****

Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет інформатики та обчислювальної техніки

Кафедра автоматики та управління в технічних системах

# Лабораторна робота №6 Основи використання Python для автоматизації тестування. Генерація мережевого трафіку за допомогою утиліти iperf

|  |  |  |
| --- | --- | --- |
| Виконав  студент групи ІА-92: |  | Перевірила: |
|  |  |  |
|  |  |  |
| Мешко І.С. |  | Гайдай |

Київ 2022

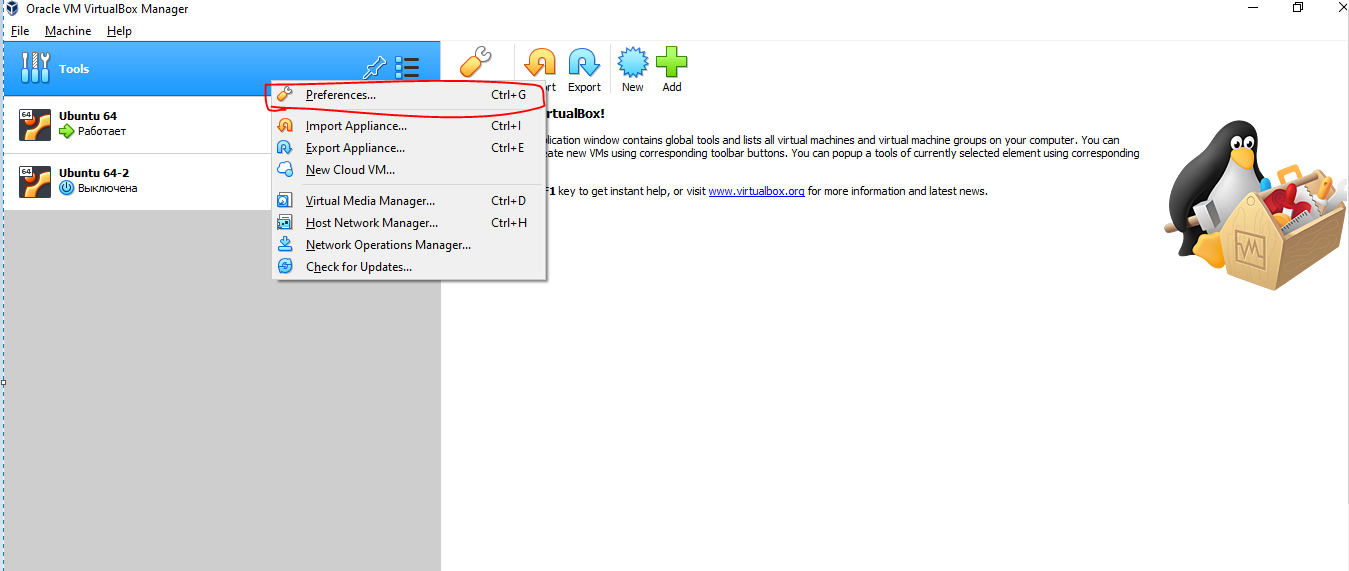
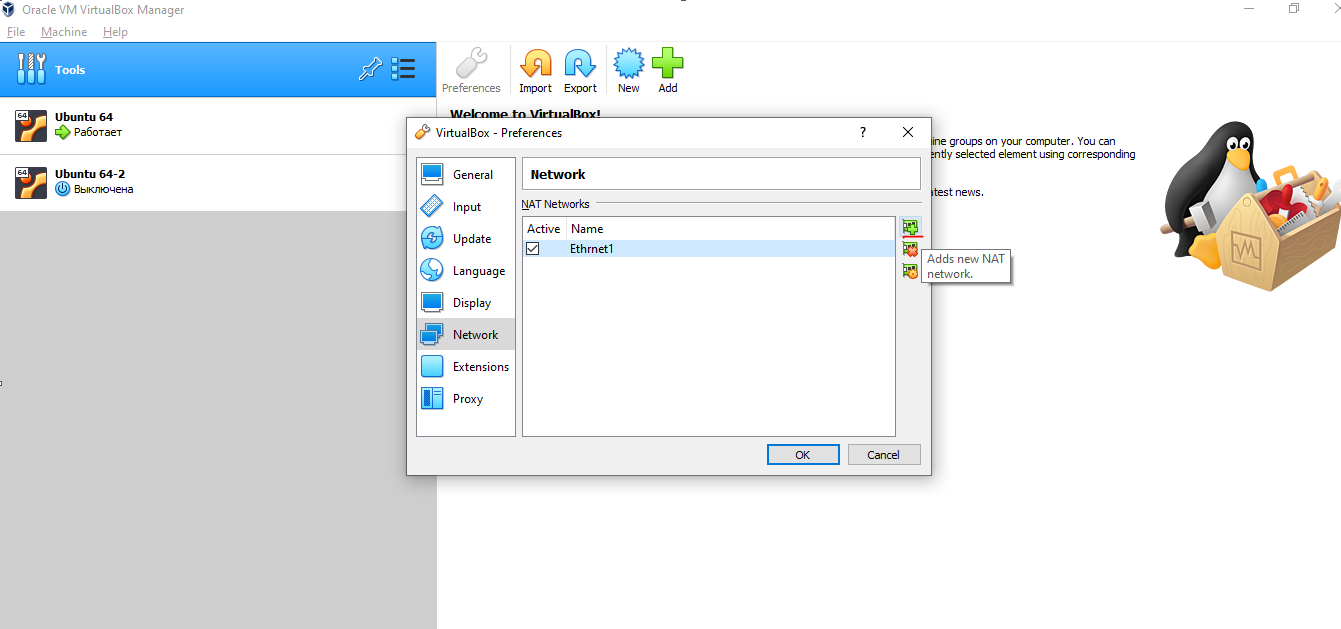
**Мета:** Ознайомитися з технологіями автоматизації тестування. Ознайомитися з

