

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ» (НГУ)

Кафедра Компьютерных систем

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Минькова Максима Сергеевича

Дистанционное управление компьютерными классами

Направление подготовки 230100.62 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Руководитель

Пищик Б.Н.

к.т.н., доцент

Автор

Миньков М.С.

ФИТ, гр.9204

Новосибирск, 2013г.

Оглавление

Введение	4
Цель работы	4
Актуальность	4
Задачи	4
Научная новизна	5
Практическая ценность	5
Глава 1. Постановка задачи	6
Глава 2. Анализ существующих систем и технологий управления компьютерными классами	8
Глава 3. Архитектура и особенности проектирования	11
Глава 4. Реализация системы дистанционного управления компьютерными классами	13
4.1 Клиентское приложение	12
4.2 Сервер приложений	14
4.3 Приложение панель администратора	16
Глава 5. Обзор базы данных	20
Заключение	22
Литература	24

ВВЕДЕНИЕ

В настоящее время при стремительном развитии компьютерной техники практически в каждом заведении, будь то учебное или производственное, располагаются компьютерные классы. В некоторых организациях количество таких классов может исчисляться десятками, для управления которыми требуется несколько администраторов, которые должны быть достаточно мобильными, чтобы обслуживать каждый из компьютеров отдельно, например, выключать их после окончания рабочего дня.

Каждая из организаций занимается своими задачами, которые могут довольно сильно различаться. Поэтому существует довольно большой спектр операционных систем, на которых эти задачи выполняются. Каждая из этих операционных систем имеет свою архитектуру, которая может быть отличной от других. Следовательно, чтобы управлять классами, на компьютерах которых стоит разное программное обеспечение, требуется система управления компьютерными классами, которая смогла бы работать в широком диапазоне операционных систем.

Цель работы.

Целью данной дипломной работы является разработка системы, позволяющей дистанционно наблюдать за компьютерными классами, управлять компьютерами, находящимися в наблюдаемых компьютерных классах, а также получать сведения о работе компьютеров. Система должна быть кроссплатформенной, то есть запускаться на нескольких операционных системах и уметь исполнять такие функции, как дистанционное выключение компьютера, показ компьютеров запущенных в данный момент, показ запущенных процессов на работающих компьютерах, просмотр статистики работы компьютеров.

Актуальность.

Так как подходящих под требование кроссплатформенности систем дистанционного управления компьютерными классами не существует, проблема обеспечения автоматизации работы администратора и упрощения сбора данных о компьютерах, использующих несколько операционных систем, является актуальной проблемой администрирования. Таким образом, создание кроссплатформенной системы дистанционного управления компьютерными классами является важной задачей для повышения эффективности работы администратора.

Задачи.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Определить требования к разрабатываемой системе на основе наиболее частых действий администраторов, а также беря за основу программные и аппаратные характеристики компьютеров, чаще всего находящихся в компьютерных классах.
2. Провести обзор существующих систем и технологий управления компьютерными классами и выявить их недостатки и преимущества. Также определить возможность их использования.
3. Найти подходящую под установленные требования программную архитектуру.
4. Спроектировать систему дистанционного управления компьютерными классами на основе полученной программной архитектуры.
5. Реализовать систему дистанционного управления компьютерными классами, используя полученные на этапе проектирования данные.

Научная новизна.

Научная новизна заключается в проектировании, отсутствующей в настоящее время, системы дистанционного управления компьютерными классами, способной обеспечить автоматизацию компьютеров, работающих на нескольких операционных системах.

Практическая ценность.

Система дистанционного управления компьютерными классами позволяет уменьшить время, которое требуется администраторам на наблюдение за компьютерами, находящимися в компьютерных классах, а также увеличить эффективность работы администраторов за счет схематичного представления расположения компьютеров на экране администраторов. Благодаря данной системе администраторы будут знать какие компьютеры работают в данный момент времени, запущенные на них процессы, и, соответственно, знать текущее состояние компьютеров.

Глава 1. Постановка задачи

Первым шагом разработки является формулирование цели. Далее необходимо провести анализ требований к системе, а также определить ее базовый функционал.

Целью проекта является проектирование и реализация системы, позволяющей дистанционно наблюдать и управлять компьютерными классами, а также просматривать статистику работы компьютеров. Управление должно заключаться в дистанционном выключении, а наблюдение за компьютерными классами должно выражаться в получении списка запущенных процессов, исполняющихся на компьютерах с установленными на них клиентскими приложениями, для того, чтобы администраторы могли обнаружить и, возможно, завершить процесс, например, потребляющий большое количество ресурсов.

Также система дистанционного управления компьютерными классами должна хранить общую статистику работы компьютеров, позволяющую обнаружить время начала и завершения работы, и действия администраторов для определения был ли выключен компьютер дистанционно, самим пользователем, или связь с ним была оборвана, например, произошла потеря сетевого соединения.

