

## Serwer pocztowy Linux – dowolność aplikacji do opracowania dla zespołu.

### 1. Opis implementacji

Aplikacja została napisana pod system operacyjny Linux. Jest napisana w języku C# w środowisku MonoDevelop 2.4 na platformie Mono.

Główną część aplikacji zawarta jest w klasie Program.cs. W niej inicjowany jest nasłuch na dwóch portach: 25 i 110. Tak jak to jest przyjęte port 25 odpowiada za działanie serwera SMTP, port 110 odpowiada za serwer POP3. Aplikacja jest wielowątkowa, dzięki czemu może korzystać z serwera jednocześnie wielu klientów podłączonych na porcie 25 i 110. Klasy SMTPSerwer.cs odpowiada za obsługę klientów chcących tworzyć i wysyłać pocztę, klasa POP3serwer.cs odpowiada przeglądanie wiadomości przez użytkownika.

Klasa User.cs odpowiada za przechowywanie adresu i hasła użytkownika. Klasa Wiadomosc.cs odpowiada za przechowywanie danych wiadomości. Klasy User i Wiadomosc są umieszczone w kolekcjach List<USER> listaUser i List<Wiadomosc> wiadomosci. Te kolekcje są serializowane poprzez metody w klasie Zapis.cs dzięki czemu przy każdym uruchomieniu aplikacji kolekcje są już zainicjowane. Kolekcje po serializacji są przechowywane w dwóch plikach messages.xml i users.xml które znajdują się w katalogu Mail Server/bin/debug/. Klasa Komunikaty.cs zawiera tablice które wysyła serwer jak i odbiera od klientów. Odbierane komunikaty służą do rozpoznawania poleceń które serwer ma wykonać. Klasa Teksty.cs zawiera metody które odpowiadają za sprawdzanie poprawności wprowadzanych adresów pocztowych.

### 2. Korzystanie z serwera

Ten serwer działa tylko dla hostów w sieci lokalnej, dla kont które są w pliku users.xml Klienci mogą łączyć się z serwerem poprzez telnet. Serwer obsługuje polecenia zaczynające się od słów kluczowych (**wielkość liter ma znaczenie**) wymienionych poniżej. Wyrażenia w <> nie są słowami kluczowymi tylko parametrami polecenia.

- Dla serwera SMTP
  - ➔ MAIL FROM: <adrespocztowy>
  - ➔ RCPT TO: <adrespocztowy>
  - ➔ SUBJECT: <temat>
  - ➔ DATA
  - ➔ QUIT
- Dla serwera POP
  - ➔ USER <adrespocztowy>
  - ➔ PASS <adrespocztowy>
  - ➔ LIST
  - ➔ RETR <numerlistu>
  - ➔ DELE <numerlistu>

### 3. Przykład działania serwera

- Zrzut działania serwera. Przy jego włączeniu wyświetlane są tylko adresy email które znajdują się w nim. On nic nie wyświetla poza tymi adresami , tylko odbiera, obsługuje i wysyła komunikaty do klientów.

```
linuxmint Debug # '/root/Pulpit/Mail Server/Server/Server/bin/Debug/Server.exe'  
jeden@poczta haslo1  
dwa@poczta haslo2  
trzy@poczta haslo3  
█
```

- Przykład obsługi klienta na serwerze SMTP

```
linuxmint ~ # telnet 192.168.2.2 25  
Trying 192.168.2.2...  
Connected to 192.168.2.2.  
Escape character is '^]'.  
220 Serwer Gotowy  
MAIL FROM: jeden@poczta  
250 OK  
RCPT TO: dwa@poczta  
250 OK  
SUBJECT: temat wiadomosci  
250 OK  
DATA  
354 Rozpoczecie przyjmowania tresci wiadomosci e-mail, zakoncz przyciskiem <Enter>  
Tresc wiadomosci  
250 OK  
QUIT  
Connection closed by foreign host.  
linuxmint ~ # █
```

- Przykład obsługi klienta na serwerze POP

```
linuxmint ~ # telnet 192.168.2.2 110
Trying 192.168.2.2...
Connected to 192.168.2.2.
Escape character is '^]'.
+OK POP3 server gotowy
USER dwa@poczta
+OK
PASS haslo2
+OK dwa@poczta maildrop has 3 messages
LIST
+OK 3 messages
RETR 3
+OK
Nadawca: jeden@poczta
Odbiorca: dwa@poczta
Data: 2010-12-21 11:12:59
Temat: temat wiadomosci
Tresc: Tresc wiadomosci
DELE 2
+OK message 2 deleted
QUIT
Connection closed by foreign host.
linuxmint ~ #
```