|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr ćwicz.  2 | Data  24.11.2016 | Dawid Wojciechowski **126798** Michał Sibera **126827** | Wydział Elektryczny | Semestr  III | grupa I2-2 |
| **Implementacja protokołu warstwy zastosowań – sprawozdanie końcowe.** | | | | | |

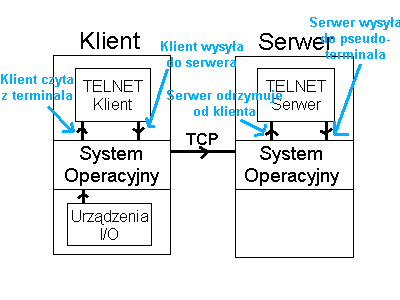
# Implementowany protokół: TELNET

Oprogramowanie klienta zostało zaimplementowane przez: Dawid Wojciechowski.

Oprogramowanie serwera zostało zaimplementowane przez: Michał Sibera.

# Specyfikacja protokołu

Telnet jest usługą, która nie cieszy się obecnie zbyt dużą popularnością wśród typowych użytkowników sieci, którzy mogą ograniczać się tylko do usług FTP, WWW i Poczty. Telnet oferuje możliwość zdalnej pracy na serwerze, który może znajdować się nawet po drugiej stronie świata. Użytkownik zalogowany do systemu dostaje do dyspozycji powłokę systemową, z której może obsługiwać system poprzez wpisywanie komend i podawanie parametrów, a sam system traktuje go jakby jego praca była przeprowadzana lokalnie na maszynie. Jedyną różnicą może być czas oczekiwania na odpowiedź serwera.



Rys.: Telnet jest aplikacją typu serwer-klient.

Praca zdalna przez usługę telnet jest możliwa tylko dla zalogowanych użytkowników, a więc aby wykonać jakiekolwiek komendy musimy posiadać na serwerze konto.

 Telnet w obu kierunkach używa sygnalizacji in-band. Bajt FF(255 - liczba ta nie mieści się w zakresie 7-bitowym, ale istnieje opcja binarna dla aplikacji Telnet pozwalająca na transmisję 8-bitową) jest nazywany IAC(ang. **I**nterpret **A**s **C**ommand). Kolejny bajt jest bajtem zawierającym polecenie.

Polecenia:



Telnet pracuje w czterech różnych trybach:

- **Tryb półdupleks**

Jest to domyślny tryb pracy obecnie rzadko stosowany. Domyślny terminal NVT jest urządzeniem półdupleksowym, które wymaga polecenia GO AHEAD (GA) wysłanego przez serwer, zanim zostaną zaakceptowane informacje podane przez użytkownika. Wpisywane przez użytkownika znaki są lokalnie odsyłane w postaci echa z klawiatury NVT na drukarkę NVT, dzięki czemu dopiero całe linie wysyłane są z klienta do serwera.

- **Tryb znakowy**

 Każdy znak wpisywany przez użytkownika jest natychmiast wysyłany do serwera. Serwer odsyła echo większości znaków, chyba że jego aplikacja wyłączyła tę funkcję. Kłopot z tym trybem pracy polega na tym, że podczas pracy na wolnych łączach lub w sieci o dużym ruchu, opóźnienie w nadsyłaniu echa jest zauważalne przez użytkownika. Tym nie mniej, ten tryb będziemy napotykali jako domyślny w większości używanych obecnie implementacji

- **Tryb wierszowy i liniowy**

Analogicznie jak tryb znakowy, lecz dotyczy wierszy lub całych linii.

# Specyfikacja implementacji

Projekt symulatora telnetu został napisany w języku C# w Microsoft Visual Studio 2015 korzystając z bibliotek takich jak Sockets i NetworkStream

**Zaimplementowaliśmy takie funkcjonalności jak:**

-logowanie i wylogowanie,

-listowanie zawartości aktywnego folderu,

-poruszanie się po drzewie katalogów,

-otwieranie plików tekstowych,

-edycja plików tekstowych,

-wyświetlenie aktualnej ścieżki

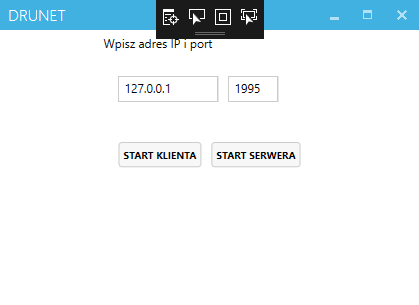
Komunikat przesyłany jest przez pojedynczy bajt, a na strukturę komendy składa się kolejka bajtów z czego pierwszy jest flagą do ustawień serwera, a kolejne bajty to tekst zakończony bajtem 254.

Serwer odbiera każdy bajt po kolei aż nie napotka bajtu 254. Następnie wykonuje metodę odpowiadającą fladze, wykorzystując odebrane dane.

