

Laboratorium **Programowanie w języku Python 2**
Wydział Elektrotechniki Automatyki I Informatyki
Politechnika Świętokrzyska

Studia: Stacjonarne I stopnia	Kierunek: Informatyka
Data wykonania: 29.04.2021	Grupa: 3ID16B
Imię i nazwisko: Arkadiusz Więclaw	Temat ćwiczenia: Programowanie sieciowe Biblioteki systemowe.

Zad 1:

Serwer:

```
import socket, sys

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
HOST = 'localhost'
PORT = 40000
server_address= (HOST, PORT)
print("Serwer wystartował na adresie {} i porcie  
{ {}".format(*server_address))
sock.bind(server_address)

while True:
    print("Serwer czeka na otrzymanie wiadomosc ")
    data, address = sock.recvfrom(4096)
    print("Serwer otrzymał wiadomosc {} o rozmiarze  
{ {}".format(address, len(data)))
    print("Tresc wiadomosc: {} ".format(data))
    if data:
        odp_serwera = b"Odpowiadam "
        sent = sock.sendto(odp_serwera, address)
        print("Serwer wysłał wiadomosc do {}".format(address))
```

Client:

```
import socket, sys

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
HOST = 'localhost'
PORT = 40000
server_address = (HOST, PORT)
wiadomosc = b" Klienta wiadomosc to jest"
try:
    print("Klient wysyłała wiadomosc do serwera")
    sent = sock.sendto(wiadomosc, server_address)
    print("Klient czeka na wiadomosc od serwera")
    data, server = sock.recvfrom(4096)
    print("Klienta otrzymał wiadomosc od serwra")
```

```
    print('Tresc wiadomosc: {}'.format(data))
finally:
    print("Klient zakonczyl komunikacje z serwerem")
    sock.close()
```

Wyniki:

Klient wysylala wiadomosc do serwera

Klient czeka na wiadomosc od serwera

Klienta otrzymal wiadomosc od serwera

Tresc wiadomosc: b'Odpowiadam '

Klient zakonczyl komunikacje z serwerem

Zad 2:

Serwer:

```
import socket, sys
sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
IP = 'localhost'
PORT = 40002
server_address = (IP, PORT)
print("Serwer wystartowal {} na porcie {}".format(*server_address))
sock.bind(server_address)
sock.listen(1)
while True:
    print('Oczekiwanie na połączenie ze stroną klienta')
    connection, client_address = sock.accept()
    try:
        print("Klient polaczyl sie z serwerem z ", client_address)
        while True:
            data = connection.recv(16)
            print('Serwer otrzymal wiadomosc od klienta "%s"' % data)
            if data:
                print('Wysłano wiadomosc zwrotna do klienta')
                connection.sendall(data)
            else:
                print('Nie ma żadnych wiadomosc od klienta',
client_address)
                break
```

```
finally:
    print("Klient zakonczyl prace")
    connection.close()
```

Client:

```
import sys, socket

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
IP = 'localhost'
PORT = 40002
server_address = (IP, PORT)
print("Klient polaczyl z serwerem {} o porcie
{}".format(*server_address))
sock.connect(server_address)
try:
    wiadomosc_kl = b'Klient wita'
    print("Klient wyslal wiadomosc do serwera ")
    sock.sendall(wiadomosc_kl)
    amount_received = 0
    amount_expected = len(wiadomosc_kl)
    while amount_received < amount_expected:
        data = sock.recv(16)
        amount_received += len(data)
        print('Klient otrzymał wiadomosc "%s"' % data)
finally:
    print('Klient zakonczyl komunikacje z serwerem')
    sock.close()
```

Wyniki:

Klient polaczyl z serwerem localhost o porcie 40002

Klient wyslal wiadomosc do serwera

Klient otrzymał wiadomosc "b'Klient wita'"

