1. Основное ТЗ

Цель
Необходимо создать мессенджер на python или C на основе сокетов от компьютера к компьютеру по примеру скайпа.
1.Этап - консольное приложение с консольным интерфейсом
2.Этап - простейший интерфейс
а)поле для ввода текста
б)поле для вывода сообщения от Вашего контакта
в)кнопка отправить сообщение
г)кнопка выбора контактов в виде списка

Возможные варианты реализации
1.клиент-сервер-клиент (весь трафик идет через сервер)
2.клиент-сервер-клиент (сервер служит только для коннекта общающихся, трафик идет напрямую)
3.клиент=сервер - клиент=сервер (коннект происходит вручную путем отправки ключа, адреса или чего-то иного друг другу через сторонние мессенджеры)
4.клиент=сервер - клиент=сервер (коннект происходит автоматически, если это вообще возможно в такой схеме)
P.S.Варианты упорядочены по порядку от самого нежелательного к самому предпочтительному

В разработке
1)не использовать ООП
2)не использовать никаких фреймворков, все делать голыми руками.
3)необходимо разобраться в самых глубинных принципах работы протоколов TCP, UPD и на самом базовом уровне отправлять сокеты
4)максимальная простота в разработке (в коде все должно быть понятно и прокомментировано по русски)
5)не использовать всякую херню типа функционального программирования, лямбда функций замыканий и т.д.
6)использовать обычный процедурный стиль (функции и глобальные переменные при необходимости)

Отчет о работе
ежедневный созвон и подробный рассказ о проделанной работе со всеми подробностями. Причем работа - это не всегда прям код, понимание каких-то важных сложных вещей тоже, естественно, будет считаться работой

2. Запуск программы из архива

2.1. Быстрый способ (предустановленные сертификаты)

В быстром способе используются сертификаты уже созданные разработчиком

- Папку cert положить на свой компьютер
- Папку client тоже положить на свой компьютер
- Папку server положить на удаленный сервер

2.2 Длинный способ (создание собственных сертификатов)

А Создание центра сертификации:

1. На локальный компьютер скопировать из архива папку cert и удалить все файлы с расширением отличным от ".sh".
2. Зайти в папку *cert/CA/* и запустить команду `bash create_cert.sh`. В процессе потребуется придумать пароль к приватному ключу центра сертификации и ввести его два раза. Этот может быть любой, его нельзя забывать. Он потребуется далее для выпуска собственных сертификатов.
3. В результате в папке *cert/CA/* появятся два файла *rootCA.key* и *rootCA.crt*. Файл *rootCA.key* нужно хранить в секрете и никому никогда не передавать. Файл *rootCA.crt* нужно положить рядом с *программой-клиентом*.

Б. Создание сертификата для программы-сервера:

1. Положить на удаленный сервер файлы *программу-сервер* и скрипт *create_key_csr.sh .*
2. Запустить скрипт командой `bash create_key_csr.sh` , в процессе потребуется вбить пароль. Этот пароль должен совпадать с паролем в коде *программы-сервера.* В результате рядом появятся два файла: *domain.key* и *domain.csr .*
3. Скачать файл *domain.csr* с удаленного компьютера на локальный и положить его в папке *cert* .
4. В папке *cert* выполнить команду `bash create_certificate.sh` , в процессе потребуется ввести пароль от приватного ключа центра сертификации (см. часть А). В результате в папке *cert* появится файл *domain.crt* , который нужно перенести на удаленный сервер и положить рядом с *программой-сервером*.

В. Создание сертификата для встроенного сервера программы-клиента:

1. На локальном компьютере зайти в папке *cert* и выполнить команду `bash create_key_csr.sh` , в процессе потребуется вбить пароль. Этот пароль должен совпадать с паролем в коде *программы-клиента.*
2. В этой же папке *cert* выполнить команду `bash create_client_certificate.sh` , в процессе потребуется ввести пароль от приватного ключа центра сертификации (см. часть А). В результате в папке *cert* появится файл *client.crt* , который нужно положить рядом с *программой-клиентом*.

TODO

Понять про Unicode, сколько байтов, как это вообще работает. Может ли символ разбиться на пакеты. Будут ли проблемы с отображением если разобьется.

Сделать чтобы при закрытии консоли крестиком программа закрывалась так же, как и при Ctrl+C

Выводить статус загрузки файла по команде пользователя в %

Обработать конфликт совпадения id. Возможно с помощью доп параметра unixtime или через политику назначения id.

