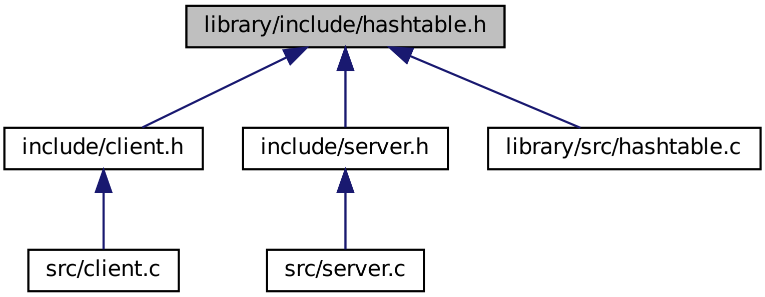
5.4 library/include/hashtable.h File Reference

```
#include "infoHashtable.h"
#include "infoList.h"
#include "list.h"
```

Include dependency graph for hashtable.h:



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



Classes

- struct SBucket

  Struct of Hashtable.
- struct SHashTable

Macros

- #define **FNV_OFFSET** 14695981039346656037UL
- #define **FNV_PRIME** 1099511628211UL
- #define **FNV_PRIME_2** 10995116282UL
- #define **CAPACITY** 64
- #define **MAX_LOAD** 0.75
- #define **GROW_FACTOR** 2
- #define **MAX_KEY_SIZE** 64

Typedefs

- typedef struct SBucket TBucket

  Struct of Hashtable.
- typedef struct SHashTable **THashTable**

Functions

- THashTable ∗ hashTableCreate ()

  Creation of a new HashTable.
- void hashTableResize (THashTable ∗ht, int n)

  Resize hashtable.
- void hashTableDestroy (THashTable ∗ht)

  destroy the HashTable
- TValueHashtable hashTableSearch (THashTable ∗ht, TKeyHashtable key)

  search in a HashTable
- void hashTableInsert (THashTable ∗ht, TKeyHashtable key, TValueHashtable value)

  Insert a new key and value in a HashTable.
- void hashTableDelete (THashTable ∗ht, TKeyHashtable key)

  Delete a key in a HashTable.
- void hashTablePrint (THashTable ∗ht)

  Print a HashTable.
- u_int64_t keyHash (TKeyHashtable)

  hash a key (function h(x))
- u_int64_t keyHashD (TKeyHashtable)

  hash a key (function d(x))
- u_int64_t keyHashExpande (TKeyHashtable, u_int64_t, int)

  double hashing return hash = h(x) + j∗d(x)

5.4.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

Copyright (c) 2022 - All Rights Reserved

5.4.2 Typedef Documentation

5.4.2.1 TBucket

```
typedef struct SBucket TBucket
```

Struct of Hashtable.

5.4.3 Function Documentation

5.4.3.1 hashTableCreate()

```
THashTable* hashTableCreate ( )
```

Creation of a new HashTable.

Returns

THashTable*

5.4.3.2 hashTableDelete()

```
void hashTableDelete (
            THashTable * ht,
            TKeyHashtable key )
```

Delete a key in a HashTable.

Parameters

ht	pointer of HashTable
key	key to delete

5.4.3.3 hashTableDestroy()

```
void hashTableDestroy (
             THashTable * ht )
```

destroy the HashTable

Parameters

ht	pointer of HashTable

5.4.3.4 hashTableInsert()

```
void hashTableInsert (
             THashTable * ht,
             TKeyHashtable key,
             TValueHashtable value )
```

Insert a new key and value in a HashTable.

Parameters

ht	pointer of HashTable
key	key to insert
value	value to insert

5.4.3.5 hashTablePrint()

```
void hashTablePrint (
             THashTable * ht )
```

Print a HashTable.

Parameters

ht	pointer of HashTable

5.4.3.6 hashTableResize()

```
void hashTableResize (
             THashTable * ht,
             int n )
```

Resize hashtable.

Parameters

ht	pointer of HashTable
n	new size

Resize hashtable.

Parameters

ht	pointer of HashTable
n	new size

5.4.3.7 hashTableSearch()

```
TValueHashtable hashTableSearch (
             THashTable * ht,
             TKeyHashtable key )
```

search in a HashTable

Parameters

ht	pointer of HashTable
key	key to find

Returns

TValueHashtable

search in a HashTable

Parameters

ht	pointer of HashTable
key	key to find

Returns

TValueHashtable

5.4.3.8 keyHash()

```
u_int64_t keyHash (
            TKeyHashtable key )
```

hash a key (function h(x))

Returns

u_int64_t

5.4.3.9 keyHashD()

```
u_int64_t keyHashD (
            TKeyHashtable key )
```

hash a key (function d(x))

Returns

u_int64_t

5.4.3.10 keyHashExpande()

```
u_int64_t keyHashExpande (
            TKeyHashtable key,
            u_int64_t hash,
            int j )
```

double hashing return hash = h(x) + j∗d(x)

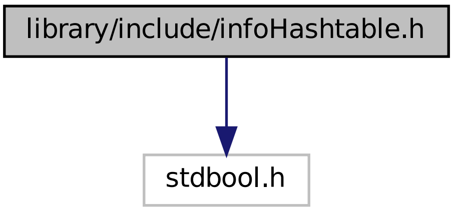
Returns

u_int64_t

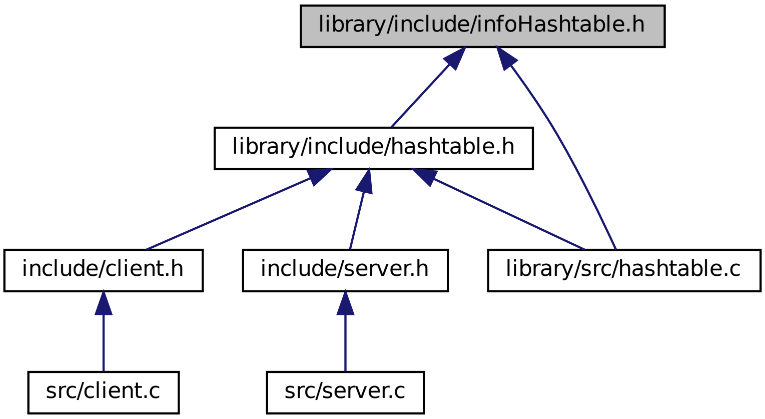
5.5 library/include/infoHashtable.h File Reference

```
#include "stdbool.h"
```

Include dependency graph for infoHashtable.h:



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



Classes

- struct TInfoHashtable

  definition of HashTable

Typedefs

- typedef char ∗ TKeyHashtable

  type of HashTable
- typedef void ∗ **TValueHashtable**

Functions

- bool infoEqual (TInfoHashtable, TInfoHashtable)

  check if frist value is equal to the second value
- bool infoGreater (TInfoHashtable, TInfoHashtable)

  check if frist value is greater to the second value
- bool infoLess (TInfoHashtable, TInfoHashtable)

  check if frist value is less to the second value
- void infoPrint (TInfoHashtable)

  structured print of the value

5.5.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

5.5.2 Typedef Documentation

5.5.2.1 TKeyHashtable

```
typedef char* TKeyHashtable
```

type of HashTable

5.5.3 Function Documentation

5.5.3.1 infoEqual()

```
bool infoEqual (
            TInfoHashtable val1,
            TInfoHashtable val2 )
```

check if frist value is equal to the second value

Returns

bool

5.5.3.2 infoGreater()

```
bool infoGreater (
            TInfoHashtable val1,
            TInfoHashtable val2 )
```

check if frist value is greater to the second value

Returns

bool

5.5.3.3 infoLess()

```
bool infoLess (
            TInfoHashtable val1,
            TInfoHashtable val2 )
```

check if frist value is less to the second value

Returns

bool

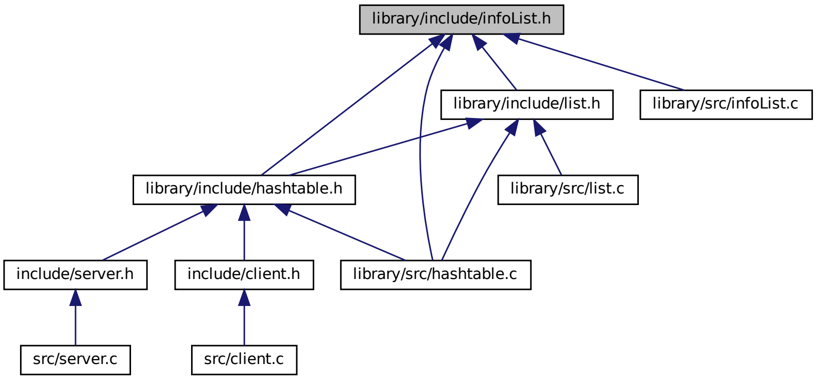
5.5.3.4 infoPrint()

```
void infoPrint (
            TInfoHashtable val1 )
```

structured print of the value

5.6 library/include/infoList.h File Reference

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



Classes

- struct SMessage

Macros

- #define **USR_MAXLEN** 64
- #define **TXT_MAXLEN** 1024
- #define **MESSAGE_STRUCT**

Typedefs

- typedef struct SMessage **Message**
- typedef Message TInfoList

    type of List

Functions

- int infoListEqual (TInfoList, TInfoList)

    check if all the information of frist value are the same of the second value
- void infoListPrint (TInfoList)

    structured print of the value

5.6.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

5.6.2 Typedef Documentation

5.6.2.1 TInfoList

```
typedef Message TInfoList
```

type of List

5.6.3 Function Documentation

5.6.3.1 infoListEqual()

```
int infoListEqual (
            TInfoList info1,
            TInfoList info2 )
```

check if all the information of frist value are the same of the second value

Returns

bool

5.6.3.2 infoListPrint()

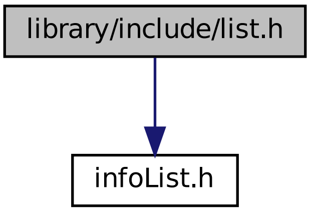
```
void infoListPrint (
            TInfoList info )
```

structured print of the value

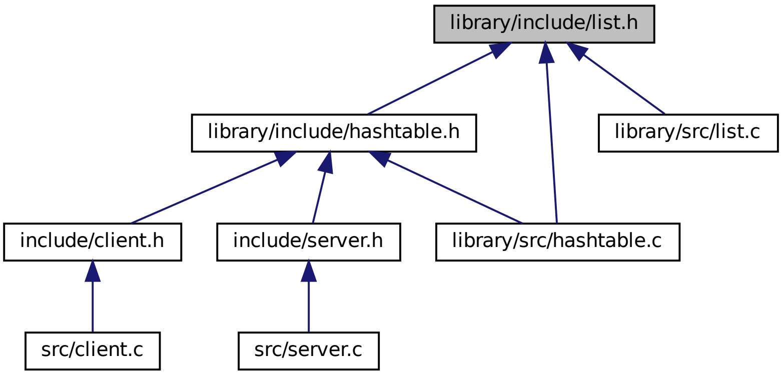
5.7 library/include/list.h File Reference

```
#include "infoList.h"
```

Include dependency graph for list.h:



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



Classes

- struct SHLNode

Typedefs

- typedef struct SHLNode THLNode

  Struct of List.
- typedef THLNode ∗ **TList**

Functions

- TList listCreate ()

  Creation of a new List.
- TList listDestroy (TList list)

  Destroy a List.
- TList listInsert (TList list, TInfoList info)

  Insert a new value in the List.
- THLNode ∗ listSearch (TList list, TInfoList info)

  Search an element into the List.
- TList listDelete (TList list, TInfoList info)

  Delete an element from a list.
- void listPrint (TList list)

  Print of an entire List.

5.7.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

5.7.2 Typedef Documentation

5.7.2.1 THLNode

```
typedef struct SHLNode THLNode
```

Struct of List.

5.7.3 Function Documentation

5.7.3.1 listCreate()

```
TList listCreate ( )
```

Creation of a new List.

Returns

TList

5.7.3.2 listDelete()

```
TList listDelete (
            TList list,
            TInfoList info )
```

Delete an element from a list.

Parameters

list	pointer of List
info	value of List

Returns

TList

5.7.3.3 listDestroy()

```
TList listDestroy (
            TList list )
```

Destroy a List.

Parameters

list	pointer of List

Returns

TList

5.7.3.4 listInsert()

```
TList listInsert (
            TList l,
            TInfoList info )
```

Insert a new value in the List.

Parameters

list	pointer of List
info	value of List

Returns

TList

5.7.3.5 listPrint()

```
void listPrint (
            TList list )
```

Print of an entire List.

Parameters

list	pointer of List

5.7.3.6 listSearch()

```
THLNode* listSearch (
            TList list,
            TInfoList info )
```

Search an element into the List.

Parameters

list	pointer of List
info	value of List

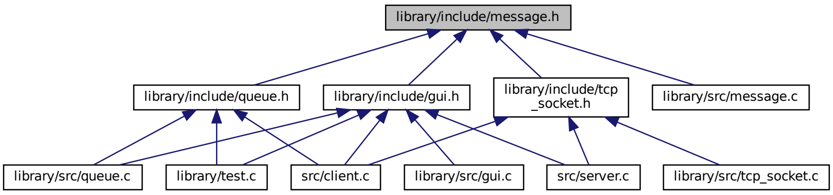
Returns

THLNode∗

5.8 library/include/message.h File Reference

Definition of struct Message.

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



Classes

- struct SMessage

Macros

- #define **USR_MAXLEN** 64
- #define **TXT_MAXLEN** 1024
- #define **MSG_MAXLEN** 32
- #define MESSAGE 0

    Message Codes for protocol.
- #define **MSG_CLI_CREATE** 1
- #define **MSG_SRV_USRACK** 2
- #define **MSG_SRV_USRNCK** 3
- #define **MSG_SRV_USRNRC** 4
- #define **MSG_CLI_LSTUSR** 5
- #define **MSG_SRV_AVLUSR** 6
- #define MESSAGE_STRUCT

    struct exchanged by parties for gagchat

Typedefs

- typedef struct SMessage **Message**

Functions

- Message ∗ message_create ()

    Creates an empty message.
- void message_constructor (Message ∗message, char ∗user, char ∗text)

    Constructs a basic exchangeable message.
- void message_code_constructor (Message ∗message, char ∗user, char ∗text, int code)

    Constructs a code message.
- void message_destroy (Message ∗message)

    Destroys a Message.

5.8.1 Detailed Description

Definition of struct Message.

