Курсовая работа: Программная реализация сетевого сервера.

Чернышев Ринат 24ПТ2

1.0

Создано системой Doxygen 1.9.1

Глава 1

Отчет о функциональном тестировании сервера

1.1 Информация о тестировании

- Дата тестирования: 2025-12-26 16:52:02
- Версия сервера: SHA-224 с суммой квадратов

1.2 Тест-кейсы функционального тестирования

ID теста	Название	Тип	Описание	Ожидаемый результат	Полученный результат	Итог
1	Базовый запуск сервера	Позитивный	Запуск сервера с тестовой БД на порту 33444	Сервер запускается и слушает порт	Успешно	ПРОЙДЕН
2	Создание лог-файла	Позитивный	Проверка создания и записи в лог-файл	Лог-файл создается при старте сервера	Успешно	ПРОЙДЕН
3	Загрузка пользовательской БД	Позитивный	Проверка загрузки пользовательской базы данных	Сервер запускается и загружает пользователей	Успешно	ПРОЙДЕН
4	Параметры командной строки	Позитивный	Проверка обработки параметров командной строки	Вывод справки при параметре -h	Успешно	ПРОЙДЕН

ID теста	Название	Тип	Описание	Ожидаемый результат	Полученный результат	Итог
5	Несуществующий файл БД	Позитивный	Обработка несуществующего файла базы данных	Сервер запускается (БД может быть пустой)	Успешно	ПРОЙДЕН

1.3 Сводка результатов

Всего тестов	Пройдено	Провалено	Успешность
5	5	0	100%

1.4 Заключение

[PASS] ВСЕ ТЕСТЫ УСПЕШНО ПРОЙДЕНЫ!

Сервер соответствует функциональным требованиям и готов к использованию.

Глава 2

Алфавитный указатель классов

2.1 Классы

Классы с их кратким описанием.

Глава 3

Список файлов

3.1 Файлы

Полный список файлов.

Глава 4

Классы

4.1 Класс Server

Класс сервера для обработки клиентских подключений.

#include <server.h>

Открытые члены

- Server (int port, const std::string &userDbPath, const std::string &logPath)
  Конструктор сервера.
- bool start ()
  Запускает сервер и начинает прослушивание порта.

Закрытые члены

- void logError (const std::string &message, bool isCritical)
  Записывает сообщение об ошибке в журнал.
- void loadUserDatabase ()
  Загружает базу данных пользователей из файла.
- std::string sha224Hash (const std::string &input)
  Вычисляет SHA-224 хэш строки.
- std::string generateSalt ()
  Генерирует случайную соль для аутентификации.
- void handleClient (int clientSocket)
  Обрабатывает подключение клиента.
- bool authenticate (int clientSocket)
  Аутентифицирует клиента по протоколу SHA-224.
- void processVectors (int clientSocket)
  Обрабатывает передачу векторов от аутентифицированного клиента.
- int16_t calculateSumOfSquares (const std::vector< int16_t > &vector)
  Вычисляет сумму квадратов элементов вектора.
- bool readExact (int socket, void *buffer, size_t size)
  Читает точное количество байт из сокета.

Закрытые данные

- int port

  Порт сервера
- std::string userDbPath

  Путь к базе пользователей
- std::string logPath

  Путь к файлу журнала
- std::unordered_map< std::string, std::string > users

  Кэш пользователей (логин:пароль)

4.1.1 Подробное описание

Класс сервера для обработки клиентских подключений.

Обеспечивает сетевую коммуникацию, аутентификацию пользователей по протоколу SHA-224 с генерацией соли на сервере, прием и обработку векторных данных в двоичном формате, а также логирование.

4.1.2 Конструктор(ы)

4.1.2.1 Server()

```
Server::Server (
            int port,
            const std::string & userDbPath,
            const std::string & logPath )
```

Конструктор сервера.

Конструктор класса Server.

