Балаковский инженерно-технологический институт - филиал

федерального государственного автономного образовательного учреждения

высшего образования

«Национальный исследовательский ядерный университет «МИФИ»

Факультет атомной энергетики и технологий

Кафедра информационные системы и технологии

КОНТРОЛЬНАЯ РАБОТА

по дисциплине

Современные технологии разработки распределенного программного обеспечения

Выполнил: студент группы ИФСТ-4з

Смирнов Денис Алексеевич

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024г.

|  |  |
| --- | --- |
|  | Проверил:  Ефремов Роман Валерьевич  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г. |

Балаково 2024

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc156177281)

[1 Понятие одноранговая сеть и ее роль в глобальной сети 4](#_Toc156177282)

[2 Структура одноранговой сети 6](#_Toc156177283)

[3 Одноранговая сеть на базе Windows 9x 10](#_Toc156177284)

[ЗАКЛЮЧЕНИЕ 14](#_Toc156177285)

[ПРАКТИЧЕСКАЯ ЧАСТЬ 15](#_Toc156177286)

# ВВЕДЕНИЕ

В случае, когда мы делимся файлами с помощью программ типа eMule(еМул), eDonkey (еДонки),и им подобных, такого рода обмен сопряжён с работой огромной машины, принципы работы которой, её истоки и идеи, на которых она построена, мы зачастую представляем себе довольно слабо.

Необходимость иметь доступ к огромному количеству информации, лежащей на других компьютерах, сначала породила локальные сети. Это помогло на некоторое время, но объединить таким образом большое количество компьютеров оказалось невозможно. Возможность же доступа к файлам, возможно имеющихся у кого-то ещё, была очень заманчива. Поэтому была предложена система, позволяющая принимать и передавать файлы с каждого компьютера, с которым есть связь и на котором имеется определенное программное обеспечение. Такие системы получили название п2п (p2p) или пиринговых (peer-to-peer) сетей. Это словосочетание можно перевести как «равноправный обмен». Это значит, что вы, имея некоторые файлы, которые могут быть интересны кому-то ещё, разрешаете желающим их у вас скопировать, а взамен получаете возможность получить файлы, необходимые вам, от тех, у кого они есть.

Так как в настоящее время Интернет получает всё большую и большую распространённость, использование пиринговых сетей является актуальнее с каждым днём. Ведь на сегодняшний день найти нужный файл (любимый фильм, новую программу для своего «электронного друга», только что вышедшею песню любимой звезды, книгу и просто какой-либо интересующий вас файл) намного проще, если вы знаете что такое пиринговые сети, и знаете как ими пользоваться.

# 1 Понятие одноранговая сеть и ее роль в глобальной сети

Однора́нговая, децентрализо́ванная или пи́ринговая (от англ. peer-to-peer, P2P — равный к равному) сеть — это оверлейная компьютерная сеть (рисунок 1), основанная на равноправии участников. В такой сети отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером. В отличие от архитектуры клиент-сервера, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов. Участниками сети являются пиры.

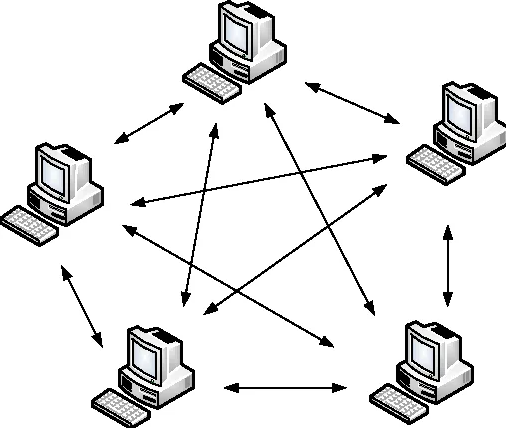


Рисунок 1 - Пиринговые (одноранговые) сети

Одноранговые сети не содержат выделенных серверов. Функции управления сетью распределены между всеми рабочими станциями сети.