Author

Giuseppe Gambardella (`g.gambardella23@studenti.unisa.it`)

Version

0.1

Date

2022-10-28

Copyright

5.8.2 Macro Definition Documentation

5.8.2.1 MESSAGE

```
#define MESSAGE 0
```

Message Codes for protocol.

5.8.2.2 MESSAGE_STRUCT

```
#define MESSAGE_STRUCT
```

struct exchanged by parties for gagchat

5.8.3 Function Documentation

5.8.3.1 message_code_constructor()

```
void message_code_constructor (
             Message * message,
             char * user,
             char * text,
             int code )
```

Constructs a code message.

Parameters

user	Username
text	Message text
code	code to append

Returns

void

5.8.3.2 message_constructor()

```
void message_constructor (
             Message * message,
             char * user,
             char * text )
```

Constructs a basic exchangeable message.

Parameters

user	Username
text	Message text

Returns

void

5.8.3.3 message_create()

```
Message* message_create ( )
```

Creates an empty message.

Returns

Message∗ pointer to the Empty message.

5.8.3.4 message_destroy()

```
void message_destroy (
             Message * message )
```

Destroys a Message.

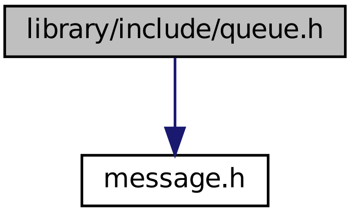
Parameters

message	the Message reference to destroy.

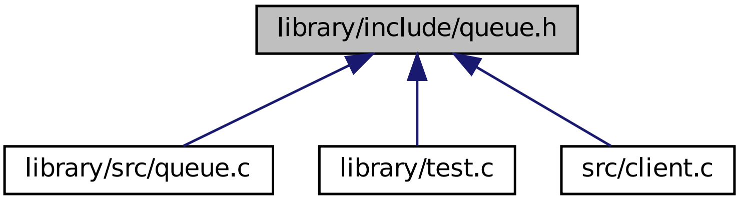
5.9 library/include/queue.h File Reference

```
#include "message.h"
```

Include dependency graph for queue.h:



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



Classes

- struct TQueue

  Struct of the queue.

Functions

- TQueue ∗ **queueCreate** ()
- void **queueDestroy** (TQueue ∗q)
- void **queueAdd** (TQueue ∗q, Message x)
- Message ∗ **queueRemove** (TQueue ∗q)
- Message ∗ **queueFront** (TQueue ∗q)
- int **queueIsEmpty** (TQueue ∗q)
- void **queuePrint** (TQueue ∗q, char ∗user_sender)

5.9.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

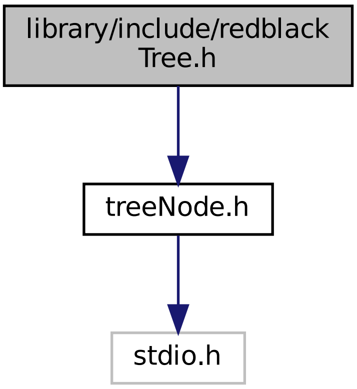
2022-11-03

Copyright

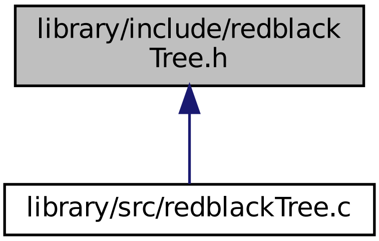
5.10 library/include/redblackTree.h File Reference

```
#include "treeNode.h"
```

Include dependency graph for redblackTree.h:



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



Classes

- struct RBTree

Functions

- RBTree ∗ **rbtree_create** (int(∗comparator)(const void ∗, const void ∗))
- void **rbtree_insert** (RBTree ∗rbtree, void ∗data)
- void **rbtree_deleteData** (RBTree ∗rbtree, void ∗data)
- void **rbtree_destroy** (RBTree ∗rbtree)

5.10.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

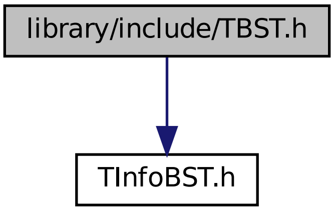
2022-11-04

Copyright

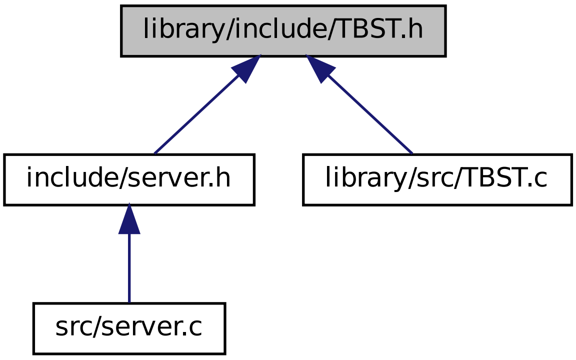
5.11 library/include/TBST.h File Reference

```
#include "TInfoBST.h"
```

Include dependency graph for TBST.h:



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



Classes

- struct SBSTnode

Typedefs

- typedef struct SBSTnode **TBSTnode**
- typedef TBSTnode ∗ **TBST**

Functions

- TBSTnode ∗ **BSTnodeCreate** (TInfoBST)
- void **BSTnodeDestroy** (TBSTnode ∗)
- TBST **BSTcreate** ()
- TBST **BSTdestroy** (TBST)
- TBST **BSTsearch** (TBST, TInfoBST)
- TBST **BSTmin** (TBST)
- TBST **BSTmax** (TBST)
- TBST **BSTinsert** (TBST, TInfoBST)
- TBST **BSTdelete** (TBST, TInfoBST)
- TBST **BSTsearchI** (TBST, TInfoBST)
- TBST **BSTinsertI** (TBST, TInfoBST)
- TBST **BSTdeleteI** (TBST, TInfoBST)
- void **BSTprint** (TBST)
- void **BSTprintStructure** (TBST)

5.11.1 Detailed Description

Author

Giuseppe Gambardella (`g.gambardella23@studenti.unisa.it`)

Version

0.1

Date

2022-11-04

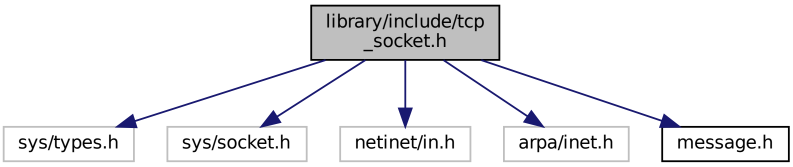
Copyright

5.12 library/include/tcp_socket.h File Reference

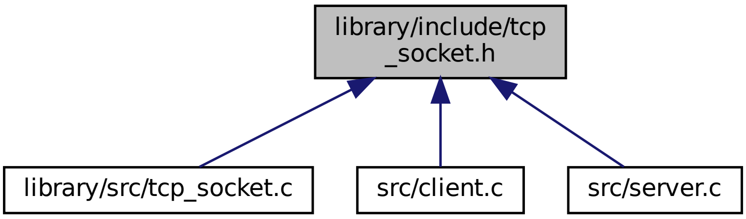
Implementation of tcp socket both for client and server.

