

2024

Second Network Programming Homework

إعداد الطلاب: سيزار أحمد شعبان 2897
جعفر سليمان علي 2621

إشراف الدكتور المهندس: مهند عيسى
السنة الخامسة اتصالات

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

برنامج السيرفر أولاً:

```
import socket
import threading

ACCOUNTS = {
    "Sezar1": 5000,
    "Jafar2": 3000}

def handle_client(conn, addr):
    print(f"New connection from {addr}")
    try:
        account_id = conn.recv(1024).decode()
        if account_id not in ACCOUNTS:
            conn.sendall("Invalid account ID".encode())
            return

        balance = ACCOUNTS[account_id]
        conn.sendall(f"Welcome to the bank ATM! Your current balance is:
{balance}".encode())

        while True:
            choice = conn.recv(1024).decode()
            if choice == "1":
                conn.sendall(f"Your balance is: {balance}".encode())
            elif choice == "2":
                amount = int(conn.recv(1024).decode())
                balance += amount
                ACCOUNTS[account_id] = balance
                conn.sendall(f"Deposit successful. Your new balance is:
{balance}".encode())
            elif choice == "3":
                amount = int(conn.recv(1024).decode())
                if amount > balance:
                    conn.sendall("Insufficient funds".encode())
                else:
                    balance -= amount
                    ACCOUNTS[account_id] = balance
                    conn.sendall(f"Withdrawal successful. Your new balance
is: {balance}".encode())
            elif choice == "4":
```

```

        conn.sendall(f"Thank you for using the bank ATM. Your
final balance is: {balance}".encode())
        break
    else:
        conn.sendall("Invalid choice".encode())

except Exception as e:
    print(f"Error: {e}")
finally:
    conn.close()

def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(("0.0.0.0", 8000))
    server_socket.listen(5)
    print("Server is listening on localhost:8000")

    while True:
        conn, addr = server_socket.accept()
        client_thread = threading.Thread(target=handle_client, args=(conn,
addr))
        client_thread.start()

if __name__ == "__main__":
    main()

```

آلية بناء وإنشائنا للكود:

- 1) قمنا باستيراد المكتبتين socket و threading لبناء المقابس وحتى نجعل السيرفر يعمل بنفس الوقت مع عدة مستخدمين وتخدمهم جميعاً.
- 2) تم بناء وإنشاء الحسابات بتعريف القاموس ACCOUNTS الذي يحوي اسم صاحب الحساب والمبلغ المالي الذي يملكه حالياً.
- 3) السيرفر يعمل على العنوان: 0.0.0.0 وتم ربطه مع رقم المنفذ 8000، طبعاً السيرفر يتعامل مع أي عنوان IP ضمن التطبيق.
- 4) عن طريق اشتقاق الغرض client_thread من الصنف Thread نستطيع التعامل مع العملاء بنفس الوقت، وذلك بإمرار التابع الذي يتعامل مع اتصالات العملاء وبارامتراته.
- 5) التابع handle_client(conn, addr) يأخذ بارمترين هما: عنوان العميل الذي يتصل مع السيرفر ومقبس العميل.

- (6) يطبع التابع رسالة إضافة الاتصال مع العميل الجديد ويطبع بجانبها عنوانه.
- (7) نقوم باستقبال اسم صاحب الحساب عبر المتحول `account_id` والتحقق منه إن كان موجود أم لا.
- (8) لأجل كل عميل هناك العديد من الخيارات للعمليات مثل: عرض إجمالي النقود في الحساب، أو إضافة مبلغ مالي على الحساب، أو سحب مبلغ.
- كود العميل الأول:

```
import socket

def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(("127.0.0.1", 8000))

    print("Enter your account ID:")
    account_id = input()
    client_socket.sendall(account_id.encode())

    response = client_socket.recv(1024).decode()
    print(response)

    while True:
        print("\nChoose an option:")
        print("1. Check balance")
        print("2. Deposit money")
        print("3. Withdraw money")
        print("4. Exit")
        choice = input()
        client_socket.sendall(choice.encode())

        if choice == "1":
            balance = client_socket.recv(1024).decode()
            print(balance)
        elif choice == "2":
            print("Enter the amount to deposit:")
            amount = int(input())
            client_socket.sendall(str(amount).encode())
            response = client_socket.recv(1024).decode()
            print(response)
        elif choice == "3":
            print("Enter the amount to withdraw:")
            amount = int(input())
```

```

        client_socket.sendall(str(amount).encode())
        response = client_socket.recv(1024).decode()
        print(response)
    elif choice == "4":
        response = client_socket.recv(1024).decode()
        print(response)
        break
    else:
        response = client_socket.recv(1024).decode()
        print(response)

client_socket.close()

if __name__ == "__main__":
    main()