Как правило, рабочие станции имеют доступ к дискам и принтерам других рабочих станций, что облегчает совместную работу пользователей в группе, но несколько снижает производительность сети. Одним из достоинств одноранговых сетей является простота создания и обслуживания сети (как правило при этом не требуется системный администратор), но это же является и недостатком – в такой сети трудно обеспечить разграничение доступа, защиту информации и централизованное обновление данных.

В качестве примеров одноранговых сетей можно выделить:

* LANtastic фирмы Artisoft;
* LANSmart фирмы D-Link Systems;
* NET-30 фирмы Invisible Software.

Особо следует отметить, что в состав самой распространенной сегодня операционной системы Microsoft Windows уже входят компоненты, необходимые для построения одноранговой сети. Операционная система сама предлагает установить их при обнаружении на компьютере сетевого адаптера. Если в сети с централизованным управлением работают компьютеры под управлением Windows, то в такой сети можно сочетать возможности централизованного управления с возможностями одноранговых сетей.

# 2 Структура одноранговой сети

Сетевая операционная система составляет основу любой вычислительной сети. Каждый компьютер в сети в значительной степени автономен, поэтому под сетевой операционной системой в широком смысле понимается совокупность операционных систем отдельных компьютеров, взаимодействующих с целью обмена сообщениями и разделения ресурсов по единым правилам - протоколам. В узком смысле сетевая ОС - это операционная система отдельного компьютера, обеспечивающая ему возможность работать в сети.

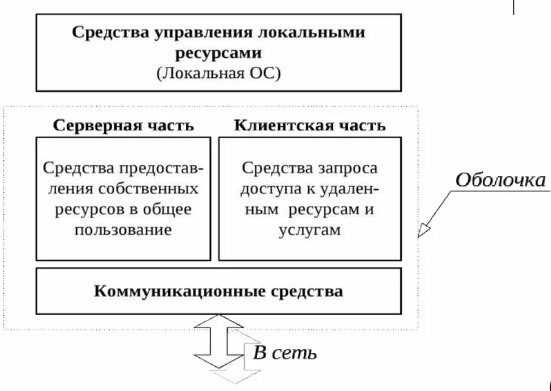


Рисунок 2 - Структура сетевой ОС

В сетевой операционной системе отдельной машины можно выделить несколько частей (рисунок 2):

1. Средства управления локальными ресурсами компьютера: функции распределения оперативной памяти между процессами, планирования и диспетчеризации процессов, управления процессорами в мультипроцессорных машинах, управления периферийными устройствами и другие функции управления ресурсами локальных ОС.
2. Средства предоставления собственных ресурсов и услуг в общее пользование - серверная часть ОС (сервер). Эти средства обеспечивают, например, блокировку файлов и записей, что необходимо для их совместного использования; ведение справочников имен сетевых ресурсов; обработку запросов удаленного доступа к собственной файловой системе и базе данных; управление очередями запросов удаленных пользователей к своим периферийным устройствам.
3. Средства запроса доступа к удаленным ресурсам и услугам и их использования - клиентская часть ОС (редиректор). Эта часть выполняет распознавание и перенаправление в сеть запросов к удаленным ресурсам от приложений и пользователей, при этом запрос поступает от приложения в локальной форме, а передается в сеть в другой форме, соответствующей требованиям сервера. Клиентская часть также осуществляет прием ответов от серверов и преобразование их в локальный формат, так что для приложения выполнение локальных и удаленных запросов неразличимо.
4. Коммуникационные средства ОС, с помощью которых происходит обмен сообщениями в сети. Эта часть обеспечивает адресацию и буферизацию сообщений, выбор маршрута передачи сообщения по сети, надежность передачи и т.п., то есть является средством транспортировки сообщений.

В зависимости от функций, возлагаемых на конкретный компьютер, в его операционной системе может отсутствовать либо клиентская, либо серверная части.