Klient zakonczyl komunikacje z serwerem

Zad 3:

```
import urllib.request
import urllib.parse

url ="https://www.netflix.com"
try:
```

```

response = urllib.request.urlopen(url)
print("Zawartosc url: ", response.read())
urlParse = urllib.parse.urlparse(url)
print("\n\nWyjście strony po sparsowaniu", urlParse)
print("\n\nZwrocony Naglowek strony ", response.info())
print("\nAdres url z odpowiedzi strony ", response.geturl())
except urllib.error.HTTPError as e:
    print("Problemy z adresem url {} {}".format(e.code, e.reason))

```

Wyniki:

```

Zawartosc url:  b'<!doctype html><html lang="en"><head><meta http-
equiv="Content-Type" content="text/html; charset=UTF-8"/><meta
charSet="utf-8"/><meta http-equiv="X-UA-Compatible"
content="IE=edge"/><meta http-equiv="origin-trial" data-feature="EME
Extension - Policy Check" data-expire="2018-11-26"
content="Aob+++752GiUzm1RNSIkM9TINnQDxTlxz02v8hFJK/uG02hmXnJqH8c/ZpI05
b2nLsHDhG03Ce2zXJUFQm07jA4AAAB1eyJvcmlnaW4iOiJodHRwczovL25ldGZsaXguY29
tOjQ0MyIsImZlYXR1cmUiOiJFbmlNyeXB0ZW50ZW50ZW50ZW50ZW50ZW50ZW50ZW50ZW50
GlyeSI6MTU0MzI0MzQyNCwiaXNTdWJkb21haW4iOnRydWV9"/><title>Netflix
Poland - Watch TV Shows Online, Watch Movies Online</title><link
rel="preload"
href="https://codex.nflxext.com/%5E3.0.0/truthBundle/webui/1.22.5-
shakti-js-
v5a12e10c/js/js/bootstrap.js,common%7Cbootstrap.js/2/0c3j032V2M062I303
i080n003k2_3e2Y3d2031340b023f00/bck/true/none" as="script"/><link
rel="preload"
href="https://codex.nflxext.com/%5E3.0.0/truthBundle/webui/1.22.5-
shakti-js-
v5a12e10c/js/js/signup%7Chome%7Cstory%7Cfuji%7CfujiFrameworkClient.
js/2/0c3j032V2M062I303i080n003k2_3e2Y3d2031340b023f00/1/true/none"
as="script"/><link type="text/css" rel="stylesheet"
href="/personalization/cl2/freeform/WebsiteDetect?source=wwwhead&f
etchType=css&modalView=nmLanding" data-ua="botLink"/><script
type="text/javascript">(function () { var request = new
XMLHttpRequest(); request.open('GET',
'/personalization/cl2/freeform/WebsiteDetect?source=wwwhead&fetchType
=js&modalView=nmLanding', true); request.send
...itd

```

```

Wyjście strony po sparsowaniu ParseResult(scheme='https',
netloc='www.netflix.com', path='', params='', query='', fragment='')

```

Zwrocony Naglowek strony Server: nq_website_nonmember-prod-release
0504ea06-9308-4497-8d89-4645de745bb7

x-frame-options: DENY

content-type: text/html; charset=utf-8

cache-control: no-cache, no-store, must-revalidate

pragma: no-cache

expires: 0

x-robots-tag: index

date: Mon, 10 May 2021 20:56:37 GMT

x-envoy-upstream-service-time: 201

x-b3-traceid: 755f903ea9de2374

x-request-id: 75b9cf98-7167-49f2-bacc-354a3ee3b30f

x-envoy-decorator-operation: lo_svc

transfer-encoding: chunked

Via: 1.1 i-01e489dbae26098a0 (eu-west-1)

X-Xss-Protection: 1; mode=block;
report=https://www.netflix.com/ichnaea/log/freeform/xssreport

X-Content-Type-Options: nosniff

Strict-Transport-Security: max-age=31536000

X-Originating-URL: http://www.netflix.com/pl-en/

Set-Cookie: flwssn=f81ba101-c800-4692-921d-76fe30c27e27;
Domain=.netflix.com; Path=/; Expires=Mon, 10 May 2021 23:56:36 GMT

