

# Zaawansowane techniki programowania

*Laboratorium 5*



**Politechnika Krakowska**  
**im. Tadeusza Kościuszki**

**Piotr Jurek**

27.03.2023

Systemy Inteligentne i Rozszerzona Rzeczywistość



[Repozytorium z zadaniem](#)

## WPROWADZENIE

W laboratorium zostały omówione różne aspekty komunikacji sieciowej oraz programowania sieciowego w języku Python.

## MATERIAŁY

Do wykonania zadania wykorzystano dołączone pliki:

- tcp\_sixteen.py
- deadlock.py
- ping\_www.py
- dns\_001.py
- dns\_mx.py

## PROCEDURA

Poniżej przedstawiono krótkie podsumowanie każdego z zadań.

1. W zadaniu tym, oparto się na pliku tcp\_sixteen.py, który zawierał prosty kod do komunikacji między klientem a serwerem za pomocą protokołu TCP. Program został zmodyfikowany tak, aby przysyłał dane klienta (nazwisko, imię, nazwę grupy i numer indeksu). Wyniki uruchomienia programu po stronie serwera wyświetlały otrzymane dane, a wyniki uruchomienia programu po stronie klienta potwierdzały udane połączenie.

```
> python3 assignment1.py server localhost
Listening at ('127.0.0.1', 1060)
Waiting to accept a new connection
We have accepted a connection from ('127.0.0.1', 55690)
  Socket name: ('127.0.0.1', 1060)
  Socket peer: ('127.0.0.1', 55690)
{'name': 'Piotr', 'surname': 'Jurek', 'group': 'SIiRZ', 'index_no':
'111111'}
Waiting to accept a new connection
> python3 assignment.py client localhost
Client has been assigned socket name ('127.0.0.1', 55690)
```

2. W zadaniu tym przeanalizowano plik `deadlock.py` i zademonstrowano, jak w przypadku dużej ilości przesyłanych danych dochodzi do zakleszczenia. Wynika to z faktu, że klient wysyła dane po 16 bajtów, a serwer odczytuje je w porcjach 1024 bajtów. Jeśli klient wyśle więcej niż 1024 bajty, zanim serwer zdąży je odczytać, to serwer stanie w miejscu. Problem ten można naprawić, umieszczając kod odczytujący dane przez serwer w pętli.

```
> python3 deadlock.py server localhost
Listening at ('127.0.0.1', 1060)
Processing up to 1024 bytes at a time from ('127.0.0.1', 55996)
  1000000 bytes processed so far
Socket closed
Processing up to 1024 bytes at a time from ('127.0.0.1', 56004)
  2835744 bytes processed so far
> python3 deadlock.py client localhost 10000000
Sending 1000000 bytes of data, in chunks of 16 bytes
  1000000 bytes sent
Receiving all the data the server sends back
  The first data received says b'CAPITALIZE THIS!CAPITALIZE
THIS!CAPITALIZE'
  1000000 bytes received
Sending 10000000 bytes of data, in chunks of 16 bytes
  5440816 bytes sent
```

3. W zadaniu tym skorzystano z gotowego programu `ping_www.py`, który wykonuje krótkie połączenie z dowolnym serwerem w sieci WAN na porcie 80. Program zakończył się powodzeniem dla próby połączenia z serwerem 1.1.1.1.

```
> python3 ping_www.py 1.1.1.1
Success: host 1.1.1.1 is listening on port 80
```

4. W zadaniu tym, za pomocą biblioteki `dns.resolver` i protokołu DNS w programie `dns_001.py`, rozwiązano nazwę DNS przykładowej domeny. Wyniki pokazały adresy IPv4 i IPv6, typ rekordu MX oraz nazwy serwerów DNS.

```
> python3 dns_001.py example.com
example.com. 31219 IN A 93.184.216.34
example.com. 86336 IN AAAA 2606:2800:220:1:248:1893:25c8:1946
example.com. 86337 IN MX 0 .
example.com. 18560 IN NS a.iana-servers.net.
example.com. 18560 IN NS b.iana-servers.net.
```

5. W zadaniu tym, za pomocą programu `dns_mx` i biblioteki `dns.resolver`, rozwiązano nazwę domenową hosta poczty e-mail. Wynik wyświetlił listę serwerów poczty e-mail dla domeny Gmail.com.

```
> python3 dns_mx.py gmail.com
This domain has 5 MX records
Priority 5
  gmail-smtp-in.1.google.com has A address 209.85.233.26
Priority 10
  alt1.gmail-smtp-in.1.google.com has A address 142.250.157.27
Priority 20
  alt2.gmail-smtp-in.1.google.com has A address 173.194.203.27
Priority 30
  alt3.gmail-smtp-in.1.google.com has A address 142.250.141.27
Priority 40
  alt4.gmail-smtp-in.1.google.com has A address 142.250.115.26
```

## PODSUMOWANIE

Podsumowując, laboratorium pozwoliło zapoznać się z podstawami programowania sieciowego w języku Python oraz pokazało, jak korzystać z różnych narzędzi do komunikacji sieciowej, takich jak protokół TCP lub DNS.