Аргументы

port	Порт для прослушивания подключений.
userDbPath	Путь к файлу базы данных пользователей.
logPath	Путь к файлу журнала сервера.

4.1.3 Методы

4.1.3.1 authenticate()

```
bool Server::authenticate (
            int clientSocket )  [private]
```

Аутентифицирует клиента по протоколу SHA-224.

Аутентифицирует клиента по протоколу SHA-224 с солью.

Аргументы

clientSocket	Дескриптор сокета клиента.

Возвращает

true если аутентификация успешна.

Протокол:

1. Клиент отправляет логин
2. Сервер генерирует и отправляет соль (16 hex символов)
3. Клиент отправляет HASH(SALT || PASSWORD)
4. Сервер проверяет хэш

Аргументы

clientSocket	Дескриптор сокета клиента.

Возвращает

true если аутентификация успешна, false в противном случае.

4.1.3.2 calculateSumOfSquares()

int16_t Server::calculateSumOfSquares (
const std::vector< int16_t > & vector) [private]

Вычисляет сумму квадратов элементов вектора.

Вычисляет сумму квадратов элементов вектора с проверкой переполнения.

Аргументы

vector	Вектор 16-битных целых чисел для обработки.

Возвращает

Сумма квадратов элементов.

При переполнении вверх возвращает 32767 (2^15), при переполнении вниз возвращает -32768 (-2^15).

Аргументы

vector	Вектор 16-битных целых чисел.

Возвращает

Сумма квадратов элементов вектора.

При переполнении вверх возвращает 32767 (2^15), при переполнении вниз возвращает -32768 (-2^15).

4.1.3.3 generateSalt()

std::string Server::generateSalt ( ) [private]

Генерирует случайную соль для аутентификации.

Возвращает

Соль в виде строки из 16 шестнадцатеричных символов (64 бита).

4.1.3.4 handleClient()

void Server::handleClient (
int clientSocket) [private]

Обрабатывает подключение клиента.

Обрабатывает подключение одного клиента.

Аргументы

clientSocket	Дескриптор сокета клиента для обмена данными.
clientSocket	Дескриптор сокета клиента.

4.1.3.5 loadUserDatabase()

void Server::loadUserDatabase ( ) [private]

Загружает базу данных пользователей из файла.

Формат файла: каждая строка содержит "логин:пароль".

Читает файл построчно, парсит строки формата "логин:пароль" и сохраняет данные во внутреннем контейнере users.

4.1.3.6 logError()

void Server::logError (
const std::string & message,
bool isCritical) [private]

Записывает сообщение об ошибке в журнал.

Записывает сообщение об ошибке в файл журнала.

Аргументы

message	Текст сообщения об ошибке.
isCritical	Флаг критичности ошибки (true для критических).
message	Текст сообщения об ошибке.
isCritical	Флаг критичности ошибки.

4.1.3.7 processVectors()

void Server::processVectors (
int clientSocket) [private]

Обрабатывает передачу векторов от аутентифицированного клиента.

Аргументы

clientSocket	Дескриптор сокета клиента для обмена данными.

Ожидает данные в двоичном формате:

- количество векторов (uint32_t)
- для каждого вектора:
  - размер (uint32_t)
  - данные (int16_t[]) Отправляет результаты в двоичном формате:
- количество результатов (uint32_t)
- результаты (int16_t[])

Аргументы

clientSocket	Дескриптор сокета клиента.

Ожидает данные в двоичном формате согласно ТЗ.

4.1.3.8 readExact()

```
bool Server::readExact (
            int socket,
            void * buffer,
            size_t size )   [private]
```

Читает точное количество байт из сокета.

Вспомогательная функция для точного чтения из сокета.

Аргументы

socket	Дескриптор сокета для чтения.
buffer	Буфер для принятых данных.
size	Количество байт для чтения.

Возвращает

true если все байты прочитаны, false при ошибке.

Аргументы

socket	Дескриптор сокета.
buffer	Буфер для данных.
size	Количество байт для чтения.