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include "message.h"
```

Include dependency graph for tcp_socket.h:



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



Classes

- struct STCPSocket

Macros

- #define **SERV_PORT** 3000
- #define **LISTENQ** 100

Typedefs

- typedef enum EConnType **ConnType**
- typedef struct STCPSocket **TCPSocket**

Enumerations

- enum **EConnType** { **CLIENT**, **SERVER** }

Functions

- TCPSocket ∗ tcp_socket_create (ConnType type, char ∗server_ip)

  Creates a TCPSocket and pass its pointer.
- void tcp_socket_server_listen (TCPSocket ∗tcp_socket)

  Configures the server in listening mode.
- int tcp_socket_server_accept (TCPSocket ∗tcp_socket)

  Accepts a connection from a requesting client.
- void tcp_socket_client_connect (TCPSocket ∗tcp_socket)

  Connects the client to the specified TCPSocket.
- int tcp_socket_send_message (int connection_fd, Message ∗message)

  Sends a Message on TCPSocket.
- int tcp_socket_recv_message (int connection_fd, Message ∗message)

  Receives a Message from TCPSocket.
- void tcp_socket_destroy (TCPSocket ∗tcp_socket)

  Destroys a TCPSocket.

5.12.1 Detailed Description

Implementation of tcp socket both for client and server.

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

Copyright (c) 2022 - All Rights Reserved

5.12.2 Function Documentation

5.12.2.1 tcp_socket_client_connect()

```
void tcp_socket_client_connect (
            TCPSocket * tcp_socket )
```

Connects the client to the specified TCPSocket.

Parameters

tcp_socket	pointer to TCPSocket struct.

5.12.2.2 tcp_socket_create()

```
TCPSocket* tcp_socket_create (
            ConnType type,
            char * server_ip )
```

Creates a TCPSocket and pass its pointer.

Parameters

type	the TCPSocket type (client or server)
server_ip	if type = CLIENT, is mandatory to specify the server's ip address formatted as string.

Returns

TCPSocket* pointer to TCPSocket struct

5.12.2.3 tcp_socket_destroy()

```
void tcp_socket_destroy (
            TCPSocket * tcp_socket )
```

Destroys a TCPSocket.

Parameters

tcp_socket	the TCPSocket reference to destroy.

5.12.2.4 tcp_socket_recv_message()

```
int tcp_socket_recv_message (
             int connection_fd,
             Message * message )
```

Receives a Message from TCPSocket.

Parameters

connection←_fd	connection file descriptor. If client, it is param sockfd of TCPSocket. If server, it is obtained by the output of tcp_socket_server_accept function.
message	pointer to the message to receive.

Returns

> int returns the number of bytes received, or -1 if an error occurred. In the event of an error, errno is set to indicate the error.

5.12.2.5 tcp_socket_send_message()

```
int tcp_socket_send_message (
             int connection_fd,
             Message * message )
```

Sends a Message on TCPSocket.

Parameters

connection←_fd	connection file descriptor. If client, it is param sockfd of TCPSocket. If server, it is obtained by the output of tcp_socket_server_accept function.
message	the Message to send.

Returns

> int On success, returns the number of bytes sent. On error, -1 is returned, and errno is set to indicate the error.

5.12.2.6 tcp_socket_server_accept()

```
int tcp_socket_server_accept (
             TCPSocket * tcp_socket )
```

Accepts a connection from a requesting client.

Parameters

tcp_socket	pointer to TCPSocket struct.

Returns

int the connection file descriptor.

5.12.2.7 tcp_socket_server_listen()

```
void tcp_socket_server_listen (
            TCPSocket * tcp_socket )
```

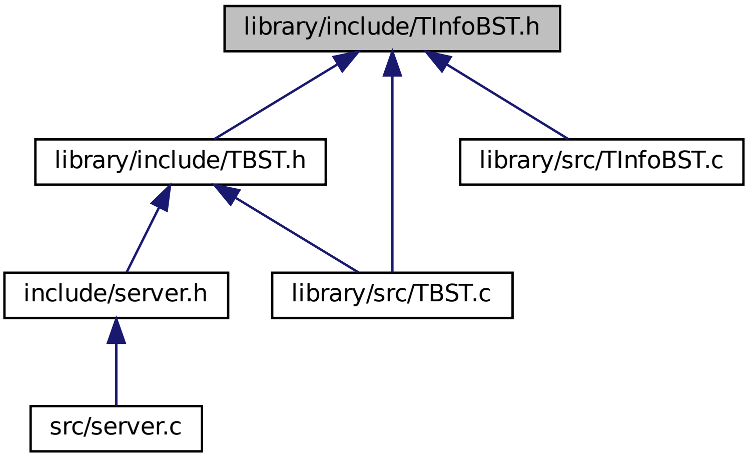
Configures the server in listening mode.

Parameters

tcp_socket	pointer to TCPSocket struct.

5.13 library/include/TInfoBST.h File Reference

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



Typedefs

- typedef char ∗ **TInfoBST**

Functions

- int **infoBSTEqual** (TInfoBST, TInfoBST)
- int **infoBSTGreater** (TInfoBST, TInfoBST)
- int **infoBSTLess** (TInfoBST, TInfoBST)
- void **infoBSTPrint** (TInfoBST)

5.13.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

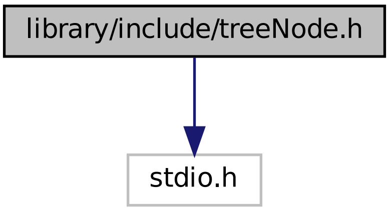
2022-11-04

Copyright

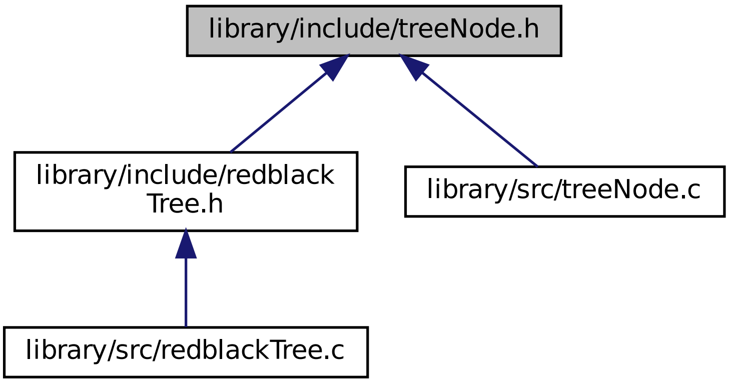
5.14 library/include/treeNode.h File Reference

```
#include <stdio.h>
```

Include dependency graph for treeNode.h:



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


library/include/treeNode.h
library/include/redblack Tree.h
library/src/treeNode.c
library/src/redblackTree.c

Classes

- struct Treenode

Typedefs

- typedef struct Treenode **Treenode**

Functions

- Treenode ∗ **treenode_createGuard** ()
- void **treenode_LeftRotate** (Treenode ∗∗root, Treenode ∗x, Treenode ∗guard)
- void **treenode_RightRotate** (Treenode ∗∗root, Treenode ∗y, Treenode ∗guard)
- void **treenode_insertion** (Treenode ∗∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_Fixedinsertion** (Treenode ∗∗root, Treenode ∗z, Treenode ∗guard)
- void **treenode_deleteTree** (Treenode ∗root, Treenode ∗guard)
- Treenode ∗ **treenode_FindData** (Treenode ∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_DeleteData** (Treenode ∗∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_FixedDelete** (Treenode ∗∗root, Treenode ∗x, Treenode ∗guard)
- void **treenode_transplant** (Treenode ∗∗root, Treenode ∗x, Treenode ∗y, Treenode ∗guard)

5.14.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

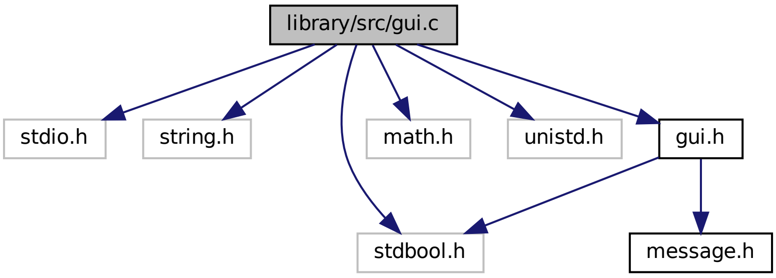
2022-11-04

Copyright

5.15 library/src/gui.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <stdbool.h>
#include <math.h>
#include <unistd.h>
#include "gui.h"
```

