

Государственное бюджетное общеобразовательное учреждение города  
Москвы «Школа № 1532»

**«Terminal Control» - система однозначного управления  
группой компьютеров в локальной сети на базе  
операционной системы Linux**

11 класс, ГБОУ школа № 1532,  
Воронов Никита Рустамович  
Руководитель:  
учитель информатики, ГБОУ школа №1532,  
Пичугин Андрей Владимирович

Москва, 2024

## Оглавление

<b>1. Введение.....</b>	<b>3</b>
1.1 О чем проект.....	3
1.2 Формирование целей проекта.....	3
1.3 Формирование задач проекта.....	4
<b>3. Методы реализации.....</b>	<b>5</b>
3.1 Логика кода.....	5
<b>4. Функциональные особенности.....</b>	<b>8</b>
4.1 Протокол удаленного доступа.....	8
4.2 Сетевое включение.....	8
4.3 Терминал.....	9
<b>5. Техническая реализация.....</b>	<b>10</b>
5.1 Мобильное приложение.....	10
5.2 Компьютерное приложение.....	10
5.3 Терминал на базе Raspberry Pi.....	11
<b>6. Результаты и выводы.....</b>	<b>13</b>
<b>7. Список используемой литературы.....</b>	<b>16</b>

# **1. Введение**

## **1.1 О чем проект**

В наше время, когда цифровая трансформация пронизывает все сферы жизни и технологии становятся неотъемлемой частью повседневности, проект "Терминальный Контроллер" приобретает важное значение в современном мире. Необходимость эффективного администрирования и управления IT-инфраструктурой становится более актуальной в условиях увеличения объема данных, сложности сетевых структур и постоянных вызовов в области кибербезопасности.

Проект предлагает современным организациям мощный инструмент для облегчения административных задач, обеспечивая возможность эффективного взаимодействия с компьютерами на базе операционных систем Linux. Соединение должно происходить в локальной сети, так что эта технология достаточно хорошо проявит себя в больших серверных центрах и относительно больших классах информатики в школах. В условиях динамичного окружения и растущей зависимости от технологий, терминальный контроллер становится важным элементом в построении гибкой, отзывчивой и устойчивой IT-инфраструктуры, что является ключевым фактором для успешного функционирования современных организаций. Его роль в обеспечении безопасности, масштабируемости и эффективности в управлении компьютерами в локальной сети делает проект незаменимым в контексте современных вызовов и требований бизнес-среды.

## **1.2 Формирование целей проекта**

*Целью* данного проекта была выбрана задача сформировать комплексное решение, нацеленное на эффективное удаленное управление информационными системами на базе компьютеров, использующих операционную систему семейства Linux. Дополнительно, настольное приложение должно совершенствоваться другими разработчиками, поэтому важно предоставлять открытый исходный код к проекту.

Ключевым этапом в реализации проекта стало также создание микро инструмента, предназначенного для администрирования отдельных зон на компьютерах. А также добавление сенсорного экрана для уменьшения степени сложности использования устройства. Процесс включал моделирование и распечатку корпуса для этого устройства, придавая проекту дополнительный аспект инноваций и физической реализации.

### **1.3 Формирование задач проекта**

Проект по распределению задач имеет критическое значение для оптимизации работы систем и повышения их эффективности в условиях современных информационных технологий. Основные задачи проекта включают:

1. Управление Ресурсами: Разработка механизмов эффективного распределения вычислительных ресурсов.
2. Отзывчивость Системы: Обеспечение быстрого отклика системы на изменения нагрузки и рабочих условий.
3. Безопасность: Имплементация мер безопасности для защиты данных и обеспечения целостности системы.
4. Разработка специального устройства контроля в локальной зоне: Функциональность такого устройства может включать в себя множество одновременных действий.
5. Разработка мобильного приложения: Разработка пользовательского мобильного интерфейса, удовлетворяющего потребности конечных

пользователей. Приложение должно упростить использование терминальных связей для обычного пользователя.

6. Разработка компьютерного приложения: Разработка пользовательского настольного интерфейса, который бы предоставлял огромные возможности по удаленному управлению и администрированию компьютеров как в локальной, так и в глобальной сети.
7. Тестирование на ОС МОС: Гарантирование функциональности и совместимости проекта с операционной системой «МОС 12» в школах.

