

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 №7

**ОСНОВИ ВИКОРИСТАННЯ ФРЕЙМВОРКУ *Pytest* ДЛЯ
АВТОМАТИЗАЦІЇ ТЕСТУВАННЯ**

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

confest.py

```
import paramiko
import subprocess
import pytest
import time

server_ip = '192.168.0.105'
password = 'root'
username = 'diana'

@pytest.fixture(scope='function')
def server():
    client_ssh = paramiko.SSHClient()
    client_ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
    client_ssh.connect(server_ip, username=username, password=password,
look_for_keys=False)
    _, _, stderr = client_ssh.exec_command('iperf3 -s -1')

    time.sleep(0.5)
    error = None
    if (stderr.channel.exit_status_ready()):
        error = stderr.read().decode()

    client_ssh.close()
    return error

@pytest.fixture(scope='function')
def client(server):
    time.sleep(0.5)
    p = subprocess.Popen(['iperf3', '-c', server_ip], stdout=subprocess.PIPE,
stderr=subprocess.PIPE, text=True)
    result, error = p.communicate()
    return result, error, server
```

parser.py

```
import re

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 parse(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

```

iperf_test.py
import parser

```

class TestSuite():
    def test_iperf3(self, client):
        stdout, error_cl, error_serv = client

        print(f'Server error: {error_serv}')
        print(f'Client error: {error_cl}')
        print(f'Results: {stdout}')

        assert not error_cl
        iperf_results = parser.parse(stdout)
        for value in iperf_results:
            assert value['Transfer'] > 2 and value['Bitrate'] > 20

```

Результати тестування

Позитивний кейс

```

diana@diana-X510UAR:~/qa/lab7$ python3 -m pytest iperf_test.py
===== test session starts =====
platform linux -- Python 3.10.6, pytest-7.2.0, pluggy-0.13.0
rootdir: /home/diana/qa/lab7, configfile: pytest.ini
collected 1 item

iperf_test.py . [100%]

===== 1 passed in 10.99s =====

```

Негативні кейси

```

diana@diana-X510UAR:~/qa/lab7$ python3 -m pytest iperf_test.py
===== test session starts =====
platform linux -- Python 3.10.6, pytest-7.2.0, pluggy-0.13.0
rootdir: /home/diana/qa/lab7, configfile: pytest.ini
collected 1 item

iperf_test.py F [100%]

===== FAILURES =====
TestSuite.test_iperf3

self = <iperf_test.TestSuite object at 0x7f87f85a9cc0>
client = ('', 'iperf3: error - unable to send control message: Bad file descriptor\n', 'iperf3: parameter error - some option you are trying to set is client o
nly\n\n')

    def test_iperf3(self, client):
        stdout, error_cl, error_serv = client

        print(f'Server error: {error_serv}')
        print(f'Client error: {error_cl}')
        print(f'Results: ', stdout)

>         assert not error_cl
E         AssertionError: assert not 'iperf3: error - unable to send control message: Bad file descriptor\n'

iperf_test.py:11: AssertionError
----- Captured stdout call -----
Server error: iperf3: parameter error - some option you are trying to set is client only

Client error: iperf3: error - unable to send control message: Bad file descriptor

Results:
===== short test summary info =====
FAILED iperf_test.py::TestSuite::test_iperf3 - AssertionError: assert not 'iperf3: error - unable to send control message: Bad file descriptor\n'

```

```

diana@diana-XS10UAR: ~/qa/lab7$ python3 -m pytest iperf_test.py
===== test session starts =====
platform linux -- Python 3.10.6, pytest-7.2.0, pluggy-0.13.0
rootdir: /home/diana/qa/lab7, configfile: pytest.ini
collected 1 item

iperf_test.py F [100%]

===== FAILURES =====
TestSuite.test_iperf3

self = <iperf_test.TestSuite object at 0x7f8492f99d20>
client = ('Connecting to host 192.168.0.105, port 5201\n[ 5] local 192.168.0.105 port 49756 connected to 192.168.0.105 port 52... sender\n[ 5] 0.00-10.0
4 sec 62.6 GBytes 53.6 Gbits/sec receiver\n\niperf Done.\n', None)

    def test_iperf3(self, client):
        stdout, error_cl, error_serv = client

        print(f'Server error: {error_serv}')
        print(f'Client error: {error_cl}')
        print(f'Results: ', stdout)

        assert not error_cl
        iperf_results = parser.parse(stdout)
        for value in iperf_results:
            > assert value['Transfer'] > 7 and value['Bitrate'] > 20
E           assert (6.3 > 7)

iperf_test.py:14: AssertionError
Server error: None
Client error:
Results: Connecting to host 192.168.0.105, port 5201
[ 5] local 192.168.0.105 port 49756 connected to 192.168.0.105 port 5201
ID Interval Transfer Bitrate Retr Cwnd
[ 5] 0.00-1.00 sec 6.30 GBytes 54.1 Gbits/sec 0 1.69 MBytes
[ 5] 1.00-2.00 sec 6.32 GBytes 54.3 Gbits/sec 0 1.69 MBytes
[ 5] 2.00-3.00 sec 6.27 GBytes 53.9 Gbits/sec 0 1.69 MBytes
[ 5] 3.00-4.00 sec 6.28 GBytes 53.9 Gbits/sec 0 1.69 MBytes
[ 5] 4.00-5.00 sec 6.27 GBytes 53.9 Gbits/sec 0 1.69 MBytes
[ 5] 5.00-6.00 sec 6.22 GBytes 53.5 Gbits/sec 0 1.69 MBytes
[ 5] 6.00-7.00 sec 6.21 GBytes 53.3 Gbits/sec 0 1.69 MBytes
[ 5] 7.00-8.00 sec 6.21 GBytes 53.3 Gbits/sec 0 1.69 MBytes
[ 5] 8.00-9.00 sec 6.26 GBytes 53.8 Gbits/sec 0 1.69 MBytes
[ 5] 9.00-10.00 sec 6.26 GBytes 53.8 Gbits/sec 0 1.69 MBytes
-----
ID Interval Transfer Bitrate Retr
[ 5] 0.00-10.00 sec 62.6 GBytes 53.8 Gbits/sec 0 sender
[ 5] 0.00-10.04 sec 62.6 GBytes 53.6 Gbits/sec 0 receiver

iperf Done.

===== short test summary info =====
FAILED iperf_test.py::TestSuite::test_iperf3 - assert (6.3 > 7)
===== 1 failed in 11.55s =====

```

Висновок

У цій лабораторній роботі ми познайомились з модулем для тестування `pytest` та модулем для роботи з протоколом `ssh paramiko`. Потім ми написали дві фікстури. В першій ми підключаємось до хосту по `ssh` та запускаємо на ньому `iperf`-сервер. В другій за допомогою модулю `subprocess` ми підключились до `iperf`-серверу. З попередньої лабораторної ми використали функцію `parse` для парсингу результатів `iperf`. Далі написаний тест-кейс аналізує результати, перевіряючи на помилку та відповідність результатів певним значенням.