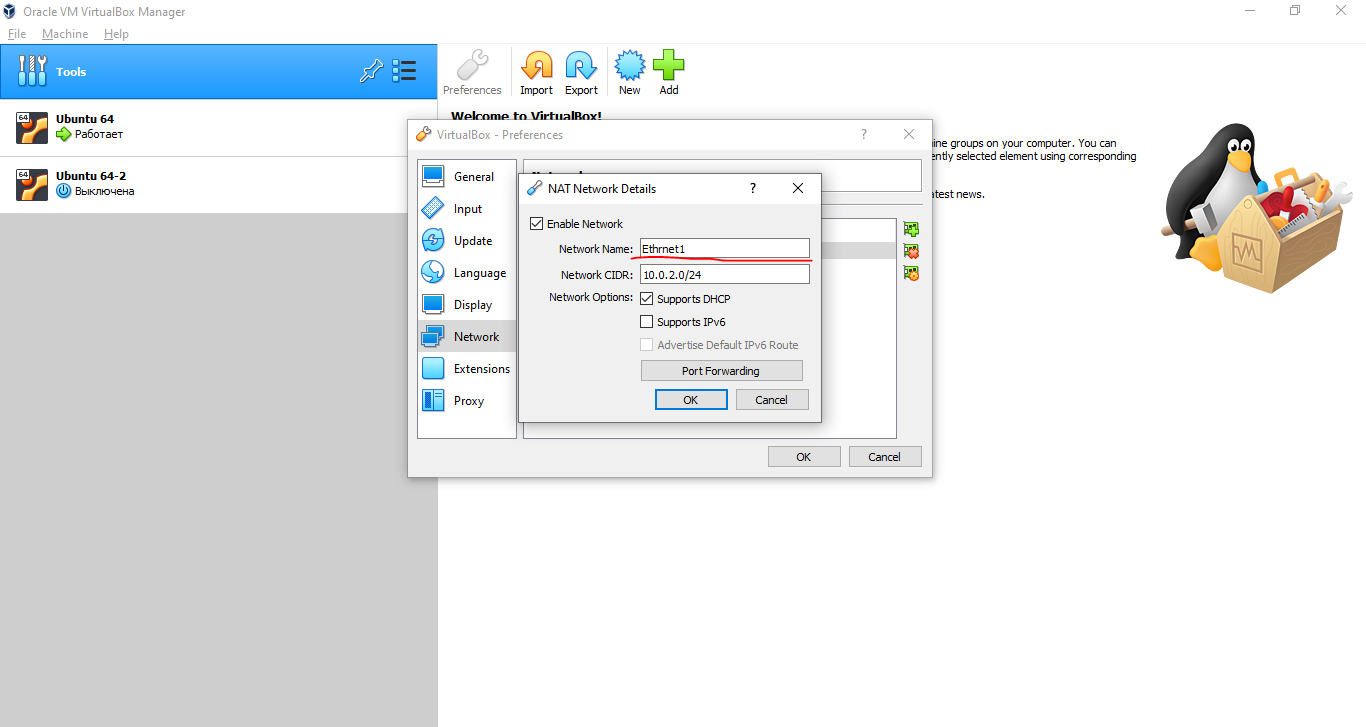
основами використання мови програмування Python для автоматизації