Необходимо определить функционал системы дистанционного управления компьютерными классами. Общими администрирующими требованиями к функциональности системы, являются: возможность управления питанием компьютеров, просмотра общей статистики работы компьютера, а также просмотра информации о его текущем состоянии. Ниже расположен подробный список функций, необходимых администраторам компьютерных классов:

1. Добавление компьютера в список наблюдаемых.
2. Просмотр статуса работы компьютерных классов, что позволит определить какие компьютеры работают, а какие выключены в данный момент времени.
3. Дистанционное выключение компьютеров, находящихся в наблюдаемых компьютерных классах.
4. Просмотр списка процессов, запущенных на наблюдаемом в данный момент компьютере. Этот список должен состоять из имени запущенного процесса, количества потребляемого им запаса оперативной памяти в текущий момент времени.
5. Просмотр статистики работы компьютерных классов, с помощью которой можно определить общее время работы любого компьютера за любой промежуток времени.

Для проведения краткого анализа требований к системе дистанционного управления компьютерными классами были использованы наиболее актуальные проблемы администрирования таких классов. В результате были установлены актуальные требования к разрабатываемой системе:

- Администратор не обязательно будет один, в следствии чего необходимо заложить в систему поддержку работы с несколькими управляющими компьютерами.
- В связи с большим количеством разных операционных систем установленных на наблюдаемых компьютерах должна быть обеспечена кроссплатформенность системы.
- Система должна быть простой в настройке и обладать простым интерфейсом.

Глава 2. Анализ существующих систем и технологий управления компьютерными классами

Обзор современных существующих решений в области управления компьютерными классами играет немаловажную роль. Он также поможет более точно оценить существующие проблемы, которые находятся перед разработчиками таких систем, узнать, какие функции позволяют совершать уже готовые решения и в чем их недостатки.

После поиска существующих систем управления компьютерами и компьютерными классами, подходящих под поставленные требования, было установлено, что таких систем на современном рынке находится много. Все они обладают разным функционалом и, как следствие, подходят для использования в разных средах.

Для исследования были выбраны наиболее популярные решения, а именно: Netop Vision, Radmin, SMART Sync Classroom manager, а также решение под названием Ammuu Admin - несколько отдаленное от прямой заданной цели, но обладающее другой архитектурой.

Большинство систем управления компьютерными классами распространяются с помощью платной лицензии и перечисленные выше не являются исключениями. Стоимость лицензии на один компьютер по состоянию на 2013 год составляет от 600 до 2000 рублей. Все рассматриваемые образцы предоставляют бесплатные ознакомительные приложения, в которых выполнение заявленных функций сильно ограничено.

Перед началом подробного обзора выбранных решений необходимо отметить их общие особенности. Все решения работают с операционной системой Windows, начиная с Windows XP и выше. Почти все рассматриваемые системы обладают клиент-серверной архитектурой, состоящей из 2 частей: клиентского приложения и серверного. Администратор управляет клиентскими приложениями из единственного в системе централизованного сервера. Поэтому если администраторов несколько, то им приходится делить один компьютер.

Начнем рассмотрение с самого популярного решения в данной области - Netop Vision. Его недостатками являются невозможность работы на операционной системе Linux и отсутствие русской локализации. Из преимуществ стоит выделить гибкую настройку системы, возможность просмотра экрана наблюдаемого компьютера, а также общение с пользователем находящимся за данным компьютером. Основные задачи системы — дистанционное наблюдение за компьютерными классами и компьютерами,

расположенными в них. Обладает такими функциями, как выключение компьютера, просмотр запущенных задач, а также мониторинг ресурсов оперативной памяти, жесткого диска и загрузки центрального процессора. Администратор сам задает, какие компьютеры будут находиться в каждом из компьютерных классов. Система состоит из двух приложений: клиента, который отдельно устанавливается на каждый из компьютеров и сервера, который устанавливается на компьютер администратора.

Radmin предоставляет несколько решений управления компьютерными классами и отдельно самими компьютерами. Из недостатков выделяется работоспособность только на операционных системах Windows и ни на каких-либо других. Также система может осуществлять активный захват компьютера, который не дает возможности работы с ним пользователю, но открывает полный доступ к управлению компьютером. Основная задача Radmin – подключение к удаленным компьютерам и просмотр информации или выполнение на нем каких-либо операций, поэтому система может выполнять функции, которые можно выполнить на компьютере. Данная система имеет помимо активного захвата компьютера такие особенности, как общение с пользователем, наблюдение сразу за несколькими компьютерами, также обладает высоким уровнем безопасности и высокой скоростью работы.

Третье исследуемое решение, система, предоставляемая SMART Sync. Из преимуществ выделяется получение расширенной информации о компьютерах. Из недостатков — невозможность работы на Unix-подобных системах. Основная задача системы — наблюдение за компьютерами, компьютерными классами, проведение компьютерных занятий. SMART Sync Classroom manager может выключать, перезагружать компьютеры или компьютерные классы, управлять запущенными процессами, а также может захватывать управление компьютером.

