Wstępna Dokumentacja Projektu: Implementacja Prostego Protokółu P2P

1. Temat zadania

Implementacja prostego protokołu P2P (Peer-to-Peer).

---

2. Treść zadania

Napisać program obsługujący prosty protokół P2P, spełniający następujące wymagania:

1. **Zasób**: Obiekt z danymi binarnymi identyfikowany nazwą (zasoby o tych samych nazwach są traktowane jako identyczne). Rozmiar zasobu mieści się w jednym datagramie protokołu UDP.
2. **Funkcjonalność**:
   - Użytkownik może dodawać zasoby z systemu plików lokalnego.
   - Pobieranie zasobów o zadanej nazwie ze wskazanego hosta.
   - Rozgłaszanie informacji o posiadanych lokalnie zasobach przy użyciu protokołu UDP.
3. **Przetwarzanie zasobów**:
   - Zasoby są propagowane poprzez kopiowanie (kopia traktowana równoznacznie z oryginałem).
   - System musi obsługiwać sytuacje wyjątkowe, np. przerwane transmisje.
4. **Interfejs użytkownika**: Tekstowy, wspierający współbieżne transfery zasobów.

---

3. Interpretacja treści zadania

Główne funkcje programu:

1. **Dodawanie zasobów lokalnych:**
   - Pobranie pliku binarnego z systemu plików lokalnego.
   - Rejestracja nazwy zasobu w katalogu lokalnym.
2. **Rozgłaszanie lokalnych zasobów:**
   - Periodyczne wysyłanie listy dostępnych zasobów przy użyciu protokołu UDP w trybie broadcast.
3. **Pobieranie zasobu:**
   - Otrzymanie nazwy zasobu od użytkownika.
   - Zapytanie zdalnych hostów o lokalizację zasobu.
   - Transfer zasobu z wybranego hosta przy użyciu UDP.
4. **Obsługa sytuacji wyjątkowych:**
   - Ponawianie zapytań w przypadku utraty datagramów.
   - Informowanie użytkownika o niepowodzeniach transferu.

---

4. Krótki opis funkcjonalny (“black-box”)

Użytkownik:

- Może dodawać zasoby do katalogu lokalnego.
- Może wywołać pobieranie zasobu o podanej nazwie z określonego hosta.
- Może przeglądać lokalnie dostępne zasoby.

Program:

- Rozgłasza informacje o lokalnych zasobach.
- Obsługuje zapytania o zasoby od zdalnych hostów.
- Przesyła i odbiera zasoby w trybie UDP.
- Zarządza transferami bez blokowania interfejsu użytkownika.

---

5. Opis i analiza protokołów komunikacyjnych

Wykorzystany protokół: UDP (User Datagram Protocol)

1. **Rozgłaszanie zasobów:**
   - **Format komunikatu:**

Pole	Typ	Opis
Typ	uint8_t	1 = Rozgłoszenie
Liczba zasobów	uint16_t	Liczba zasobów w komunikacie
Nazwa zasobu	string	Unikalna nazwa zasobu

