relay

Generated by Doxygen 1.8.11

Contents

Chapter 1

README

First Commit

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

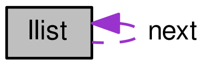
Chapter 4

Data Structure Documentation

4.1 llist Struct Reference

Linked List to store socket file descriptors.

Collaboration diagram for llist:



Data Fields

- int fileDesc
- struct llist ∗ next

4.1.1 Detailed Description

Linked List to store socket file descriptors.

4.1.2 Field Documentation

4.1.2.1 int llist::fileDesc

4.1.2.2 struct llist∗ llist::next

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

- src/dispatcher.c

4.2 socketStruct Struct Reference

Socket structure used to get socket information to thread.

Data Fields

- int socketFd
- struct sockaddr ∗ address
- int sockaddrlen

4.2.1 Detailed Description

Socket structure used to get socket information to thread.

4.2.2 Field Documentation

4.2.2.1 struct sockaddr∗ socketStruct::address

4.2.2.2 int socketStruct::sockaddrlen

4.2.2.3 int socketStruct::socketFd

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

- src/dispatcher.c

Chapter 5

File Documentation

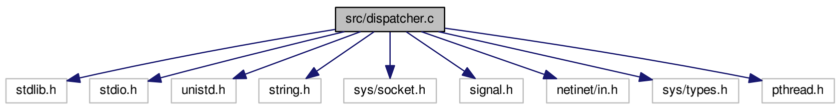
5.1 README.md File Reference

5.2 src/dispatcher.c File Reference

Message Server Software. Will allow any number of hosts to connect and recieve messages.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <string.h>
#include <sys/socket.h>
#include <signal.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <pthread.h>
```

Include dependency graph for dispatcher.c:



Data Structures

- struct llist

  Linked List to store socket file descriptors.
- struct socketStruct

  Socket structure used to get socket information to thread.

Typedefs

- typedef struct llist llist
- typedef struct socketStruct socketStruct
- typedef void ∗(∗ func_f) (void ∗)

  typedef used to simplify function cast

Functions

- void ∗ listenConnection (socketStruct ∗mySock)

  Used to run in thread. Will listen on port specified by RESET environment variable and add new connections to linked list.
- void addConnection (int socketNum)

  Used to add file descriptor to linked list.
- llist ∗ newConnection (int fileDesc)

  Creates a new linked list node with socket(2) file descriptor.
- void removeConnection (int fileDesc)

  Removes file descriptor from linked list.
- void destroyLL (void)

  Frees memory from linked list.
- void ignoreSIGINT (__attribute__((unused)) int sig_num)

  Signal handler to catch CTRL+C.
- int main (int argc, __attribute__((unused)) char ∗∗argv)

5.2.1 Detailed Description

Message Server Software. Will allow any number of hosts to connect and recieve messages.

Author

Jack Spence

Date

18 Jan 2019

5.2.2 Typedef Documentation

5.2.2.1 typedef void∗(∗ func_f) (void ∗)

typedef used to simplify function cast

5.2.2.2 typedef struct llist llist

5.2.2.3 typedef struct socketStruct socketStruct

5.2.3 Function Documentation

5.2.3.1 void addConnection (int *socketNum*)

Used to add file descriptor to linked list.

Parameters

socketNum	Number returned from socket(2)

Iterator used for sending to each file descriptor

Adding first connection to linked list

Finding end of linked list

Creating new node and adding to end of linked list

5.2.3.2 void destroyLL (void)

Frees memory from linked list.

Iterator used for sending to each file descriptor

Loop used to free each element in linked list

5.2.3.3 void ignoreSIGINT (__attribute__((unused)) int *sig_num*)

Signal handler to catch CTRL+C.

5.2.3.4 void ∗ listenConnection (socketStruct ∗ *mySock*)

Used to run in thread. Will listen on port specified by RESET environment variable and add new connections to linked list.

Parameters

mySock	socketStruct that has socket details

Int used for file descriptor

Starts listening on sockets file descriptor

Accepts incoming connections from clients

Grabs lock and adds connection to linked list

5.2.3.5 int main (int *argc*, __attribute__((unused)) char ∗∗ *argv*)

Used for getopt and switch case.

Used for string verification from command line

Using getopt(3) to parse command line arguments

Limit switch

Checking for other garbage not caught by getopt(3)

Sigaction used to call signal hander function

Setting program to call sighandler for CTRL+C

Allows program to continue running after SIGPIPE from closed connection.

End pointer for getenv(3) verification.

String pulled from getenv(3).

Port number as a long to connect to.

File descriptor returned from socket(2).

Setting the socket options.

Initialization of socket structure

Initialization of socketStruct for thread.

Binds to port specified in RELAY

Thread created to handle incoming connections

initialze mutex lock

Spawns thread to run and handle incoming connection

Buffer to store input into.

Set by getline(3).

Used to control how much data gets sent.

Exit condition. User entered CTRL+D

Not goint to talk to myself.

Iterator used for sending to each file descriptor

Grabbing lock for critical code section

Sending data to clients

Checking if message was received. Removing connection from list if not received.

Releasing lock.

5.2.3.6 llist ∗ newConnection (int *fileDesc*)

Creates a new linked list node with socket(2) file descriptor.

Parameters

fileDesc	File Descriptor

Returns

New node for linked list

New node for linked list

Obligatory checking return of malloc

5.2.3.7 void removeConnection (int *fileDesc*)

Removes file descriptor from linked list.

Iterators used for sending to each file descriptor

Checking if first in list is to be removed

Testing to see if only one in linked list.

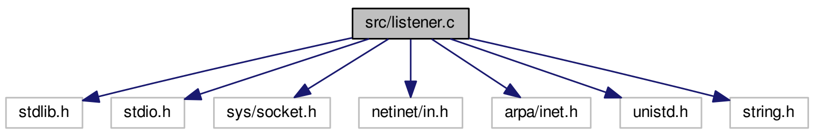
If the first one needs to be removed.

Looping to find fileDesc to remove from list

5.3 src/listener.c File Reference

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <unistd.h>
#include <string.h>
```

Include dependency graph for listener.c:



Functions

- int main (int argc, __attribute__((unused)) char ∗∗argv)

5.3.1 Function Documentation

5.3.1.1 int main (int *argc*, __attribute__((unused)) char ** *argv*)

End pointer for getenv(3) verification.

String pulled from getenv(3).

Port number as a long to connect to.

File descriptor returned from socket(2).

Structure used to connect to server

Used to get localhost IP into bits

Attempting to connect to server with previous struct

Buffer to store data from server

Integer used to break loop if server closes.

Error if RELAY isn't found

Index