Include dependency graph for gui.c:



Functions

- void gui_set_color (char ∗color)

  set the color of text or background, apply a text style
- void gui_print_n_times (char ∗str, int n)

  Print.
- void gui_print_horizontal_line ()

  Print the upper and lower separator of box.
- void gui_print_horizontal_msg_line (bool is_sender)

  Print the upper and lower separator of the text box.
- void gui_print_user_msg (char ∗user, bool is_sender)
- void **gui_print_msg** (char ∗msg, bool is_sender)
- void gui_print_message (Message ∗msg, bool is_sender)
- void **gui_clear_Screen** ()
- void **gui_print_line** (char ∗line)
- void **gui_print_menu** (char ∗user)
- void **gui_print_list_users_header** ()
- void **gui_print_list_users_footer** ()
- void **gui_print_list_user** (char ∗user)

5.15.1 Detailed Description

Author

Gianluca (`g.canzolino3@studenti.unisa.it`)

Version

0.1

Date

2022-11-03

Copyright

Copyright (c) 2022 - All Rights Reserved

This file is part of gagchat. gagchat is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

gagchat is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with gagchat. If not, see `http://www.gnu.org/licenses/`.

5.15.2 Function Documentation

5.15.2.1 gui_print_horizontal_line()

```
void gui_print_horizontal_line ( )
```

Print the upper and lower separator of box.

5.15.2.2 gui_print_horizontal_msg_line()

```
void gui_print_horizontal_msg_line (
             bool is_sender )
```

Print the upper and lower separator of the text box.

Parameters

is_sender	true if is a message receved, else false

5.15.2.3 gui_print_message()

```
void gui_print_message (
             Message * msg,
             bool receved )
```

Parameters

msg	
receved	

5.15.2.4 gui_print_n_times()

```
void gui_print_n_times (
             char * str,
             int n )
```

Print.

Parameters

n	time the string
str	
str	string to print
n	times

5.15.2.5 gui_print_user_msg()

```
void gui_print_user_msg (
            char * user,
            bool is_sender )
```

Parameters

user	
is_sender	true if is a message receved, else false

5.15.2.6 gui_set_color()

```
void gui_set_color (
            char * color )
```

set the color of text or background, apply a text style

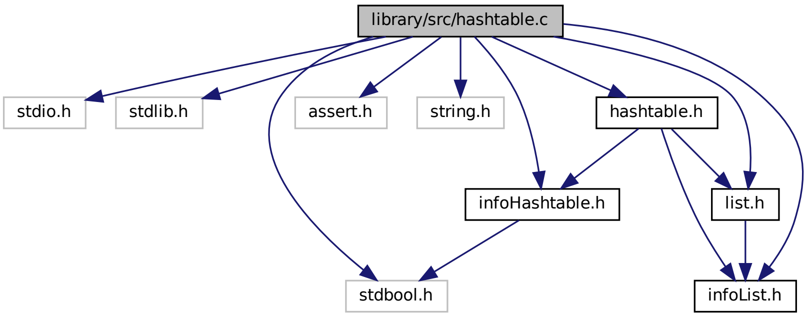
Parameters

color	

5.16 library/src/hashtable.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#include <assert.h>
#include <string.h>
#include "infoList.h"
#include "list.h"
#include "hashtable.h"
#include "infoHashtable.h"
```

Include dependency graph for hashtable.c:



Functions

- THashTable ∗ hashTableCreate ()

  Creation of a new HashTable.
- void hashTableResize (THashTable ∗ht, int n)

  resizing Hashtable
- void hashTableDestroy (THashTable ∗ht)

  destroy the HashTable
- TValueHashtable hashTableSearch (THashTable ∗ht, TKeyHashtable key)

  search a key in a HashTable, return the list
- void hashTableInsert (THashTable ∗ht, TKeyHashtable key, TValueHashtable value)

  Insert a new key and value in a HashTable.
- void hashTableDelete (THashTable ∗ht, TKeyHashtable key)

  Delete a key in a HashTable.
- void hashTablePrint (THashTable ∗ht)

  Print a HashTable.
- u_int64_t keyHash (TKeyHashtable key)

  hash a key (function h(x))
- u_int64_t keyHashD (TKeyHashtable key)

  hash a key (function d(x))
- u_int64_t keyHashExpande (TKeyHashtable key, u_int64_t hash, int j)

  double hashing return hash = h(x) + j∗d(x)

5.16.1 Detailed Description

Author

Gianluca (`g.canzolino3@studenti.unisa.it`)

Version

0.1

Date

2022-10-29

Copyright

5.16.2 Function Documentation

5.16.2.1 hashTableCreate()

```
THashTable* hashTableCreate ( )
```

Creation of a new HashTable.

Returns

THashTable*

5.16.2.2 hashTableDelete()

```
void hashTableDelete (
            THashTable * ht,
            TKeyHashtable key )
```

Delete a key in a HashTable.

Parameters

ht	pointer of HashTable
key	key to delete

5.16.2.3 hashTableDestroy()

```
void hashTableDestroy (
            THashTable * ht )
```

destroy the HashTable

Parameters

ht	pointer of HashTable

5.16.2.4 hashTableInsert()

```
void hashTableInsert (
            THashTable * ht,
            TKeyHashtable key,
            TValueHashtable value )
```

Insert a new key and value in a HashTable.

Parameters

ht	pointer of HashTable
key	key to insert
value	value to insert

5.16.2.5 hashTablePrint()

```
void hashTablePrint (
            THashTable * ht )
```

Print a HashTable.

Parameters

ht	pointer of HashTable

5.16.2.6 hashTableResize()

```
void hashTableResize (
            THashTable * ht,
            int n )
```

resizing Hashtable

Resize hashtable.

Parameters

ht	pointer of HashTable
n	new size

5.16.2.7 hashTableSearch()

```
TValueHashtable hashTableSearch (
            THashTable * ht,
            TKeyHashtable key )
```

search a key in a HashTable, return the list

search in a HashTable

Parameters

ht	pointer of HashTable
key	key to find

Returns

TValueHashtable

5.16.2.8 keyHash()

```
u_int64_t keyHash (
            TKeyHashtable key )
```

hash a key (function h(x))

Returns

u_int64_t

5.16.2.9 keyHashD()

```
u_int64_t keyHashD (
            TKeyHashtable key )
```

hash a key (function d(x))

Returns

u_int64_t

5.16.2.10 keyHashExpande()