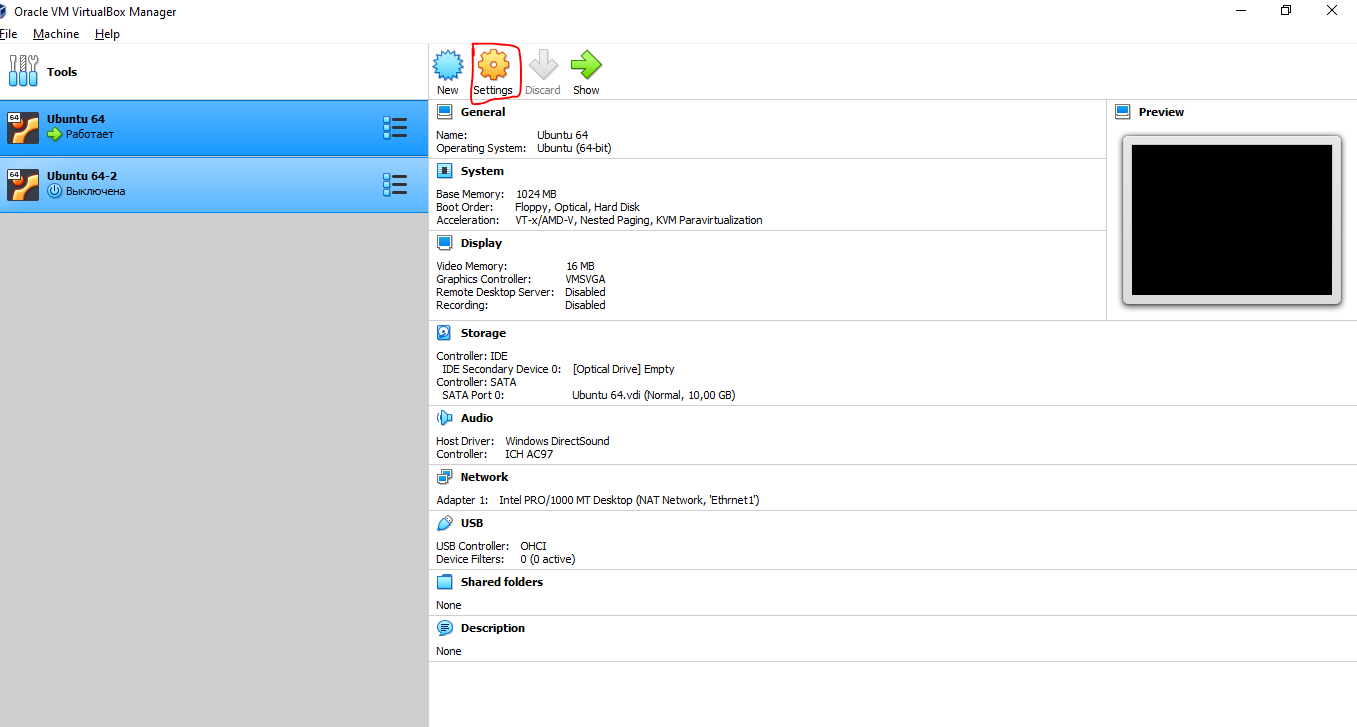
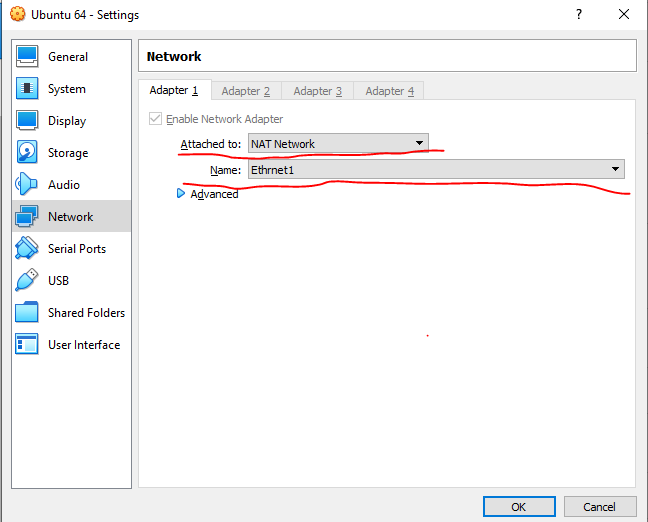
тестування.

**Тестова документація:**

**Налаштування мережі**Для виконання завдання лабораторної роботи, нам пострібно з’єднанайте два PC один з одним через інтерфейс NAT Network.  
Step:

1. У VirtualBox натисніть у верхній частині Tools правою кнопкою мищі та виберіть Preferences   
   
2. Далі, вам виведе вікно, після якого потрібно натисніть Nework та натиснути “Adds new NAT network”   
   
3. Після натискання у “Списку мереж” з’явиться нова мережа, у випадках перейменування, потрібно натиснути на неї двічі, і після редагування натисніть “OK”



1. Далі виберіть любу віртуальну машину, та натисніть Setting  
   
2. Після натискання перейдіть до вкладки Network, та у “Attached to”  
   Виберіть NAT Network   
   
3. Далі натисніть “OK”

### **Завдання на лабораторну роботу**

1) Проаналізувати трафік для сесії на ваш вибір (Skype, SMTP, ftp та ін.)

