

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ПО ПРОФИЛЮ «ИНЖЕНЕРНОЕ ДЕЛО»

36315

регистрационный номер

Секция: «Информационные технологии»

Программное обеспечение для администрирования в локальной сети
Контроль Удалённых Рабочих Компьютеров 5.0

Автор:

Смагин Иван Сергеевич
МБОУ Центр Образования №2, 8Е

Научный руководитель:

Коркунова Наталья Ивановна
МБОУ Центр Образования №2
Учитель информатики

Аннотация

Данная работа представляет описание процесса и разработку программного обеспечения для администрирования локальных сетей. Цель работы – создать полноценное программное обеспечение для администрирования компьютеров в локальной сети с использованием доступных в нём инструментов.

В работе изложены основные методы и приёмы, такие как: паттерны проектирования программного обеспечения, многопоточность, парадигмы программирования. В программе разработаны следующие инструменты: файлообменник (или сетевой организатор передачи), функциональная консоль, телефония и видеосвязь, конфигурация параметров. Разработка проводилась в несколько этапов.

Первый этап заключался в проектировании программы. Необходимо было определиться с файловой структурой, взаимосвязями программы. Желательно проектировать её так, чтобы можно было в дальнейшем поддерживать и дополнять различными инструментами, ведь иначе проект потеряет ценность и будет со временем непригоден к использованию. Программа должна быть кроссплатформенной, но пока что она работает только на операционных системах Windows 10 и в дальнейшем планируется перейти на Linux.

Второй этап заключался в самой разработке программы. Нужно было выбрать язык программирования и библиотеки для решения поставленных задач. Затем нужно создать файловую структуру программы – разместить файлы в понятные каталоги, систематизировать разработку программного обеспечения. Разработать объекты, которые выполняют только свою работу, разработать модули, которые используют эти объекты и представляют основную работу программы.

Выводом данной работы будет готовое программное обеспечение, которое позволит пользователям удалённо управлять компьютером в локальной сети, а также то, что важно изначально правильно спроектировать программу, чтобы в этапе разработки не возникло никаких проблем, которые требуют переписи всего программного обеспечения для их решения.

Содержание

Введение	4
1 Теория	5
1.1 Выбор языка программирования и библиотек	5
1.2 Специальные объекты и их функции	5
1.3 Основные инструменты администрирования	6
1.4 Таблица для сравнения программы с аналогами на рынке	8
2 Практика	9
2.1 Алгоритмы работы подпрограмм безопасности	9
2.1.1 Подпрограмма проверки подлинности подключения	10
2.1.2 Подпрограмма для шифрования байтов	11
2.1.3 Подпрограмма для шифрования команд	11
2.2 Демоверсии программного обеспечения	12
Заключение	14
Список литературы	15

Введение

Данное программное обеспечение актуально тем, что позволяет администрировать удалённый компьютер в локальной сети. Для меня в первую очередь данный проект – это новый опыт в разработке и проектировании программного обеспечения, знакомство с новыми возможностями и осознание большего.

Данная работа будет описывать процесс разработки программного обеспечения, какие методы и приёмы конкретно использовались, и какие задачи каким способом были выполнены.

Цель работы – создать полноценное программное обеспечение для администрирования компьютеров в локальной сети.

Основными задачами в разработке стали:

1. Проектирование файловой структуры, взаимосвязей и возможностей программы.
2. Разработка интерфейса всплывающих диалогов, основных окон и модулей.
3. Программирование работы основного приложения и его связь с модулями.
4. Программирование основной работы модулей (или инструментов).
5. Программирование дополнительных возможностей программы.
6. Внедрение дополнительных обработчиков ошибок во время использования.

Во время выполнения каждой задачи выполняется тестирование с целью исправить ошибки.

К примерам применения данного программного обеспечения, можно отнести работу системного администратора. Когда на компьютере сотрудника происходит какая-то нештатная ситуация, системный администратор может подключиться к данному компьютеру и исправить её, используя доступные инструменты для администрирования в программе.

Теория

Перед тем, как перейти в практическую часть (практику) или разработку, необходимо прояснить некоторые важные теоретические материалы, которые будут упомянуты дальше.

В данном разделе будут описаны технические инструменты, которые использовались в работе: язык программирования, планируемая работа специальных объектов и их использование модулями. Также будет представлена таблица для сравнения данной программы с аналогами.