Последнее решение, Ammyu Admin, не может наблюдать сразу за несколькими компьютерами, а лишь способно управлять каждым по отдельности. Из преимуществ системы выделяется тот факт, что система состоит из одного приложения, которое не требует установки. Из недостатков — плохое обеспечение безопасности системы и частые потери соединения. Система позволяет выключать, перезагружать компьютеры, полноценно работать с файлами на компьютере клиента.

Подводя итог краткому исследованию существующих решений можно выделить общий ряд преимуществ и недостатков. Все рассматриваемые системы имеют большинство требуемых функций, а также функций, с помощью которых можно получить требуемый результат. Также все рассматриваемые комплексы не являются

кроссплатформенными, так как не поддерживают операционную систему Linux, что является минусом. Некоторые решения не поддерживают русский язык, а некоторые требуют достаточно много ресурсов. Исходя из этого, подходящей под основные поставленные требования системы дистанционного управления компьютерными классами, а именно кроссплатформенности, среди исследуемых нет.

Глава 3. Архитектура и особенности проектирования

Архитектура разрабатываемой системы будет базироваться на трехуровневой архитектуре, состоящей из клиентских приложений, серверного приложения и сервера базы данных. Ниже предоставлена структура системы:

- Клиентское приложение, устанавливаемое на компьютеры в компьютерных классах соответствует требованиям безопасности, масштабируемости и надежности, которое достигается за счет отсутствия доступа к базе данных, отсутствия основной бизнес-логики и отсутствием хранения состояния приложения. Данное клиентское приложение взаимодействует только с сервером приложений, передавая данные о состоянии компьютера и исполняя команды, полученные от сервера приложений.
- Клиентское приложение для администраторов, именуемое далее панелью администратора, которое с точки зрения архитектуры является копией обычного клиентского приложения с отличием в логике, и которое заключается в управлении компьютерами с клиентскими приложениями через сервер приложений.
- Сервер приложений, содержащий основную бизнес-логику, заключающуюся в обработке данных, получаемых от клиентских приложений и обработке команд, полученных от панели администратора. Вне сервера приложений находятся фрагменты данных, хранимых в базе данных.
- Сервер базы данных обеспечивает хранение данных, получаемых от сервера приложений.

В результате использования вышеописанной структуры системы можно выделить основные особенности проектирования системы дистанционного управления компьютерными классами:

- Полученная система будет гибкой в модернизации.
- Логика системы будет разделена на соответствующие части: клиентскую, серверную и базу данных.
- Система будет обладать высокой безопасностью и надежностью.
- Система будет гибко конфигурироваться.

Общие требования к разрабатываемой системе должны строиться на основе функционала системы и актуальных для администраторов проблем. В связи с большим количеством операционных систем должно быть обеспечено основное требование — кроссплатформенность. Проектируемая система должна уметь работать с

операционными системами Windows и Linux. Работоспособность с другими операционными системами должна быть заложена.

Для обеспечения кроссплатформенности необходимо использовать язык высокого уровня Java, так как именно он предоставляет независимость от платформы. Также язык Java предоставляет широкий спектр средств для обмена сообщениями между приложениями и гибкую и тонкую в настройке графическую систему.

Глава 4. Реализация системы дистанционного управления компьютерными классами

В текущей главе будут изложены детали реализации системы дистанционного управления компьютерными классами, а также полное описание действий системы в различных ситуациях.

Как было установлено ранее, система должна состоять из базы данных и трех приложений, а именно: клиентского приложения, сервера приложений и панели администратора.

4.1 Клиентское приложение

Начнем рассмотрение системы с клиентского приложения, которое устанавливается на компьютеры в компьютерных классах.

Клиентское приложение — это процесс-«демон», который запускается со старта операционной системы или ручным способом и постоянно исполняется в фоновом режиме. Механизм работы клиентского приложения заключается в постоянной отправке специальных пакетов серверу приложений через заданное время, а также в получении запроса от сервера приложений на отправку списка запущенных процессов на данном компьютере серверу приложений и в обработке приказа от сервера приложений о выключении компьютера.

Остановимся подробнее на внутреннем устройстве отправки пакетов Серверу. Эта функция программно реализована с помощью создания отдельного потока, в котором постоянно исполняется цикл. В цикле находится ожидание отведенного времени, по прошествии которого происходит установка TCP соединения с сервером и передача ему текущего IP-адреса и имени компьютера клиентского приложения. После отправки пакета поток снова переходит в состояние сна.

Обработка команд, получаемых от сервера приложений также осуществляется в отдельном потоке. Данный поток находится в режиме сна, пока сервер приложений не установит с ним прямое соединение. После установки соединения поток получает команду от сервера приложений следующего типа: «shutdown». После обработки полученной команды поток начинает ее исполнение. В случае команды, запрашивающей список запущенных процессов, поток после получения списка процессов передает его серверу приложений.

