

министерство образования и науки российской федерации  
**федеральное государственное бюджетное образовательное  
учреждение высшего профессионального образования**  
**«Московский государственный индустриальный университет»**  
**(ФГБОУ ВПО «МГИУ»)**  
кафедра информационных систем и технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Проектирование и разработка корпоративных  
информационных систем»

на тему «Система управления и контроля сетевых устройств на основе простого  
протокола управления сетями»

Группа	111131
Студент	Д.Г. Грачев
Руководитель работы	Ю.В. Курасов

## **Аннотация**

Работа посвящена созданию приложения для мониторинга устройств в локальной вычислительной сети ФГБОУ ВПО «МГИУ» по протоколу SNMP v3 на языке программирования C++. А так же WEB сервера для удобного использования и отображения данных, панели управления с использованием фреймворка Ruby on Rails.

Требуется реализовать следующие подзадачи:

- Программа должна работать в фоновом режиме.
- Опрос устройств в асинхронном режиме.
- Работу с базой данных SQLite3.
- К приложению можно подключиться через TCP/IP
- Необходима авторизация как через приложение, так и на WEB сервере используя данные о пользователях из одной базы данных.

WEB сервер должен иметь возможность предоставления отчета как в виде графика, так и в виде списка, создания/редактирования/удаления пользователей, вспомогательных словарей, устройств из базы данных.

## Содержание

- Введение ..... 4
- Литературный обзор ..... 5
- Проектирование. .... 8
- Список используемой литературы ..... 13

## ***Введение***

В сложившемся современном мире усложняются информационные системы. Вместе с ними возрастают требования к вычислительным сетям, благодаря которым происходит передача и обработка данных. В нынешнее время для успешной работы компаний, организаций необходима стабильная информационная система. Так как от нее может зависеть огромное количество факторов от которых зависит функциональность организации. К примеру это может быть внутренний почтовый сервис или система документооборота. Ежедневно по вычислительной информационной системе может передавать огромное количество информации. Потому возникает необходимость в административной поддержки.

Для успешного администрирования вычислительной системы, далее ЛС, возрастает необходимость в инструментах позволяющих эффективно отслеживать состояние ЛС. Так же для успешного администрирования сети необходимо знать состояние каждого ее элемента с возможностью изменять параметры его функционирования. Обычно сеть состоит из устройств различных производителей и управлять ею было бы нелегкой задачей если бы каждое из сетевых устройств понимало только свою систему команд. Поэтому возникла необходимость в создании единого языка управления сетевыми ресурсами, который бы понимали все устройства, и который, в силу этого, использовался бы всеми пакетами управления сетью для взаимодействия с конкретными устройствами.

Подобным языком стал SNMP - Simple Network Management Protocol. Разработанный для систем, ориентированных под операционную систему UNIX, он стал фактически общепринятым стандартом сетевых систем управления и поддерживается подавляющим большинством производителей сетевого оборудования в своих продуктах. В силу своего названия - Простой Протокол Сетевого Управления - основной задачей при его разработке было добиться максимальной простоты его реализации. В результате возник протокол, включающий минимальный набор команд, однако позволяющий выполнять практически весь спектр задач управления сетевыми устройствами - от получения информации о местонахождении конкретного устройства, до возможности производить его тестирование.

Но так как SNMP требует определенных навыков в использовании UNIX систем, возникает необходимость в квалифицированных работниках. Часто важно иметь сотрудника который способен быстро устранить неисправность в ЛС, отследить состояние, а так же перенастроить оборудование. Для обучения молодого персонала требуется время и средства. Соответственно возникает потребность в сокращении времени на обучение сотрудника. Либо средства позволяющие на интуитивном уровне администрировать ЛС.

Цель работы – разработка эффективной, интуитивно-понятной системы для мониторинга локальной вычислительной системы в рамках ИВЦ ФГБОУ ВПО МГИУ.

## Литературный обзор

### Постановка задачи

Необходимо реализовать систему мониторинга, далее СМ, устройств входящих в локальную вычислительную сеть ФГБОУ ВПО МГИУ, по протоколу SNMP v3. Она должна поддерживать шифрование передаваемых данных по сети. Делать асинхронный опрос устройств на определенные параметры задаваемые в MIB файлах и записывать в базу данных ответы от устройств. Данная программа должна работать в фоновом режиме, т.е. быть демоном. Так же необходимо реализовать WEB интерфейс предоставляющий возможность просмотра статистики, добавлении устройств и настройки их параметров. Более подробное описание приведено ниже.

### Обзор существующих решений

Существует несколько решений, для мониторинга устройств в ЛВС по протоколу SNMP.

Один из самых известных и наиболее мощных является **Zabbix**.