1.1 Выбор языка программирования и его библиотек

В данной разработке используется высокоуровневый язык программирования Python, так как я более-менее хорошо его знаю и могу написать на нём программное обеспечение. Теперь самое время представить его библиотеки, которые я использую для выполнения поставленных задач в проекте:

1. PyQt5 – библиотека, представляющая собою набор расширений и функций для разработки графических интерфейсов на Python.
2. Socket – библиотека, представляющая программный интерфейс для обмена данными между процессами, которые могут работать как и на одном компьютере, так и на другом. Главное – чтобы их связывала сеть.
3. Threading – библиотека, которая позволяет реализовывать многопоточность на языке программирования Python.

1.2 Специальные объекты и их функции

Ниже представлен список специальных объектов. Они были разработаны для того, чтобы основные файлы не были огромными. Их легко читать, поддерживать – они выполняют строго свою роль в программе, не зависят от других компонента.

1. Аудио – объект для манипуляции потоками аудио (входом и выходом). Запись байтов звука в поток вывода позволяет выводить звук через устройство-вывода (наушники, колонки и так далее). А запись потока входа получает байты звука из устройства-ввода (микрофона).

2. **Контакты** – объект, который несёт ответственность за контактную книгу. Позволяет добавлять новый контакт, редактировать существующий, получать сетевое имя компьютера по контакту и контакт по сетевому имени.
3. **Конфигурация** – объект, который манипулирует конфигурационными файлами программы. В конфигурационные файлы входят: данные профиля пользователя, данные настроек. Этот объект создаёт конфигурационные файлы в случае их отсутствия, может читать параметры и записывать их. Данные профиля и данные настроек шифруются подпрограммами безопасности.
4. **Электронная почта** – объект для отправки электронных писем по шаблону на Яндекс. Почту, используя аккаунт программы и отправляя на аккаунт создателя.

1.3 Основные инструменты администрирования

Они представляют основной функционал программы и с помощью них пользователь может удалённо управлять компьютерами, используя возможности каждого инструмента.

1. **Сетевой организатор передачи (или Файлообменник)** – инструмент, предоставляющий пользователю возможность управлять файловой структурой на своём компьютере и на удалённом компьютере через интерфейс. Инструмент позволяет: создавать и удалять файлы и папки, редактировать содержимое и переименовывать. В интерфейсе есть возможность перетаскивания файлов и папок в другие пути или в другую файловую систему, что начинает процесс обмена файлами между компьютером и удалённым компьютером.
2. **Внутри-компьютерная почта** – инструмент, который реализует почту на удалённом компьютере. Данная почта доступна к просмотру и изменению только в программе. Пользователи, которые включили данный инструмент смогут отсортировывать письма по меткам и просматривать их содержимое. Письмо может оставить каждый пользователь, например, для описания нештатных ситуаций на компьютере или делать какие-то пометки. На письма также можно отвечать и просматривать оставленные ответы.

3. **Функциональная консоль** – инструмент, который позволяет работать с командной строкой. Можно выбрать участников работы, которые будут получать эти команды. Пользователь, отправляя команду в консоли удалённым компьютерам, получает ответ. Помимо отправки команд, инструмент может исполнять Python команды. Можно открывать браузерные страницы и активировать режим администратора для больших привилегий.
4. **Телефония и видеосвязь** – инструмент, который позволяет установить коммуникацию между удалённым компьютером. Пользователь может вещать голосом, управлять мышкой и клавиатурой, просматривать удалённый рабочий стол и его веб-камеру.
5. **Конфигурация параметров** – инструмент, который позволяет управлять буфером обмена удалённого компьютера, удалённо печатать документы и изменять параметры – громкость, яркость и так далее.

Все инструменты передают данные по локальной сети через специальные объекты подключения, которые обрабатывают команды через подпрограммы защиты и отправляют их.

1.4 Таблица сравнения программы с аналогами на рынке

В данном разделе приведена таблица сравнения разрабатываемой программы с другими аналогами на рынке для администрирования локальных компьютеров. Проект (Контроль Удалённых Рабочих Компьютеров 5.0, или Контроль У.Р.К-5.0) будет сравниваться с такими программами, как: LiteManager, Radmin, TeamViewer (см. список литературы).