```
u_int64_t keyHashExpande (
            TKeyHashtable key,
            u_int64_t hash,
            int j )
```

double hashing return hash = h(x) + j∗d(x)

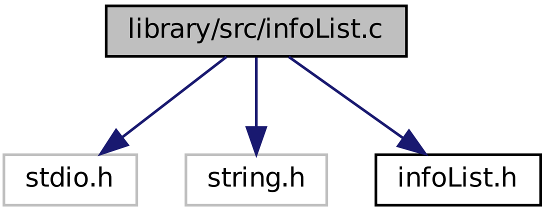
Returns

u_int64_t

5.17 library/src/infoList.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "infoList.h"
```

Include dependency graph for infoList.c:



Functions

- int infoListEqual (TInfoList info1, TInfoList info2)

  check if all the information of frist value are the same of the second value
- void infoListPrint (TInfoList info)

  structured print of the value

5.17.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

Copyright (c) 2022 - All Rights Reserved

5.17.2 Function Documentation

5.17.2.1 infoListEqual()

```
int infoListEqual (
            TInfoList info1,
            TInfoList info2 )
```

check if all the information of frist value are the same of the second value

Returns

bool

5.17.2.2 infoListPrint()

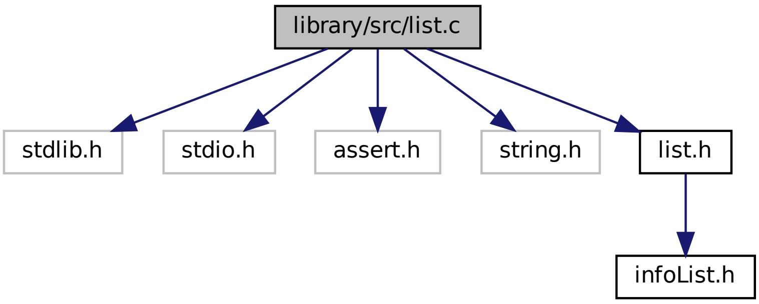
```
void infoListPrint (
            TInfoList info )
```

structured print of the value

5.18 library/src/list.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>
#include <string.h>
#include "list.h"
```

Include dependency graph for list.c:



Functions

- THLNode ∗ nodeCreate (TInfoList info)
- void nodeDestroy (THLNode ∗node)
- TList listCreate ()

  Creation of a new List.
- TList listDestroy (TList list)

  Destroy a List.
- TList listInsert (TList l, TInfoList info)

  Insert a new value in the List.
- THLNode ∗ listSearch (TList list, TInfoList info)

  Search an element into the List.
- TList listDelete (TList list, TInfoList info)

  Delete an element from a list.
- void listPrint (TList list)

  Print of an entire List.

5.18.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

5.18.2 Function Documentation

5.18.2.1 listCreate()

```
TList listCreate ( )
```

Creation of a new List.

Returns

TList

5.18.2.2 listDelete()

```
TList listDelete (
            TList list,
            TInfoList info )
```

Delete an element from a list.

Parameters

list	pointer of List
info	value of List

Returns

TList

5.18.2.3 listDestroy()

```
TList listDestroy (
            TList list )
```

Destroy a List.

Parameters

list	pointer of List

Returns

TList

5.18.2.4 listInsert()

```
TList listInsert (
            TList l,
            TInfoList info )
```

Insert a new value in the List.

Parameters

list	pointer of List
info	value of List

Returns

TList

5.18.2.5 listPrint()

```
void listPrint (
            TList list )
```

Print of an entire List.

Parameters

list	pointer of List

5.18.2.6 listSearch()

```
THLNode* listSearch (
            TList list,
            TInfoList info )
```

Search an element into the List.

Parameters

list	pointer of List
info	value of List

Returns

THLNode∗

5.18.2.7 nodeCreate()

```
THLNode* nodeCreate (
            TInfoList info )
```

Parameters

info	

Returns

THLNode∗

5.18.2.8 nodeDestroy()

```
void nodeDestroy (
            THLNode * node )
```

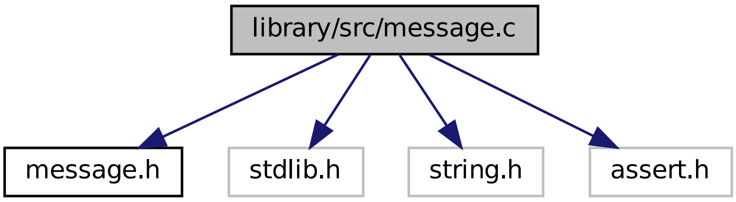
Parameters

node	

5.19 library/src/message.c File Reference

```
#include "message.h"
#include <stdlib.h>
#include <string.h>
#include <assert.h>
```

Include dependency graph for message.c:



Functions

- Message ∗ message_create ()

  Creates an empty message.
- void message_constructor (Message ∗msg, char ∗user, char ∗text)

  Constructs a basic exchangeable message.
- void message_code_constructor (Message ∗msg, char ∗user, char ∗text, int code)

  Constructs a code message.
- void message_destroy (Message ∗message)

  Destroys a Message.

5.19.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

2022-10-29

Copyright

5.19.2 Function Documentation

5.19.2.1 message_code_constructor()

```
void message_code_constructor (
            Message * message,
            char * user,
            char * text,
            int code )
```

Constructs a code message.

Parameters

user	Username
text	Message text
code	code to append

Returns

void

5.19.2.2 message_constructor()

```
void message_constructor (
            Message * message,
            char * user,
            char * text )
```

Constructs a basic exchangeable message.

Parameters

user	Username
text	Message text

Returns

void

5.19.2.3 message_create()

```
Message* message_create ( )
```

Creates an empty message.

Returns

Message∗ pointer to the Empty message.

5.19.2.4 message_destroy()

```
void message_destroy (
            Message * message )
```

Destroys a Message.

Parameters

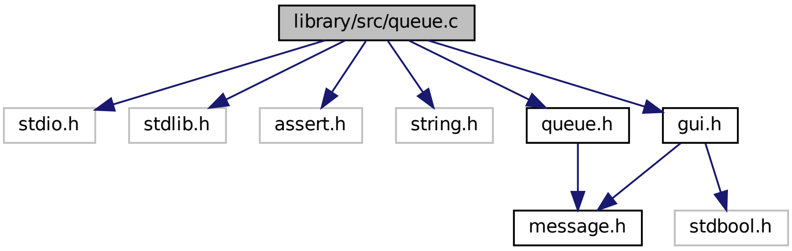
message	the Message reference to destroy.

5.20 library/src/queue.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
```

```
#include <assert.h>
#include <string.h>
#include "queue.h"
#include "gui.h"
```

Include dependency graph for queue.c:



Functions

- TQueue ∗ **queueCreate** (int queue_size)
- void **queueDestroy** (TQueue ∗q)
- void **queueAdd** (TQueue ∗q, Message x)
- Message ∗ **queueRemove** (TQueue ∗q)
- Message ∗ **queueFront** (TQueue ∗q)
- int **queueIsEmpty** (TQueue ∗q)
- void **queuePrint** (TQueue ∗q, char ∗user_sender)

