

System zdalnego zamykania systemów operacyjnych (1osoba)

Sieci Komputerowe 2 – Sprawozdanie z projektu

Wykonanie: Paulina Pacura, 142179

1. Opis protokołu komunikacyjnego

Zadanie projektowe zostało zrealizowane w architekturze **klient-serwer** z wykorzystaniem protokołu TCP.

Protokół TCP – protokół kontroli transmisji. Jest to protokół połączeniowy, w którym przed rozpoczęciem przesyłania danych wymaga nawiązania połączenia między komunikującymi się urządzeniami.

W celu nawiązania połączenia między urządzeniami w zadaniu wykorzystano interfejs gniazd BCD. Serwer nasłuchuje na wybranym porcie; w tym czasie klient uzyskuje jego deskryptor oraz nawiązuje z nim połączenie. Następnie możliwym jest przesyłanie danych w obie strony.

2. Przyjęta metoda rozwiązania problemu

W przesyłaniu danych ustalono następujący standard:

Przesyłanie urządzeń do zamknięcia	Ładowanie listy dostępnych urządzeń
<ul style="list-style-type: none">• klient wysyła komendę,• serwer wysyła potwierdzenie,• klient wysyła treść,• serwer wysyła potwierdzenie,• klient wysyła sygnał zakończenia,• serwer zamyka połączenie.	<ul style="list-style-type: none">• klient wysyła komendę,• serwer wysyła potwierdzenie,• serwer wysyła treść,• klient wysyła sygnał zakończenia,• serwer zamyka połączenie.

gdzie:

komenda = „0,” (ładowanie listy *load*) lub „1,” (żądanie zamykania *shutdown*);

treść = lista urządzeń (dostępnych online lub przekazanych do zamknięcia);

potwierdzenie = „accept”;

sygnał zakończenia = „end”.

Klient z urządzeniami do zamknięcia komunikuje się z wykorzystaniem protokołu **ssh**. Zadanie wykonano z założeniem o wykorzystaniu w sieci, gdzie wszystkie urządzenia mają schematyczny adres oraz znane hasło.

Serwer przeszukuje sieć pod kątem urządzeń, z którymi możliwe jest nawiązanie połączenia. Następnie ich listę wysyła danemu klientowi na żądanie. Klient wybiera z listy urządzenia, które chce zamknąć i wysyła listę do serwera. Serwer łączy się ze wskazanymi urządzeniami poprzez ssh i wykonuje polecenie **shutdown –now**.

3. Sposób implementacji problemu

Całość zadania zrealizowano w języku C++. W przypadku klienta zaprojektowano interfejs graficzny do obsługi programu, serwer wykonywany jest w terminalu.

3.1 Opis zawartości plików źródłowych

3.1.1 Program klienta

Aplikacja okienkowa umożliwia klientowi podanie adresu, na którym działa serwer. Przycisk „Refresh” powoduje przeładowanie listy dostępnych urządzeń, „Shutdown” pozwala zamknąć wybrane z listy.

Wciśnięcie przycisku wywołuje funkcję **handle_command**, która inicjuje nawiązanie połączenia z podanym serwerem i następnie w zależności od przycisku wywołuje funkcję **load_available_devices** lub **shutdown_devices**. Obie funkcje polegają na przesłaniu serwerowi komend oraz tekstu z nazwami urządzeń w przypadku funkcji shutdown. Na końcu następuje wysłanie do serwera sygnał końca połączenia.

3.1.2 Program serwera

Interakcja z klientem jest obsługiwana przez wątek. Odbiera on nowe połączenie od klienta i obsługuje je funkcją **client_connection**. Wczytuje w niej komendę klienta i na jej podstawie przechodzi do następnych funkcji: **handle_load** – w której sukces nawiązania połączenia ssh z urządzeniami w sieci oznacza ich dostępność - lub **handle_shutdown** – w której przetwarza uzyskany tekst na schematyczne nazwy urządzeń w sieci i łączy się z nimi poprzez ssh, wysyłając polecenie shutdown –now. Odebranie sygnału „end” powoduje zakończenie pracy wątku.

4. Sposób kompilacji, uruchomienia i obsługi programów projektu

Aplikacja klienta została tak skonfigurowana, aby jej uruchomienie możliwe było poprzez wywołanie pliku **ClientSystemyApp.exe** w folderze release z wszystkimi potrzebnymi zależnościami (skrót umieszczono w głównym folderze). Obsługa programu polega na wprowadzeniu adresu na którym działa serwer, załadowaniu przyciskiem **Refresh** listy dostępnych w sieci urządzeń oraz wybraniu urządzeń do zamknięcia. Osiągalne urządzenia znajdują się na jednej liście, a urządzenia do zamknięcia na drugiej – identyfikatory przenosi się między listami przy użyciu strzałek.

Aplikacja serwera uruchamiana jest z konsoli następującymi poleceniami:

- g++ server.cpp -o server -w -lssh -pthread
- ./server

Po uruchomieniu serwer nasłuchuje i obsługuje przychodzące sygnały bez ingerencji użytkownika.