Рассмотрим более подробно команды, с помощью которых выполняется выключение или получение списка запущенных процессов.

Для выключения компьютера, установленный на нем Клиент выполняет команду в консоле Windows «shutdown -s». Если клиент установлен на операционную

систему Linux, то команда будет выглядеть следующим образом: «shutdown -p now». Так как команды на выключение компьютера на разных операционных системах различаются, потребовалось обеспечить проверку текущей операционной системы и исполнить для нее соответствующую команду. Ниже, на рисунке 1, приведен участок кода, отвечающий за выключение компьютера с установленным на нем Клиентом.

```
String property = System.getProperty("os.name");
System.out.println( property );

if ( property.toLowerCase().contains("windows") ) {
    String[] commands = {"shutdown", "-s"};
    Runtime.getRuntime().exec(commands);
}
else if ( property.toLowerCase().contains("linux") ) {
    String[] commands = {"shutdown", "-p", "now"};
    Runtime.getRuntime().exec(commands);
}
```

Рисунок 1. Реализация в клиентском приложении выключения компьютера

Получение списка запущенных в данный момент процессов происходит по той же схеме, что и выключение компьютера: исполняется команда «tasklist.exe /fo csv /nh», результатом которой является таблица с информацией о процессах. Эта таблица состоит из имени процесса, его идентификатора, имени сессии, номера сеанса и оперативной памяти, которую потребляет процесс. В данной таблице нас интересует только имена процессов и их потребление памяти. После выборки из таблицы необходимых данных образуется строка, в которой находятся все данные и которая будет отправлена серверу.

4.2 Сервер приложений

Работу сервера приложений можно описать следующими словами: постоянное получение информации от клиентских приложений, ее обработка и запись в базу данных, а также перенаправление операций, получаемых из панели администратора к клиентским приложениям и обратно. В число таких операций входят выключение компьютера клиентского приложения и получение его списка процессов.

Сервер приложений также является процессом-«демоном», который запускается либо при старте операционной системы, либо ручным способом. При запуске сервер приложений автоматически устанавливает статусы компьютеров клиентских приложений в базе данных в отключенное положение. Сервер приложений состоит из четырех потоков, каждый из которых работает независимо.

Первый поток постоянно получает данные от клиентских приложений. Реализован данный механизм следующим образом. Клиентское приложение

устанавливает соединение и передает свой пакет серверу приложений. Сервер приложений получает пакет и обрабатывает, выделяя из него имя компьютера и IP-адрес компьютера с клиентским приложением, после чего записывает полученную информацию в базу данных, устанавливая статус компьютера и добавляя текущее время для дальнейшей проверки последней отправки пакета от клиентского приложения. После этого соединение разрывается и сервер приложений переходит в режим слушателя, в котором он ждет новое соединение на заданном порту.

Результатом работы второго потока является проверка статуса компьютеров с клиентскими приложениями, находящихся в базе данных, которая происходит раз в 20 секунд. По прошествии данного времени происходит получение времени последней отправки пакета клиентским приложением. Если разница во времени отличается от текущего более чем на 30 секунд, происходит обнуление статуса компьютера с клиентским приложением. В результате в панели администратора данный компьютер будет помечен как выключенный. Участок кода, показывающий реализацию логики данного потока, расположен на рисунке 2.

```
Statement stat = Launcher.getConnection().createStatement();
ResultSet result = stat.executeQuery("SELECT Name, IP, Time FROM CMMask WHERE
    Status=1");
if(result.next()) {
    long time = System.currentTimeMillis() - (long)result.getLong("Time");
    if(time > HALF_MINUTE){
        Logger.addToStats(result.getString("Name"), result.getString("IP"), "NOT
            RESPONDING");
        stat.executeUpdate("UPDATE CMMask SET Status=0, Time="
            System.currentTimeMillis() + " WHERE Name="
            + result.getString("Name") + "");
    }
}
sleep(TICK);
```

Рисунок 2. Проверка статуса работы компьютера с клиентским приложением

Третий поток занимается передачей данных из базы данных в панель администратора по его запросу. Пакет для передачи формируется из имени компьютера и его статуса в данный момент.

Четвертый поток осуществляет передачу команд из панели администратора к клиентским приложениям. Передача также происходит с помощью TCP соединения. Поток в зависимости от полученной команды обращается к клиенту по определенному порту. В случае с завершением работы компьютера с клиентским приложением данный поток устанавливает соединение по определенному порту. Для исполнения запроса на получение списка запущенных процессов, поток устанавливает соединение с