Возможности	Контроль У.Р.К-5.0	LiteManager	Radmin	TeamViewer
Просмотр рабочего стола удалённого компьютера	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Передача файлов	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Телефония и видеосвязь	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Подключение по Интернету	Отсутствует (-)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Шифрование, безопасность	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Удалённое управление	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Диспетчер задач	Отсутствует (-)	Присутствует (+)	Отсутствует (-)	Отсутствует (-)
Командная строка	Присутствует (+)	Присутствует (+)	Присутствует (+)	Присутствует (+)
Редактор реестра	Отсутствует (-)	Присутствует (+)	Отсутствует (-)	Отсутствует (-)
Платформы	Windows 10	Все	Windows	Все
Дата регистрации	2024, 15.02	2016, 16.02	2010, 15.12	2014, 25.03
Бесплатное пользование	Да	Условное	Нет	Условное

Рисунок 1. Таблица сравнения моей программы с другими аналогами на рынке

Практика

При разработке программы применялись следующие похожие паттерны проектирования на: **шаблон функционального дизайна** (каждый модуль программы имеет только одну обязанность и исполняет её с минимумом побочных эффектов на другие части программы), **шаблон модель-представление-контроллер** (модель отвечает за саму логику работы программы, представление выводит информацию в интерфейс, контроллер обрабатывает ввод от пользователя, меняет поведение модели и обрабатывает её вывод).

Использовалась парадигма программирования ООП (Объектно-Ориентированное Программирование), так как оно повышает читаемость программы и позволяет лучше выстраивать взаимосвязи, разбивая всю программы на объекты и функции.

В программном обеспечении также использовалась многопоточность. С помощью неё можно работать сразу с помощью нескольких инструментов. Она не блокирует интерфейс, ожидая данные сервера - с помощью неё можно ставить выполнение задач на задний план.

В этом разделе будут представлены алгоритмы работы модулей, некоторых важных объектов и основной связи взаимодействия программы со всеми модулями. Алгоритмы, которые будут дальше изображены, являются соображением автора программного обеспечения

2.1 Алгоритмы работы подпрограмм безопасности

Подпрограммы безопасности были разработаны для того, чтобы защитить передаваемые данные по локальной сети и обеспечить надёжность в подключении к серверам, установленных на компьютерах пользователей. В этом подразделе будут описаны виды подпрограмм и их алгоритмы.

2.1.1 Подпрограмма проверки подлинности подключения

Данная подпрограмма несёт ответственность за проверку подлинности подключения. С помощью неё сервер распознает кто к нему хочет подключиться – клиентское приложение Контроль У.Р.К-5.0 (данное программное обеспечение, официальное) или стороннее вредоносное ПО. Алгоритм работы приведён ниже:

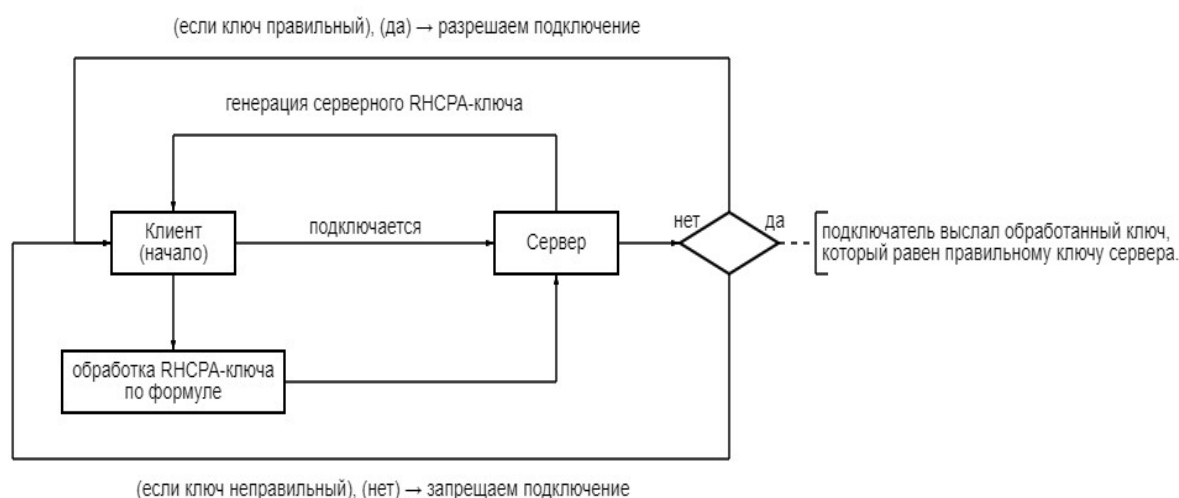


Рисунок 2. Алгоритм работы подпрограммы проверки подлинности подключения

Суть проверки заключается в том, что тому, кто подключается к серверу, необходимо выслать обработанный по формуле (2) РНСРА-ключ из псевдослучайно сгенерированного по формуле (1) числа, который прислал сервер. Если тот, кто подключается, не выслал обработанный ключ или высылал неправильный, то подключение запрещают. Формулы представлены ниже:

Формула 1 состоит в несколько этапов: $192.168.33.100$ (IPv4 адрес сервера) = $19216833100 = 19216833100 + 978307$ (число в диапазоне от 0 до 1000000) = $19217811407 * 0.(\text{день})(\text{месяц})(\text{год})0804 = 19217811407 * 0.100320240804 = 1927935468.076098$

Формула 2 состоит в несколько этапов: 1927935468.076098 (полученный ключ) = $1927935468.076098 / 0.(\text{день})(\text{месяц})(\text{год})0804 = 1927935468.076098 / 0.100320240804 = 19217811407 = 19217811407 * 2709 = 52061051101563 =$ заменяем все 3 в числе 52061051101563 на 9 = 52061051101569

2.1.2 Подпрограмма для шифрования байтов

Данная подпрограмма безопасности была разработана для шифрования байтов для их безопасной передачи по локальной сети. Алгоритм подпрограммы представлен ниже:

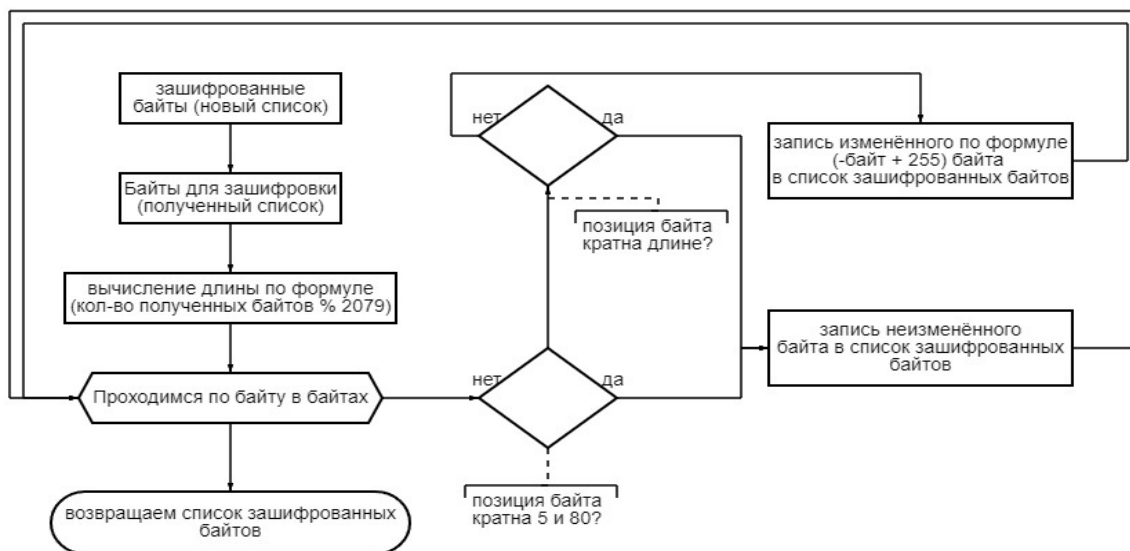


Рисунок 3. Алгоритм работы подпрограммы для шифрования байтов

2.1.3 Подпрограмма для шифрования команд

Данная подпрограмма безопасности была разработана для шифрования текстовых команд для их безопасной передачи по локальной сети. Алгоритм данной подпрограммы представлен ниже:

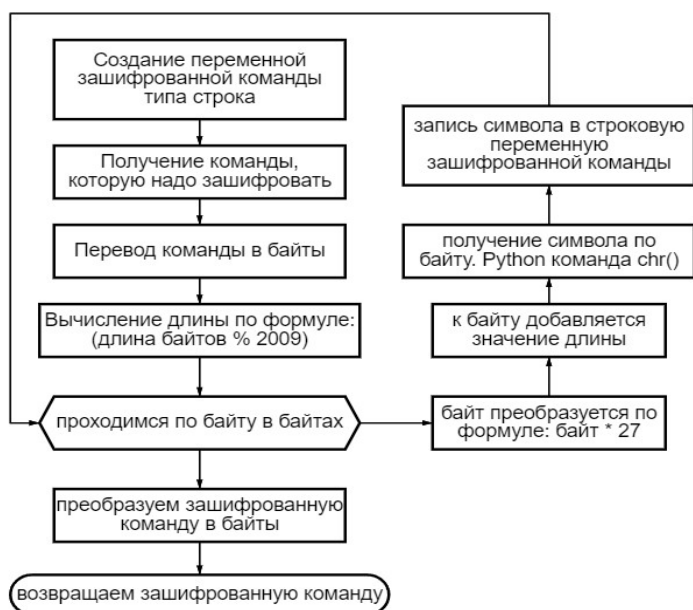


Рисунок 4. Алгоритм работы подпрограммы для шифрования команд

2.2 Демоверсии программного обеспечения