### **3. Методы реализации**

#### **3.1 Логика кода**

При выборе языка программирования главным критерием была выбрана кроссплатформенность. Для этого лучше всего подходят интерпретируемые языки программирования, которые не привязаны к определенной платформе или к определенным компиляторам. Самым популярным из таких языков программирования является Python.

Python привлекателен для построения управляющей системы в локальной сети из-за нескольких ключевых преимуществ. Во-первых, его простота и читаемость кода облегчают разработку и поддержку. Python также обладает обширной экосистемой библиотек, что делает интеграцию различных функциональных возможностей, таких как работа с сетью и многозадачность, более удобной. Его кроссплатформенность обеспечивает возможность запуска приложения на различных операционных системах без изменений. Гибкость и расширяемость Python позволяют использовать

различные стили программирования и интегрировать библиотеки, написанные на других языках.

Кроме того, активное сообщество разработчиков Python предоставляет доступ к обширным ресурсам и решениям для работы с локальной сетью. По сравнению с C# Python обладает простым и лаконичным синтаксисом, а также предоставляет богатый выбор библиотек, что делает его удобным для разработки сетевых приложений. Однако, в некоторых случаях Python может проявлять меньшую производительность по сравнению с C#. В сравнении с C++, Python предоставляет легкость в разработке сетевых приложений, но может уступать в производительности, особенно в случаях высоких требований к скорости передачи данных или обработке больших объемов информации в локальной сети.

В целом, Python представляет собой привлекательный выбор для управления системами в локальной сети, обеспечивая простоту, богатую библиотеку и активное сообщество, но требования проекта могут также подразумевать использование C# или C++, особенно в случае высоких требований к производительности.

Библиотеки, используемые в проекте предоставляют большие возможности по организации локальной и глобальной сети в контексте администрирования и контроля. Если рассматривать каждую библиотеку по отдельности, то можно получить следующую картину:

1. *Wake-on-LAN*: Для осуществления удаленного включения компьютеров в локальной сети использовалась библиотека *wakeonlan*. Она позволяет отправлять “магические пакеты” компьютеру на его mac адрес.
2. *Paramiko*: Для выполнения удаленных команд на серверах по протоколу SSH использовалась библиотека *paramiko*. Эта библиотека

обеспечивает безопасное и эффективное выполнение команд на удаленных устройствах.

3. *Threading*: Для повышения производительности приложения и обеспечения отзывчивости интерфейса мы использовали многозадачность с помощью библиотеки *threading*. Она позволяет выполнять различные задачи параллельно, улучшая общую производительность.
4. *SCP* (Secure Copy Protocol): Для копирования файлов между устройствами в локальной сети выбрана библиотека *scp*. Она обеспечивает безопасную передачу файлов между устройствами по протоколу SSH.
5. *Ping3*: Для проверки доступности устройств в локальной сети была взята библиотека *ping3*. Эта библиотека позволяет осуществлять ICMP-пинги к устройствам и определять их статус - доступны или нет. Это полезный инструмент для мониторинга состояния устройств в сети.
6. *Subprocess*: Для выполнения произвольных системных команд используется *subprocess*. Эта библиотека предоставляет интерфейс для запуска внешних процессов, что позволяет нам взаимодействовать с операционной системой, выполнять команды и получать результат их выполнения. *Subprocess* является мощным инструментом для автоматизации различных операций на уровне системы.

Эффективное использование этих библиотек позволяет обеспечить удобное и безопасное управление компьютерами и серверами в локальной сети.

## **3.2 Реализация логики в пользовательских интерфейсах**

Три разнообразных приложения, разработанных на языке программирования Python, демонстрируют универсальность и гибкость этого языка в создании разнообразных пользовательских интерфейсов. Мобильное приложение, созданное с использованием фреймворка Kivy, обеспечивает интуитивно понятное управление заметками, включая возможность добавления мультимедийных вложений. Настольное приложение, основанное на PyQt5, превращается в органайзер с удобным интерфейсом для управления задачами и календарем. Специальное приложение для Raspberry Pi, также с использованием PyQt5, предназначено для мониторинга и управления датчиками окружающей среды.