```

الزبون (العميل) يقوم بإجراءات محددة بعد إدخال اسمه، وهي توافق أرقام الخيارات 1،2،3،4. ويقوم بإرسال هذه القيم للسيرفر لمعالجتها.

التنفيذ:

```

Server is listening on localhost:8000
New connection from ('127.0.0.1', 1943)

```

```

█

```

```

Enter your account ID:
Sezar1
Welcome to the bank ATM! Your current balance is: 5000

```

```

Choose an option:

```

1. Check balance
2. Deposit money
3. Withdraw money
4. Exit

```

2

```

```

Enter the amount to deposit:

```

```

2000

```

```

Deposit successful. Your new balance is: 7000

```

```

Choose an option:

```

1. Check balance
2. Deposit money
3. Withdraw money
4. Exit

```

_

```

Question 2: Chat Text Analyzer:

```
import re

def analyze_text(file_path):
    with open(file_path, 'r') as file:
        text = file.read()

    words = text.split()
    word_count = len(words)

    char_count = len(text)

    sentences = re.split(r'[.!?]+', text)
    sentence_count = len(sentences)

    return word_count, char_count, sentence_count

file_path = 'text_file.txt'
word_count, char_count, sentence_count = analyze_text(file_path)

print(f"Word count: {word_count}")
print(f"Character count: {char_count}")
print(f"Sentence count: {sentence_count}")
```

هذا البرنامج يقوم بتحليل العبارات والجمل الموجودة في ملف TXT، ويقوم بطباعة عدد الكلمات، وعدد المحارف...

- (1) import re: يتم استيراد المكتبة re التي توفر وظائف لمطابقة ومعالجة التعابير العادية.
- (2) def analyze_text(file_path): تعريف الدالة analyze_text التي تقوم بتحليل النص الموجود في ملف نصي معين بواسطة.
 - (3) with open(file_path, 'r') as file: يتم فتح الملف الموجود في file_path في وضع القراءة ('r')
 - (4) text = file.read(): يتم حفظه في المتغير text.
 - (5) words = text.split(): يتم تقسيم النص إلى كلمات وحفظها في قائمة من خلال استخدام الفراغ كفاصل.

(6) `word_count = len(words)`: يتم حساب عدد الكلمات الكلي في النص من خلال عدد العناصر في القائمة `words`.

(7) `char_count = len(text)`: تم حساب عدد الأحرف في النص الكلي من خلال عدد الأحرف في المتغير `text`.

(8) `sentences = re.split(r'[.!?]+', text)`: تقسيم النص إلى جمل من خلال استخدام `re.split` وتحديد النمط `[?!.]` الذي يعني التقسيم بناءً على العلامات الترقيم.

(9) `sentence_count = len(sentences)`: يتم حساب عدد الجمل في النص من خلال عدد العناصر في القائمة `sentences`.

(10) `return word_count, char_count, sentence_count`: الدالة ترجع عدد الكلمات، عدد الأحرف، وعدد الجمل.

(11) `file_path = 'text_file.txt'`: تعيين المسار للملف النصي المراد تحليله.

(12) `word_count, char_count, sentence_count = analyze_text(file_path)`: استدعاء الدالة `analyze_text` وتخزين قيمة الكلمات، الأحرف، والجمل في المتغيرات المقابلة.

(13) `print(f"Word count: {word_count}")`, `print(f"Character count: {char_count}")`, `print(f"Sentence count: {sentence_count}")`: يتم طباعة

عدد الكلمات، عدد الأحرف، وعدد الجمل بواسطة تنسيق السلسلة `f-string` في كل تمريرة.

```
≡ text_file.txt
1 Hello I'm little Dev...
2 How can I help you?
```

```
≡ text_file.txt
1 Hello I'm little Dev...
2 How can I help you?
3 ?????????????????????????????????????
```

Word count: 10

Character count: 67

Sentence count: 3