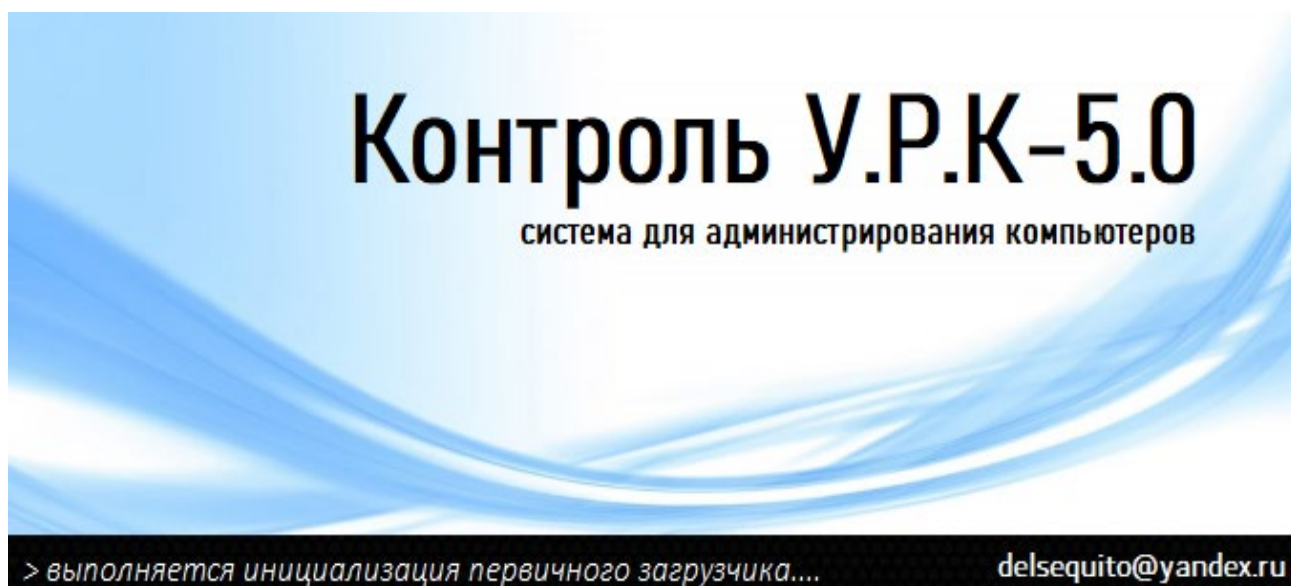


Рисунок 5. Первоначальный загрузчик программы

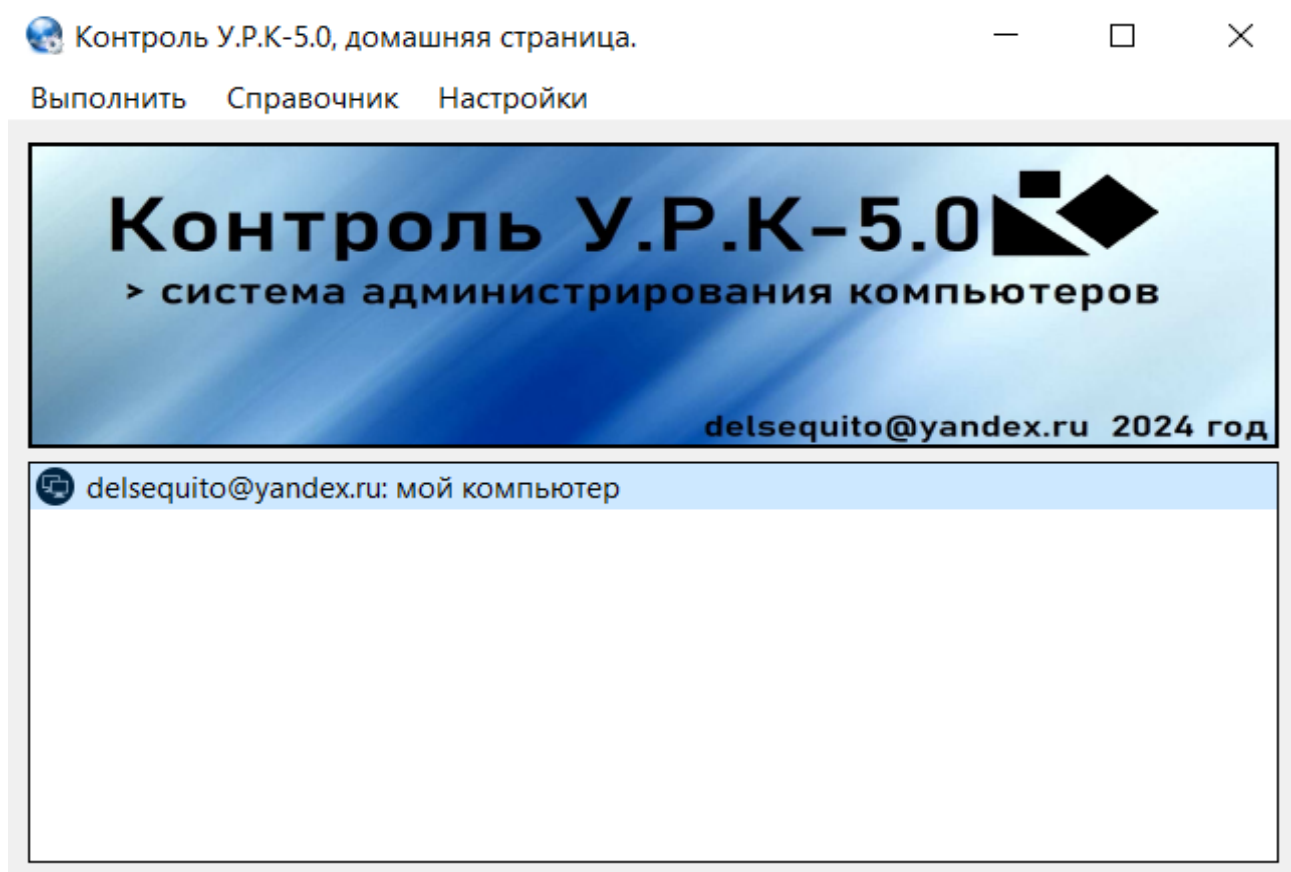


Рисунок 6. Домашняя страницы программы.