Все приложения воспользовались богатством библиотек Python для эффективной реализации своих функций. Kivy, как кроссплатформенный фреймворк, обеспечил легкость создания мобильного интерфейса. PyQt5, мощная библиотека для создания графических интерфейсов, была использована как для настольного, так и для специального приложения для Raspberry Pi. Эти библиотеки значительно упростили процесс разработки, обеспечивая стабильность и согласованный стиль между приложениями, что важно для легкости сопровождения и обновлений.

## **4. Функциональные особенности**

### **4.1 Протокол удаленного доступа**

Протокол SSH (Secure Shell) представляет собой криптографический механизм для обеспечения безопасного удаленного доступа и выполнения команд на удаленных устройствах. Он обладает ключевыми



характеристиками, такими как шифрование для конфиденциальности данных, проверка целостности данных, а также различные методы аутентификации, включая использование паролей, открытых и закрытых ключей.

SSH поддерживает управление ключами, что обеспечивает высокий уровень безопасности и удобства, особенно в сценариях автоматизации процессов. Обычно используется на порту 22, хотя администраторы могут изменять порт для повышения безопасности. Протокол также обладает функцией проброса портов, позволяя создавать безопасные туннели для перенаправления трафика.

SSH может использовать сжатие данных для оптимизации производительности, и он поддерживает две основные версии протокола: SSH-1 и более современную и безопасную SSH-2. Протокол является стандартным средством обеспечения безопасного удаленного доступа в различных операционных системах, включая Linux, Unix и Windows. Его широкое применение в области сетевой безопасности делает SSH важным инструментом для защиты удаленных соединений от несанкционированного доступа и атак.

## **4.2 Сетевое включение**

Wake-on-LAN (WoL) представляет собой протокол, используемый для удаленного включения компьютеров в локальной сети. В рамках приложения, доступного под лицензией MIT, интегрирована библиотека wakeonlan. Эта библиотека обеспечивает функционал отправки "магических пакетов" на MAC-адреса целевых компьютеров.

С помощью wakeonlan пользователи могут удаленно управлять питанием компьютеров в сети, отправляя специальные сигналы, которые пробуждают компьютер из режима сна или отключения. Этот функционал приложения полезен для эффективного управления энергопотреблением и

обеспечивает удобство в удаленной работе с компьютерами в локальной сети.

### 4.3 Терминал

В операционных системах семейства Linux терминал представляет собой интерфейс командной строки, который позволяет пользователям взаимодействовать с операционной системой посредством текстовых команд. Терминал предоставляет доступ к широкому спектру системных ресурсов и утилит, что делает его мощным инструментом для администрирования и настройки системы.

Один из ключевых аспектов терминала в Linux - это выполнение процессов под разными идентификаторами процессов (PID). Каждый процесс в Linux имеет уникальный идентификатор процесса, который используется для его управления и отслеживания. Пользователь может запустить процесс в терминале, указав определенный PID или используя команды, такие как `sudo` для выполнения процессов с повышенными привилегиями.

Процессы в Linux также могут выполняться в фоновом режиме, что позволяет пользователю продолжать вводить команды в терминале, не ожидая завершения выполнения определенной задачи. Для управления процессами в фоновом режиме можно использовать команды, такие как `bg`, `fg`, и `jobs`.

Кроме того, терминал поддерживает множество командных оболочек, таких как Bash, Zsh, и Fish, каждая из которых предоставляет свои собственные возможности и синтаксис команд. Это позволяет пользователям выбирать тот интерфейс командной строки, который наилучшим образом соответствует их потребностям.

Таким образом, терминал в Linux представляет собой мощный инструмент для взаимодействия с операционной системой, обеспечивая гибкость в управлении процессами и ресурсами системы.

## **5. Техническая реализация**

### **5.1 Мобильное приложение**

Разработанное мобильное приложение на базе Python с использованием библиотеки KivyMD, предназначено для удаленного управления процессами и устройствами в локальной сети. Приложение включает в себя функционал кнопок для включения/выключения устройств, запуска различных программ, а также возможность считывания данных из файлов локального диска мобильного устройства и их рассылки по всем компьютерам в сети.

Дополнительно была предусмотрена возможность выполнения операций с использованием только одного компьютера. Особенностью приложения является "клон" терминала Linux устройства, к которому произведено подключение, предоставляя возможность удаленного взаимодействия с системой.

