

DATABASEQUERYSERVER

1.0

Generated by Doxygen 1.7.1

Fri Feb 14 2014 03:04:19

Contents

1	QueryPage	1
1.1	File Names	1
1.2	Running Server	1
1.3	Running Client	1
1.4	Execution Directive	1
2	File Index	3
2.1	File List	3
3	File Documentation	5
3.1	code/server.c File Reference	5
3.1.1	Define Documentation	6
3.1.1.1	BUF_SIZE	6
3.1.1.2	PORT	6
3.1.1.3	QUEUE	6
3.1.2	Function Documentation	6
3.1.2.1	catch_signal	6
3.1.2.2	handle_shutdown	6
3.1.2.3	main	6
3.1.3	Variable Documentation	6
3.1.3.1	listener_id	6
3.1.3.2	sem	6
3.2	code/service.c File Reference	7
3.2.1	Define Documentation	7
3.2.1.1	BUF_SIZE	7
3.2.2	Function Documentation	8
3.2.2.1	addRecord	8
3.2.2.2	bind_to_port	8
3.2.2.3	connect_to_port	8

3.2.2.4	error	8
3.2.2.5	find_command	8
3.2.2.6	hear	9
3.2.2.7	open_listener_socket	9
3.2.2.8	print_record	9
3.2.2.9	remove_spaces	9
3.2.2.10	say	10
3.2.2.11	searchByEntry	10
3.2.2.12	searchByName	10
3.2.2.13	searchByYear	10
3.2.2.14	strnicmp	10
3.3	code/service.h File Reference	11
3.3.1	Function Documentation	11
3.3.1.1	addRecord	11
3.3.1.2	bind_to_port	12
3.3.1.3	catch_signal	12
3.3.1.4	connect_to_port	12
3.3.1.5	error	12
3.3.1.6	find_command	12
3.3.1.7	handle_shutdown	12
3.3.1.8	hear	13
3.3.1.9	open_listener_socket	13
3.3.1.10	print_record	13
3.3.1.11	remove_spaces	13
3.3.1.12	say	13
3.3.1.13	searchByEntry	14
3.3.1.14	searchByName	14
3.3.1.15	searchByYear	14

Chapter 1

QueryPage

This is home page for documantation of QueryServer

1.1 File Names

1. [server.c](#)
2. [service.c](#)
3. [service.h](#)
4. makefile

1.2 Running Server

1. \$ gcc -o QueryServer [server.c](#) [service.c](#) -lpthread
2. \$ make

1.3 Running Client

1. \$ telnet <hostaddress> <portno>

1.4 Execution Directive

1. name <NAME>
2. entry <ENTRYNO>
3. return *
4. return <YEAR>
5. add <NAME> <ENTRYNO> <EMAILID>
6. quit

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

code/ server.c	5
code/ service.c	7
code/ service.h	11

Chapter 3

File Documentation

3.1 code/server.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include "service.h"
#include <sys/socket.h>
#include <arpa/inet.h>
#include <semaphore.h>
```

Defines

- #define [PORT](#) 30000
- #define [QUEUE](#) 1000
- #define [BUF_SIZE](#) 255

Functions

- int [catch_signal](#) (int sig, void(*handler)(int))
- void [handle_shutdown](#) (int sig)
- int [main](#) ()

Variables

- sem_t [sem](#)
- int [listener_id](#)

3.1.1 Define Documentation

3.1.1.1 `#define BUF_SIZE 255`

Definition at line 13 of file server.c.

3.1.1.2 `#define PORT 30000`

Definition at line 11 of file server.c.

3.1.1.3 `#define QUEUE 1000`

Definition at line 12 of file server.c.

3.1.2 Function Documentation

3.1.2.1 `int catch_signal (int sig, void(*) (int) handler)`

Signal Handler for QueryServer

Definition at line 25 of file server.c.

3.1.2.2 `void handle_shutdown (int sig)`

this function is used for shutting down the Query Server

Definition at line 36 of file server.c.

3.1.2.3 `int main ()`

This is main function in this query server is implemented

Definition at line 47 of file server.c.

3.1.3 Variable Documentation

3.1.3.1 `int listener_id`

Socket Descriptor Identifier

Definition at line 21 of file server.c.

3.1.3.2 `sem_t sem`

semaphore used for mutual exclusive excess to file

Definition at line 17 of file server.c.

3.2 code/service.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <errno.h>
#include <signal.h>
#include <ctype.h>
#include <sys/socket.h>
#include <arpa/inet.h>
#include <semaphore.h>
#include "service.h"
```