На рисунке 3 показано взаимодействие сетевых компонентов. Здесь компьютер 1 выполняет роль "чистого" клиента, а компьютер 2 - роль "чистого" сервера, соответственно на первой машине отсутствует серверная часть, а на второй - клиентская. На рисунке отдельно показан компонент клиентской части - редиректор. Именно редиректор перехватывает все запросы, поступающие от приложений, и анализирует их. Если выдан запрос к ресурсу данного компьютера, то он переадресовывается соответствующей подсистеме локальной ОС, если же это запрос к удаленному ресурсу, то он переправляется в сеть. При этом клиентская часть преобразует запрос из локальной формы в сетевой формат и передает его транспортной подсистеме, которая отвечает за доставку сообщений указанному серверу. Серверная часть операционной системы компьютера 2 принимает запрос, преобразует его и передает для выполнения своей локальной ОС. После того, как результат получен, сервер обращается к транспортной подсистеме и направляет ответ клиенту, выдавшему запрос. Клиентская часть преобразует результат в соответствующий формат и адресует его тому приложению, которое выдало запрос.

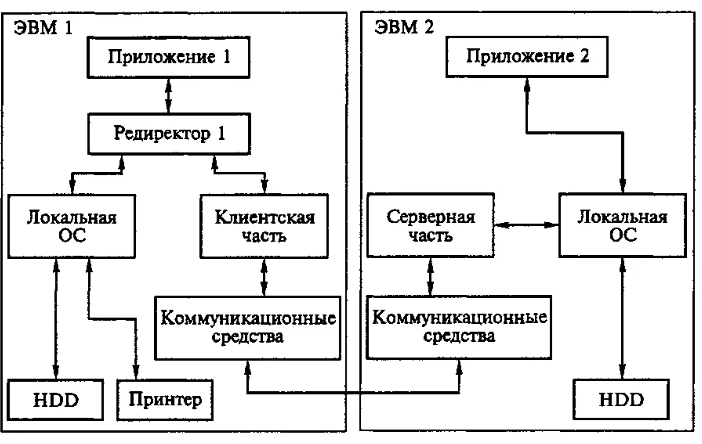


Рисунок 3 - Взаимодействие компонентов операционной системы при взаимодействии компьютеров

На практике сложилось несколько подходов к построению сетевых операционных систем (рисунок 4).

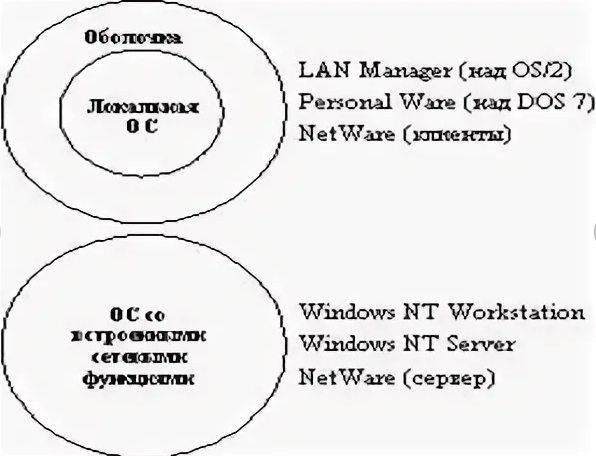


Рисунок 4 - Варианты построения сетевых ОС

Первые сетевые ОС представляли собой совокупность существующей локальной ОС и надстроенной над ней сетевой оболочки. При этом в локальную ОС встраивался минимум сетевых функций, необходимых для работы сетевой оболочки, которая выполняла основные сетевые функции. Примером такого подхода является использование на каждой машине сети операционной системы MS DOS (у которой начиная с ее третьей версии появились такие встроенные функции, как блокировка файлов и записей, необходимые для совместного доступа к файлам). Принцип построения сетевых ОС в виде сетевой оболочки над локальной ОС используется и в современных ОС, таких, например, как LANtastic или Personal Ware.

