ИТОГОВАЯ РАБОТА К МОДУЛЮ 3
ИНСТРУКЦИЯ

1. Для выполнения этого итогового задания можно выбрать один из двух способов:
    a. просканировать на открытые порты локальную виртуальную машину и вторую виртуальную машину
    b. просканировать на открытые порты локальную виртуальную машину и основную ОС
2. Для начала рассмотрим сканирование портов для двух виртуальных машин. Вторая виртуальная машина должна была создаваться из готового образа Image согласно инструкции Практического задания 2.5. Если же у вас на данный момент есть только одна виртуальная машина в Virtual Box, то вторую можно создать самым простым способом - Клонированием.
   Для этого нужно зайти в VirtualBox, выбрать виртуальную машину, которую нужно клонировать (Kali Linux), нажать на нее правой кнопкой мыши, выбрать “Клонироовать”, далее выбрать название для новой ВМ, а также место расположения (эти настройки можно оставить по-умолчанию), “Продолжить” и выбрать Полное клонирование.
   Далее дождаться создания клона виртуальной машины, процесс может занять некоторое время.
3. Чтобы виртуальные машины были доступны друг для друга, можно воспользоваться добавлением общего адаптера сети и указать IP адреса каждой виртуальной машины по отдельности. Инструкция по добавлению сетевого адаптера и выяснению адресов виртуальных машин также находится в Практическом задании 2.5

4. Теперь модернизируем одну из написанных в этом модуле программ (сканер состояния портов) под требуемые задачи.
   Для этого нам нужно, чтобы программа проверяла не один хост, а несколько

```
import socket
#сканирование всех IP-адресов
def scan_hosts(hosts, port_list):
    for host in hosts:
        scan_ports(host, port_list)
```

```
def scan_ports(host, port_list):
    print(f"Scan started. Host:{host}")

    for port in port_list:
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            result = sock.connect_ex((host, port))

            if result == 0:
                print(f"Port {port} is open")
            else:
                print(f"Port {port} is closed")

            sock.close()

        except socket.error:
            print(f"Could not connect to {host}:{port}")
    print("Scan is finished!")

hosts = ["127.0.0.1", "127.0.0.2"]  # сюда нужно вставить IP адрес сканируемых хостов, то есть либо вторую ВМ, либо основную ОС
port_list = [80, 443, 22, 3389]  # список портов (можно изменять для проверок)
scan_hosts(hosts,port_list)
```

Снимок экрана для проверки отступов:

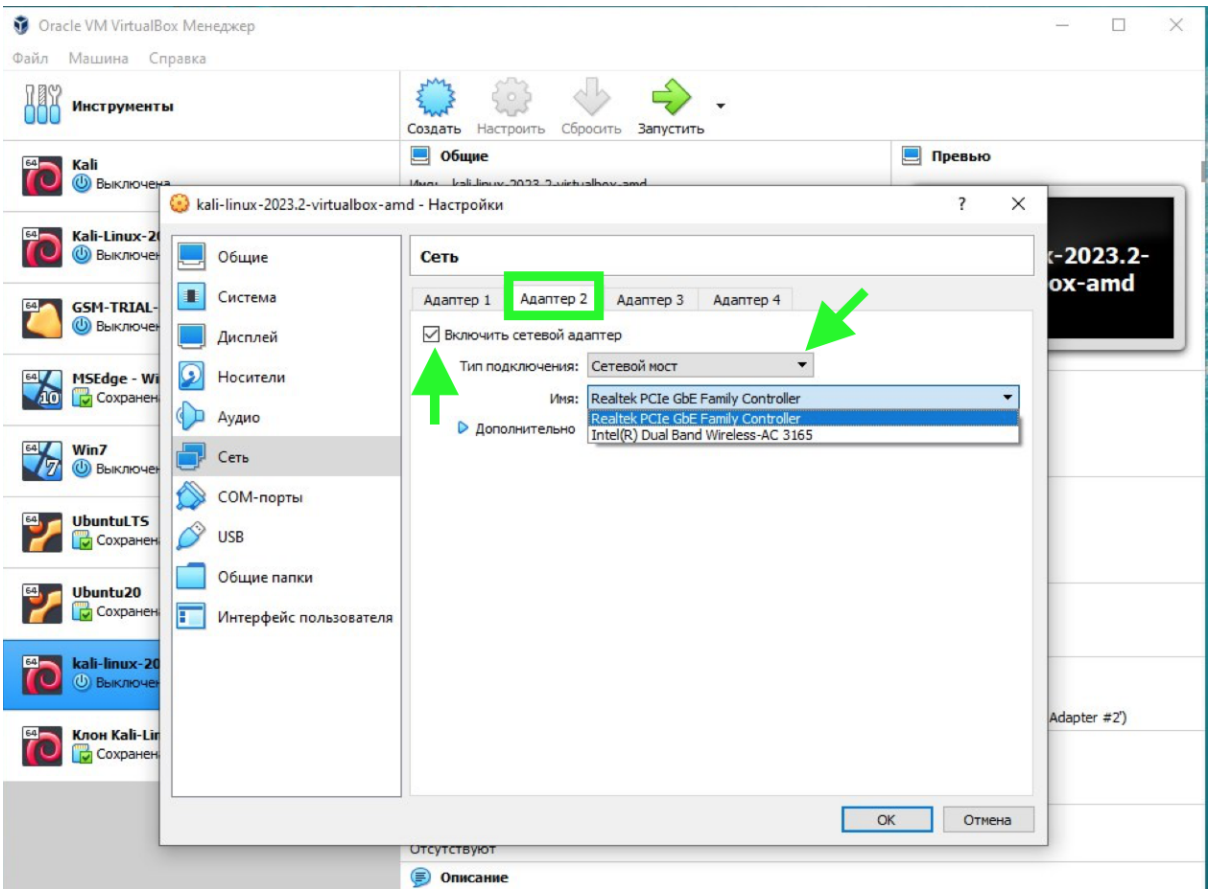
main.py

```python
import socket


1 usage
def scan_hosts(hosts, port_list):
    for host in hosts:
        scan_ports(host, port_list)
1 usage
def scan_ports(host, port_list):
    print(f"Scan started. Host:{host}")
    for port in port_list:
        try:
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            result = sock.connect_ex((host, port))

            if result == 0:
                print(f"Port {port} is open")
            else:
                print(f"Port {port} is closed")

            sock.close()

        except socket.error:
            print(f"Could not connect to {host}:{port}")
    print("Scan is finished!")
```

```python


hosts = ["127.0.0.1",
         "127.0.0.2"]
port_list = [80, 443, 22, 3389]
scan_hosts(hosts, port_list)
```

Эта программа формирует результаты сканирования по указанным хостам, проверяя нужные порты. В нее добавлена функция scan_hosts, которая отвечает за проверку каждого хоста по очереди, а в ней вызывается функция такой же проверки портов на каждом хосте. Результат выводится в консоль.

5. Теперь разберем случай, когда в качестве второго сканируемого хоста будет выступать основная Операционная система компьютера. Для этого нам нужно зайти в Virtual Box , в настройки нашей виртуальной машины Kali и подключить следующий сетевой адаптер:



Нужно включить сетевой адаптер и Типом подключения выбрать сетевой мост. (Скорее всего, имя адаптера выберется корректное по-умолчанию)

Далее зайти в виртуальную машину, выполнить команду $ifconfig и увидеть новый интерфейс с выделенным IP адресом. Этот IP адрес можно указать в коде программы как адрес локальной машины.

6. Теперь нужно узнать IP-адрес основной ОС. И если вы используете Windows, то нужно открыть Командную строку и ввести команду $ipconfig и команда выведет IP-адрес компьютера. Он будет в той же подсети как и выданный IP-адрес виртуальной машине.

   (то есть, например, 192.168.1.15 и 192.168.1.16)

   Именно эти IP адреса нужно внести в код программы, параметр hosts = ["127.0.0.1", "127.0.0.2"] и запустить сканирование. Программа проверит порты на основной ОС и на виртуальной машине.