Defines

- #define [BUF_SIZE](#) 255

Functions

- void [error](#) (char *msg)
- int [strnicmp](#) (const char *pStr1, const char *pStr2, size_t Count)
- int [open_listener_socket](#) ()
- void [remove_spaces](#) (const char *input, char *result)
- void [bind_to_port](#) (int socket, int port)
- void [connect_to_port](#) (int socket, char *address, int port)
- int [say](#) (int socket, char *s)
- int [hear](#) (int socket, char *buf, int len)
- int [find_command](#) (char *bigString)
- void [print_record](#) (int s)
- void [searchByName](#) (char *name, int s)
- void [searchByEntry](#) (char *name, int s)
- void [searchByYear](#) (char *name, int s)
- void [addRecord](#) (char *name, char *entry, char *email, int s)

3.2.1 Define Documentation

3.2.1.1 #define BUF_SIZE 255

Defines the The Size of Buffer

Definition at line 15 of file service.c.

3.2.2 Function Documentation

3.2.2.1 void addRecord (char * *name*, char * *entry*, char * *email*, int *s*)

This function is used for inserting student record in database

Parameters

name name of Student

entry entry number of Student

email email of Student

Definition at line 324 of file service.c.

3.2.2.2 void bind_to_port (int *socket*, int *port*)

this function is used for binding socket to port

Parameters

socket socket descriptor

port port no for binding

Definition at line 81 of file service.c.

3.2.2.3 void connect_to_port (int *socket*, char * *address*, int *port*)

This is used for connecting socket to a port

Parameters

socket socket identifier

address ip address of system

port port number

Definition at line 106 of file service.c.

3.2.2.4 void error (char * *msg*)

This function is used for printing error message to stderr stream

Parameters

msg it is pointer to the error mesaage

Definition at line 20 of file service.c.

3.2.2.5 int find_command (char * *bigString*)

This is used for extracting the code corresponding to a command

Parameters

bigString pointer to message containing command

Definition at line 166 of file service.c.

3.2.2.6 int hear (int *socket*, char * *buf*, int *len*)

This method is used for reading message form Socket

Parameters

socket socket descriptor

buf pointer to Message

len length of message

Definition at line 138 of file service.c.

3.2.2.7 int open_listener_socket ()

This function is used for creating socket descriptor

Returns

handle of socket

Definition at line 52 of file service.c.

3.2.2.8 void print_record (int *s*)

this method is used for printing all records of database

Parameters

s socket descriptor

Definition at line 208 of file service.c.

3.2.2.9 void remove_spaces (const char * *input*, char * *result*)

this function is used for returning a string from which spaces has been removed

Parameters

input original string

result final string

Definition at line 66 of file service.c.

3.2.2.10 int say (int *socket*, char * *s*)

This method is used for writing message to the socket

Parameters

socket socket identifier
s message to send to socket

Definition at line 123 of file service.c.

3.2.2.11 void searchByEntry (char * *name*, int *s*)

this method this used for searching records in database using enrollment Number of student and printing all matched records

Parameters

name enrollment Number of student

Definition at line 260 of file service.c.

3.2.2.12 void searchByName (char * *name*, int *s*)

this method this used for searching records in database using name of student and printing all matched records

Parameters

name name of student

Definition at line 229 of file service.c.

3.2.2.13 void searchByYear (char * *name*, int *s*)

this method this used for searching records in database using enrollment Year of student and printing all matched records

Parameters

name enrollment Year of student

Definition at line 289 of file service.c.

3.2.2.14 int strnicmp (const char * *pStr1*, const char * *pStr2*, size_t *Count*)

This function is used for Case-Insensitive Comparison of string pointed by *pStr1* and *pStr2* upto length of *Count* Character

Parameters

pStr1 pointer to first string
pStr2 pointer to second string

Definition at line 31 of file service.c.

3.3 code/service.h File Reference

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

Functions

- void [error](#) (char *msg)
- int [open_listener_socket](#) ()
- int [catch_signal](#) (int signal, void(*handler)(int))
- void [handle_shutdown](#) (int signal)
- void [bind_to_port](#) (int socket, int port)
- void [connect_to_port](#) (int socket, char *addres, int port)
- int [say](#) (int socket, char *s)
- int [hear](#) (int socket, char *buf, int len)
- int [find_command](#) (char *bigString)
- void [remove_spaces](#) (const char *input, char *result)
- void [searchByName](#) (char *name, int s)
- void [searchByEntry](#) (char *entry, int s)
- void [searchByYear](#) (char *year, int s)
- void [print_record](#) (int s)
- void [addRecord](#) (char *a, char *b, char *c, int s)

3.3.1 Function Documentation

3.3.1.1 void addRecord (char * *name*, char * *entry*, char * *email*, int *s*)

This function is used for inserting student record in database

Parameters

name name of Student

entry entry number of Student

email email of Student

Definition at line 324 of file service.c.