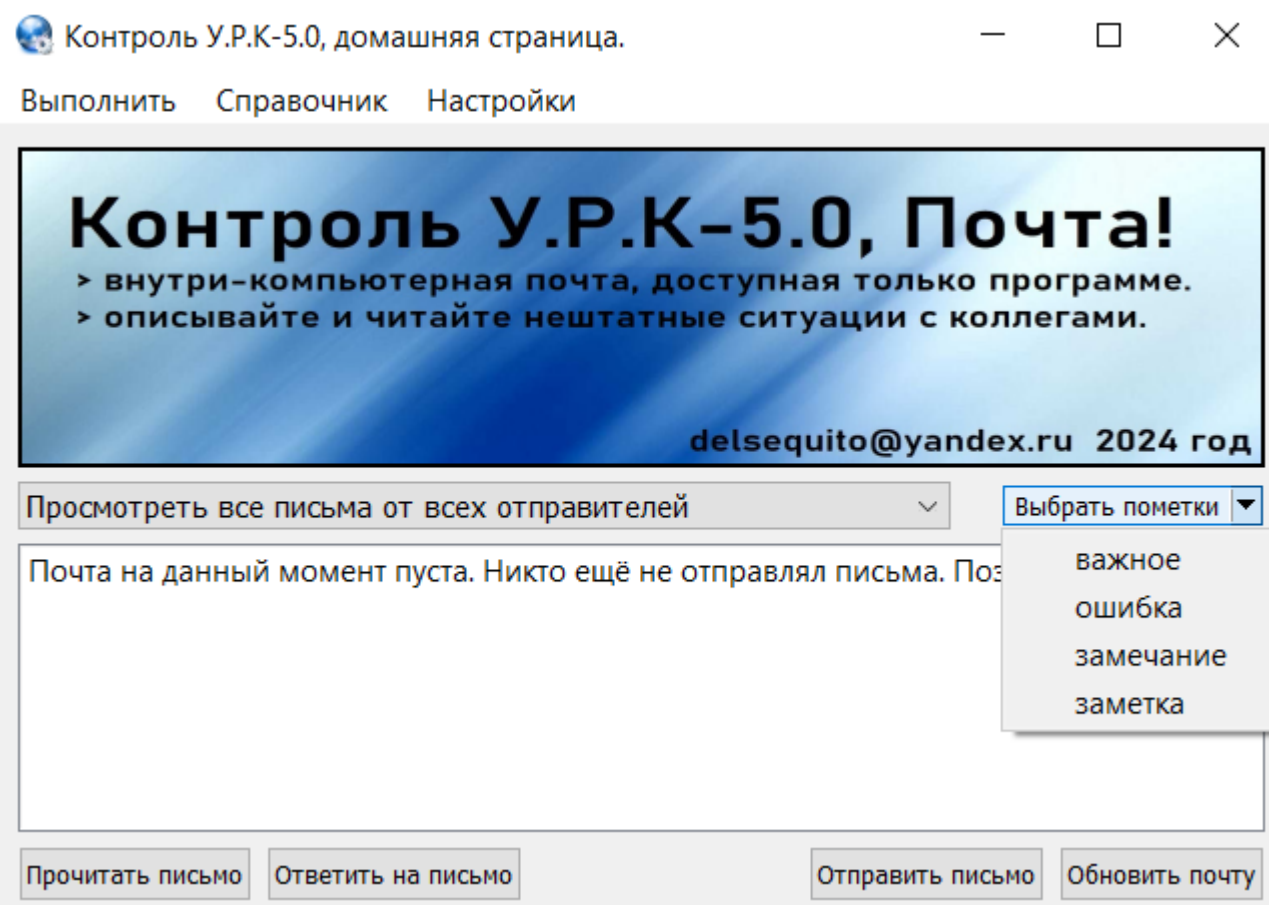


Рисунок 7. Внутри-компьютерная почта в программе.