**Zabbix**—свободная система мониторинга и отслеживания статусов разнообразных сервисов компьютерной сети, сетевого оборудования .

*Структура:*

- **Zabbix сервер** — это ядро программного обеспечения Zabbix. Сервер может удаленно проверять сетевые сервисы, является хранилищем, в котором хранятся все конфигурационные, статистические и оперативные данные, и он является тем субъектом в программном обеспечении Zabbix, который оповестит администраторов в случае возникновения проблем с любым контролируемым оборудованием.
- **Zabbix прокси** — собирает данные о производительности и доступности от имени Zabbix сервера. Все собранные данные заносятся в буфер на локальном уровне и передаются Zabbix серверу, к которому принадлежит прокси-сервер. Zabbix прокси является идеальным решением для централизованного удаленного мониторинга мест, филиалов, сетей, не имеющих локальных администраторов. Он может быть также использован для распределения нагрузки одного Zabbix сервера. В этом случае, прокси только собирает данные, тем самым на сервер ложится меньшая нагрузка на ЦПУ и на ввод-вывод диска.
- **Zabbix агент** — контроль локальных ресурсов и приложений (таких как жесткие диски, память, статистика процессора и т. д.) на сетевых системах, эти системы должны работать с запущенным Zabbix агентом. Zabbix агенты являются чрезвычайно эффективными из-за использования родных системных вызовов для сбора информации о статистике.
- **Веб-интерфейс** — интерфейс является частью Zabbix сервера, и, как правило (но не обязательно), запущен на том же физическом сервере, что и Zabbix сервер. Работает на PHP, требует веб сервер (напр. Apache). Неочевидной особенностью веб-интерфейса Zabbix является тот факт, что он **не** является фронтэндом в традиционном понимании этого слова: все операции чтения/записи веб-интерфейс осуществляет напрямую с

базой данных, минуя собственно сервер zabbix. Таким образом, если не учитывать гипотетическую возможность записи пользователем в СУБД напрямую (что сильно осложняется отсутствием гарантий совместимости структуры базы данных от версии к версии), то во-первых сервер zabbix без веб-интерфейса оказывается просто нефункционален, а во-вторых - сторонние разработчики на практике не могут написать "альтернативный" веб-интерфейс, поскольку тот должен будет привязываться к базе данных, спецификация которой может меняться без уведомления со стороны разработчиков Zabbix совершенно произвольным образом.

#### *Обзор возможностей:*

- Распределенный мониторинг вплоть до 1000 узлов. Конфигурация младших узлов полностью контролируется старшими узлами, находящимися на более высоком уровне иерархии.
- Сценарии на основе мониторинга
- Автоматическое обнаружение
- Централизованный мониторинг лог-файлов
- Веб-интерфейс для администрирования и настройки
- Отчетность и тенденции
- SLA мониторинг
- Поддержка высокопроизводительных агентов (zabbix-agent) практически для всех платформ
- Комплексная реакция на события
- Поддержка SNMP v1, 2, 3
- Поддержка SNMP ловушек
- Поддержка IPMI
- Поддержка мониторинга JMX приложений из коробки
- Поддержка выполнения запросов в различные базы данных без необходимости использования скриптовой обвязки
- Расширение за счет выполнения внешних скриптов
- Гибкая система шаблонов и групп
- Возможность создавать карты сетей

и многое другое входит в возможности данного продукта. Данный продукт разрабатывается и поддерживается с 1998 года. Он включает мощный инструментарий для мониторинга устройств в локально вычислительной сети. Но его мощность и есть его недостаток — он достаточно сложен в освоении и требует определенного опыта как в программировании так и в понимании взаимодействия устройств. Так же можно к минусам отнести и то что приложение от сторонних разработчиков. SNMP v1 и v2 не поддерживают шифрования. Да и многое из этого приложения просто не является востребованным для «ИВЦ МГИУ». Конечно у Zabbix есть поддержка, но она платная. И чем больше необходимо инструментов, расширенного функционала, тем дороже выйдет.

Есть еще одно решение, такое как **PowerSNMP Free Manager**.

**PowerSNMP Free Manager** - это бесплатный, полнофункциональный SNMP-менеджер, построенный с использованием PowerSNMP для .NET. Для работы с узлами сети, с возможностью просматривать MIB деревья и анализировать сетевые запросы. Идеально подходит для легких умеренных задач управления.

**MIB** - виртуальная база данных, используемая для управления объектами в сети связи. Наиболее часто это понятие связывают с Simple Network Management Protocol (SNMP).

*Особенности:*

- Простой, легкий в использовании, доступный интерфейс
- Работа по сети.
- Определяет SNMP ловушки
- Поддерживает SNMP версии 1, 2 и 3
- Создает автоматические уведомления по электронной почте при переменных выпадающих из диапазона
- Построен с использованием надежной системой PowerSNMP для компонентов .NET

Так как в «ИБЦ МГИУ» используют только Unix подобные системы, а платформа .NET для семейства MS-DOS уже не подходит для использования. Так же как и в Zabbix для расширения функционала, поддержки, необходима определенная плата.