Однако более эффективным представляется путь разработки операционных систем, изначально предназначенных для работы в сети. Сетевые функции у ОС такого типа глубоко встроены в основные модули системы, что обеспечивает их логическую стройность, простоту эксплуатации и модификации, а также высокую производительность. Примером такой ОС является система Windows NT фирмы Microsoft, которая за счет встроенности сетевых средств обеспечивает более высокие показатели производительности и защищенности информации по сравнению с сетевой ОС LAN Manager той же фирмы (совместная разработка с IBM), являющейся надстройкой над локальной операционной системой OS/2.

# 3 Одноранговая сеть на базе Windows 9x

Как уже было сказано выше, наличие операционной системы Windows 9x на подключенных к сети компьютерах достаточно для организации одноранговой сети. При добавлении к компьютеру сетевого адаптера на Рабочем столе Windows появляется значок "Сетевое окружение", а также появляется возможность редактировать свойства сети (с помощью соответствующего значка на Панели управления). Рассмотрим возможности настройки сети.

В свойствах сети на вкладке Конфигурация присутствует список установленных компонентов (а также кнопки управления, позволяющие добавлять, удалять компоненты или редактировать их свойства), переключатель способа входа в сеть, а также кнопка настройки доступа к общим папкам и принтерам. Коррекция любых параметров должна завершаться перезагрузкой операционной системы, чтобы сделанные изменения вступили в силу.

Среди сетевых компонентов системы прежде всего должен быть определен драйвер сетевого адаптера. Если система не смогла правильно распознать сетевой адаптер после установки, обратитесь к документации на это устройство.

Кроме драйвера должен быть определен один из протоколов для работы с сетью. Нажав кнопку "Добавить" выберите тип устанавливаемого компонента "Протокол" и среди предлагаемых фирмой Microsoft протоколов выберите либо IPX/SPX-совместимый протокол, либо протокол TCP/IP. Можно выбирать любой протокол, лишь бы на всех компьютерах сети был установлен один и тот же протокол, иначе они не смогут общаться друг с другом. Если в сети предполагается использование программ работы с Internet, то им понадобится протокол TCP/IP, который придется после установки настроить в соответствии с условиями работы вашей сети.

Для обращения к сети кроме протокола понадобится и клиент. Среди клиентов, предлагаемых фирмой Microsoft, следует воспользоваться "Клиентом для сетей Microsoft". После его установки будет выставлен и способ входа в сеть: Вход в сеть Microsoft.

Нажав на кнопку "Доступ к файлам и принтерам" можно определить, будут ли файлы данного компьютера или подключенные к нему принтеры передаваться в общее пользование в пределах создаваемой сети. После утвердительного ответа на этот вопрос система добавит к установленным компонентам Службу доступа к файлам и принтерам сетей Microsoft.

Для правильной ориентации среди компьютеров сети следует на вкладке Идентификация указать имя компьютера, рабочую группу и описание компьютера. Имя компьютера должно быть уникальным. Имя рабочей группы должно быть одинаковым для тех компьютеров, которым предстоит постоянное взаимодействие друг с другом (в одной сети можно организовать несколько рабочих групп так, чтобы они не мешали друг другу). Описание компьютера может помочь в выборе нужного компьютера при просмотре списка всех компьютеров рабочей группы в окне Сетевого окружения.

После настройки свойств сети на всех компьютерах можно приступать к ее эксплуатации. Для общения пользователей локальной сети можно воспользоваться программой WinPopUp или другими ей подобными, которые должны быть установлены на всех компьютерах. Среди возможностей локальной одноранговой сети рассмотрим работу с "общими ресурсами", в качестве которых могут выступать жесткие диски и принтеры.

