My Project

Создано системой Doxygen 1.9.1

Глава 1

Алфавитный указатель классов

1.1 Классы

Классы с их кратким описанием.

Глава 2

Список файлов

2.1 Файлы

Полный список документированных файлов.

Глава 3

Классы

3.1 Класс Server

Класс, управляющий работой сервера

#include <mdfile.h>

Открытые члены

- int self_addr (string error, string file_error, int port)
  конструктор класса
- int client_addr (int s, string error, string file_error)
  Функция принимающая подключение клиента

3.1.1 Подробное описание

Класс, управляющий работой сервера

3.1.2 Методы

3.1.2.1 client_addr()

```
int Server::client_addr (
                int s,
                string error,
                string file_error )
```

Функция принимающая подключение клиента

```
int Server::client_addr(int s, string error, string file_error){
        sockaddr_in * client_addr = new sockaddr_in;
        socklen_t len = sizeof (sockaddr_in);
        int work_sock = accept(s, (sockaddr*)(client_addr), &len);
        if(work_sock == -1) {
            std::cout « "Error #2\n";
            error = "error #2";
            errors(error, file_error);
            return 1;
        }
        else {
            //Успешное подключение к серверу
            std::cout « "Successfull client connection!\n";
            return work_sock;
        }
        }
```

Аргументы

sockaddr	Адрес сокета
client_addr	Адрес клиента

3.1.2.2 self_addr()

```
int Server::self_addr (
                string error,
                string file_error,
                int port )
```

конструктор класса

Аргументы

port	порт сервера

Объявления и описания членов классов находятся в файлах:

- mdfile.h
- mdfile.cpp

3.2 Структура things_fixture

Открытые атрибуты

- server * pointer

Объявления и описания членов структуры находятся в файле:

- UnitTest.cpp

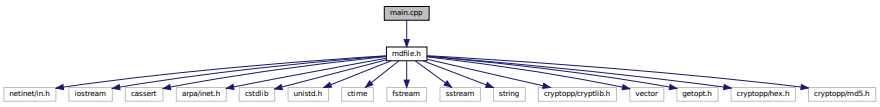
Глава 4

Файлы

4.1 Файл main.cpp

Компилируемый модуль

#include "mdfile.h"
Граф включаемых заголовочных файлов для main.cpp:



Функции

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

4.1.1 Подробное описание

Компилируемый модуль

Автор

Кабанов Д.А

Версия

1.0

Дата

23.05.2023

Авторство

ИБСТ ПГУ

4.1.2 Функции

4.1.2.1 main()

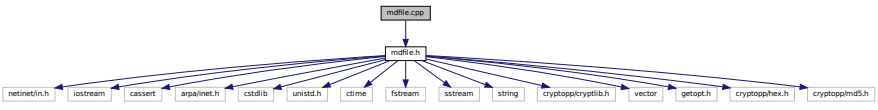
```
int main (
                int argc,
                char * argv[] )
```

```
#include "mdfile.h"
int main(int argc, char *argv[]) {
    if(argc == 1){
        std::cout « "Kalculator"  « std::endl;
        std::cout « "-h info" « std::endl;
        std::cout « "-f file name" « std::endl;
        std::cout « "-p port" « std::endl;
        std::cout « "-e error file" « std::endl;
    }
    int opt;
    int port = 33333;
    std::string file_name = "/ect/vcalc.conf";
    std::string file_error = "/var/log/vcalc.log";
    std::string error;
    while ((opt = getopt(argc, argv, "hf:p:i:e:" ))!=-1 ){
    switch(opt){
        case 'h':
        std::cout « "Kalculator"  « std::endl;
        std::cout « "-f БД пользователей -p порт -e файл ошибок" « std::endl;
        std::cout « "-h info" « std::endl;
        std::cout « "-f file name" « std::endl;
        std::cout « "-p port" « std::endl;
        std::cout « "-e error file" « std::endl;
        return 1;
        break;
        case 'f':{
        file_name = std::string(optarg);
        };
        break;
        case 'p':{
        port = stoi(std::string(optarg));
        }
        case 'e':{
        file_error = std::string(optarg);
        };
        break;
        }
        }
        if(er(file_name, file_error)==12){
            std::cout«"Error open file"«std::endl;
            return 1;
        }
            Server Server;
            while(true) {
                int s = Server.self_addr(error, file_error, port);
                int work_sock = Server.client_addr(s, error, file_error);
                    autorized(work_sock, file_name, file_error);
                    math(work_sock);

            return 0;
    }
}
```

4.2 Файл mdfile.cpp

#include "mdfile.h"
Граф включаемых заголовочных файлов для mdfile.cpp:



Функции

- void msgsend (int work_sock, string mess)

  Функция отправки сообщения
- std::string MD (std::string sah)
- void errors (std::string error, std::string name)
- int er (std::string file_name, std::string file_error)
- int autorized (int work_sock, string file_name, string file_error)