клиентским приложением и ждет получения данных. После получения он пересылает их панели администратора. Такая же логика работы и с получением журналов последних действий. На рисунке 3 изображена часть метода, показывающая логику работы третьего потока.

```
byte buf[] = new byte[1024];
String removeIP = socket.getInetAddress().toString();
int r = socket.getInputStream().read(buf);

String parseInputString[] = new String(buf, 0, r).split(":");

if(parseInputString[0].equals("listing")){
    String data = requesting(parseInputString[1]);
    socket.getOutputStream().write(data.getBytes());
    Logger.addToLog(parseInputString[1], "ListOfProcesses", removeIP);
}
else if(parseInputString[0].equals("shutdown")){
    shutdown(parseInputString[1]);
    Logger.addToLog(parseInputString[1], "Shutdown", removeIP);
}
else if(parseInputString[0].equals("delete")){
    deleteRecord(parseInputString[1]);
    Logger.addToLog(parseInputString[1], "DeleteRecord", removeIP);
}
else if(parseInputString[0].equals("Statistic")){
    getStats(socket);
}
else if(parseInputString[0].equals("Logs")){
    getLogs(socket);
}
```

Рисунок 3. Реализация в сервере приложений перенаправления запросов от панели администратора к клиентским приложениям

4.3 Приложение панель администратора

Панель администратора является единственным приложением в данной системе дистанционного управления компьютерными классами, обладающим графическим интерфейсом. Интерфейс состоит из 3 частей. Первая часть это меню навигации, которое располагается в верхней части окна. В нем находятся кнопка выхода, а также кнопки просмотра журналов. Вторая часть интерфейса расположена в крайней правой части окна приложения и отображает текущие вкладки, которые являются именами компьютерных классов. При нажатии на вкладку в центральной части приложения начинают отображаться компьютеры в выбранном компьютерном классе в третьей части интерфейса. На рисунке 4 изображено главное окно интерфейса панели администратора.

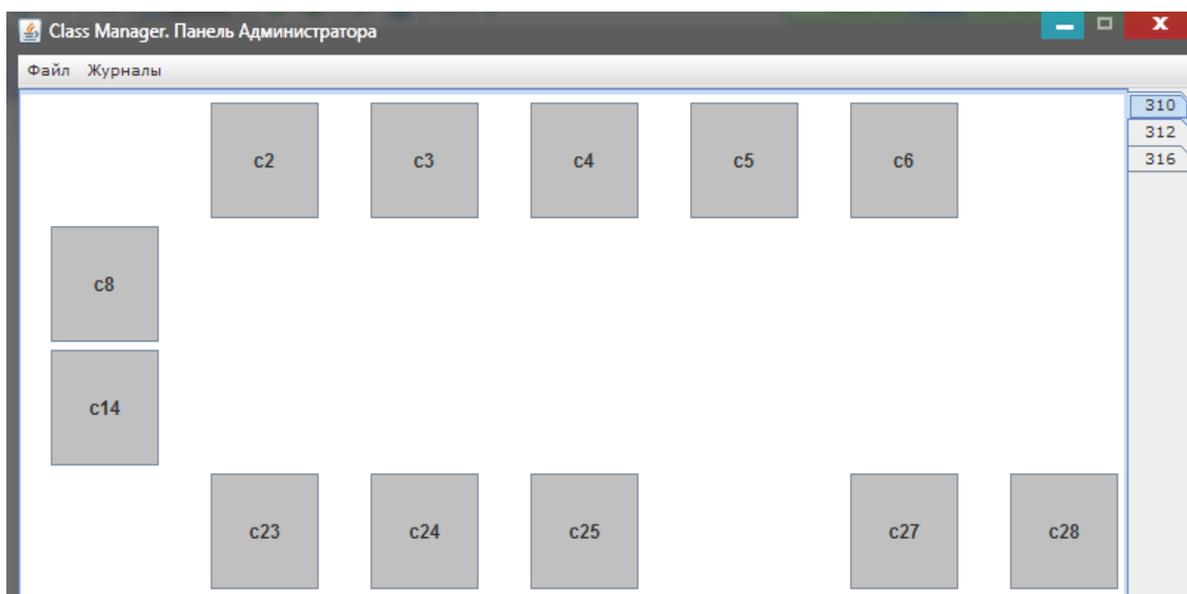


Рисунок 4. Пример интерфейса главного окна панели администратора

Компьютеры выглядят как кнопки, при нажатии на которых отображается меню, состоящее из пунктов: выключить выбранный компьютер и получить его список процессов. При выборе элемента в данном меню сначала появляется окно с уведомлением, в котором запрашивается подтверждение выбранного действия. При выборе элемента в данном меню сначала появляется окно с уведомлением, в котором запрашивается подтверждение выбранного действия. В этом окне уведомления можно согласиться и подтвердить операцию, или отказаться, нажав на соответствующую кнопку. При подтверждении выполняется выбранная операция следующим способом: передается собранный специально для данной команды пакет серверу приложений и, если было выбрано меню «Получение списка процессов», то панель администратора переходит в прослушивающий режим, в котором приложение ждет данные от сервера приложений, а именно список процессов. На рисунке 5 изображено окно для случая с выключением компьютера.