### **5.2 Компьютерное приложение**

Десктопное приложение, основанное на Python с использованием библиотеки PyQt5 и разработанное с применением технологии Qt Designer, предназначено для обширного управления процессами и устройствами в локальной сети. Визуальный интерфейс, спроектированный в Qt Designer, обеспечивает простоту использования и удобство в управлении устройствами, а также выполнении различных операций.

Приложение включает функционал кнопок для включения/выключения устройств, запуска программ и выполнения операций на компьютерах в сети. Важным элементом является встроенный "клон" терминала Linux, обеспечивающий возможность удаленного взаимодействия с системой. Этот терминал позволяет одновременное выполнение всех прописанных команд в активных терминалах, предоставляя удобный способ удаленного администрирования.

Дополнительно, приложение обеспечивает функционал для чтения данных с локального диска ПК пользователя и их рассылки по всем компьютерам в локальной сети, что улучшает управление информацией.

Следует отметить, что данное десктопное приложение на основе PyQt5 и Qt Designer обладает максимальным функционалом для ПК, предоставляя удобный и мощный инструмент для удаленного администрирования локальной сети.

### **5.3 Терминал на базе Raspberry Pi**

Разработанный инструмент для администрирования конкретных зон построенное на базе Python с применением библиотеки PyQt5 и интеграции технологии Qt Designer, предназначено для удобного и эффективного удаленного управления процессами и устройствами в локальной сети с использованием Raspberry Pi 3. Интеллектуальное приложение предоставляет разносторонний функционал, включая кнопки для включения и выключения устройств, запуска различных программ, а также уникальную возможность считывания данных из файлов, хранящихся на локальном диске мобильного устройства. Через простой и интуитивно понятный интерфейс Qt Designer пользователь может легко настраивать интерфейс приложения, а также просматривать и изменять локальные файлы, отправляя их по всей локальной сети.

Созданное приложение обеспечивает надежное и безопасное управление различными процессами и устройствами в локальной сети, предоставляя пользователю удобство и гибкость в управлении системой.

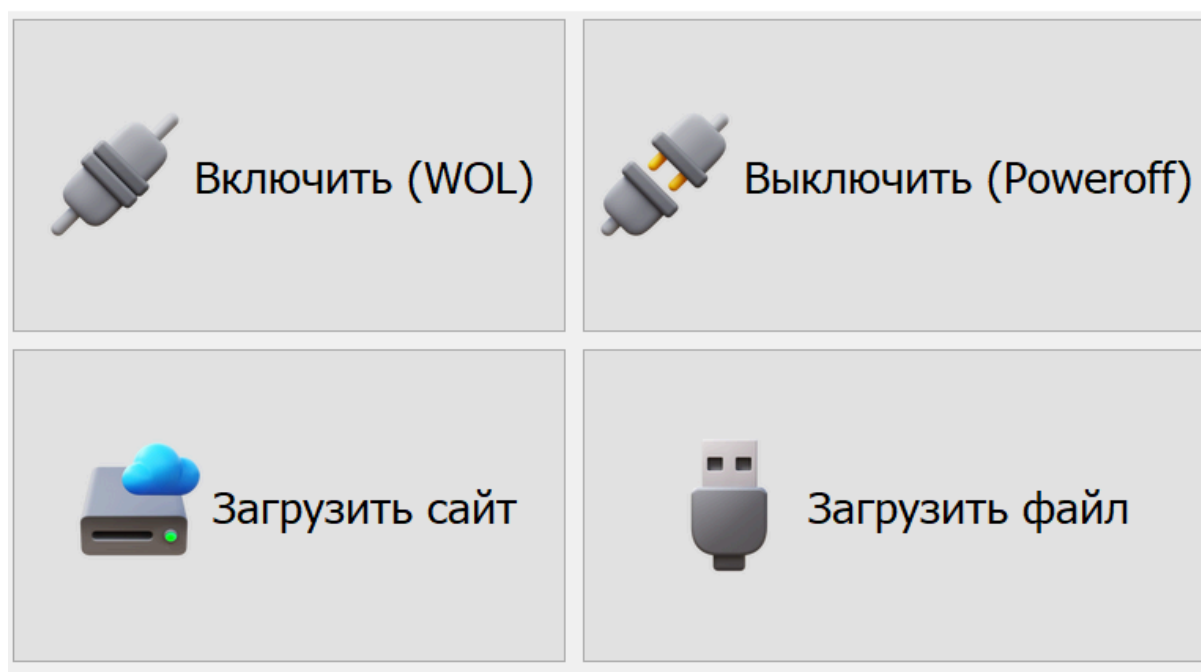


Рисунок 1 - внешний вид приложения для raspberrу pi

Из-за размеров платы функциональная схема приложения ограничена, но можно реализовать более полезный функционал с использованием дополнительных технологий, такие как администрирование с помощью внешнего монитора или удаленное подключение конкретно к плате raspberrу.



