



Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Лабораторна робота №9

РІЗНІ ВИДИ ВЗАЄМОДІЇ ДОДАТКІВ:
CLIENT-SERVER, PEER-TO-PEER,
SERVICE-ORIENTED ARCHITECTURE

Виконала
студентка групи ІА – 13:
Майданюк Анастасія

Перевірив:
Мягкий М.Ю.

Київ 2023

Завдання:

1. Ознайомитися з короткими теоретичними відомостями.
2. Реалізувати частину функціоналу робочої програми у вигляді класів і їх взаємодій для досягнення конкретних функціональних можливостей.
3. Реалізувати взаємодію програми в одній з архітектур відповідно до обраної теми.

Варіант:

..11 Web crawler (proxy, chain of responsibility, memento, template method, composite, p2p)

Веб-сканер повинен вміти розпізнавати структуру сторінок сайту, переходити за посиланнями, збирати необхідну інформацію про зазначений термін, видаляти не семантичні одиниці (рекламу, об'єкти javascript і т.д.), зберігати знайдені дані у вигляді структурованого набору html файлів вести статистику відвіданих сайтів і метадані.

Хід роботи

Кожна інстанція даної програми здатна функціонувати одночасно як сервер, який приймає з'єднання від клієнтів, так і як клієнт, що спілкується з іншими серверами. Ця властивість програми дозволяє формувати децентралізовані мережі, де кожен вузол має можливість спілкування з рештою вузлів мережі.

Коли скрипт активується, відбуваються наступні процеси:

Серверний Компонент:

- Метод `start_server` запускає серверний сокет і чекає на з'єднання від клієнтів.
- Для кожного нового з'єднання ініціюється окремий потік (`client_thread`), який використовує функцію `handle_client_peer_wrapper` для обслуговування клієнта. Ця функція керує взаємодією з клієнтом, обробляє його команди та директує їх на подальшу обробку в функції `handle_peer`.

Клієнтський Компонент:

- Функція `start_client` створює клієнтський сокет і здійснює підключення до сервера.
- Клієнт має змогу надсилати команди, які потім передаються на сервер для їх обробки.

Обробка Команд:

- Всі команди, що призначені для взаємодії з сервером, передаються через сокети. Вони обробляються за допомогою функції `handle_peer`.

```

7  def handle_client_peer_wrapper(client_socket):
8      try:
9          handle_peer(client_socket)
10     except (ConnectionAbortedError, ConnectionResetError) as exp:
11         pass
12     finally:
13         client_socket.close()
14
15
16  def start_server():
17      free_port = find_free_port()
18      if free_port is None:
19          return
20
21
22      db_manager = DatabaseManager()
23
24      server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
25      server_socket.bind(('0.0.0.0', free_port))
26      server_socket.listen(1)
27
28      while True:
29          client_socket, client_address = server_socket.accept()
30
31          ip, port = client_address
32          db_manager.insert_user(ip, port)
33
34          client_thread = threading.Thread(target=handle_client_peer_wrapper, args=(client_socket,))
35          client_thread.start()

```

Висновок:

Розроблено два основні компоненти: серверний та клієнтський. Серверний компонент ініціює серверний сокет і обробляє з'єднання від клієнтів. Клієнтський компонент включає створення клієнтського сокета та підключення до сервера. Команди між клієнтом та сервером передаються через сокети і обробляються відповідно.