5.20.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

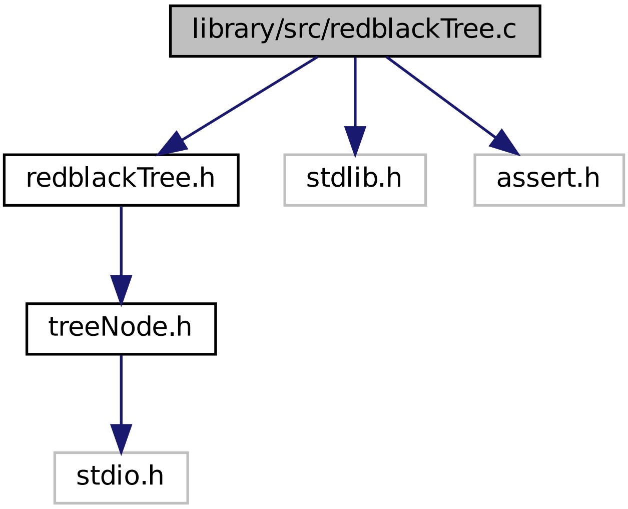
2022-11-03

Copyright

5.21 library/src/redblackTree.c File Reference

```
#include "redblackTree.h"
#include <stdlib.h>
#include <assert.h>
```

Include dependency graph for redblackTree.c:



Functions

- RBTree ∗ **rbtree_create** (int(∗comparator)(const void ∗, const void ∗))
- void **rbtree_insert** (RBTree ∗rbtree, void ∗data)
- void **rbtree_deleteData** (RBTree ∗rbtree, void ∗data)
- void **rbtree_destroy** (RBTree ∗rbtree)

5.21.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

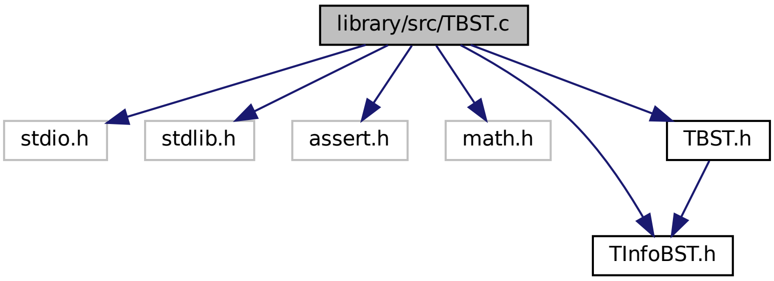
Date

2022-11-04

5.22 library/src/TBST.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>
#include <math.h>
#include "TInfoBST.h"
#include "TBST.h"
```

Include dependency graph for TBST.c:



Functions

- TBSTnode ∗ **BSTnodeCreate** (TInfoBST info)
- void **BSTnodeDestroy** (TBSTnode ∗node)
- TBST **BSTcreate** ()
- TBST **BSTdestroy** (TBST tree)
- TBST **BSTsearch** (TBST tree, TInfoBST info)
- TBST **BSTmin** (TBST tree)
- TBST **BSTmax** (TBST tree)
- TBST **BSTinsert** (TBST tree, TInfoBST info)
- TBST **BSTdelete** (TBST tree, TInfoBST info)
- void **BSTprint** (TBST tree)
- TBST **BSTsearchI** (TBST tree, TInfoBST info)
- TBST **BSTinsertI** (TBST tree, TInfoBST info)
- TBST **BSTdeleteI** (TBST tree, TInfoBST info)
- int **BSTdepth** (TBST tree)
- void **BSTprintStructure1** (TBST tree, int o, TInfoBST ∗n, int ∗m)

5.22.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

2022-11-04

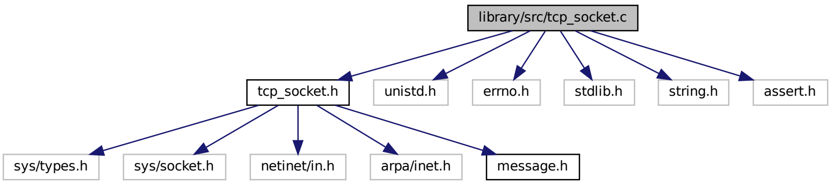
Copyright

Copyright (c) 2022 - All Rights Reserved

5.23 library/src/tcp_socket.c File Reference

```
#include "tcp_socket.h"
#include <unistd.h>
#include <errno.h>
#include <stdlib.h>
#include <string.h>
#include <assert.h>
```

Include dependency graph for tcp_socket.c:



Functions

- TCPSocket ∗ tcp_socket_create (ConnType type, char ∗server_ip)

  Creates a TCPSocket and pass its pointer.
- void tcp_socket_server_listen (TCPSocket ∗tcp_socket)

  Configures the server in listening mode.
- int tcp_socket_server_accept (TCPSocket ∗tcp_socket)

  Accepts a connection from a requesting client.
- void tcp_socket_client_connect (TCPSocket ∗tcp_socket)

  Connects the client to the specified TCPSocket.
- int tcp_socket_send_message (int connection_fd, Message ∗message)

  Sends a Message on TCPSocket.
- int tcp_socket_recv_message (int connection_fd, Message ∗message)

  Receives a Message from TCPSocket.
- void tcp_socket_destroy (TCPSocket ∗tcp_socket)

  Destroys a TCPSocket.

5.23.1 Detailed Description

Author

Giuseppe Gambardella (`g.gambardella23@studenti.unisa.it`)

Version

0.1

Date

2022-10-30

Copyright

Copyright (c) 2022 - All Rights Reserved

This file is part of gagchat. gagchat is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

gagchat is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with gagchat. If not, see `http://www.gnu.org/licenses/`.

5.23.2 Function Documentation

5.23.2.1 tcp_socket_client_connect()

```
void tcp_socket_client_connect (
            TCPSocket * tcp_socket )
```

Connects the client to the specified TCPSocket.

Parameters

tcp_socket	pointer to TCPSocket struct.

5.23.2.2 tcp_socket_create()

```
TCPSocket* tcp_socket_create (
             ConnType type,
             char * server_ip )
```

Creates a TCPSocket and pass its pointer.

Parameters

type	the TCPSocket type (client or server)
server↩_ip	if type = CLIENT, is mandatory to specify the server's ip address formatted as string.

Returns

TCPSocket* pointer to TCPSocket struct

5.23.2.3 tcp_socket_destroy()

```
void tcp_socket_destroy (
             TCPSocket * tcp_socket )
```

Destroys a TCPSocket.

Parameters

tcp_socket	the TCPSocket reference to destroy.

5.23.2.4 tcp_socket_recv_message()

```
int tcp_socket_recv_message (
             int connection_fd,
             Message * message )
```

Receives a Message from TCPSocket.

Parameters

connection↩_fd	connection file descriptor. If client, it is param sockfd of TCPSocket. If server, it is obtained by the output of tcp_socket_server_accept function.
message	pointer to the message to receive.

Returns

> int returns the number of bytes received, or -1 if an error occurred. In the event of an error, errno is set to indicate the error.

5.23.2.5 tcp_socket_send_message()

```
int tcp_socket_send_message (
            int connection_fd,
            Message * message )
```

Sends a Message on TCPSocket.

Parameters

connection↩_fd	connection file descriptor. If client, it is param sockfd of TCPSocket. If server, it is obtained by the output of tcp_socket_server_accept function.
message	the Message to send.

Returns

> int On success, returns the number of bytes sent. On error, -1 is returned, and errno is set to indicate the error.

5.23.2.6 tcp_socket_server_accept()

```
int tcp_socket_server_accept (
            TCPSocket * tcp_socket )
```

Accepts a connection from a requesting client.

Parameters

tcp_socket	pointer to TCPSocket struct.

Returns

> int the connection file descriptor.

5.23.2.7 tcp_socket_server_listen()