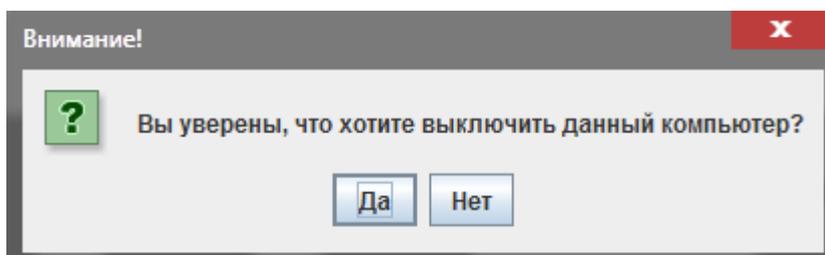


Рисунок 5. Окно, появляющееся при дистанционном выключении компьютера в панели администратора.

Вся основная логика удаленного выключения компьютера с клиентским приложением и получение его списка процессов уже была описана в пунктах 4.1 и 4.2.

При получении списка процессов панелью администратора, он преобразует ее в табличную форму и отображает окно с данной таблицей. Также нельзя выключить уже выключенный компьютер, как и нельзя получить его список процессов.

В верхней части приложения в меню находится закладка, называемая «Журналы». При нажатии на нее выпадает список из двух кнопок: «Системные действия» и «Действия администраторов». На рисунке 6 показан список журналов.

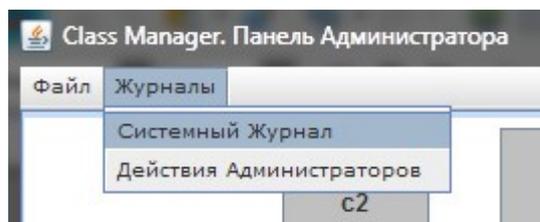


Рисунок 6. Демонстрация списка журналов

После выбора журнала появляется сам журнал. Как и окно со списком процессов, окно с журналом состоит из таблицы с несколькими столбцами. Для случая с системным журналом столбцы имеют следующие названия: Имя Компьютера, IP, Событие и Время. В имени компьютера находится полное имя компьютера клиентского приложения, состоящее из названия класса и имени самого компьютера. В случае с журналом последних действий администраторов заголовки столбцов следующие: Имя Компьютера, Команда, IP Администратора и Время. Ознакомиться с внешним видом журнала можно на рисунке 7.

Имя Компьютера	IP	Событие	Время
SERVER	192.168.100.6	TURN ON	03 июн 2013 01.32
310-Exception	192.168.100.4	NOT RESPONDING	03 июн 2013 01.32
310-Exception	192.168.100.4	UPDATE	03 июн 2013 01.28
310-Exception	192.168.100.4	UPDATE	03 июн 2013 01.28
310-Exception	192.168.100.4	UPDATE	03 июн 2013 01.28

Рисунок 7. Демонстрация журнала последних системных событий

Панель администратора использует конфигурационный файл, в котором находятся общие настройки отображения компьютеров в приложении. Таким образом администратор должен самостоятельно задать компьютерные классы и имена компьютеров, а также их расположение. После запуска панели администратора приложение автоматически считывает конфигурации в данном файле и при дальнейшем отображении компьютеров, они будут находиться в том порядке, который задал

администратор. Если какого-либо компьютера нет в данном файле, то отображаться на экране он не будет.

После успешного считывания данных с конфигурационного файла происходит обращение к базе данных через сервер приложений. После получения списка компьютеров, указанных в базе данных, информация о них обновляется на экране панели администратора. Ознакомиться с примером конфигурации можно на рисунке 8. На рисунке 4 предоставлен результат отображения компьютеров в компьютерном классе номер 310.

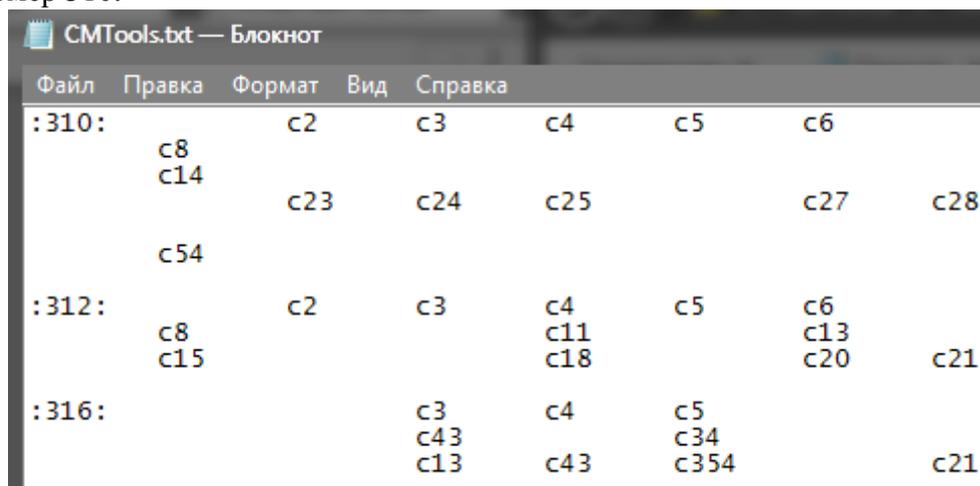


Рисунок 8. Пример расположения компьютеров в компьютерных классах.