Рисунок 2 - Устройство на базе Raspberry pi 3 в пластиковом корпусе

Значительные усилия приложены по решению на данном устройстве технических проблем: установка драйвера для экрана и подбор размеров корпуса для печати.

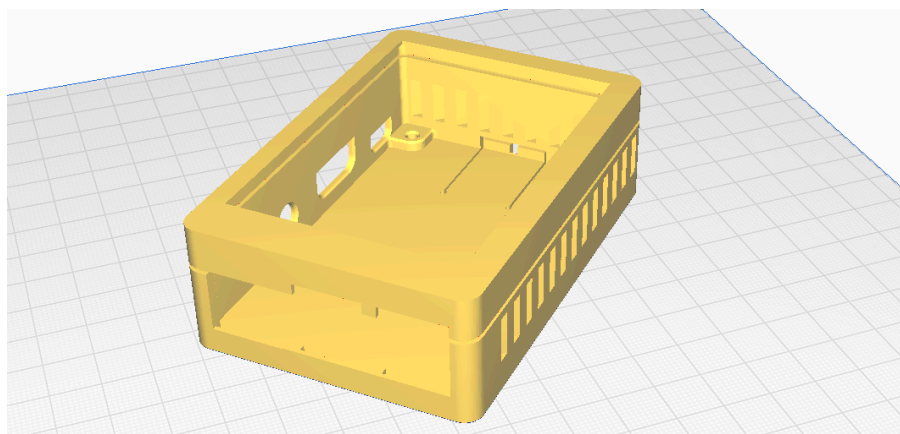


Рисунок 3 - 3D модель корпуса для raspberry pi

## 6. Результаты и выводы

В итоге было разработано мобильное приложение для удаленного управления персональными компьютерами на базе операционной системы Linux. Данное решение было успешно протестировано в локальной сети московской школы, которая насчитывает более 100 устройств подключены к одной сети. Параллельно было создано приложение для компьютеров, предоставляющее более расширенный функционал по сравнению с мобильной версией. Для компьютеров была добавлена возможность управления отдельными экранами пользователей. Также разработано устройство для администрирования отдельных зон с использованием Raspberry Pi 3, оборудованного сенсорным экраном. Данное устройство можно разместить в любой точке помещения и подключить к локальной сети, имея возможность быстрого администрирования всех устройств в конкретной зоне. Проект включает в себя не только программное, но и аппаратное решение, включая моделирование в программе Autodesk Fusion 360 и печать корпуса для устройства на 3D принтере, а также последующую сборку.

В ходе тестирования и внедрения приложений и разработанного устройства, получены положительные отзывы от пользователей и администраторов систем. Проект демонстрирует эффективность удаленного управления, а также гибкость и масштабируемость,



Рисунок 4 - распечатанное и собранное устройство на базе raspberry

поддерживая как мобильные, так и настольные платформы. Общий



результат подчеркивает успешную интеграцию программных и аппаратных компонентов для оптимизации администрирования информационных систем в различных сценариях использования.