Для того, чтобы "отдать" в общее пользование весь диск своего компьютера или отдельный каталог (со всеми подкаталогами) проще всего выделить его имя и выбрать из контекстного меню команду Доступ. В окне настройки доступа к общему ресурсу следует прежде всего активизировать переключатель "Общий ресурс" и указать его сетевое имя и способ доступа к ресурсу. Сетевое имя определяет, как будет виден ваш каталог в Сетевом окружении на других компьютерах. Каталог может быть открыт другим пользователям сети только на чтение или на чтение и запись (последнее означает и возможность стирать файлы на вашем компьютере!). В пределах одноранговой сети разграничение доступа к общему ресурсу возможно лишь на основании пароля (при подключении к вашему ресурсу другому пользователю придется ввести правильный пароль). Можно определить разные пароли "только для чтения" и "полный доступ" (т.е. чтение и запись). Если одноранговая сеть Windows функционирует в пределах сети с централизованным управлением появляется возможность разграничения доступа на уровне пользователей. В этом случае (указав заранее имя сервера, с которого следует брать список пользователей, на вкладке Управление доступом в свойствах сети) можно определить пользователей, которым разрешен доступ "только на чтение" и пользователей с "полным доступом". В этом случае за безопасностью ваших данных будет следить уже сервер локальной сети.

После создания "общего ресурса" соответствующий каталог вашего компьютера изменит свое изображение. Для того, чтобы проверить полный список открытых каталогов лучше всего "взглянуть" на компьютер с точки зрения сети, воспользовавшись окном Сетевого окружения.

Для того, чтобы воспользоваться общим каталогом на другом компьютере следует найти этот компьютер в Сетевом окружении. Дважды щелкнув мышью по его имени можно получить в окне список всех его "общих" каталогов (не все они могут быть доступны для вас). Если "войти" в один из каталогов (введя соответствующий пароль в случае разграничения доступа на уровне ресурсов), то можно работать с файлами и подкаталогами так же , как и на локальном диске своего компьютера (разве что там может быть разрешено только чтение, но не запись). При подключении к общему каталогу можно попросить Windows запомнить пароль (на будущее). Можно также общий каталог подключить как сетевой диск. В этом случае содержимое этого каталога будет выглядеть так же, как содержимое локальных дисков, и обращаться к нему можно будет по выбранному имени диска. Windows может также автоматически подключать этот каталог как сетевой диск при каждом включении компьютера. Разумеется, он будет доступен только при условии, что включен тот компьютер, на котором физически расположен этот каталог.

Сделать принтер своего компьютера "общим ресурсом" можно аналогично каталогу. Для того, чтобы воспользоваться общим принтером на другом компьютере следует прежде всего установить у себя драйвер этого принтера, указав при установке, что это сетевой принтер, и указав его в Сетевом окружении. После установки драйвера можно отправлять задания на печать на этот принтер так же, как и на обычный локальный принтер. Печататься они будут на удаленном принтере, если, конечно, там включен и компьютер, и принтер.

# ЗАКЛЮЧЕНИЕ

Процессор обычной клиентской машины мало загружен. Особенно в офисах, где машины используются преимущественно для подготовки документов, для набора текстов и т.п. То же касается и подавляющего большинства домашних компьютеров.

Пиринговые сети разнообразны. Основной целью одних является обмен музыкальными и видео файлами. В других реализуются проекты поиска лекарства от рака, третьи тренируются во взломе известных шифров на основе распределенных вычислений, четвертые ищут внеземные цивилизации на основе данных, получаемых с радиотелескопов.

С математической точки зрения пиринговая сеть может быть представлена графом неопределенного вида: нет какой-либо стандартной архитектуры сети (например, звезды или кольца). Более того, этот граф – динамический, так как отдельные пользователи включаются в сеть и выходят из ее состава в произвольные моменты времени. Любой пользователь, играющий роль сервера, в любой момент времени может превратиться в клиента на некоторый отрезок времени. Но может и пребывать одновременно в положении и сервера и клиента.

