OS_P2

Jaider Andrés Pinto Pinto
Cesar Esteban Díaz Medina
Diego Esteban Quintero Rey

Thu Jun 3 2021

Table of Contents

Data Structure Index

Data Structures

Here are the data structures with brief descriptions:

File Index

File List

Here is a list of all files with brief descriptions:

Data Structure Documentation

dato Struct Reference

Data Fields

- int **client**
- pthread_t **hilo**
- int **state**

Field Documentation

int client

pthread_t hilo

int state

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

- **p2-server.c**

viaje Struct Reference

Data Fields

- int **origen**
- int **destino**
- int **hora**
- float **media**
- float **desviacion**
- float **med_geo**
- float **desv_geo**

Field Documentation

int destino

float desv_geo

float desviacion

int hora

float med_geo

float media

int origen

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

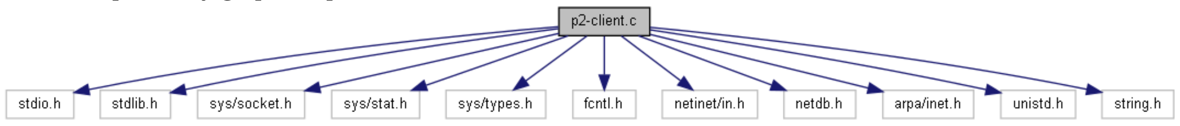
- **p2-server.c**

File Documentation

p2-client.c File Reference

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

Include dependency graph for p2-client.c:



Macros

- #define **PORT**   3535
- #define **RES_SIZE**   9
- #define **REQ_SIZE**   13
- #define **PERMISSIONS**   0666

Functions

- void **menu** ()
  This function prints the menu to the console It's called from ***main()***

- int **main** ()
  This function starts by allocating memory for the input data. Then it creates the client file descriptor and assign the fields of the client structure Once it's done, it attempt to establish a connection with the server. If it's successful, it will print the menu by calling the ***menu()*** *function. Once the user enter the correct three input values and selects option [4] from menu, a request will be sent to the server. The server will perform the search and send a response which is printed by this function. The program stops if the user selects the option [5] from menu.*

Macro Definition Documentation

#define PERMISSIONS   0666

#define PORT   3535

#define REQ_SIZE   13

#define RES_SIZE   9

Function Documentation

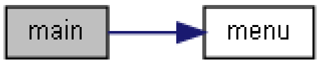
int main ()

This function starts by allocating memory for the input data. Then it creates the client file descriptor and assign the fields of the client structure Once it's done, it attempt to establish a connection with the server. If it's successful, it will print the menu by calling the **menu()** function. Once the user enter the correct three input values and selects option [4] from menu, a request will be sent to the server. The server will perform the search and send a response which is printed by this function. The program stops if the user selects the option [5] from menu.

Return values

void	

Here is the call graph for this function:



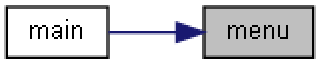
void menu ()

This function prints the menu to the console It's called from **main()**

Return values

void	

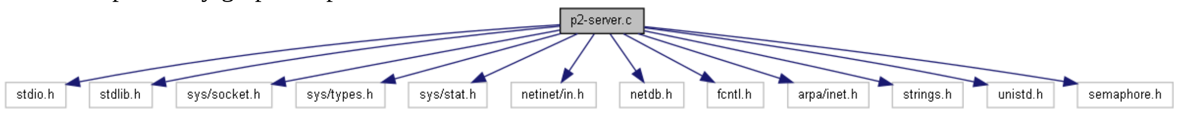
Here is the caller graph for this function:



p2-server.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/socket.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <netinet/in.h>
#include <netdb.h>
#include <fcntl.h>
#include <arpa/inet.h>
#include <strings.h>
#include <unistd.h>
#include <semaphore.h>
```

Include dependency graph for p2-server.c:



Data Structures

- struct **viaje**
- struct **dato**

Macros

- #define **PORT** 3535
- #define **BACKLOG** 32
- #define **RES_SIZE** 9
- #define **REQ_SIZE** 13
- #define **PERMISSIONS** 0666

Functions

- * **proceso** (void *arg)

  This is the search function that will be perform by each thread in order to give a response to the client. First it allocates memory for the received data, then it loads the hash table to memory It receive the data with recv() and performs the search with fseek(), fread() and the reg structure. It finally sends the response to the client based on the result of the search process.

- int **main** ()

  *The **main()** function starts by creating and configuring the server socket. Once configuration is done it starts listening. It opens the files for the hash table and the data blocks and accepts connections only if the number of clients is below the BACKLOG. Then it creates the threads and join them when execution ends. Each thread will execute the search process.*

Variables

- FILE * **infile**
- FILE * **hash**
- int **r**
- int **num_client**
- struct **dato datos** [**BACKLOG**+1]

Macro Definition Documentation

#define BACKLOG   32

#define PERMISSIONS   0666

#define PORT   3535

#define REQ_SIZE   13

#define RES_SIZE   9

Function Documentation

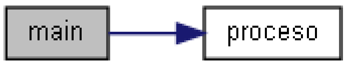
int main ()

The **main()** function starts by creating and configuring the server socket. Once configuration is done it starts listening. It opens the files for the hash table and the data blocks and accepts connections only if the number of clients is below the BACKLOG. Then it creates the threads and join them when execution ends. Each thread will execute the search process.

Return values

void	

Here is the call graph for this function:



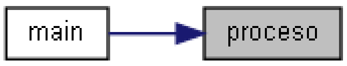
* proceso (void *   *arg*)

This is the search function that will be perform by each thread in order to give a response to the client. First it allocates memory for the received data, then it loads the hash table to memory It receive the data with recv() and performs the search with fseek(), fread() and the reg structure. It finally sends the response to the client based on the result of the search process.

Return values

void	

Here is the caller graph for this function:



Variable Documentation

struct dato datos[BACKLOG+1]

FILE * hash

FILE* infile

int num_client

int r

Index

INDEX