

Wersja Javy: 1.8.

Program należy najpierw skompilować uruchamiając plik compile.bat, a następnie uruchomić w linii poleceń poprzez

```
java DAS <port> <number>
```

lub uruchamiając jeden z plików testowych Test\_DAS formatu .bat.

Zagadnienia protokołu UDP:

Klasa DAS zawiera pola statyczne: SOCKET typu DatagramSocket, PACKET typu DatagramPacket oraz PORT i NUMBER typu prostego int, opisujące odpowiednio port, na którym otwiera się dany DatagramSocket i wartość, którą dane uruchomienie programu będzie wysyłać w pakiecie klasy DatagramPacket.

Dla pomocy przy logach utworzyłem pole PREFIX typu String z pomocniczą metodą log().

Na każdym nowo utworzonym gnieździe UDP ustawiłem 80-sekundowy TimeOut również dla ułatwienia testowania, aby można było obserwować nadchodzące pakiety do wariantu MASTER, jednocześnie nie trzymając go włączonego zbyt długo.

Każda komunikacja pomiędzy uruchomionymi wariantami programu DAS odbywa się poprzez wysyłanie DatagramPacket'ów z liczbą typu int przetworzoną przez metody klasy ByteBuffer na tablicę bajtów. Adres IP odbiorcy wiadomości to localhost (127.0.0.1), a port zależy od wartości NUMBER.

Wariant MASTER, oczekując na pakiety, ma w gotowości bufor o rozmiarze 1400 bajtów. Po odebraniu pakietu logowane są adres ip i port nadawcy oraz odebrana wartość typu int wyciągnięta z buforu klasy ByteBuffer.

Wariant SLAVE powstaje na skutek obsługi wyjątku SocketException – który mówi, że nie można stworzyć nowego DatagramSocket, ponieważ na danym porcie jest już otwarte gniazdo UDP. Nowe gniazdo tworzone jest poprzez domyślny konstruktor, który wymyślenie nowego portu powierza systemowi operacyjnemu.

Problemy napotkane podczas pracy:

- Problem z wyliczeniem adresu rozgłoszeniowego. Na początku próbowałem wyliczyć go ręcznie, prowadząc operacje bitowe na poszczególnych oktetach adresów, próbując wyliczyć szukany adres poprzez dodanie negacji binarnej maski do adresu sieciowego.
- Okazało się jednak, że nie było takiej potrzeby, ponieważ klasa InterfaceAddress zapewnia metodę getBroadcast() zwracającą adres rozgłoszeniowy dla danego interfejsu sieciowego - metoda pomocnicza showBroadcastAdr().