

الجمهورية العربية السورية وزارة التعليم العالي جامعة تشرين







إعداد الطالب: إبراهيم محمد حمدو

2436

Second Network Programming Homework

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

Project Description:

Build a TCP server and client Bank ATM application using Python. The server should handle multiple client connections simultaneously using multi-threading. The application should allow clients to connect, perform banking operations (such as check balance, deposit, and withdraw), and receive their updated account status upon completion.

Requirements:

- A. The server should be able to handle multiple client connections concurrently.
- B. The server should maintain a set of pre-defined bank accounts with balances.
- C. Each client should connect to the server and authenticate with their account details.
- D. Clients should be able to perform banking operations: check balance, deposit money, and withdraw money.
- E. The server should keep track of the account balances for each client.
- F. At the end of the session, the server should send the final account balance to each client.

Guidelines:

- Use Python's socket module without third-party packages.
- · Implement multi-threading to handle multiple client connections concurrently.
- Store the account details and balances on the server side.

Notes:

- Write a brief report describing the design choices you made and any challenges faced during implementation.
- You can choose to create a TCP Server/Client Bank ATM application or any other appropriate
 application that fulfills all requirements.

Project ∨

□ pythonProject C.Up

□ pythonProject C.Up

□ python Project C.Up

□ python Pro

كود السيرفر:

• كود الكلاينت:

```
import socket

def main():

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:

client.connect(('127.0.0.1', 4445))

account_number = input("Enter your account number: ")

client.send(account_number.encode())

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

except socket.error as massageerror:

print(massageerror)

while True:

print("Options: \n 1. Check Balance \n 2. Deposit \n 3. Withdraw \n 4. Exit")

option = input("Enter option: ")

if option == '4':

break

client.send(option.encode())

if option == '1':

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

elif option == '2' or option == '3':

amount = input("Enter amount: ")

client.send(amount.encode())

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

client.send(amount.encode())

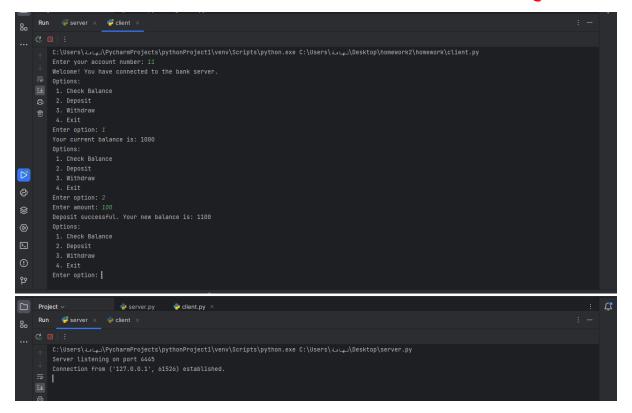
print(client.recv(1024).decode())

client.close()

if __name__ == "__main__":

main()
```

• الخرج:



الشرح:

يتضمن الكود قسمين القسم الأول هو كود السيرفر سنتحدث عنه أولا 1-بعد تعريف المكاتب عرفنا قاموس بحيث مفاتيحه هي ارقام حسابات 5 مشتركين اما القيم المقابلة هي قيم المبالغ المالية في حسابات المشتركين 2-يبدا تنفيذ البرنامج من السطر 40 وهو استدعاء التابع الفرعي ()start_server الذي يقوم بتشغيل السيرفر.

3-عرفنا بعدها مقبس وبذلك حددنا نوع السيرفر وهو TCP

4- قمنا بتوصيل السيرفر على العنوان 127.0.0.1 وهو عنوان داخلي حيث ينتظر السيرفر على الممنفذ 4445

5- نحدد عدد العملاء اللذين يستطيع السيرفر تخديمهم سوية وهو 8

6- ندخل لحلقة تقوم بداية بتعريف مقبس للعميل الأول المنضم وتحفظ عنوانه أيضا

7- ننشئ غرض باسم CLIENT_THREAD من الصنف threading.Thread ونحدد الوظيفة التي سيشغلها وهي التابع client_socket وهو مقبس العميل المنضم على السيرفر

8- التابع handel_client

9- نعرف متحول باسم account_number قيمته تساوي قيمة الرسالة المستلمة من العميل بعد فك تشفير ها

10- ندخل إلى مجموعة شروط if وهي الأول: نبحث من خلاله عن مقارنة بين الرقم المدخل من العميل و المفاتيح في القاموس المعرف مسبقا في حال التطابق ترسل رسالة ترحيب الى العميل اما في حال عدم تطابق أي الحساب غير موجود مسبقا ترسل رسالة للإعلام بذلك ثم يقوم السير فر بإغلاق المقبس الخاص بالعميل أي يفصل العميل ليبحث عن عميل منتظر أخر ليضيفه إلى مجموعة العملاء الذين يقوم بتخديمهم

11-ندخل في حلقة هدفها هو معرفة الاجراء الذي يريد من خلاله العميل ان يدير حسابه البنكي في حال تطابقه

12-يقوم المتحول option الذي قيمته تساوي الرسالة المستقبلة من العميل والتي تحدد نوع الاجراء المتبع (1،2،3،4 رقم الاجراء مشروح في كود العميل)

13-نقارن قيمة القيمة المدخلة مع الإجراءات 1-2-3-4 في حال تطابقها مع 1-نرسل للعميل قيمة الحالية للحساب، 2- حيث انه يمثل خيار إضافة مال إلى الحساب عندها يطلب من العميل ادخال القيمة المراد اضافتها، 3- الخيار يمثل سحب أموال عندها أيضا نطلب من العميل ادخال القيمة المراد سحبها مع مراعاة ان القيمة لايجب لها ان تتجاوز قيمة الرصيد الكلي، 4- الخروج دون أي إجراء وعندها يقوم السيرفر بإغلاق المقبس الخاص بالعميل

القسم الثاني: كود العميل

1-ضمنا المكتبة

يبدأ الكود من السطر قبل الأخير عند تشغيل التابع ()main الذي يعيدنا لبداية الكود من أجل التنفيذ

2- عرفنا مقبس خاص بالعميل

3-كي نتجنب توقف التطبيق قمنا بفحص الاتصال داخل try

4- تعليمات الادخال الخاصة برقم الحساب ثم ارسلناها للسيرفر

5-تلقي رسالة من السيرفر بعد الاتصال به

6-تعليمة except تحفظ الخطا في حال وقوعه في متحول و من ثم نطبع رسالة الخطا 7-ندخل إلى حقلة 8-تعليمة طباعة توضح للمستخدم أن عليه ان يدخل أحد الأرقام ليقوم السيرفر بمعرفة الاجراء الذي يريد أن يقوم به المستخدم (1- استعلام الرصيد، 2-إيداع مبلغ، 3- سحب مبلغ، 4-الخروج) المتحول option يحمل قيمة الإدخال من العميل ثم ندخل في مجموعة شروط وهي تناقش الموضوع ان الاجراء 4- يعني أن الزبون يريد إنهاء العميلة، 2 أو 3 يعني ان الزبون سيدخل او يسحب مال عندها يطلب منه إدخال القيمة المرادة وترسل للسيرفر عدا ذلك يقوم يقوم بارسال القيمة المدخلة إلى السيرفر