

UNIwersYTET RZESZOWSKI
WYDZIAŁ NAUK ŚCISŁYCH I TECHNICZNYCH
INSTYTUT INFORMATYKI



Krystian Filipek
134907

Informatyka

Komunikator „Koliber”

Praca projektowa

Praca wykonana pod kierunkiem
mgr inż. Jarosław Szkoła

Rzeszów 2026

Spis treści

1. Wprowadzenie	6
2. Opis założeń projektu	7
2.1. Wymagania funkcjonalne	7
2.2. Wymagania нефункционалне	7
3. Opis struktury projektu	8
3.1. Struktura plików	8
3.2. Najważniejsze metody	8
4. Prezentacja działania i testy	9
4.1. Serwer Ubuntu: komunikaty rozpoczęcia pracy i dołączenia	9
4.2. Serwer Ubuntu: komunikaty rozłączania się użytkowników	9
4.3. Działanie aplikacji z perspektywy maszyn klienckich	9
5. Instrukcja uruchomieniowa	10
6. Podsumowanie	11
Bibliografia	12
Spis rysunków	13

1. Wprowadzenie

Projekt prostego, anonimowego komunikatora sieciowego został zrealizowany na potrzeby przedmiotu **Sieci Komputerowe** na 2 roku Informatyki na **Uniwersytecie Rzeszowskim**. Głównym celem pracy było zaprojektowanie i implementacja systemu wielowątkowego wymiany informacji w architekturze **Klient-Serwer-Klient**. System pozwala na równoczesną komunikację wielu jednostek klienckich za pośrednictwem centralnego węzła pośredniczącego, którym w naszym przypadku jest serwer **Ubuntu**. Zarządza on przepływem danych w izolowanym środowisku sieciowym na potrzeby symulacji realnej sieci.

The project of a simple, anonymous network messenger was carried out for the purposes of the **Computer Networks** course in the second year of Computer Science at the **University of Rzeszów**. The main goal of the work was to design and implement a multithreaded information exchange system in the **Client-Server-Client** architecture. The system allows for simultaneous communication between multiple client units via a central intermediary node, which in our case is an Ubuntu server. It manages data flow in an isolated network environment for the purposes of simulating a real network.

2. Opis założeń projektu

Celem projektu było stworzenie bezpiecznego i anonimowego komunikatora sieciowego przy użyciu języka programowania **Python**. Program ma za zadanie umożliwić wielu użytkownikom jednoczesną wymianę wiadomości tekstowych w czasie rzeczywistym, wykorzystując architekturę wielowątkową również zaimplementowaną używając wspomnianego wcześniej języka **Python** dobierając wcześniej odpowiednie moduły takie jak `threading` czy `socket`. Program ma działać w środowisku rozproszonym, łącząc maszyny o różnych systemach operacyjnych pracując wewnątrz wirtualnej infrastruktury sieciowej.

Podstawowym problemem, który zostanie rozwiązany przez realizację tego projektu, jest zagrożenie prywatności w sieci oraz konieczność udostępniania danych osobowych w tradycyjnych systemach komunikacji. Obecnie wiele krajów zachodnich takich jak chociażby *Wielka Brytania* coraz częściej z wrogością zwraca się do prywatnej wymiany informacji w obawie o to iż użytkownicy będą komunikować między sobą informacje nie będące po drodze narracji rządowej. Projekt inspirowany się forum obrazkowym *4chan* rozwiązuje ten problem poprzez brak konieczności zakładania kont oraz wykorzystanie tymczasowych identyfikatorów sesji, które są generowane losowo przez serwer i usuwane natychmiast po rozłączeniu użytkownika z pamięci operacyjnej węzła.

Aby problem został skutecznie rozwiązany, potencjalny zespół musi posiadać wiedzę z zakresu architektury sieciowej **TCP/IP**, modelu **ISO/OSI** (w szczególności warstwy transportowej i aplikacji) oraz podstawowej obsługi wątków.

Rozwiązanie problemu przebiegło w kilku zdefiniowanych krokach. W pierwszej kolejności nastąpiło zaprojektowanie topologii sieciowej w środowisku `VirtualBox`. Kolejnym krokiem było zaimplementowanie logicznej warstwy serwera, odpowiedzialnego za akceptowanie połączeń i przesyłanie komunikatów. Następnie wdrożono obsługę wielu wątków oraz system anonimowego przydzielania identyfikatorów. Końcowym etapem było stworzenie podstawowego interfejsu **CLI** dla klientów, gdzie Ci będą mogli odróżniać wiadomości swoje od cudzych za pośrednictwem kolorów.

2.1. Wymagania funkcjonalne

2.2. Wymagania niefunkcjonalne

3. Opis struktury projektu

3.1. Struktura plików

3.2. Najważniejsze metody

4. Prezentacja działania i testy

4.1. Serwer Ubuntu: komunikaty rozpoczęcia pracy i dołączenia

4.2. Serwer Ubuntu: komunikaty rozłączania się użytkowników

4.3. Działanie aplikacji z perspektywy maszyn klienckich

5. Instrukcja uruchomieniowa

Aby poprawnie uruchomić skrypty a co za tym idzie kompletną aplikację, należy przeprowadzić następujące kroki:

1. Zainstalować środowisko **VirtualBox**, najlepiej w najnowszej wersji.
2. Utworzyć maszynę wirtualną - Ubuntu Server. W tym przypadku wersja Ubuntu 24.04.3 LTS.
3. Utworzyć klienckie maszyny - w przypadku projektu Windows 11 LTSC oraz Linux Mint.
4. Plik `client.py` przesłać do maszyn klienckich np. na pulpit.
5. Plik `server.py` przesłać na serwer Ubuntu.
6. Plik `server.py` uruchomić na serwerze poprzez komendę `python3 server.py` - po wykonaniu komendy powinien przywitać nas odpowiedni komunikat.
7. Plik `client.py` uruchomić poprzez komendę `python client.py` (Windows) / `python3 client.py` (Linux) uprzednio przechodząc w odpowiednią ścieżkę w terminalu - CMD / Bash.
8. Po wykonaniu wcześniejszych kroków całość powinna działać poprawnie.

6. Podsumowanie

Zrealizowany projekt pozwolił na stworzenie w pełni funkcjonalnego, wielowątkowego komunikatora sieciowego. Wszystkie założenia oraz cele zostały w pełni osiągnięte. Podczas prac szczególną uwagę poświęcono stabilności połączenia oraz czytelności "GUI" w terminalu poprzez kolory odróżniające komunikaty. W przyszłości można dodać bardziej rozbudowany interfejs użytkownika poprzez odpowiednie moduły oraz opcję przesyłania plików między użytkownikami.

Bibliografia

- [1] ChatGPT, 2025. Wykorzystano do wygenerowania dummy text. <https://openai.com/chatgpt>.
- [2] Coolors, 2025. <https://coolors.co/>.
- [3] Freepik, 2025. <https://www.freepik.com/>.
- [4] Google Fonts, 2025. <https://fonts.google.com/>.
- [5] Pexels, 2025. <https://www.pexels.com/>.
- [6] Stack Overflow, 2025. <https://stackoverflow.com/>.
- [7] Unsplash, 2025. <https://unsplash.com/>.
- [8] W3schools, 2025. <https://www.w3schools.com/>.

Spis rysunków