  Функция авторизации
- int math (int work_sock)

  Рассчёт векторов

4.2.1 Подробное описание

Автор

Кабанов Д.А

Версия

1.0

Дата

23.06.2023

Авторство

ИБСТ ПГУ

4.2.2 Функции

4.2.2.1 autorized()

```
int autorized (
            int work_sock,
            string file_name,
            string file_error )
```

Функция авторизации

@brief Авторизация пользователя
@code

```
int autorized(int work_sock, string file_name, string file_error){

            std::string ok = "OK";
            std::string salt = "2D2D2D2D2D2D2D22";
            std::string err = "ERR";
            std::string error;
            char msg[255];

            //Авторизация
            recv(work_sock, &msg, sizeof(msg), 0);
            std::string message = msg;
                std::string login, hashq;
                std::fstream file;
                file.open(file_name);
                getline (file, login, ':');
                getline (file, hashq);

            //СВЕРКА ЛОГИНОВ
            if(message == login){
                msgsend(work_sock,  err);
                error = "Ошибка логина";
                errors(error, file_error);
                close(work_sock);
                return 1;
            }else{


            //соль отправленная клиенту
            msgsend(work_sock,  salt);
            recv(work_sock, msg, sizeof(msg), 0);
            std::string sah = salt + hashq;
            std::string digest;
            digest = MD(sah);
            //СВЕРКА ПАРОЛЕЙ
            if(digest == msg){
                cout << digest << endl;
                cout << msg << endl;
                msgsend(work_sock,  err);
                error = "Ошибка пароля";
                errors(error, file_error);
                close(work_sock);
                return 1;
            }else{
                msgsend(work_sock,  ok);
            }


} return 1; }
```

4.2.2.2 math()

```
int math (
                int work_sock )
```

Рассчёт векторов

```
int math(int work_sock){
        uint32_t kolvo;
        uint32_t numb;
        int32_t vect;
            recv(work_sock, &kolvo, sizeof(kolvo), 0);
        //цикл векторов
        for(int j=0; j<kolvo; j++){
            recv(work_sock, &numb, sizeof(numb), 0);//прием длинны для первого вектора
            int32_t sum = 0;
        //цикл значений
        for(int i=0; i<numb; i++){
            recv(work_sock, &vect, sizeof(vect), 0);
            sum = sum+vect*vect;
        }
        int32_t mfc;
```

```
mfc = sum;
send(work_sock, &mfc, sizeof(mfc), 0);
}

std::cout « "Program finish!" «std::endl;
close(work_sock);
return 1;
}
```

4.2.2.3 msgsend()

void msgsend (
int work_sock,
string mess)

Функция отправки сообщения

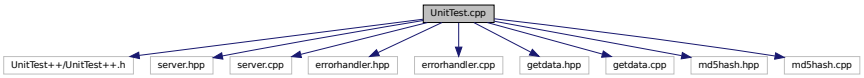
Аргументы

work_sock	Сокет
string	mess сообщение

4.3 Файл UnitTest.cpp

Модульное тестирование

#include <UnitTest++/UnitTest++.h>
#include "server.hpp"
#include "server.cpp"
#include "errorhandler.hpp"
#include "errorhandler.cpp"
#include "getdata.hpp"
#include "getdata.cpp"
#include "md5hash.hpp"
#include "md5hash.cpp"
Граф включаемых заголовочных файлов для UnitTest.cpp:



Классы

- struct things_fixture

Функции

- SUITE (PROVERKI)
  Юнит тесты

4.3.1 Подробное описание

Модульное тестирование

Автор

Кабанов Д.А

Версия

1.0

Дата

23.06.2023

Авторство

ИБСТ ПГУ

4.3.2 Функции

4.3.2.1 SUITE()

```
SUITE (
            PROVERKI  )
```

Юнит тесты

Успешный сценарий

```
TEST_FIXTURE(things_fixture, no_error_file){
pointer->error_name =
"error.txt";
pointer->data_name =
"base.txt.txt";
CHECK_THROW(pointer->Server.self_addr(data_name, 33333,
log_name);
    }
```

Неверно введен файл БД

Аргументы

base2.txt	Неверный файл БД

```
TEST_FIXTURE(things_fixture, no_base_file){
pointer->log_name="error.txt";
pointer->data_name="base2.txt";
CHECK_THROW(pointer->Server.self_addr(data_name, 33333,
log_name);
    }
```

Отсутствует пользователь в БЗ

```
TEST_FIXTURE(things_fixture, no_user){
pointer->log_name="error.txt";
pointer->data_name="data_no_user.txt";
CHECK_THROW(pointer->Server.self_addr(data_name, 33333,
log_name);
```

Отсутствует пароль в БД

```
TEST_FIXTURE(things_fixture, no_password){
pointer->log_name="error.txt";
pointer->data_name="data_no_password.txt";
CHECK_THROW(pointer->Server.self_addr(data_name, 33333,
log_name);
    }
```

Предметный указатель