# ПРАКТИЧЕСКАЯ ЧАСТЬ

Клиентское приложение разработана с помощью javax, код программы представлен ниже:

import java.awt.\*;

import java.awt.event.\*;

import java.io.BufferedReader;

import java.io.DataOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.net.HttpURLConnection;

import java.net.URI;

import java.net.URL;

import java.net.http.HttpClient;

import java.net.http.HttpRequest;

import java.nio.charset.StandardCharsets;

import javax.swing.\*;

public class FormApp extends JFrame{

    private JButton button = new JButton("Отправить");

    private JLabel label1 = new JLabel("Имя:");

    private JLabel label2 = new JLabel("Фамилия:");

    private JLabel label3 = new JLabel("email:");

    private JTextField input1 = new JTextField("",5);

    private JTextField input2 = new JTextField("",5);

    private JTextField input3 = new JTextField("",5);

    public FormApp(){

        super("Форма");

        this.setBounds(100,100,300,200);

        this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

        Container container = this.getContentPane();

        container.setLayout(new GridLayout(4,2,2,2));

        container.add(label1);

        container.add(input1);

        container.add(label2);

        container.add(input2);

        container.add(label3);

        container.add(input3);

        button.addActionListener(new ButtonEvent());

        container.add(button);

    }

    class ButtonEvent implements ActionListener{

        public void actionPerformed(ActionEvent e){

            String message = "";

            message += "имя: " + input1.getText() + "\nФамилия: " + input2.getText() + "\nemail: "+ input3.getText();

            JOptionPane.showMessageDialog(null,message,"Output",JOptionPane.PLAIN\_MESSAGE);

            try {

                URL url = new URL("http://127.0.0.1:8000");

                HttpURLConnection conn = (HttpURLConnection) url.openConnection();

                conn.setRequestMethod("POST");

                conn.setDoOutput(true);

                String params = "name:"+input1.getText()+";surname:"+input2.getText()+";email:"+input3.getText();

                byte[] postData = params.getBytes(StandardCharsets.UTF\_8);

                try (DataOutputStream wr = new DataOutputStream(conn.getOutputStream())) {

                    wr.write(postData);

                }

                try (BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream(), "UTF-8"))) {

                    String inputLine;

                    while ((inputLine = in.readLine()) != null) {

                        System.out.println(inputLine);

                    }

                }

            } catch (IOException ee) {

                ee.printStackTrace();

            }

        }

    }

}

При заполнении формы (рисунок 5) и нажатии кнопки «Отправить» отправляется запрос на серверное приложение, которое расположено по адресу http://127.0.0.1:8000 и имеет конечную точку, в которую будет передаваться строка с данными.

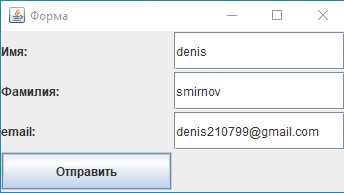


Рисунок 5 – Клиентское приложение

Серверное приложение использует пакет com.sun.net.httpserver, который представляет базовую функциональность для создания HTTP-сервера. Так же используется пакет java.sql для работы с различными СУБД. Ниже представлен код серверного приложения:

import com.sun.net.httpserver.HttpExchange;

import com.sun.net.httpserver.HttpHandler;

import com.sun.net.httpserver.HttpServer;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.net.InetSocketAddress;

import java.sql.\*;

import java.text.ParseException;

import java.util.StringTokenizer;

public class Main {

    public static void main(String[] args) throws IOException {

        HttpServer server = HttpServer.create(new InetSocketAddress(8000), 0);

        server.createContext("/", new MyHandler());

        server.setExecutor(null); // creates a default executor

        server.start();

    }

    static class MyHandler implements HttpHandler {

        @Override

        public void handle(HttpExchange t) throws IOException {

            user user = new user();

            StringBuilder sb = new StringBuilder();

            InputStream ios = t.getRequestBody();

            int i;

            while ((i = ios.read()) != -1) {

                sb.append((char) i);

            }

            StringTokenizer st1 = new StringTokenizer(sb.toString(), ";");

            while (st1.hasMoreTokens())

            {

                String token = st1.nextToken();

                String[] element = token.split(":");

                switch (element[0]){

                    case "name":

                        user.setName(element[1]);

                        break;

                    case "surname":

                        user.setSurname(element[1]);

                        break;

                    case "email":

                        user.setEmail(element[1]);

                        break;

                }

            }

            System.out.println(user.getName());

            System.out.println(user.getSurname());

