

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

«Тестування та контроль якості (QA) вбудованих систем»

Лабораторна робота №6

**ОСНОВИ ВИКОРИСТАННЯ Python ДЛЯ АВТОМАТИЗАЦІЇ
ТЕСТУВАННЯ.
ГЕНЕРАЦІЯ МЕРЕЖНОГО ТРАФІКУ ЗА ДОПОМОГОЮ
УТИЛІТИ iperf**

Виконала:
студентка групи ІО-91
Тимошенко Діана

Київ
2022 р.

Мета: Ознайомитися з технологіями автоматизації тестування. Ознайомитися з основами використання мови програмування *Python* для автоматизації тестування. Написати функцію для підключення клієнта до сервера, використовуючи модуль *subprocess*.

Посилання на репозиторій:

<https://github.com/diana-tym/qa-labs>

Виконання лабораторної роботи

Налаштування мережного оточення

```
import subprocess
import re

server_ip = '192.168.0.105'
regex = '[0-9.-]*\s*sec\s*[0-9.]*\s*[A-Z]?Bytes\s*[0-9.]*\s*[A-Z]?bits/sec\s*[0-9]' \
        '*\s*[0-9.]*\s*[A-Z]?Bytes'
keys = ['Interval', 'Transfer', 'Bitrate', 'Retr', 'Cwnd']

def client(server_ip):
    p = subprocess.Popen(['iperf3', '-c', server_ip], stdout=subprocess.PIPE,
stderr=subprocess.PIPE, text=True)
    result, error = p.communicate()
    return result, error

def parser(result):
    iperf_results = []
    parsed_list = re.findall(regex, result)
    for i in parsed_list:
        values = []
        values.append(re.search('[0-9.-]*(?=\s*sec)', i).group())
        values.append(float(re.search('[0-9.]*(?=\s*[A-Z]?Bytes)', i).group()))
        values.append(float(re.search('[0-9.]*(?=\s*[A-Z]?bits/sec)', i).group()))
        values.append(float(re.search('\s[0-9]\s', i).group()))
        values.append(float(re.search('[0-9.]*(?=\s*[A-Z]?Bytes\s*$)', i).group()))
        iperf_results.append(dict(zip(keys, values)))
    return iperf_results

result, error = client(server_ip)

if not error:
    iperf_results = parser(result)
    for value in iperf_results:
        if value['Transfer'] > 2 and value['Bitrate'] > 20:
            print(value)
else:
    print(error)
```

Результати

```
diana@diana-X510UAR:~$ iperf3 -s
-----
Server listening on 5201
-----
Accepted connection from 192.168.0.105, port 58518
[ 5] local 192.168.0.105 port 5201 connected to 192.168.0.105 port 58532
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-1.00 sec      5.86 GBytes    50.3 Gbits/sec
[ 5] 1.00-2.00 sec      6.13 GBytes    52.6 Gbits/sec
[ 5] 2.00-3.00 sec      6.17 GBytes    53.0 Gbits/sec
[ 5] 3.00-4.00 sec      6.16 GBytes    52.9 Gbits/sec
[ 5] 4.00-5.00 sec      6.17 GBytes    53.0 Gbits/sec
[ 5] 5.00-6.00 sec      5.80 GBytes    49.8 Gbits/sec
[ 5] 6.00-7.00 sec      6.20 GBytes    53.3 Gbits/sec
[ 5] 7.00-8.00 sec      6.22 GBytes    53.4 Gbits/sec
[ 5] 8.00-9.00 sec      6.20 GBytes    53.3 Gbits/sec
[ 5] 9.00-10.00 sec     6.24 GBytes    53.6 Gbits/sec
[ 5] 10.00-10.04 sec    259 MBytes     52.4 Gbits/sec
-----
[ ID] Interval            Transfer        Bitrate
[ 5] 0.00-10.04 sec     61.4 GBytes    52.5 Gbits/sec
receiver
```

```
diana@diana-X510UAR:~/qa/lab6$ python3 lab6.py
{'Interval': '0.00-1.00', 'Transfer': 6.11, 'Bitrate': 52.5, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '1.00-2.00', 'Transfer': 6.13, 'Bitrate': 52.6, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '2.00-3.00', 'Transfer': 6.15, 'Bitrate': 52.8, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '3.00-4.00', 'Transfer': 6.17, 'Bitrate': 53.0, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '4.00-5.00', 'Transfer': 6.13, 'Bitrate': 52.7, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '5.00-6.00', 'Transfer': 5.83, 'Bitrate': 50.1, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '6.00-7.00', 'Transfer': 6.2, 'Bitrate': 53.2, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '7.00-8.00', 'Transfer': 6.23, 'Bitrate': 53.5, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '8.00-9.00', 'Transfer': 6.2, 'Bitrate': 53.3, 'Retr': 0.0, 'Cwnd': 2.19}
{'Interval': '9.00-10.00', 'Transfer': 6.25, 'Bitrate': 53.7, 'Retr': 0.0, 'Cwnd': 2.19}
```

Помилка в підключенні

```
diana@diana-X510UAR:~/qa/lab6$ python3 lab6.py
iperf3: error - unable to send control message: Bad file descriptor
```

Опис скрипта у вигляді тест-кейсу

Test ID: TC_iperf_1

Summary: Перевірити пропускну здатність домашньої мережі.

Attachments: lab6.py, iperf_server.png, results.png

Setup description:



Preconditions: Запущений iperf-сервер на PC1.

Steps:

1. Підключитися до iperf-серверу на PC1 з PC1 за допомогою модулю subprocess та зафіксувати вивід результату або повідомлення про помилку.

Команда підключення: `iperf3 -c <PC1_IP>`

ER: Отримання результату тестування або повідомлення про помилку.

2. Вивести повідомлення про помилку з п.1 (якщо така є).

ER: Відображення повідомлення про помилку.

3. Зробити парсинг результатів тестування з п.1 (якщо вони були отримані).

ER: Отримання списку словників зі значеннями результату iperf тестування на кожному інтервалі.

4. Відфільтрувати результати iperf тестування з п.3 (якщо вони були отримані) за умовами `Transfer > 2 GBytes` та `Bitrate > 20 GBits/sec`.

ER: Відображення результатів, які відповідають вимогам.

Висновок

В цій лабораторній роботі ми познайомились з модулем subprocess, який дозволяє запускати зовнішні програми та переглядати їхній вивід. За допомогою нього ми автоматизували підключення клієнта до iperf-сервера. Також ми написали функцію для парсингу результатів та відфільтрували їх. Написаний скрипт ми описали у тест-кейсі.