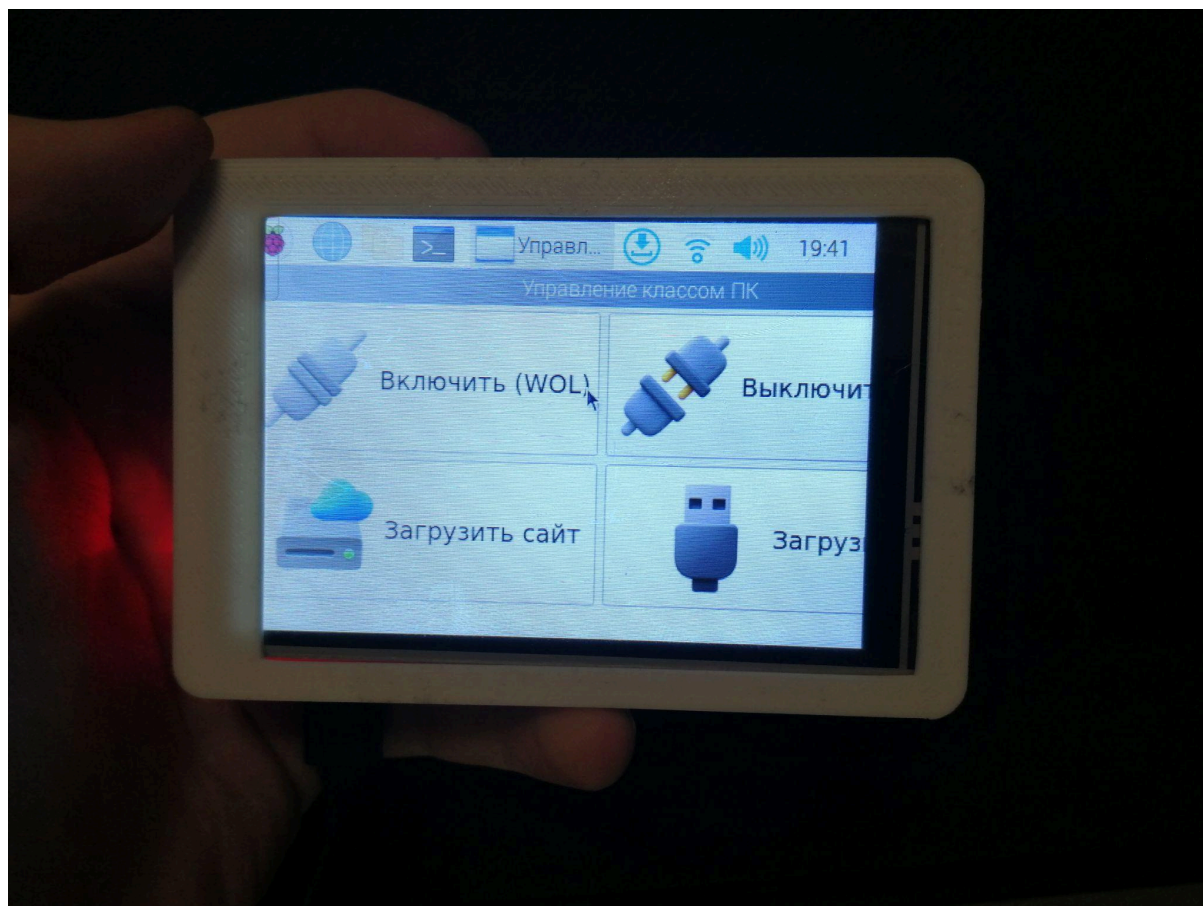


Рисунок 5 - тестирование работы приложения на raspberry pi 3

Как видно из рисунка 5, над разрешением экрана еще предстоит поработать, однако это сделать совершенно не сложно и не потратит много времени.

## 7. Список используемой литературы

1. Денис Колисниченко - "Linux. От новичка к профессионалу"  
<https://static-sl.insales.ru/files/1/443/10019259/original/725852593.pdf>
2. Карла Шрёдер - "Linux. Книга рецептов. Всё необходимое для администраторов и пользователей"



<http://d978729a.beget.tech/wp-content/uploads/2022/11/Linux.-Книга-цептов.-2-е-изд.Rescuer.pdf>

3. Андрей Робачевский - "Операционная система Unix"

[http://www.8361.ru/5sem/books/Robachevskii-Operatsionnaya\\_sistema\\_UNIX.pdf](http://www.8361.ru/5sem/books/Robachevskii-Operatsionnaya_sistema_UNIX.pdf)

4. Эндрю Хоутон - "Raspberry Pi. Настройка и использование"

<https://microtechnics.ru/raspberry-pi-obmen-dannymi-po-interfejsu-spi/>

5. Тимур Машнин - "Создание настольных Python приложений с графическим интерфейсом пользователя"

<https://www.litres.ru/book/timur-mashnin-301845/sozдание-nastolnyh-python-prilozheniy-s-graficheskim-65059196/>