Set-Cookie: clSharedContext=ebe311be-1550-45c5-9ded-b21561141cd8;
Domain=.netflix.com; Path=/

Set-Cookie: nfvdid=BQFmAAEBEBh0hRnH9KfFs04UeHV2gVRA17-
w8YxYkKbLCWnqzmFK4vGvXUuOgc_9Fycm05UG38Rf0x203aiV2vH3n6joNlaFVSgvL0G3C
G08uctrISZVdA%3D%3D; Domain=.netflix.com; Path=/; Max-Age=31536000

Set-Cookie: SecureNetflixId=v%3D2%26mac%3DAQEAEQABABTFmZVRAo-
BZ8qk9dXNkr0e-M6ZhOhJVLo.%26dt%3D1620680197070; Domain=.netflix.com;

Expires=Tue, 10 May 2022 20:56:37 GMT; Path=/; HttpOnly; Secure;
SameSite=Strict

Set-Cookie: NetflixId=v%3D2%26ct%3DBQA0AAEBEDE7J0r6PWUTUtELjC-
_uF6BAM0pKOEyaUgv-
wfh9MDw8V0wc9WVVEDUgnCkN4oZSCPDw9r2hPEPP3TGGqxrTZKshigIlaf-
vvwH1F84jejxed8eHaqXgZvehGu1AR6boDet-z1gqg9t6ZuFyUPHV9PSzVU6WK4Bq-
iSz7fdYTQ_sTN6p3BgfldXUXNgfYwKytFBDOP05JK_QY7BeigHlfng9c0ZaQ004Vypo6-
X9dtQnFLSvpZ7gqWnPpav7piAHM3mj-
dk59kYpr0BMT1PwgdTfSnEUVanSdZ1sUJ1IikFiVEkZf6PvqF4GMpnL91XtiYeIehPcZ18
wdTob0axsfhqdyU0PnnUVu2lcTi3TDNq2rs.%26bt%3Ddev%26mac%3DAQEAEABABT1oL
46XQVlFTZjm7NanJPPmcr1vWnzoZk.; Domain=.netflix.com; Expires=Tue, 10
May 2022 20:56:37 GMT; Path=/; HttpOnly; Secure; SameSite=Lax

X-Netflix.nfstatus: 1_1

Set-Cookie: memclid=15cdfc08-10ef-4f4b-8388-70ff1a10bfea; Max-
Age=31536000; Expires=Tue, 10 May 2022 20:56:37 GMT; Path=/;
Domain=.netflix.com

X-Netflix.proxy.execution-time: 289

Connection: close

Adres url z odpowiedzi strony <https://www.netflix.com/pl-en/>

Zad 4:

```
import ftplib
"""
```

```
Program wyswietla zawartosc serwera ftp nastepnie przechodzi do  
wybranego katalogu wskazanego przez uzytkownika  
a na koncu pyta go jaki plik pobrac  
"""
```

```
def wyswietl_zawartoc_katalogu():  
    list_directory = []  
    ftp.dir(list_directory.append)  
    ftp.quit  
    for line in list_directory:  
        print("-", line)
```

```

def zmien_katalog():
    directory = input("Do ktorego katalogu chcesz przejsc ")
    try:
        ftp.cwd(directory)
    except:
        print("Nie ma takiego katalogu")

def lokalizacja():
    print("Obecny katalog w którym jesteś ", ftp.pwd())

def pobierz():
    filename = input("Podaj nazwę pliku który chcesz скачать ")
    try:
        ftp.retrbinary("RETR " + filename ,open(filename, 'wb').write)
        print("Pobrano plik")
    except:
        print("Error")

if __name__ == "__main__":
    login = "anonymous"
    haslo = "ftplib-example-1"
    server_ftp = "ftp.nluug.nl"
    ftp = ftplib.FTP(server_ftp)
    ftp.login(login, haslo)
    wyswietl_zawartosc_katalogu()
    zmien_katalog()
    lokalizacja()
    wyswietl_zawartosc_katalogu()
    pobieranie()

    ftp.quit()