Написать комментарии к коду
Написать комментарии к коду в main

В функции display сбрасывать и повторять текущий ввод при отображении сообщений(похоже что невозможно именно в таком виде)

Что будет в случае внезапного разрыва соединения (интернета) при передаче большого файла. Может ли он потом докачаться, когда скажем через 3 сек соединение восстановится. Чтобы не скачивать заново

Кол-во сеансов на сервере (примерно понять сколько). Проверить с помощью автоматизированного теста.

Завершить текущий чат (переключиться на невалидного собеседника)

Сейчас общение через интернет. Рассмотреть вариант, когда вообще нет доступа в интернет, т.е. все в одной замкнутой локальной сети.

Можно реализовать проверку на наличие клиента в белом списке, чтобы не было автоконнекта

Аутентификация клиентов

DONE

Отображение байтов понормальному в 16 и 2 виде. (с помощью f строк)

Сделали заглушки для шифрования, нужно используя эти функции сделать шифрование файлов перед отправкой и при получении. Файл копируется и шифруется, затем отправляется и расшифровывается.

Добавить метку supertext для глобальных констант

Возможно лучше сделать клиента с помощью библиотеки asyncio, чтобы не было переключения между ожидающими потоками (Версия с потоками тоже работает нормально если использовать неблокирующие режимы и очереди)

? Сделать отправку из двух очередей в разных потоках (две очереди оставил, чтобы не пилить приоритетную, но отправка идет из одного потока, чтобы не было конкурентного доступа к сокету на запись)

? Добавить больше Event-ов для отображения пользователю что происходит

Проблема с ситуацией прерывания коннекта при пересылке большого сообщения
? Сделать чтение файла отдельным потоком, так как это занимает много времени

Посмотреть проблему с отправкой сообщений пока передаются файлы. Со стороны отправителя сейчас есть задержка

Автоматом создавать папку downloads для загрузки

Вынести взаимодействие с пользователем из main
Сделать отправку от сервера к клиенту сигнала QUITED, если данный клиент неожиданно отключился
Решить проблему с отправкой файлов (чтобы доходили и не терялись если буфер заполнен)
Идентификация клиентов
Логика выбора собеседника
Выпадение списка, выбор номера собеседника
Обработка ошибок
Формат пакетов (команды / файлы / текст)

TODO GUI

1) [До 1 дня] Уникальность запуска. Обработка на сервере уникальных id.
Проверка на стороне сервера совокупности параметров: ip, port, source_id, unique_key.
Запрет подключения дважды с этими параметрами.
Еще надо будет сделать, что если в системе запущена одна программа с уникальным ключом, то полная копия этой программы с тем же ключом не должна запуститься

Сервер должен отправлять команду на восстановление id

2) [До 2 дня] Шифрование пакетов.RSA
Будем ассиметричным шифрованием зашифровывать сообщения с помощью какой-то стандартной штуки, а внутри будет уже наш ключ для аутентификации и шифрование данных. SSL/TLS для UDP посмотреть, есть ли такое.????

Шифрование для встроенного сервера с уникальным ключом

ПОКАЗАТЬ КАК ПАКЕТЫ ШИФРУЮТСЯ В WIRESHARK

3) [До 1 дня] Доработка интерфейса и кода.

Подготовить интерфейс к работе с фреймворком через сокет, но не делать еще.

Насчет дехассемблирования, возможно стоит еще прогнать код прграммы через “порчу”. Надо посмотреть такие программы.

Группы контактов как в тг, папка спам. Подумать насколько сложно сделать.
В папку спам попадают те чаты, которые пользователь закрыл крестиком. Если к нему начинают ломиться, то не засоряются рабочие папки, а чат продолжает отображаться в папке спам. Можно сделать через фильтр и дополнительную внутреннюю переменную.

странный баг, общение 2-х клиентов идет, но у 2-ого из них написан О про коннект первого. Это случилось при повторном подключении, а не при самом первом

в коде будет проверка на истечение срока пользования программой - в интерфейсе выводить информацию о завершении лицензии и о предстоящем завершении лицензии

Сделать полоску сверху для показа сообщения об истечении через месяц, до этого не показывать.
Сделать запрос времени сервера и проверять срок лицензии по нему, а не по локальному времени компьютера.

Сделать чтобы в поле ввода id можно было вводить только цифры (игнорировать другие символы) и ограничить по длине. Короче проверку надо сделать с выводом сообщений об ошибках

подумать как сообщить пользователю, что он вбил некорректный/несуществующий id. Белый список и окошко для подключения. Как-то надо отображать, когда пользователь пишет в чат, в котором нет подключения.