Другие продукты либо малофункциональны, либо платные, либо устарели. В связи с этим необходимо реализовать свой собственный программный продукт удовлетворяющий требованиям мониторинга сети и имеющий интуитивно-понятный интерфейс, но имеющий достаточный функционал для анализа полученных данных.

**Объекты системы и принципы их взаимодействия**

- Приложение для опроса устройств.
- База данных SQLite3.
- Web сервер
- Браузер на стороне клиента.

При запуске программа будет переходит в фоновый режим, в котором создаст два дополнительных потока в одном из которых поднимается асинхронный SNMP manager, а в другом сервер для обработки запросов от пользователя. При инициализации менеджера данные об устройстве должны считываются из базы данных. Набор запросов к каждому из устройств будет считан из MIB файлов. Администратор системы будет иметь возможность активировать/деактивировать любую из команд.

Данные будут передаваться в зашифрованном виде благодаря поддержки SNMP v3.

Полученные данные будут проверены на стороне приложения. После проверки валидации данные записываются в базу данных. Приложение предоставит пользовательский интерфейс после авторизации, который будет иметь возможность настройки приложения, просмотра отчетов, сведений об устройствах и т.п.

## Проектирование

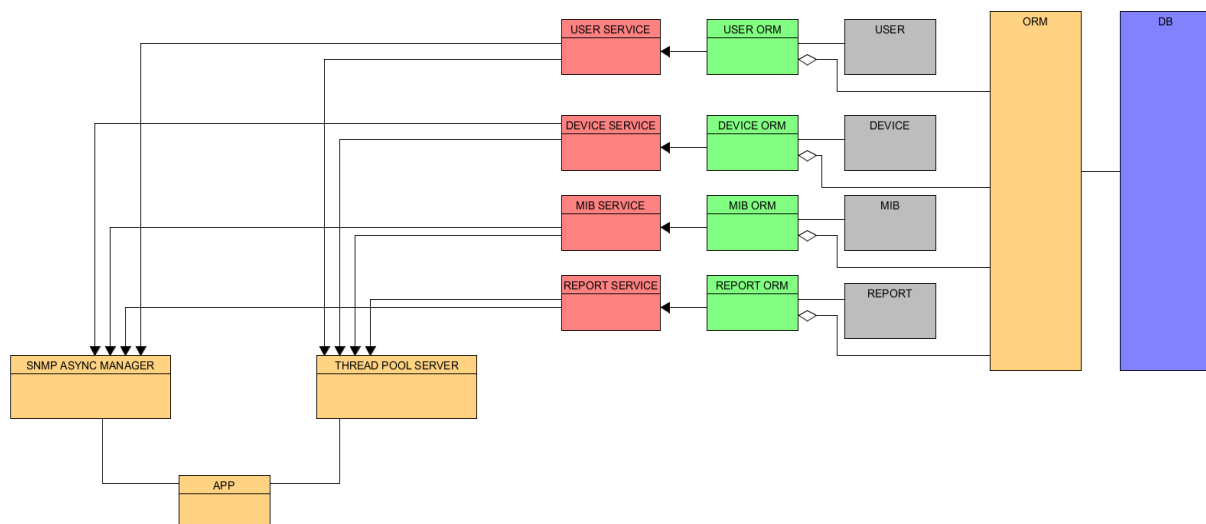
Первоначально необходимо было разобраться с тем как будет выглядеть конечный продукт. Работа состоит из двух частей: реализации приложения для мониторинга устройств и WEB сервера, который должен уметь только работать с данными из базы данных. Логично то что начинать разработку надо с первого.

Известно что приложение будет реализовано на языке программирования C++ и оно должно уметь работать с базой данных SQLite3. Соответственно применяем DAO шаблон проектирования и ORM технологию.

**ORM** — технология программирования, которая связывает базы данных с концепциями объектно-ориентированного программирования, создавая «виртуальную объектную базу данных».

**DAO** — это объект, который предоставляет абстрактный интерфейс к какому-либо типу базы данных или механизму хранения

Осталось только изобразить конечный вид структуры приложения.  
Для этого использовался язык UML:



Данная диаграмма дает общее представление о связях между объектами внутри приложения. Особенность в свою очередь скрыты и остаются на совести программиста. Серым цветом выделены модели объектов, зеленым объекты позволяющие полученные данные из базы данных и превращать их в объекты



определенного типа. Это и есть ORM технология. Розовым отмечены обертки для более удобной работы с объектами. Именно их и будет использовать приложение.