3.3.1.2 void bind_to_port (int *socket*, int *port*)

this function is used for binding socket to port

Parameters

socket socket descriptor

port port no for binding

Definition at line 81 of file service.c.

3.3.1.3 int catch_signal (int *sig*, void(*)(int) *handler*)

Signal Handler for QueryServer

Definition at line 25 of file server.c.

3.3.1.4 void connect_to_port (int *socket*, char * *address*, int *port*)

This is used for connecting socket to a port

Parameters

socket socket identifier

address ip address of system

port port number

Definition at line 106 of file service.c.

3.3.1.5 void error (char * *msg*)

This function is used for printing error message to stderr stream

Parameters

msg it is pointer to the error mesaage

Definition at line 20 of file service.c.

3.3.1.6 int find_command (char * *bigString*)

This is used for extracting the code corresponding to a command

Parameters

bigString pointer to message containing command

Definition at line 166 of file service.c.

3.3.1.7 void handle_shutdown (int *sig*)

this function is used for shutting down the Query Server

Definition at line 36 of file server.c.

3.3.1.8 int hear (int *socket*, char * *buf*, int *len*)

This method is used for reading message form Socket

Parameters

socket socket descriptor

buf pointer to Message

len length of message

Definition at line 138 of file service.c.

3.3.1.9 int open_listener_socket ()

This function is used for creating socket descriptor

Returns

handle of socket

Definition at line 52 of file service.c.

3.3.1.10 void print_record (int *s*)

this method is used for printing all records of database

Parameters

s socket descriptor

Definition at line 208 of file service.c.

3.3.1.11 void remove_spaces (const char * *input*, char * *result*)

this function is used for returning a string from which spaces has been removed

Parameters

input original string

result final string

Definition at line 66 of file service.c.

3.3.1.12 int say (int *socket*, char * *s*)

This method is used for writing message to the socket

Parameters

socket socket identifier

s message to send to socket

Definition at line 123 of file service.c.

3.3.1.13 void searchByEntry (char * *name*, int *s*)

this method this used for searching records in database using enrollment Number of student and printing all matched records

Parameters

name enrollment Number of student

Definition at line 260 of file service.c.

3.3.1.14 void searchByName (char * *name*, int *s*)

this method this used for searching records in database using name of student and printing all matched records

Parameters

name name of student

Definition at line 229 of file service.c.

3.3.1.15 void searchByYear (char * *name*, int *s*)

this method this used for searching records in database using enrollment Year of student and printing all matched records

Parameters

name enrollment Year of student

Definition at line 289 of file service.c.

Index

addRecord
 service.c, 8
 service.h, 11

bind_to_port
 service.c, 8
 service.h, 11

BUF_SIZE
 server.c, 6
 service.c, 7

catch_signal
 server.c, 6
 service.h, 12

code/server.c, 5
code/service.c, 7
code/service.h, 11

connect_to_port
 service.c, 8
 service.h, 12

error
 service.c, 8
 service.h, 12

find_command
 service.c, 8
 service.h, 12

handle_shutdown
 server.c, 6
 service.h, 12

hear
 service.c, 9
 service.h, 12

listener_id
 server.c, 6

main
 server.c, 6

open_listener_socket
 service.c, 9
 service.h, 13

PORT

server.c, 6

print_record
 service.c, 9
 service.h, 13

QUEUE
 server.c, 6

remove_spaces
 service.c, 9
 service.h, 13

say
 service.c, 9
 service.h, 13

searchByEntry
 service.c, 10
 service.h, 13

searchByName
 service.c, 10
 service.h, 14

searchByYear
 service.c, 10
 service.h, 14

sem
 server.c, 6

server.c
 BUF_SIZE, 6
 catch_signal, 6
 handle_shutdown, 6
 listener_id, 6
 main, 6
 PORT, 6
 QUEUE, 6
 sem, 6

service.c
 addRecord, 8
 bind_to_port, 8
 BUF_SIZE, 7
 connect_to_port, 8
 error, 8
 find_command, 8
 hear, 9
 open_listener_socket, 9
 print_record, 9

- remove_spaces, [9](#)
- say, [9](#)
- searchByEntry, [10](#)
- searchByName, [10](#)
- searchByYear, [10](#)
- strnicmp, [10](#)
- service.h
 - addRecord, [11](#)
 - bind_to_port, [11](#)
 - catch_signal, [12](#)
 - connect_to_port, [12](#)
 - error, [12](#)
 - find_command, [12](#)
 - handle_shutdown, [12](#)
 - hear, [12](#)
 - open_listener_socket, [13](#)
 - print_record, [13](#)
 - remove_spaces, [13](#)
 - say, [13](#)
 - searchByEntry, [13](#)
 - searchByName, [14](#)
 - searchByYear, [14](#)
- strnicmp
 - service.c, [10](#)