Возвращает

true если все байты прочитаны, false при ошибке.

4.1.3.9 sha224Hash()

```
std::string Server::sha224Hash (
            const std::string & input )   [private]
```

Вычисляет SHA-224 хэш строки.

Вычисляет SHA-224 хэш для входной строки.

Аргументы

input	Входная строка для хэширования.

Возвращает

SHA-224 хэш в шестнадцатеричном формате (56 символов, верхний регистр).

Аргументы

input	Входная строка для хэширования.

Возвращает

SHA-224 хэш строки в шестнадцатеричном формате (верхний регистр, 56 символов).

4.1.3.10 start()

bool Server::start ( )

Запускает сервер и начинает прослушивание порта.

Запускает основной цикл работы сервера.

Возвращает

true если сервер успешно запущен, false при ошибке.
true если сервер успешно запущен, false при критической ошибке.

4.1.4 Данные класса

4.1.4.1 logPath

std::string Server::logPath [private]

Путь к файлу журнала

4.1.4.2 port

int Server::port [private]

Порт сервера

4.1.4.3 userDbPath

std::string Server::userDbPath [private]

Путь к базе пользователей

4.1.4.4 users

std::unordered_map<std::string, std::string> Server::users [private]

Кэш пользователей (логин:пароль)

Объявления и описания членов классов находятся в файлах:

- server.h
- server.cpp

Глава 5

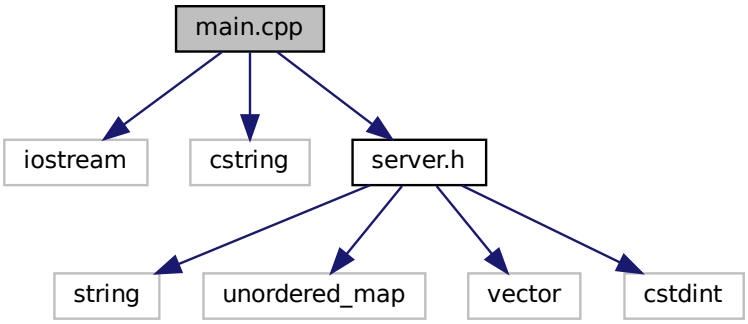
Файлы

5.1 Файл final_test_report.md

5.2 Файл main.cpp

Точка входа серверного приложения.

#include <iostream>
#include <cstring>
#include "server.h"
Граф включаемых заголовочных файлов для main.cpp:



Функции

- void showHelp ()

  Выводит справочную информацию о параметрах командной строки.
- int main (int argc, char *argv[])

  Основная функция серверного приложения.

5.2.1 Подробное описание

Точка входа серверного приложения.

Автор

Чернышев Ринат Рустямович

Дата

26.12.2025

Основная программа, которая парсит аргументы командной строки, создает и запускает экземпляр сервера для обработки клиентских подключений.

5.2.2 Функции

5.2.2.1 main()

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

Основная функция серверного приложения.

Аргументы

argc	Количество аргументов командной строки.
argv	Массив строк-аргументов.

Возвращает

Код завершения программы:

- 0: успешное завершение или показ справки
- 1: ошибка запуска сервера

5.2.2.2 showHelp()

```
void showHelp ( )
```

Выводит справочную информацию о параметрах командной строки.

Отображает список доступных опций для запуска сервера и их значения по умолчанию.

5.3 Файл server.cpp

Реализация методов класса Server для обработки клиентских подключений.

```
#include "server.h"
#include <iostream>
#include <fstream>
#include <sstream>
#include <vector>
#include <cstring>
#include <ctime>
#include <iomanip>
#include <random>
#include <sys/socket.h>
#include <netinet/in.h>
#include <unistd.h>
#include <arpa/inet.h>
#include <openssl/evp.h>
#include <cstdlib>
```

Граф включаемых заголовочных файлов для server.cpp:

5.3.1 Подробное описание

Реализация методов класса Server для обработки клиентских подключений.