2) Розробити тестову документацію.

QA завдання:

*Main.py*

# Імпортуємо необхідний модуль, для виконання Linux команд із скрипта

import re

import subprocess

# Ip-адреса сервера

server\_ip = 'server'

def client():

# З підключенням до серверу допомагає модуль subprocess та клас subprocess.Popen.

pro = subprocess.Popen(

# -c – запуск клієнта та підключення до сервера, -i – інтервал між виведенням результату тестування,-u - підключення до протоколу UDP

        ['iperf', '-c', server\_ip, '-i', '1', '-u'],

# STDOUT (стандартний висновок)

stdout=subprocess.PIPE,

# STDERR (стандартний висновок помилок)

        stderr=subprocess.PIPE,

    )   
# wait() повертає ідентифікатор процесу та код завершення, упакований у 16-бітове значення.

    pro.wait()

# Повертая вивід та помилку.

    return pro.stdout.read().decode('utf8'), pro.stderr.read().decode('utf8')

def parser():

    # Парсер повертає список усіх інтервалів з iperf, щоб у подадьшому їх можна було проаналізувати.  
 keys = ['Interval', 'Transfer', 'Bandwidth']

    parsed\_result = []

    stdout, stderr = client()

    parser\_result\_interval = re.compile(r'[0-9].[0-9]{4}-[0-9].[0-9]{4}')

    parser\_result\_transfer = re.compile(r'[0-9]+\sKBytes')

    parser\_result\_bandwidth = re.compile(r'[0-9].[0-9]{2}\sMbits/sec')

    # Угрупованний список інтервалів

grouped\_result = list(

        zip(

            parser\_result\_interval.findall(stdout),

            parser\_result\_transfer.findall(stdout),

            parser\_result\_bandwidth.findall(stdout)

        )

    )

# Цикл з додаванням даних до parsed result

    for res in grouped\_result:

        parsed\_result.append({key: value for key, value in list(zip(keys, res))})

# список усіх інтервалів iperf

    return parsed\_result, stderr

# Використовується для виконання деякого коду, лише якщо файл було запущено безпосередньо, а не імпортовано.

if \_\_name\_\_ == '\_\_main\_\_':

    result, error = parser()

**Результати виконання практичного завдання**

Title: Аналіз трафіку для сесії UDP

Description: Переконайтеся, що клієнт підключаєтеся до iperf-серверу по протоколу UDP

Setup Description:  
PC1 ----etherne1 ---- PC2

PC1: 10.0.2.15

PC2: 10.0.2.7

Prerequisites:

1)Щоб дізнатися який IP у віртуальні машині потрібно в терміналі самій віртуальці ввести “ip -a”, та виберіть IP в залежності від мережі

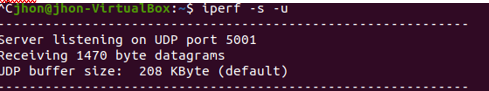
2)Для запуску python коду потрібно на PC2 в консолі ввести:

sudo apt install python

3) Основний код m*ain.py* повинен бути на головній сторінці

Steps:

1. На PC1 запустіть UDP iperf-сервер командою:  
   iperf -s -u



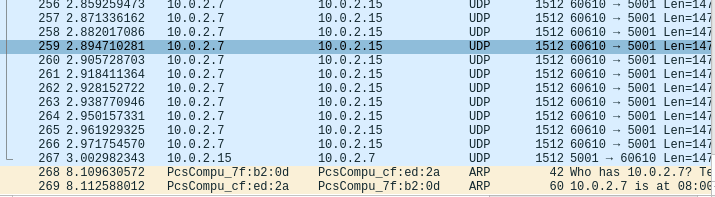
1. Далі запустіть на Wireshark на PC1:

sudo wireshark

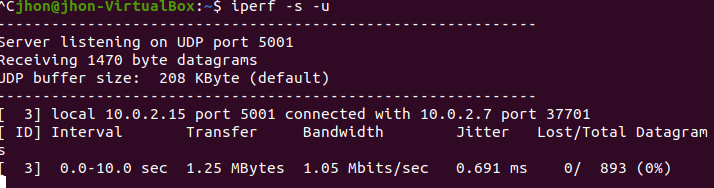
1. Після цього на PC2 запустіть main.py ввівши у консоль:

python main.py  


1. Перевіряємо сигнал UDP протоколу на PC1 у Wireshark:



1. Перевіряємо результат ітервалу на сервері



<https://github.com/Jou2323/QA-Lab6.git>

Висновок: У даній лабораторній роботі я зрозумів як робити автоматизації тестування. Та ознайомитися з основами використання мови програмування Python для автоматизації тестування.