а еще если во время загрузки неожиданно один из 2-х общающихся закроет чат - должно быть какое-то сообщения на месте прогресс бара не complete как в случае удачного завершения, а erro

Отображение времени отправки и времени получения сообщения (для времени получения видимо надо сделать управляющий сигнал)
Для каждого диалога можно поставить галку сохранять историю при закрытии программы.

Отдельное поле для показа отправленных и скачанных файлов с отображением статуса загрузки и кнопкой показать все Или через фильтр файловых сообщений

Выбор языка интерфейса (хотя бы из 2х вариантов русский-английский) с перспективой на добавление любых новых языков

Надо будет сделать шестеренку где-нибудь - при нажатии на которую будут открываться настройки чата - и там что-то можно будет понастраивать (тот же язык) и т.д.

4) [До 1 дня] Восстановление отправления/принятия файла после закрытия отправителя или получателя. Сохранение в файл траты лимитов.

5) [До 3 дня] Сохранение и загрузка истории сообщений + отображение в интерфейсе нового дня. По умолчанию надо сохранять + при наведении на галочку - подсказка, что она значит
При закрытии программы выводить информационное сообщение, что все неотправленные сообщения будут потеряны.
Подсказка для галочки сохранения чатов

6) [До 2 дня] Фичи + КРАСИВОСТИ
Скругление уголков в сообщениях, ….

важно еще будет добавить скорость загрузки файла во время загрузки - постоянно ее пересчитывать вместе с прогресс баром, а по завершении отобразить среднюю скорость загрузки. И отобразить информацию о размере файла.

Доработать Label чтобы он мог выделять текст или научиться изменять динамически размер Text

Редактирование отправленного сообщения (Если новое сообщение больше одного пакета, то обрубить до одного пакета)

Возможно поиск по контактам (после всего)

Ctrl+z добавить, чтобы работало

Ответ на сообщение (reply)

7) P2P через udp

8) Потоковое аудио, видео

DONE GUI

Проверить чтобы не было зависания при запуске скомпилированной программы с одинаковым уникальным ключем

Элемент показа состояния собеседника сделать словом и с подсветкой слова. Онлайн - зеленый, оффлайн - бледно красный, отказ - серый

Сделать чтобы при открытии gui консоль не открывалась

Вывод сообщения для копирования рядом с сообщением в теле чата.

картинку и иконку намертво в код вписать в виде бинарных файлов? Чтобы при упаковке в exe они были частью exe

Перышко лучше убрать и заменить на свой символ

Подтверждение соединения надо вывести в тело чата, чтобы они постоянно не возникали перед пользователем.

Проработать механизм для оповещения собеседника о нежелании принимать файл

Проработать механизм для оповещения собеседника об ошибке внутри отправки файла

6) в эти функции надо добавить логику внутрь, если пришел ключ равный условно (-1), то надо вывести всплывающее сообщение над полем ввода типа (ключ не найден) и не отправлять

7) надо продумать систему лимитов на общение каждой пары пользователей в виде количества пакетов, которые можно потратить на сообщения и на файлы. Пока что хранить эти лимиты будем в текстовом файлике рядом и вручную добавлять, когда кончатся. по умолчанию какое-то количество при 1-м коннекте должно даваться - скажем 1000 и записываться в файлик
лимит на сообщения и лимит на файлы - разные лимиты
сколько осталось по каждому лимиту надо где-то рисовать. Либо в чате наверху с данным пользователем, либо в менюшке слева - надо подумать где красивее будет

Если текстовое сообщение большое (больше одного пакета), то тоже надо в теле чата запросить подтверждение траты пакетов.

Со стороны получателя надо сделать проверку целостности файла. Надо сделать чтобы получатель мог определить какой пакет ему нужно дополучить в случае разрыва соединения.

Сделать трату пакетов сразу при отправке, а не пакет за пакетом. Перед отправкой в теле чата под сообщением вывести вопрос с подтверждением траты кол-ва пакетов. Если пользователь согласился, то при отмене пакеты все равно потратятся. Надо сделать отдельный тип пакета для передачи размера файла в нешифрованном виде.

Строчку с указанными лимитами надо тоже покрасить в какой-то другой близкий цвет.

Кнопку с закрытием чата надо сделать менее заметной

Сделать в левой панели подсвечивание фона как в тг с тем, с кем сейчас открыт чат.