   - **Diagram komunikacji:**

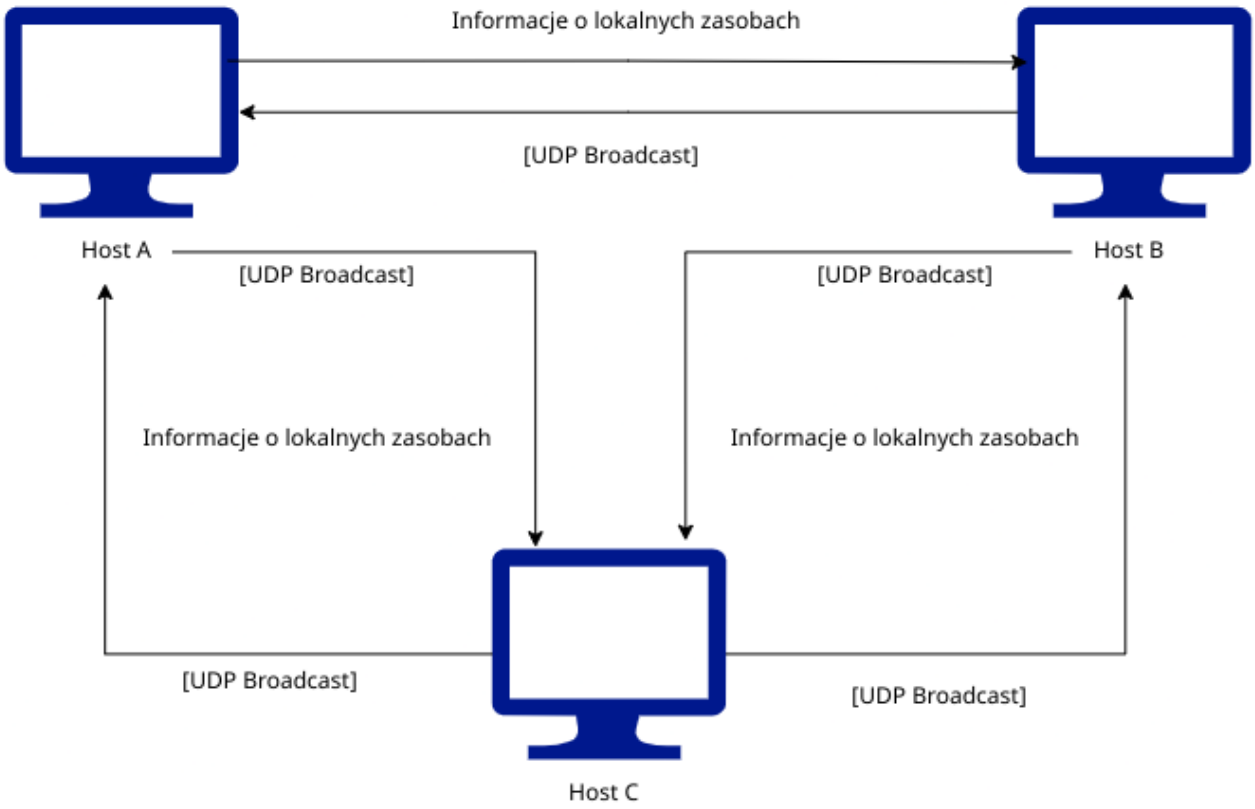


Figure 1: Broadcast diagram

2. **Pobieranie zasobu:**
   - **Format zapytania:**

Pole	Typ	Opis
Typ	uint8_t	2 = Zapytanie
Nazwa zasobu	string	Unikalna nazwa zasobu

   - **Diagram komunikacji:**

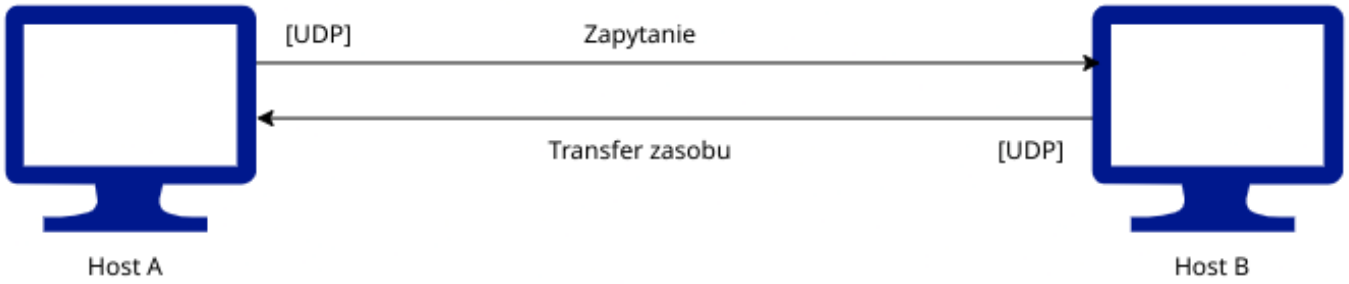


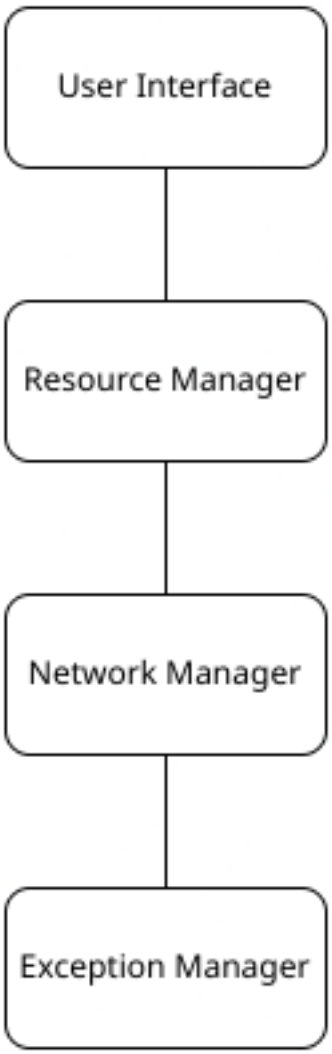
Figure 2: Transfer diagram

6. Planowany podział na moduły

Moduły:

1. **Moduł zarządzania zasobami:**
   - Obsługuje dodawanie, usuwanie i przegląd zasobów lokalnych.
2. **Moduł sieciowy:**
   - Rozgłaszanie zasobów.
   - Obsługa zapytań i transferu zasobów przy użyciu UDP.
3. **Moduł interfejsu użytkownika:**
   - Prosty tekstowy interfejs umożliwiający wykonywanie operacji przez użytkownika.
4. **Moduł obsługi wyjątków:**
   - Zarządza retransmisją utraconych datagramów i obsługą błędów.

Rysunek struktury:



7. Zarys koncepcji implementacji

Język programowania:

- **C++**

Biblioteki:

- **Boost.Asio**: Obsługa gniazd sieciowych i operacji asynchronicznych.
- **STL**: Przechowywanie i zarządzanie danymi lokalnymi.

Narzędzia:

- **CMake**: System budowy projektu.
- **GCC/Clang**: Kompilator.

Ogólne podejście do implementacji:

1. Klasa `Resource` reprezentująca zasób.
2. Implementacja klasy `NetworkManager` do obsługi rozgłaszania i transferów.
3. Implementacja klasy `ResourceManager` do zarządzania zasobami.
4. Implementacja klasy `ExceptionManager` do obługi wyjątków.
5. Wdrożenie prostego interfejsu tekstowego opartego na pętlach zdarzeń.