

	Datum	05.10.2022
Informatik Web Application Engineering		Übungsblatt 1
Thomas Feilhauer		Zu lösen bis 19.10.2022, 8:00 Uhr

Vorbereitungsaufgabe (in der Übungsstunde zu lösen):

Implementieren Sie einen TCP-Server, der auf Port 4711 hört und darüber einen String von einem Client entgegennimmt. Der Server wandelt den empfangenen String in Großbuchstaben um und sendet ihn an den Client zurück. Testen Sie Ihren TCP-Server, indem Sie einen entsprechenden Client implementieren, der dem Server den String "Hello" sendet und nach Empfangen der Antwort des Servers auf der Konsole den Text "Response from server:" vor dem empfangenen String ausgibt.

Aufgabe 1: Sockets**(10 Punkte)**

Implementieren Sie ein vereinfachtes Ping in Java, das nur aus Komponenten für Client und Server besteht, die Router-Komponente aber nicht berücksichtigt. Tatsächlich verwendet Ping das Protokoll ICMP type 8 (echo message) and type 0 (echo reply message), also weder TCP noch UDP. Ihre Ping-Implementierung soll auf UDP aufsetzen und den Serverport 7777 verwenden.

Die Applikation soll aus zwei Komponenten bestehen. Die Server-Komponente soll auf Port 7777 hören. Die Client-Komponente soll über die Konsole aufrufbar sein und über ein Command Line Interface (CLI) verfügen, das sich an das von Ping anlehnt. Ihr Client-Programm sollte auf der Konsole über

```
java PING [-s] {IP_address | host_name} [size] [quantity]
```

aufrufbar sein. Die Groß-/Kleinschreibung aus der Vorlage ist zu berücksichtigen.

Details zum erwarteten Verhalten der Applikation entnehmen Sie bitte

<https://datatracker.ietf.org/doc/html/rfc1739#section-2.2>

PING

Ping is one of the most widely available tools bundled with TCP/IP software packages. Ping uses a series of Internet Control Message Protocol (ICMP) Echo messages to determine if a remote host is active or inactive, and to determine the round-trip delay in communicating with it. The Ping command, referred to as the Packet Internetwork Groper in some references, has the following general format:

```
PING [-s] {IP_address | host_name} [size] [quantity]
```

In the first test below, we ping the host "thumper.bellcore.com" to determine whether it is up and running. This simple use of the command contains no optional parameters.

In the second test, the "-s" parameter tells the system to send an ICMP Echo message every second. The optional "size" parameter specifies that each message should be 64 bytes in length (which is the default size); the optional "quantity" parameter indicates that this test will only send 12 messages (the default is to run the test continuously until interrupted). The results of the second test displays the round-trip delay of each Echo message that is returned to the sending host; at the end of the test, summary statistics are displayed.

=====

```
** SMCVAX$ PING THUMPER.BELLCORE.COM
thumper.bellcore.com is alive

** SMCVAX$ PING -s THUMPER.BELLCORE.COM 64 12
PING THUMPER.BELLCORE.COM (128.96.41.1): 56 data bytes
64 bytes from 128.96.41.1: icmp_seq=0 time=150 ms
64 bytes from 128.96.41.1: icmp_seq=1 time=110 ms
64 bytes from 128.96.41.1: icmp_seq=2 time=130 ms
64 bytes from 128.96.41.1: icmp_seq=3 time=130 ms
64 bytes from 128.96.41.1: icmp_seq=4 time=320 ms
64 bytes from 128.96.41.1: icmp_seq=5 time=110 ms
64 bytes from 128.96.41.1: icmp_seq=6 time=440 ms
64 bytes from 128.96.41.1: icmp_seq=7 time=90 ms
64 bytes from 128.96.41.1: icmp_seq=9 time=100 ms
64 bytes from 128.96.41.1: icmp_seq=10 time=110 ms

---THUMPER.BELLCORE.COM PING Statistics---
12 packets transmitted, 10 packets received, 16% packet loss
round-trip (ms)  min/avg/max = 90/169/440
```

Testen Sie Ihre Applikation, indem Sie Client und Server auf unterschiedlichen Hosts starten. Geben Sie zu Ihrem Programm sowohl den Sourcecode als auch die kompilierten Class-Dateien ab. Dokumentieren Sie, wie die Server-Komponente gestartet werden kann.