



الجمهورية العربية السورية

وزارة التعليم العالي والبحث العلمي

جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

السنة الخامسة

## Second Network Programming Homework

### إعداد الطالبتين:

سدره زين تنزكلي (2816) - رشا شادي جبيلي (2939)

إشراف الدكتور المهندس: مهدي عيسى

### Question 1: Bank ATM Application with TCP Server/Client and Multi-threading

```
import socket, threading

accounts = {
    "Sedra": 1000,
    "Rasha": 1500
}

def handle_client(client_socket, client_address):
    print("Accepted connection from {}".format(client_address))

    client_socket.sendall(b"Welcome to the Bank ATM. Please enter your account name: ")
    account_name = client_socket.recv(1024).decode().strip()

    if account_name not in accounts:
        client_socket.sendall(b"Account not found. Connection closed.")
        client_socket.close()
        return

    client_socket.sendall("Hello, {}. Please enter your PIN:".format(account_name).encode())
    pin = client_socket.recv(1024).decode().strip()

    while True:
        client_socket.sendall(b"Available operations:\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit\nEnter your choice: ")
        choice = client_socket.recv(1024).decode().strip()

        if choice == "1":
            client_socket.sendall("Your current balance is: {}".format(accounts[account_name]).encode())
        elif choice == "2":
            amount = float(client_socket.recv(1024).decode().strip())
            accounts[account_name] += amount
            client_socket.sendall("Deposit successful. Your updated balance is: {}".format(accounts[account_name]).encode())
        elif choice == "3":
            amount = float(client_socket.recv(1024).decode().strip())
            if amount > accounts[account_name]:
                client_socket.sendall(b"Insufficient funds.")
            else:
                accounts[account_name] -= amount
```

```

        client_socket.sendall("Withdrawal successful. Your updated
balance is: {}".format(accounts[account_name]).encode())
        elif choice == "4":
            break

    print("Closing connection with {}".format(client_address))
    client_socket.close()

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(("0.0.0.0", 4444))
server.listen(5)
print("Server listening on port 0.0.0.0")

while True:
    client_socket, client_address = server.accept()
    client_thread = threading.Thread(target=handle_client,
args=(client_socket, client_address))
    client_thread.start()

```

تم استيراد المكتبتين socket و threading لإنشاء المقابس وجعل السيرفر يخدم العملاء بنفس الوقت. باستخدام المتغير accounts تم بناء الحسابات بحيث يكون اسم صاحب الحساب هو المفتاح Key للقاموس والمبلغ المالي في الحساب هو القيمة Value المقابلة للمفتاح.

جعلنا السيرفر يعمل على أي عنوان IP في التطبيق باستخدام server.bind(("0.0.0.0", 5555)) ومع ملاحظة أننا جعلنا رقم المنفذ للسيرفر هو: 4444.

باستخدام التابع handle\_client(client\_socket, client\_address) تم التعامل مع اتصالات العملاء جميعها كما يلي: تم تمرير البارمتريين client\_socket و client\_address للذان يمثلان مقبس العميل وعنوان العميل وذلك عند قبول الاتصال معه، وعند قبول الاتصال نقوم بإرسال رسالة ترحيبية للمستخدم نطلب فيها إدخال اسمه (اسم صاحب الحساب) ونخزن الاسم في المتحول account\_name ثم نتحقق من كون اسم صاحب الحساب موجود فعلاً في الحسابات في البنك أم لا ونطبع في حال لم يكن موجودا الرسالة: Account not found. Connection closed.

إن كان اسم المستخدم موجود فعلاً أي لديه حساب نطلب منه إدخال رمز أمان له PIN، ومن ثم نعرض عليه جميع الخيارات الممكنة: Available operations:\n1. Check Balance\n2. Deposit\n3. Withdraw\n4. Exit

نتابع حسب رقم الخيار الذي يختاره العميل: بالعملية المرادة.

```
import socket

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect(("127.0.0.1", 4444))

response = client.recv(1024).decode()
print(response)
account_name = input()
client.sendall(account_name.encode())

response = client.recv(1024).decode()
print(response)
pin = input()
client.sendall(pin.encode())

while True:
    response = client.recv(1024).decode()
    print(response)
    choice = input()
    client.sendall(choice.encode())

    if choice == "4":
        break

    if choice in ["2", "3"]:
        amount = input("Enter amount: ")
        client.sendall(amount.encode())

    operation_result = client.recv(1024).decode()
    print(operation_result)

client.close()
```

Welcome to the Bank ATM. Please enter your account name:

Sedra

Hello, Sedra. Please enter your PIN:

123

Available operations:

1. Check Balance
2. Deposit
3. Withdraw
4. Exit

Enter your choice:

2

Enter amount: 2564

Deposit successful. Your updated balance is: 3564.0

Available operations:

1. Check Balance
2. Deposit
3. Withdraw
4. Exit

Enter your choice:

1

Your current balance is: 3564.0

Available operations:

1. Check Balance
2. Deposit
3. Withdraw
4. Exit

Enter your choice:

4

Server listening on port 0.0.0.0

Accepted connection from ('127.0.0.1', 12450)

Closing connection with ('127.0.0.1', 12450)

□

Question 2: Website:



برنامج سيرفر Flask:

```
from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template('index.html')

@app.route('/info')
def about():
    return render_template('about.html')

@app.route('/login')
def login():
    return render_template('login.html')

if __name__ == '__main__':
    app.run(debug=True)
```

**Username****Password**[Forgot password?](#)**My Contact Information**

Name	Number
سندھ زین نمبر	2816
رہا شادی نمبر	2939