Задания квалификационного этапа.

1. Скачать и установить специальную версию SDK KasperskyOS для хакатона:

https://ds.mai307.ru/s/CrM38T9X2LtABnZ.
ВНИМАНИЕ! Стандартная версия, скачанная с сайта KasperskyOS, может быть использована для выполнения задания квалификационного этапа, но **не позволит выполнять задания основного этапа**!

SDK KasperskyOS может быть установлено на Debian GNU/Linux "Buster" 10 x64 или на Ubuntu 20.04 x64.
На другие версии Linux можно установить SDK KasperskyOS в Docker контейнер с Debian 10.
На Windows можно установить Debian 10 в Virtual Box.

KasperskyOS SDK включает:
— кросс систему разработки, которая позволяет собирать приложения
— KasperskyOS
Кросс система разработки включает компилятор gcc/g++, утилиты для сборки cmake, программу эмулятор qemu для запуска приложений KasperskyOS для архитектуры arm на машинах с архитектурой x86 и необходимый набор библиотек.
KasperskyOS CE может работать на raspberry pi 4 или в эмуляторе qemu, который входит в состав SDK.
Другие аппаратные платформы не поддерживаются.

Инструкция по установке KasperskyOS CE может быть найдена в курсе “Разработка для KasperskyOS”
https://stepik.org/course/73418/promo
или в онлайн документации:
https://click.kaspersky.com/?hl=en-us&link=online_help&pid=kos&version=1.1/

2. Разработать и запустить приложение, которое будет выводить в консоль "Hello from KasperskyOS"

В качестве примера возможно использовать приложение example\hello из SDK Kaspersky OS.
Собирающийся исходный код приложения долен быть выложен на github. Результаты работы программы в qemu должны быть скопированы и сохранены в отдельный файл results.txt и добавлены в git репозиторий.

Инструкция по запуску примера hello в KasperskyOS CE может быть найдена в курсе “Разработка для KasperskyOS” (https://stepik.org/course/73418/promo) или в онлайн документации (https://click.kaspersky.com/?hl=en-us&link=online_help&pid=kos&version=1.1/).

3. Скомпилировать и запустить в KasperskyOS пример mqtt_subscriber из состава SDK.

Установить и запустить на машине разработки с Debian linux MQTT-брокер mosquitto. Записать в топик строку “hello” при помощи mosquitto_pub. Получить данные из топика и отобразить их на экране.

Для установки mosquito можно воспользоваться командами:

apt-get install mosquitto
apt-get install mosquitto-clients

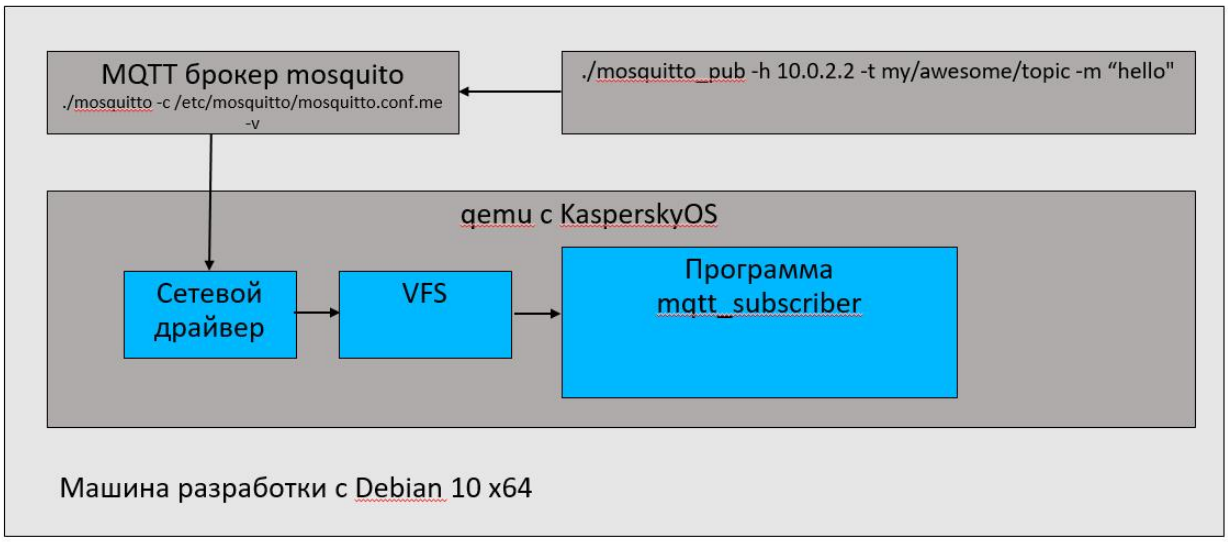

Рис.1. Взаимодействие приложений при запуске mqtt_subscriber

Желательно скорректировать конфигурационный файл для брокера mosquito /etc/mosquitto/mosquitto.conf для обеспечения возможности принимать входящие соединения.
Пример конфигурационного файла mosquitto.conf:

listener 1883 **10.0.2.2**
persistence false
persistence_location /path_to_persistent/mqtt
persistence_file mosquitto.db
log_dest syslog
log_dest stdout
log_dest topic
log_type error
log_type warning
log_type notice
log_type information
connection_messages true
log_timestamp true
allow_anonymous true
password_file /path_to_pwdfile/pwfile