```
void tcp_socket_server_listen (
            TCPSocket * tcp_socket )
```

Configures the server in listening mode.

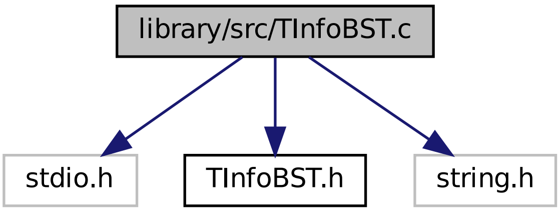
Parameters

tcp_socket	pointer to TCPSocket struct.

5.24 library/src/TInfoBST.c File Reference

```
#include <stdio.h>
#include "TInfoBST.h"
#include <string.h>
```

Include dependency graph for TInfoBST.c:



Functions

- int **infoBSTEqual** (TInfoBST info1, TInfoBST info2)
- int **infoBSTGreater** (TInfoBST info1, TInfoBST info2)
- int **infoBSTLess** (TInfoBST info1, TInfoBST info2)
- void **infoBSTPrint** (TInfoBST info)

5.24.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

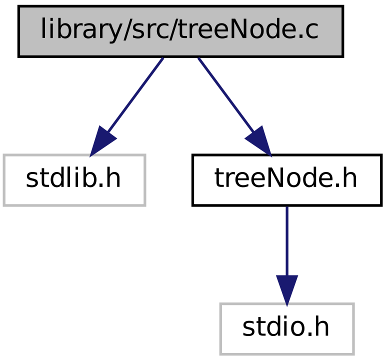
2022-11-04

Copyright

5.25 library/src/treeNode.c File Reference

```
#include <stdlib.h>
#include "treeNode.h"
```

Include dependency graph for treeNode.c:



Functions

- void **treenode_LeftRotate** (Treenode ∗∗root, Treenode ∗x, Treenode ∗guard)
- void **treenode_RightRotate** (Treenode ∗∗root, Treenode ∗x, Treenode ∗guard)
- void **treenode_insertion** (Treenode ∗∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_Fixedinsertion** (Treenode ∗∗root, Treenode ∗z, Treenode ∗guard)
- Treenode ∗ **treenode_FindData** (Treenode ∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_DeleteData** (Treenode ∗∗root, void ∗data, int(∗comparator)(const void ∗, const void ∗), Treenode ∗guard)
- void **treenode_transplant** (Treenode ∗∗root, Treenode ∗x, Treenode ∗y, Treenode ∗guard)
- void **treenode_FixedDelete** (Treenode ∗∗root, Treenode ∗x, Treenode ∗guard)
- void **treenode_deleteTree** (Treenode ∗root, Treenode ∗guard)
- void **treenode_printTree** (FILE ∗f, Treenode ∗root, Treenode ∗guard)
- Treenode ∗ **treenode_createGuard** ()

5.25.1 Detailed Description

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

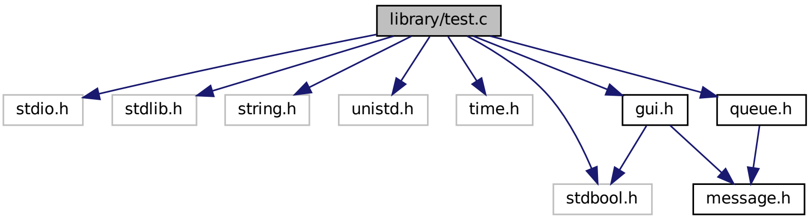
2022-11-04

Copyright

5.26 library/test.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <time.h>
#include <stdbool.h>
#include "gui.h"
#include "queue.h"
```

Include dependency graph for test.c:



Functions

- int **main** (int argc, char **argv)

5.26.1 Detailed Description

Author

Gianluca (g.canzolino3@studenti.unisa.it)

Version

0.1

Date

2022-10-29

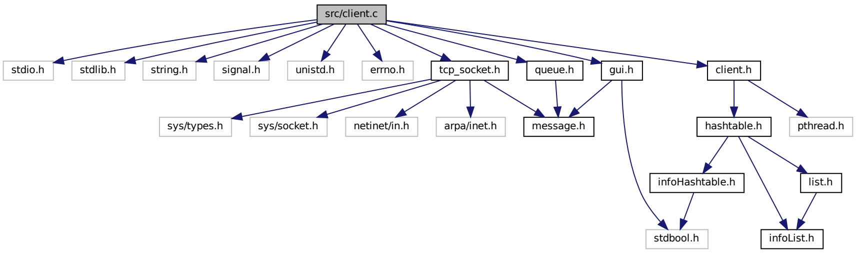
Copyright

5.27 src/client.c File Reference

Client Implementation of gagchat.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <errno.h>
#include "tcp_socket.h"
#include "client.h"
#include "queue.h"
#include "gui.h"
```

Include dependency graph for client.c:



Functions

- void ∗ client_message_receiver (void ∗arg)

  thread for message receiving.
- int **main** (int argc, char ∗∗argv)

Variables

- TCPSocket ∗ **sock**

5.27.1 Detailed Description

Client Implementation of gagchat.

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

2022-10-28

Copyright

5.27.2 Function Documentation

5.27.2.1 client_message_receiver()

```
void* client_message_receiver (
            void * arg )
```

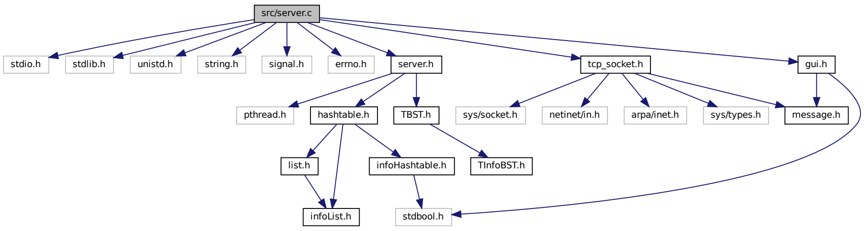
thread for message receiving.

5.28 src/server.c File Reference

Server implementation of gagchat.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <signal.h>
#include <errno.h>
#include "server.h"
#include "tcp_socket.h"
#include "gui.h"
```

Include dependency graph for server.c:



Functions

- void server_sendToAll (THashTable ∗ht, TBST tree, Message ∗message)

  Sends a message to all clients.
- void server_sendUsersTo (int connection_fd, TBST users)

  Sends the users list to specified client.
- void ∗ server_manage_client (void ∗arg)

  Thread for managing a generic client.
- int **main** (int argc, char ∗∗argv)

Variables

- TCPSocket ∗ **sock**

5.28.1 Detailed Description

Server implementation of gagchat.

Author

Giuseppe Gambardella (g.gambardella23@studenti.unisa.it)

Version

0.1

Date

2022-10-30

Copyright

5.28.2 Function Documentation

5.28.2.1 server_manage_client()

```
void* server_manage_client (
            void * arg )
```

Thread for managing a generic client.

5.28.2.2 server_sendToAll()

```
void server_sendToAll (
            THashTable * ht,
            TBST tree,
            Message * message )
```

Sends a message to all clients.

Parameters

ht	the hashtable containing connections.
tree	tree list of users.
message	the message to send.

5.28.2.3 server_sendUsersTo()

```
void server_sendUsersTo (
            int connection_fd,
            TBST users )
```

Sends the users list to specified client.

Parameters

connection←↩ _fd	connection file descriptor.
users	tree with all of users.

Index