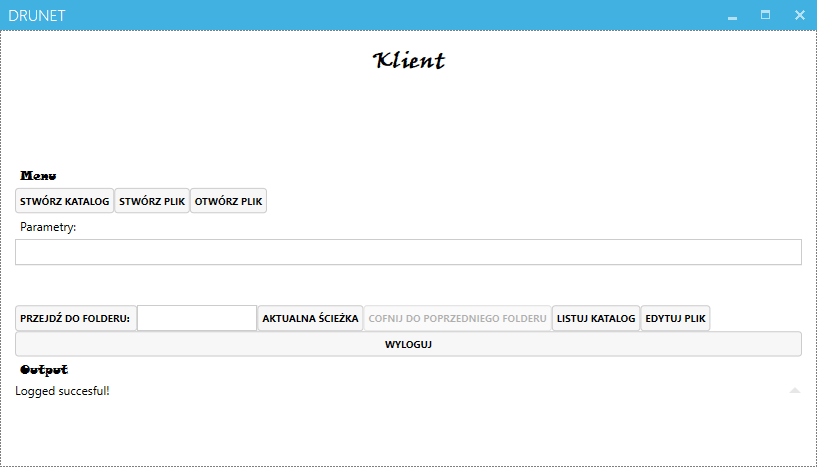
System logowania opiera się na zamku, który znajduje się na serwerze. Gdy dane logowania są zaakceptowane zamek ustawia się na wartość 0 udostępniając wszystkie funkcjonalności klientowi. W przeciwnym wypadku jego wartość to 1, a funkcje serwera zablokowane, również po wylogowaniu.

Przechodzenie do katalogu wyżej zostało zablokowane na poziomie Root poprzez semafor dołączony do przycisku, który jest inkrementowany i dekrementowany podczas operacji poruszania się po drzewie katalogów.

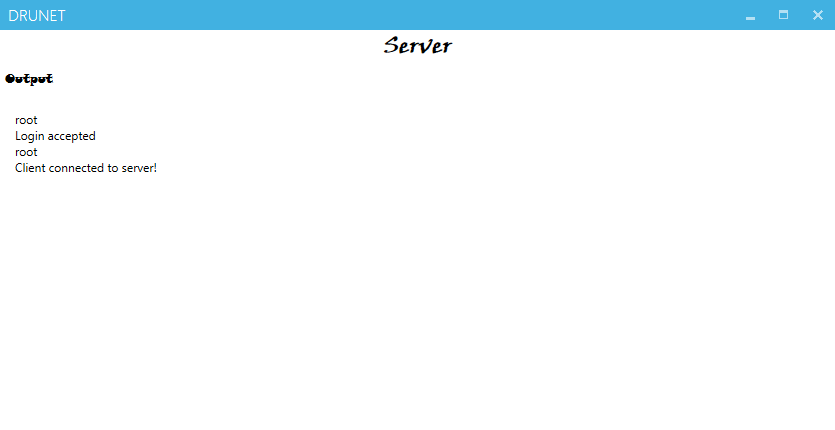
Cały projekt zawiera około 700 linii kodu oraz GUI oparte na WPF.



Widok główny po uruchomieniu programu z opcjami wyboru klient/serwer oraz ustawienie adresu IP i portu.



Widok klienta



Widok serwera

# Testy

W przeprowadzonych testach naszego projektu z użyciem programu przechwytującego Wireshark zauważyliśmy, że jest on w stanie przechwycić i odczytać każdy przesłany komunikat, ponieważ bajty nie są w żaden sposób zaszyfrowane.

# Podsumowanie

Użyteczność telnetu nie zamyka się w wąskim kręgu. Odnajduje się w wielu zastosowaniach, np. dostęp do bazy danych, ściągnięcie pliku przy słabym łączu, szybka aktualizacja www. Dodatkowo użytkownik telnetu nie może być zawirusowany, ponieważ nie odbiera żadnych danych, a jedynie wyniki operacji serwera.

Niestety posiada też wady, z których największą jest możliwość podsłuchania danych do logowania, co daje nieuprawnionej osobie dostęp do wszystkich funkcji serwera.

Podsumowując doszliśmy do wniosku, że przy dzisiejszym rozwoju technologii oraz faktem, że prawie każdy ma dostęp do szybkiego łącza, telnet może spokojnie odejść na emeryturę.

Wygodniejszym rozwiązaniem jest połączenie się z serwerem na przykład przy pomocy

Pulpitu Zdalnego.

# Bibliografia

<http://www.kozgan.neostrada.pl/koziol/telnet/index.html>

<https://pl.wikipedia.org/wiki/Telnet>

<https://tools.ietf.org/html/rfc855>

https://tools.ietf.org/html/rfc854