```

Wyniki:

```

- lrwxrwxrwx    1 0      0          1 Nov 13  2012 ftp -> .
- lrwxrwxrwx    1 0      0          3 Nov 13  2012 mirror ->
pub
- drwxr-xr-x   23 0      0         4096 Dec 14  2018 pub
- drwxr-sr-x   89 0      450        4096 May 08 10:55 site
- drwxr-xr-x    9 0      0         4096 Jan 23  2014 vol

```

Do ktorego katalogu chcesz przejsc lrwxrwxrwx

nie ma takiego katalogu

Obecny katalog w którym jesteś /

```
- lrwxrwxrwx    1 0      0          1 Nov 13  2012 ftp -> .
- lrwxrwxrwx    1 0      0          3 Nov 13  2012 mirror ->
pub
- drwxr-xr-x   23 0      0        4096 Dec 14  2018 pub
- drwxr-sr-x   89 0      450       4096 May 08 10:55 site
- drwxr-xr-x    9 0      0        4096 Jan 23  2014 vol
```

Podaj nazwę pliku który chcesz scignac /

Error

Zad 5:

Serwer:

```
import asyncio, websockets
async def server(websocket, path):
    message = await websocket.recv()
    print(f"Wiadomosc od klienta: {message}")
    print("Serwer wyslal odpowiedz do klienta")
    greet = "Otrzymałem twoja wiadomosc"
    await websocket.send(greet)
start_server = websockets.serve(server, "localhost", 8001)
asyncio.get_event_loop().run_until_complete(start_server)
asyncio.get_event_loop().run_forever()
```

Client:

```
import asyncio, websockets
#prosty klient
async def client():
    uri = "ws://localhost:8001"
    async with websockets.connect(uri) as websocket:
        message = input("Podaj wiadomosc dla serwera ")
        await websocket.send(message)
        print("Klient wyslal wiadomosc do serwera")
        odpowiedz_ser = await websocket.recv()
        print(f"Wiadomosc od serwer: {odpowiedz_ser}")
    asyncio.get_event_loop().run_until_complete(client())
```

Wyniki:

Podaj wiadomosc dla serwera fghjkl

Klient wyslal wiadomosc do serwera

Wiadomosc od serwer: Otrzymałem twoja wiadomosc

Zad 6:

```
# Przyklad tworzy prosty serwer www wykorzystujac klase
simpleHTTPRequestHandler
import http.server, socketserver
PORT = 11022
handler = http.server.SimpleHTTPRequestHandler
with socketserver.TCPServer("", PORT), handler) as httpd:
    print("Serwer wystartowal na porcie ", PORT)
    httpd.serve_forever()
```

Wyniki:

Serwer wystartowal na porcie 11022

Zad 7:

```
from http.server import BaseHTTPRequestHandler, HTTPServer
import time

hostName = "localhost"
serverPort = 8094
class MyServer(BaseHTTPRequestHandler):
    def do_GET(self):
        self.send_response(200)
        self.send_header("Content-type", "text/html")
        self.end_headers()

self.wfile.write(bytes("<html><head><title>python</title></head>",
"utf-8"))
        self.wfile.write(bytes("<p>Request: %s</p>" % self.path, "utf-
8"))
        self.wfile.write(bytes("<body>", "utf-8"))
        self.wfile.write(bytes("<p>This is an example web server</p>",
"utf-8"))
        self.wfile.write(bytes("</body></html>", "utf-8"))
```

```
if __name__ == "__main__":  
    webServer = HTTPServer((hostName, serverPort), MyServer)  
    print("Server startd https://%s:%s " % (hostName, serverPort))  
    try:  
        webServer.serve_forever()  
    except KeyboardInterrupt:  
        pass  
    webServer.server_close()  
    print("Server stopped")
```

Wyniki:

Server startd https://localhost:8094