Стоит упомянуть структуру данного файла. Компьютерный класс должен находиться в начале строки и выделяться с обеих сторон двоеточиями. После последнего двоеточия необходимо поставить табуляцию. Далее через табуляцию необходимо указать точные имена компьютеров, находящихся в данном компьютерном классе. Число компьютеров в одной строке не должно превышать число 7. В противном случае отображаться на экране они будут последовательно.

Также можно оставлять свободное место. На данном месте будет пустое место. Эта функция необходима для более удобного представления компьютеров. Например, если компьютеры в классе находятся по сторонам комнаты, образуя тем самым полукруг, администратор может задать расположение компьютеров соответствующим образом.

Глава 5. Обзор базы данных

В данной системе дистанционного управления компьютерными классами база данных необходима для хранения данных о компьютерах, а именно их имена и адреса, а также для хранения всех журналов. Так как количество сведений о компьютерах, хранимых в базе невелико, то нам не требуется оперировать большими массивами данных, и следовательно, можно использовать быструю и легкую систему управления базами данных. Рассмотрим некоторые из них.

Oracle является одной из лучших СУБД, что подкреплено десятками лет успешной деятельности компании и многочисленным числом соответствующих наград. Высокая скорость работы с огромными массивами данных. Обладает широким спектром возможностей по управлению базами данных. Из минусов стоит выделить платную лицензию и длительное обучение работы с системой.

MySQL одна из самых быстрых при оперировании небольшими массивами данных систем управления базами данных, обладающая большим количеством различных приложений, упрощающих работу с СУБД, а также простотой в освоении и частыми обновлениями. Максимальная скорость работы достигается на небольших массивах данных, до 100000 записей. Обладает кешем запросов, который увеличивает скорость обработки запросов при неоднократном повторении запросов на чтение.

PostgreSQL направлена на предоставление широкого функционала пользователям. Но недостаток данного подхода заключается в плохой реализации данного функционала, в следствии сего обладает в разы меньшей аудиторией по сравнению с СУБД Oracle и MySQL. Также существуют проблемы с работоспособностью на операционных системах Windows. Из преимуществ стоит отметить, что PostgreSQL может моментально сжимать и разжимать данные.

MongoDB является документо-ориентированной СУБД не требующая описания схемы таблиц. С одной стороны, она позволяет делать очень быстрые операции над объектом, зная его идентификатор, а с другой, предоставляет мощный инструмент для сложных взаимодействий. Из недостатков выделяется невозможность сделать join и накладные расходы на хранение названий свойств.

Теперь рассмотрим применение в нашей системе. У нас нет больших массивов данных, поэтому для нашей системы дистанционного управления компьютерными классами достаточно использовать быструю и легкую систему управления базами данных, такую как MySQL или MongoDB. Обе СУБД удовлетворяют нашим требованиям, но язык запросов Mongo уступает SQL по гибкости и возможностям, поэтому окончательный выбор остановится на MySQL.

Рассмотрим внутреннее строение базы данных. В базе данных будет находиться таблица CMMask, содержащая в себе необходимые данные, идентифицирующие компьютер в компьютерном классе, а также требуемые поля для определения его текущего состояния. Полный список полей таблицы следующий: Name, IP, Status и Time (соответственно имя компьютера, IP-адрес, статус, отображающий включен или выключен компьютер, и время получения последнего пакета). Данный набор позволит идентифицировать компьютер среди других и определить его IP-адрес для непосредственного управления им.

Добавлением данных в таблицу CMMask, занимается сервер приложений. Он получает имена компьютеров и IP-адреса Клиентов, к которым затем прибавляет статус и время получения пакета с данными. После этого он проверяет наличие записи о данном компьютере по его имени. Если данного компьютера в таблице нет, то он создает новую запись, а если было обнаружено совпадение имени компьютера, то просто обновляет запись, используя текущие IP-адрес, статус и время получения пакета.

Сбором данных из таблицы занимается также сервер приложений, который получает все записи и интерпретирует их следующим образом: разделяет имя на 2 части — название компьютерного класса и имя компьютера.

В базе данных также располагаются два журнала: журнал действий администраторов и журнал всей истории операций в системе, не зависящих от администраторов. Они называются соответственно CMLog и CMStatistic. Поля у них такие же как и в таблицах в панели администратора (Смотрите раздел 4.3).

Стоит также отметить, что всеми операциями с базой данных занимается только сервер приложений. Другие приложения не имеют доступа к базе данных в соответствии с трехуровневой клиент-серверной архитектурой.