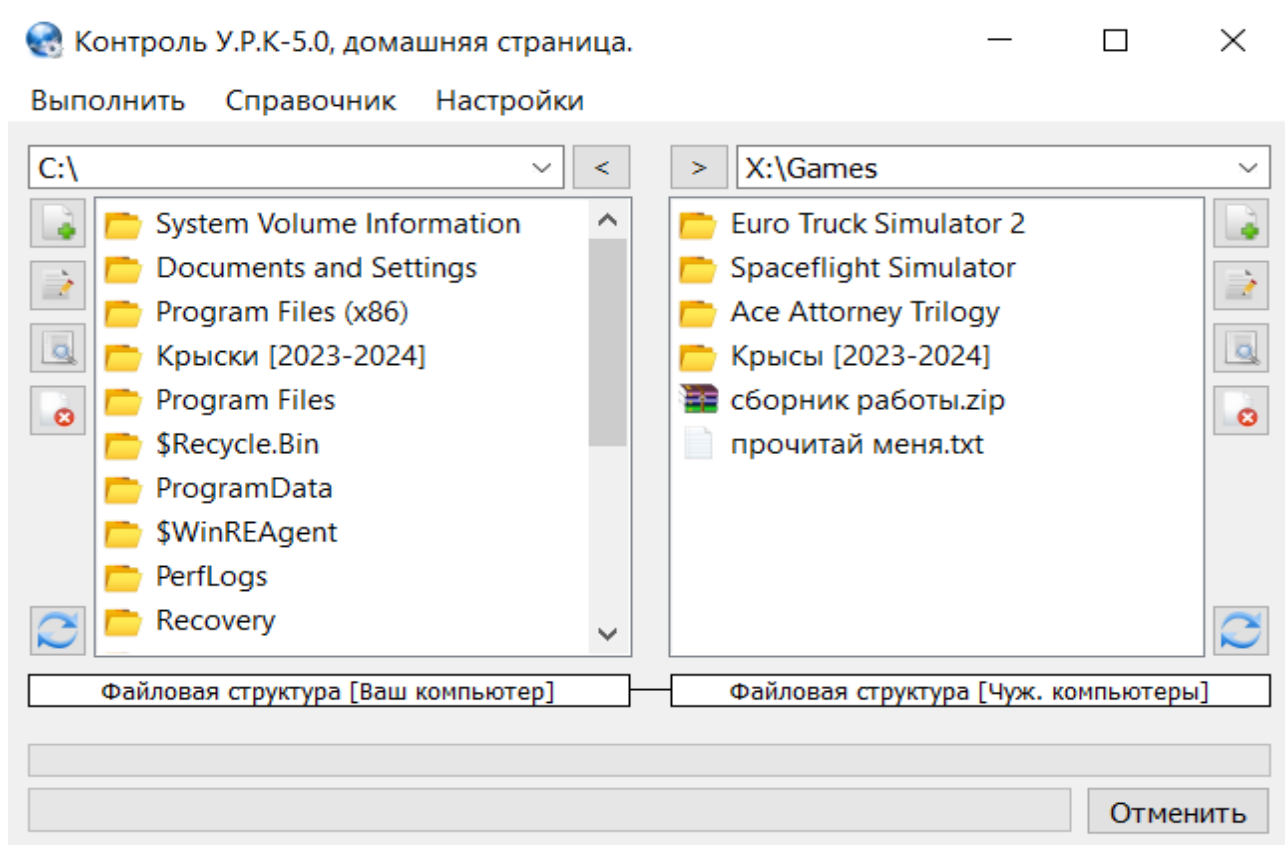


Рисунок 8. Сетевой организатор передачи (или Файлообменник)

Заключение

Итогом проделанной работы является программное обеспечение для администрирования компьютеров в локальной сети.

Из этого опыта можно извлечь следующее:

1. Для разработки программного обеспечения существуют много паттернов программирования, которые позволяют разработать продукт, который можно будет легко поддерживать в будущем.
2. Перед разработкой всегда нужно продумать основные взаимосвязи программы с её компонентами, чтобы во время этапа создания не возникло никаких проблем, требующих переписать всю программу с нуля.
3. Всегда необходимо разбивать программу на объекты и на модули, необходимо сократить дублирование кода.

В дальнейшем можно будет написать программное обеспечение – **Менеджер разработки программного обеспечения**, который включает в себя систему для упрощения разработки программ на Python: среда разработки, система автоматического и удалённого тестирования, система визуального проектирования и так далее.

Список литературы

Документация по языку программирования Python

<https://docs.python.org/3.11/>

Паттерны проектирования и шаблоны

https://ru.wikipedia.org/wiki/Шаблон_проектирования

Модель-представление-контроллер

<https://ru.wikipedia.org/wiki/Model-View-Controller>

Информация о ПО “Radmin”

<https://ru.wikipedia.org/wiki/RAdmin>

Информация о ПО “LiteManager”

<https://ru.wikipedia.org/wiki/LiteManager>

Информация о ПО “TeamViewer”

<https://ru.wikipedia.org/wiki/TeamViewer>

ООП (Объектно – Ориентированное Программирование)

https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование

Документация по PyQt5

<https://doc.qt.io/qtforpython-5/contents.html>

Документация по Socket

<https://docs.python.org/3/library/socket.html>

Документация по Threading

<https://docs.python.org/3/library/threading.html>