Добавить картинку на начальный фон вместо зеленого фона

3) надо сделать всплывающее окно, над полем ввода, куда выводить в виде всплывающей, а потом исчезающей подсказки отправляемое "типа зашифрованное" сообщение в виде 0 и 1, причем в 2-х видах - до типа шифрования в виде 0 и 1 и после типа шифрования тоже в виде 0 и 1 (тоже на стороне получателя)
пользователь должен наглядно видеть, как его сообщение превратилось в 0 и 1, потом типа зашифровалось (в нашем случае развернулось), а потом эта штука должна сама исчезнуть. И эта опция должна быть включаема/отключаема

4)сделать 5 байтный id-шник,чтобы дипазон был 2^40 степени

5)надо сделать 2 разных функции шифрования/дешифрования - одну для пакетов сообщений, другую для пакетов файлов. У них будет разное шифрование

2)при начале отправки большого файла должен быть способ отменить эту загрузку.
Отобразить пользователю сразу, чтобы он не пугался

Отображение падения сервера и способ реконнекта

resize сообщений при первом открытии

Перенести виджет отправки сообщения в каждый чат

вот еще странный баг - первое сообщение "да" у одного написано во сколько ушло, а у другого вместо "recieved" пустой "sent" почему-то

Сочетания клавиш ctrl+c, ctrl+v, ctrl+a, ctrl+x на русской раскладке

еще после двойного клика по сообщению - появляется окошко с этим сообщением. И в этом окошке выделить мышкой сообщение можно - а вот скопировать нельзя никаким способом:
1)не работает ни ctrl+c

2)не работает контекстное меню правой кнопкой мыши

еще есть 1 косяк, который неочевидный, но надо обработчик написать. При определенной внутренней зашифровке содержимого пакета - на расшифровке при неправильном способе расшифровки ф-ия просто не справляется с расшифровкой в utf-8 и падает с ошибкой, а должна выводить хотя бы, скажем, сами биты или кракозябры, а не падать с ошибкой

еще 1 странный баг - если одного клиента закрыть, а потом открыть заново - у него в списке контактов появляется странная строка id:0 наряду с прошлым контактом, с кем он общался до этого.

еще надо будет сделать так, чтобы selfid, над полем коннект который - тоже поддавался копированию

еще надо будет сделать, чтобы отправленные и полученные по цвету фона различались - ну это легко видимо. Ну и для красоты скруглить уголки у цветных квадратиков

Сделать, чтобы по завершению загрузки прогресс бар заменялся на Completed или типа того.

Отображение кол-ва новых (непрочитанных) сообщений

Прокручивание по колесику

dest_id будут назначаться по определенной политике. И тогда можно создать базу данных (текстовый файл) со списком контактов. Можно там же в этом файле хранить человекочитаемые имена. Можно реализовать по двойному нажатию на диалог

Копирование (выделение) текста сообщений или по правой кнопке мыши. (В tk самым адекватным вариантом оказалось сделать доп окно)

Удаление виджета (или другое отображение) при отсутсвии подключения к собеседнику

Словарь с лэйблами сообщений (ключ-contentID) для каждого собеседника. Отдельные словари должны быть для отправленных сообщений, отдельный - для принятых

Отображение своего id

Сделать для прогресс бара ограничение по максимальной длине

Заменить Enter на Shift-Enter

Растягивание поля ввода при превышении длины строки

Окно для выбора dest_id, можно либо ввод, либо выбор из списка

Кнопка выбора файла

Растягивание на весь экран

3. (СДЕЛАТЬ) Реализация

Особенности компиляции

pyinstaller –onefile -w client.py
pyinstaller –onefile --noconsole client.py
pyinstaller –onefile --windowed client.py

Для создания png можно использовать доп скрипт png_to_base64.py
python3 png_to_base64.py file.png

2.1 Общее описание

Есть две программы: центральный сервер (он же координационный и сигнальный) и программа клиента

Клиент реализует

- поток для общения с центральным сервером
- поток который будет отвечать за установку соединения с другим клиентом (не важно через сервер или напрямую. Главная задача предоставить какой-то интерфейс (сокет?) для общения
- поток для отправки файлов через очередь файловых пакетов
- поток для сборки файлов из файловых пакетов
- поток для отправки текстовых сообщений
- поток для чтения и диспетчеризации принятых файлов (текстовые сообщения выводятся пользователю, файловые сообщения добавляются в файловую очередь, управляющие сообщения обрабатываются)

Текстовые и файловые сообщения пересылаются через отдельный поток / вызов асинхронной функции. Или возможно это будет не через очередь, а через БД.
Должен быть какой-то механизм принятия файла (разрешения на принятие)

Пакеты через сокет отправляются переменной длины: фиксированный хедер + данные
Структура хедера:

- 4 байта SOURCE_ID
- 4 байта DESTINATION_ID
- 4 байта CONTENT_ID
- 2 байта CONTENT_TYPE (текст / файл)
- 2 байта PACKET_DATA_SIZE

2.2 Установка и настройка

TODO

2.3 Работа и выполнение

Программа при запуске создает несколько нитей (threads) с воркерами
Воркеры:

Worker	Source data	Produced data
supertext_listener	sockets	
supertext_sender	очередь из пакетов на отправку	
supertext_filesaver	очередь из принятых файловых пакетов	
supertext_controller	Управляющие пакеты с типом контента 0	
supertext_displayer	очередь пришедших текстовых сообщений и очередь событий	
supertext_connector	очередь dest_id для которых нужно установить соединение	

- sender: отправляет заранее подготовленные пакеты из очереди

Далее можно развивать программу в двух направлениях:
- Функциональность воркеров
- Функциональность интерфейса пользователя

Если файл большой, то сейчас блокируется интерфейс
Каждый воркер обрабатывает свою очередь и складывает результаты в другую очередь

Реализация клиента через потоки (библиотека threading):

Должны быть процедуры подключения и отключения клиента к серверу.

Подключение клиента к серверу:
Сервер должен идентифицировать и аутентифицировать клиента, установить с ним TCP соединение и добавить его в список активных клиентов.

Затем должен поддерживать tcp соединение и разрывать его либо по ошибке, либо по команде клиента.

При разрыве соединения клиент удаляется из списка активных клиентов.

Процедура коннекта:
клиент добавляет сокет, готовый для прослушивания в список и направляет сигнал PING
listener прослушивает все сокеты из списка и если на какой-то из сокетов приходит сигнал PONG, добавляет отправителя в список активных клиентов

2.4 Текст программы

TODO

Заметки

В качестве сервера будет последний debian 11 (или даже 12)

Все отправлять 0 и 1
До 21 июля 2 часа в день
21-22 полностью два дня в дороге
с 23 полный день
Курс приоритетнее
Если нужен сервер - написать, по настройке тоже
В идеале за месяц
Хоть 100 гб отправлять
Основная ОС Windows, затем MacOS и Linux

Рисунки

https://docs.google.com/drawings/d/1sli4jUKhLTvHuadJeO8H15ANfv_BqFqvzwsOsft3dGs/edit

Ссылки на материалы по теме

https://habr.com/ru/articles/596983/
https://compress.ru/article.aspx?id=23161
https://mixmag.io/post/internet-messenger/

Есть сервисы, которые позволяют передавать файлы p2p типа не сохраняя у себя. Эти сервисы используются только для установки соединения. По факту это то что хотелось бы.
Пример: https://toffeeshare.com/

https://habr.com/ru/companies/ruvds/articles/416821/

> Одна из главных сложностей, связанных с P2P-соединениями браузеров заключается в том, что браузерам сначала надо обнаружить друг друга, после чего — установить сетевое соединение, основанного на сокетах для обеспечения двунаправленной передачи данных. Предлагаем обсудить сложности, связанные с установкой подобных соединений.
>
> Когда веб-приложению нужны какие-то данные или ресурсы, оно загружает их с сервера и на этом всё заканчивается. Адрес сервера известен приложению. Если же речь идёт, например, о создании P2P-чата, работа которого основана на прямом соединении браузеров, адреса этих браузеров заранее неизвестны. В результате для того, чтобы установить P2P-соединение, придётся справиться с некоторыми проблемами.

Посмотрим как работает NAT. Если вы находитесь в корпоративной сети и подключились к WiFi, вашему компьютеру будет назначен IP-адрес, который существует только за вашим NAT-устройством. Предположим, что это — IP-адрес 172.0.23.4. Для внешнего мира, однако, ваш IP-адрес может выглядеть как 164.53.27.98. Внешний мир, в результате, видит ваши запросы как приходящие с адреса 164.53.27.98, но, благодаря NAT, ответы на запросы, выполненные вашим компьютером к внешним сервисам, будут отправлены на ваш внутренний адрес 172.0.23.4. Это происходит с использованием таблиц трансляций. Обратите внимание на то, что в дополнение к IP-адресу для организации сетевого взаимодействия нужен ещё и номер порта.

https://itnan.ru/post.php?c=1&p=304150

https://www.voip-info.org/stun/
https://zerotier.atlassian.net/wiki/spaces/SD/pages/7405571/Using+Linux+STUN+Tools+to+Characterize+NAT+Behavior

https://tailscale.com/blog/how-nat-traversal-works/