ЗАКЛЮЧЕНИЕ

Цель настоящей работы заключается в проектировании и реализации системы, позволяющей дистанционно наблюдать за компьютерными классами, управлять компьютерами, находящимися в наблюдаемых компьютерных классах, а также получать сведения о работе компьютеров. Для достижения указанной цели перед работой был поставлен ряд задач.

При решении задачи определения требований к системе дистанционного управления компьютерными классами в работе проведен анализ характеристик компьютеров, расположенных в компьютерных классах, и актуальных проблем их администрирования, в результате которого было выявлено, что основными требованиями являются обеспечение кроссплатформенности системы и поддержка управления системой несколькими администраторами.

При решении задачи проведения обзора существующих систем и технологий управления компьютерными классами были выявлены преимущества и недостатки готовых решений, а также было установлено, что реализованные технологии в исследуемых решениях не подходят под установленные требования.

Решая задачи, направленные на разработку архитектуры и проектирование системы дистанционного управления компьютерными классами было принято решение использовать существующую трехуровневую архитектуру клиент-сервер. Спроектированная система выполняет все выявленные общие требования.

В результате выполнения всех вышеописанных задач решается главная задача, направленная на реализацию системы дистанционного управления компьютерными классами, и которая выполняет основную цель работы, все поставленные требования, и которая обладает функционалом, необходимым для администрирования компьютерных классов.

Таким образом, задачи решены в полном объеме, цель работы достигнута.

Подводя итог всей проделанной работе отметим некоторые моменты, которые являются особенностями разработанной системы и на которые пришлось уделить дополнительное внимание.

Во-первых, это архитектура системы, состоящая из трех приложений и базы данных. Таким образом, она является клиент-серверной архитектурой, то есть имеется выделенный сервер, через который происходит управление всей системой, и который является единственным объектом в системе, имеющим доступ к базе данных. Также использование трех приложений добавляет функцию, которая выполняет основное

требование: увеличивает количество администраторов, одновременно управляющих системой.

Во-вторых, особенности реализации получения списка процессов и завершение работы компьютера клиентского приложения. Сначала было предложено реализовать данный функционал с помощью готовых библиотек, которые для разных операционных систем собирают разные данные. Но в этом случае мы столкнулись с двумя проблемами: требованием использования разных библиотек для различных операционных систем и сбором общих для всех библиотек данных.

В результате, использование библиотек было заменено исполнением обычных команд операционных систем, это «shutdown» и «tasklist». Данные команды и подобные им, существуют почти на всех современных операционных системах, что делает систему более кроссплатформенной и легкой в модернизации.

В-третьих, это ведение журналов всех событий, произошедших в системе, что позволяет полностью наблюдать за системой и получать последние действия администраторов. Также можно открыть журнал последних событий произошедших в самой системе. Например, запуск сервера приложений или получение пакета от клиентского приложения.

Данная система дистанционного управления компьютерными классами успешно прошла ряд тестирований. Все тестирования были проведены на операционных системах Windows XP и Windows 7 и закончились подтверждением функционала системы.

На будущее можно запланировать повышение безопасности системы путем шифрования обмена данных внутри системы. Также можно посоветовать вести более полную статистику работы всех наблюдаемых компьютеров, чтобы узнать, какие из компьютеров загружены работой больше остальных, а какие включают всего несколько раз в месяц.

Литература

1. Features overview: [Электронный ресурс] // Netop. 2013. URL: <http://www.netop.com/classroom-management-software/products/netop-vision/features-overview.htm> (дата обращения: 16.01.2013)
2. Возможности Radmin 3: [Электронный ресурс] // «Фалматек Трейд». 1999-2013. URL: <http://www.radmin.ru/products/radmin/features.php> (дата обращения: 18.01.2013)
3. Технология удаленного подключения с помощью Ammy Admin: [Электронный ресурс] // «Амми». 2013. URL: http://www.ammyu.com/ru/admin_technologies.html (дата обращения: 18.01.2013)
4. Программное обеспечение SMART Sync для управления учебным процессом: [Электронный ресурс] // SMART Technologies. 2013. URL: <http://ru.smarttech.com/solutions/products/software/smart+syn+classroom+management+software?cat=&subCat=&count=31&searchType=Products> (дата обращения: 20.01.2013)
5. Сравнение PostgreSQL и Oracle: [Электронный ресурс] // «Аксифт». М., 2009-2013. URL: <http://oracle.axoft.ru/catalog/PSQLvsOracle.php> (дата обращения: 27.02.2013)
6. Справочное руководство по MySQL: [Электронный ресурс] // MySQL AB. 2010. URL: <http://www.mysql.ru/docs/man/Comparisons.html> (дата обращения: 27.02.2013)
7. The MongoDB 2.4 Manual: [Электронный ресурс] // 10gen. 2011-2013. URL: <http://docs.mongodb.org/manual/> (дата обращения: 6.03.2013)