            System.out.println(user.getEmail());

            String response = "Hello, World123!";

            DB db = null;

            try {

                db = new DB();

            } catch (SQLException e) {

                System.out.println(e.getLocalizedMessage());

                throw new RuntimeException(e);

            } catch (ClassNotFoundException e) {

                System.out.println(e.getLocalizedMessage());

                throw new RuntimeException(e);

            }

            try {

                db.createPersonal(

                        user.getName(),

                        user.getSurname(),

                        user.getEmail()

                );

            } catch (SQLException e) {

                System.out.println(e.getLocalizedMessage());

                throw new RuntimeException(e);

            } catch (ParseException e) {

                throw new RuntimeException(e);

            }

            t.sendResponseHeaders(200, response.length());

            OutputStream os = t.getResponseBody();

            os.write(response.getBytes());

            os.close();

        }

    }

}

Подключение и работа с БД реализована в классе DB.java, ниже представлен код:

import java.sql.\*;

import java.text.ParseException;

public class DB {

    private Connection connection;

    public  DB() throws SQLException, ClassNotFoundException {

        this.connGet();

    }

    public void connGet () throws SQLException, ClassNotFoundException {

        String dbUser = "root";

        String dbPass = "";

        String dbName = "javadb";

        String connUrl = "jdbc:mysql://127.0.0.1:3306/javadb";

        Class.forName("com.mysql.cj.jdbc.Driver");

        this.connection = DriverManager.getConnection(connUrl,dbUser,dbPass);

    }

    public void createPersonal(String name, String surname, String email) throws SQLException, ParseException {

        PreparedStatement statement = this.connection.prepareStatement("INSERT INTO user (id, name, surname, email) VALUES (NULL,?,?,?)");

        statement.setString(1,name);

        statement.setString(2,surname);

        statement.setString(3,email);

        statement.execute();

    }

}

При нажатии на кнопку «Отправить» пользователю выводиться информация (рисунок 6), которую он ввел, после чего данные записываются в БД (рисунок 7).

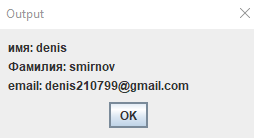


Рисунок 6 – Информационное окно

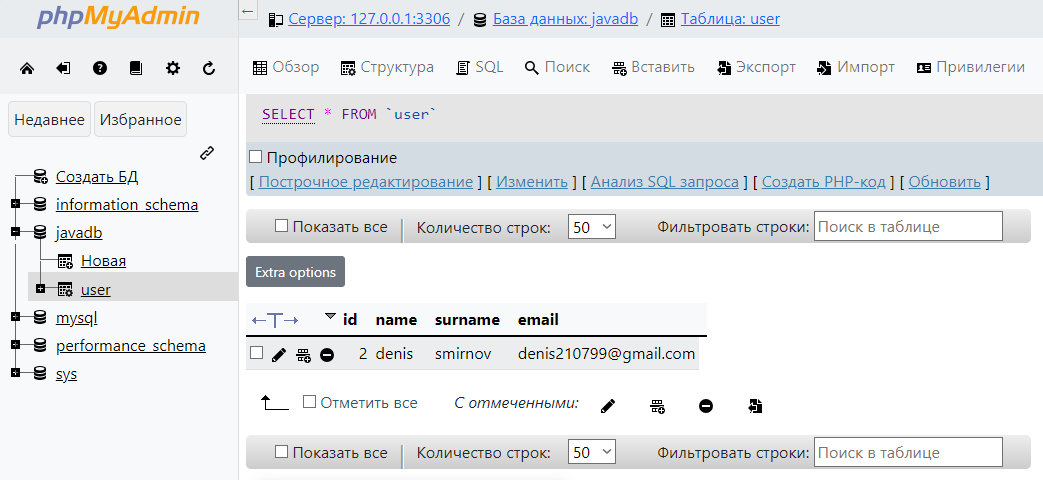


Рисунок 7 – База данных

База данных развернута с помощью локального сервера OpenServer.