Представим как пользователь будет взаимодействовать с конечным продуктом. Приложение будет постоянно «общаться» с устройствами внутри локальной вычислительной сети. Вести отчеты. У клиента есть два способа взаимодействовать с базой данных.

Первый это через приложение. Администратор будет подключаться к приложению по TCP/IP проходить авторизацию и получать доступ к интерфейсу.

Второй, самый простой, с помощью веб интерфейса. Который будет предоставлять удобный доступ к необходимым данным и дружелюбным для пользователя способом.

## **Результат работы полученного ПО**

### **Приложение**

Так как приложение переходит в фоновый режим и все данные записывает в базу данных, то нет возможности визуально отобразить процесс работы, только если как в диспетчере задач. Конечно же процесс работы приложения можно видеть по результатам — записям в базу данных. Данные из которой можно отобразить и через браузер, используя WEB сервер.

### **WEB сервер**

Данное приложение реализовано с помощью фреймворка Ruby on Rails. Его архитектура основана на шаблоне проектирования MVC. Коротко описывая проделанную работу, надо сказать о том что были сделаны миграциями, для взаимодействия с существующей базой данных. Валидации на уровне моделей. Сессий, использовались определенные дополнительные пакеты и плагины. Снимки в момент работы.

## Авторизация

Login

Denis

Password

.....

Войти

Авторизация пользователя при входе в систему.

Просмотр статистики

Устройства

Мибы

Словарь команд

Пользователи

Выход

## Просмотр статистики

Название устройства

first

Тип запроса

Всего использованно RAM

Период отчетности

с

1

11

2014

по

1

12

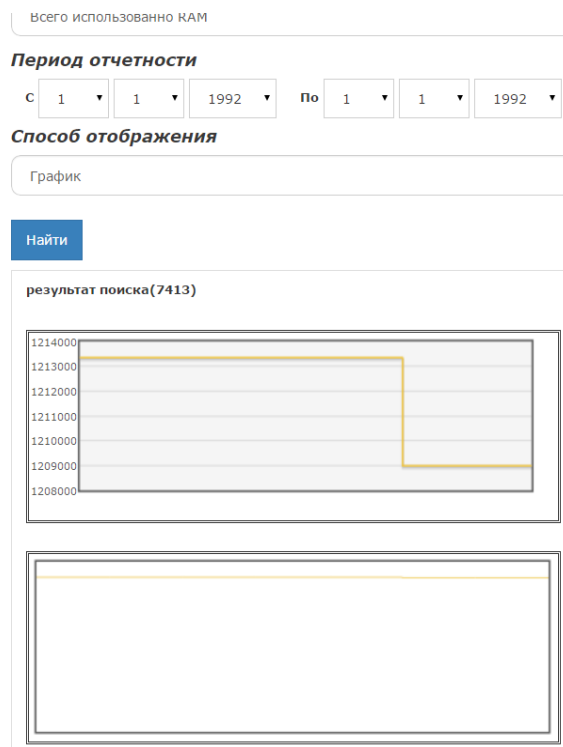
2014

Способ отображения

График

Найти

Фильтр для отображения отчетов.



При работе приложения можно видеть результат работы.

Просмотр статистики	Устройства	Мибы	Словарь команд	Пользователи	Выход
---------------------	------------	------	----------------	--------------	-------

## Устройства

first			
second			

New device

localhost:3000/devices/1

Список доступных устройств.

## New device

**Name**

**Description**

**Room**

**Mac**

**Serial number**

**Model**

**Peername**

**Port**

**Login**

**Password**

**Mib**

[Back](#)

Добавление нового устройства.

## Список Oids

Наименование: <b>Всего использованно RAM Oid: .1.3.6.1.4.1.2021.4.6.0</b>	<input type="button" value="🔍"/> <input type="button" value="🗑️"/>
Наименование: <b>Свободно RAM Oid: 1.3.6.1.4.1.2021.4.11.0</b>	<input type="button" value="🔍"/> <input type="button" value="🗑️"/>

New Old

Словарь команд для перевода из машинного представления в человеко-читаемый вид.

## Список используемой литературы

1. [https://ru.wikipedia.org/wiki/Data\\_Access\\_Object](https://ru.wikipedia.org/wiki/Data_Access_Object)
2. <https://ru.wikipedia.org/wiki/ORM>
3. <https://ru.wikipedia.org/wiki/Zabbix>
4. <http://www.zabbix.com/ru/> - ресурс посвященный Zabbix
5. <http://www.net-snmp.org/> - ресурс посвященный библиотеке для работы с SNMP.
6. <http://www.dart.com/snmp-free-manager.aspx> — ресурс посвященный SNMP FREE MANAGER.
7. <https://www.google.ru/> - поисковая система google.