Необходимо отметить, что в примере mqtt_subscriber подключение осуществляется к MQTT брокеру, расположенному по адрес 10.0.2.2. Таким образом необходимо установить статический IP адрес сетевой карты локальной машины в 10.0.2.2.
Информация по настройке сетевого взаимодействия между KasperskyOS, работающей в qemu, и приложением на машине разработке может быть найдена в презентации Михаила Демченко. Как сделать продукт на KasperskyOS:
https://www.youtube.com/watch?v=TvK-EfV2XJQ, временная метка 10:30

4. Выделить в примере mqtt_subscriber для KasperskyOS отображение принятых данных в отдельное приложение для KasperskyOS, связанное с исходным приложением по IPC.

Приложение subscriber должно принять целое число по MQTT и передать его в программу showapp при помощи IPC. Приложение showapp должно принять число по IPC и отобразить его на экран.

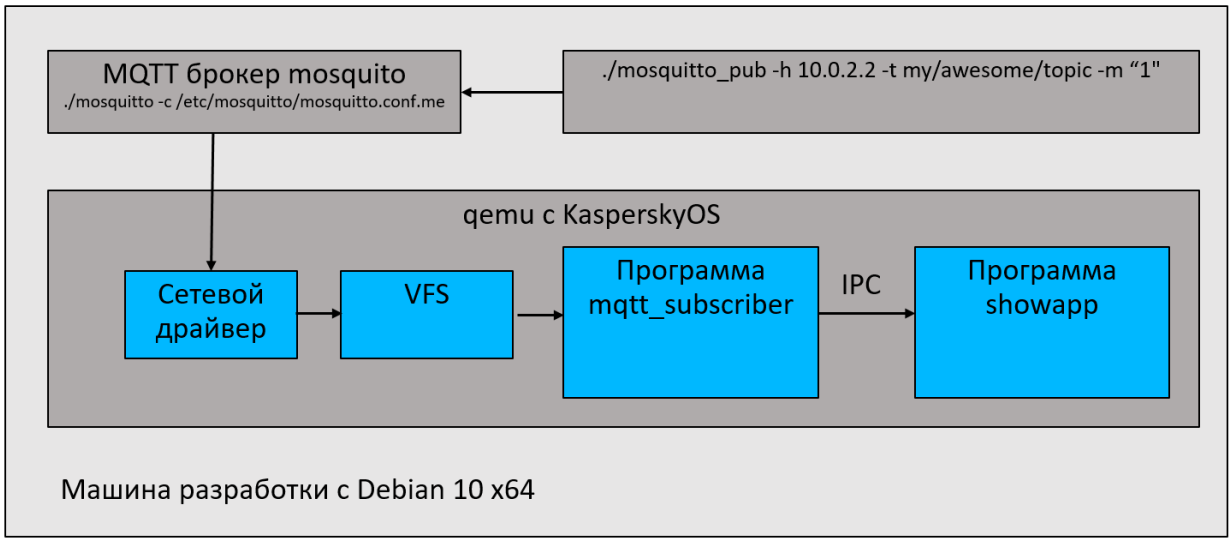

Рис.2. Взаимодействие приложений про запуске mqtt_subscriber и showapp

Для получения данных по IPC программа showapp может реализовать метода для получения целых чисел. В качестве примера работы с IPC возможно взять пример echo, входящий в состав SDK. Дополнительная информация по работе с IPC может быть найдена в онлайн документации https://click.kaspersky.com/?hl=en-us&link=online_help&pid=kos&version=1.1/.

Необходимо создать на github репозиторий для квалификационных заданий 2 и 4 и прислать ссылку на репозиторий на почту hackathon_307@mail.ru до 20:00 16.11.2022 включительно.

Обязательным для допуска к очному этапу хакатона является выполнение заданий 1 и 2. Выполнение заданий 3 и 4 дает по 5 дополнительных баллов в основной хакатон за каждое.

Задание основного этапа хакатона посвящено разработке программы траекторного управления роботом Alphabot. Потребуется работа с MQTT, IPC, GPIO и OpenCV в KasperskyOS. Задание основного этапа будет опубликовано 18.11.2022

Дополнительная информация

1. Руководство "KasperskyOS Community Edition":
https://support.kaspersky.com/help/KCE/1.1/ru-RU/KasperskyOS-CE.pdf
2. Бесплатный видеокурс "Разработка для KasperskyOS":
https://stepik.org/course/73418/promo
3. Видеолекция Михаила Демченко "Как сделать продукт на KasperskyOS":
https://www.youtube.com/watch?v=TvK-EfV2XJQ