Автор

Чернышев Ринат Рустямович

Дата

26.12.2025

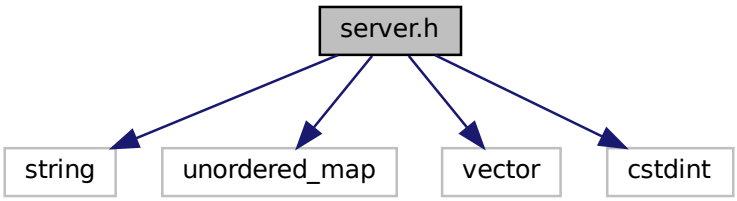
Содержит реализацию сетевого сервера с аутентификацией пользователей по протоколу SHA-224, обработкой векторных данных и логированием событий.

5.4 Файл server.h

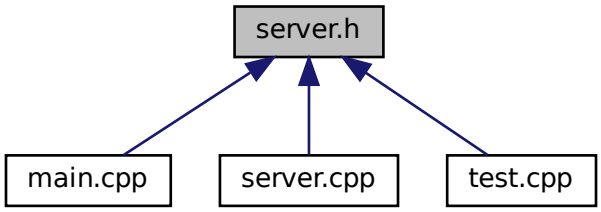
Заголовочный файл класса Server.

```
#include <string>
#include <unordered_map>
#include <vector>
```

#include <cstdint>
Граф включаемых заголовочных файлов для server.h:



Граф файлов, в которые включается этот файл:



Классы

- class Server

  Класс сервера для обработки клиентских подключений.

5.4.1 Подробное описание

Заголовочный файл класса Server.

Автор

Чернышев Ринат Рустямович

Дата

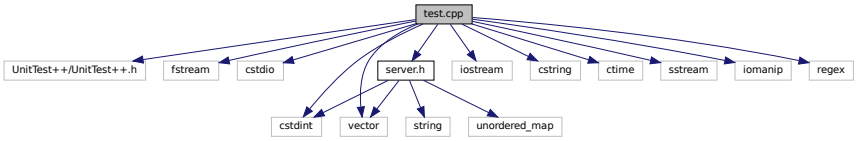
26.12.2025

Объявление класса сервера для обработки сетевых подключений, аутентификации клиентов по протоколу SHA-224 и вычисления суммы квадратов векторов.

5.5 Файл test.cpp

```
#include <UnitTest++/UnitTest++.h>
#include <fstream>
#include <cstdio>
#include <cstdint>
#include <vector>
#include <iostream>
#include <cstring>
#include <ctime>
#include <sstream>
#include <iomanip>
#include <regex>
#include "server.h"
```

Граф включаемых заголовочных файлов для test.cpp:



Функции

- string createTempUserDb (const vector< pair< string, string >> &users)
- void deleteTempFile (const string &filename)
- SUITE (CalculationTest)
- SUITE (SHA224HashTest)
- SUITE (SaltGenerationTest)
- SUITE (UserDatabaseTest)
- SUITE (SimpleConstructorTest)
- SUITE (AuthIntegrationTest)
- int main ()

5.5.1 Функции

5.5.1.1 createTempUserDb()

```
string createTempUserDb (
            const vector< pair< string, string >> & users )
```

5.5.1.2 deleteTempFile()

```
void deleteTempFile (
            const string & filename )
```

5.5.1.3 main()

int main ( )

5.5.1.4 SUITE() [1/6]

SUITE (
AuthIntegrationTest)

5.5.1.5 SUITE() [2/6]

SUITE (
CalculationTest)

5.5.1.6 SUITE() [3/6]

SUITE (
SaltGenerationTest)

5.5.1.7 SUITE() [4/6]

SUITE (
SHA224HashTest)

5.5.1.8 SUITE() [5/6]

SUITE (
SimpleConstructorTest)

5.5.1.9 SUITE() [6/6]

SUITE (
UserDatabaseTest)

Предметный указатель