

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	

## Aufgabe 1a:

### Programmierung eines einfachen Client/Server-Systems

Es ist eine einfache Client-/Server-Anwendung für das TCP-Transportprotokoll zu implementieren. Die Realisierung kann in Java oder Python, (aber auch in C oder C++) erfolgen.

Auf eine GUI für den Client sollte aus Zeitgründen verzichtet werden, es genügt z.B. eine einfache Bedienung über die Kommandozeile mit geeigneten Parametern und dann während der Laufzeit die Eingabe über STDIN.

Mit dem Programm NETCAT kann man beide zu entwickelnden Komponenten zurzeit gut testen! Tatsächlich brauchen Sie – wenn Sie NETCAT verwenden – keinen Client, Sie könnten alles damit „erledigen“. Jedoch geht es ja gerade darum, dass Sie Clients \*UND\* Server programmieren können. Also bitte beide Komponenten implementieren.

#### Funktion des Clients:

Der Client wird unter Angabe der Adresse (IP bzw. Rechnername) des Servers, der die Anfragen beantworten soll, sowie des TCP-Ports, auf dem der Server-Prozess auf Anfragen wartet, gestartet. Damit Sie das alles einfach lokal entwickeln und testen können, verwenden Sie einfach die LOOPBACK-Adresse (127.0.0.1), die auf jedem TCP/IP-System definiert ist und quasi eine Schleife implementiert, die Daten an einen Server auf dem eigenen Rechner schickt, ohne dass diese überhaupt diesen Rechner verlassen.

Der Client nimmt nach erfolgreichem Aufbau einer TCP-Verbindung zum angegebenen TCP-Port auf dem angegebenen Server interaktiv vom Benutzer eine beliebige Zeichenkette mit definierter maximalen Länge entgegen. In dieser Zeichenkette steht zunächst ein Kommando, das bestimmt, was der Server-Prozess ausführen soll (siehe dazu die Tabelle der Kommandos auf der folgenden Seite).

Alle Antworten des Servers – Ergebnisse bzw. Fehlermeldungen – sowie Fehlermeldungen des Betriebssystems – z.B. bei Verbindungsfehlern – sind vom Client ohne weitere Veränderungen auszugeben.

#### Funktion des Servers:

Der Server-Prozess, der die Anfragen des Clients beantworten soll, wird gestartet und wartet auf einem beim Aufruf definierten TCP-Port auf Verbindungen von beliebigen Clients. Beim Aufruf wird auch ein Passwort übergeben, dass zur Prüfung von kritischen Kommandos der Clients genutzt werden kann. Der Server soll von mehreren Clients gleichzeitig genutzt werden können.

Der Server nimmt nach erfolgreichem Aufbau einer TCP-Verbindung von dem jeweiligen Client beliebige Zeichenketten bis zu der definierten maximalen Länge und definiertem Ende-

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	

Zeichen an. Innerhalb der Zeichenkette wird am Beginn nach dem Schlüsselwort eines unterstützten Kommandos gesucht und dieses dann auf den Rest der Zeichenkette angewendet. Abhängig von dem Kommando wird evtl. auch ein Passwort übergeben, dass dann mit dem beim Aufruf vorgegebenem Passwort verglichen werden muss.

Der Server-Prozess sendet einen Status über die erfolgreiche Ausführung eines Kommandos mit entsprechender Ausgabe oder eine Fehlermeldung mit Details zum aufgetretenen Fehler an den Client zurück.

### Protokollspezifikation:

Jede der Nachrichten von Client bzw. Server ist in UTF-8-Codierung als Bytefolge zu senden, die mit Newline (\n bzw. ASCII #10) abgeschlossen ist. Jede Nachricht darf einschließlich des abschließenden Newline-Zeichens, das nicht weiterverarbeitet wird, maximal 255 Bytes lang sein.

Jede Nachricht eines Clients beginnt mit einem Kommando spezifiziert durch das zugeordnete Schlüsselwort (anzugeben in GROSSBUCHSTABEN gemäß folgender Tabelle). Wenn dem Kommando weitere Parameter mitgegeben werden können, muss nach dem Kommando ein Leerzeichen als Trennzeichen folgen.

Kommando	Schlüsselwort	Parameter	Bemerkungen
Umwandeln einer Zeichenkette in Kleinbuchstaben	<b>LOWERCASE</b>	<string>	Andere Zeichen als Buchstaben bleiben unverändert.  Leerer <string> ist nicht erlaubt!
Umwandeln einer Zeichenkette in Großbuchstaben	<b>UPPERCASE</b>	<string>	Andere Zeichen als Buchstaben bleiben unverändert.  Leerer <string> ist nicht erlaubt!
Umkehrung der Zeichenkette „abc“ → „cba“	<b>REVERSE</b>	<string>	Leerer <string> ist nicht erlaubt!
Sitzungsende	<b>BYE</b>	–	Es wird nur das Schlüsselwort übergeben.
Sitzungsende und Herunterfahren des Servers	<b>SHUTDOWN</b>	<password>	Der Zeichensatz für das Passwort ist nicht eingeschränkt.  Leeres <password> ist nicht erlaubt!  Wenn das Passwort übereinstimmt, wird bzgl. des Clients verfahren, als wenn „BYE“ übermittelt worden wäre.  Es werden keine neuen Verbindungen mehr angenommen.

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	

	Sobald der Server keine weiteren aktiven Verbindungen (aktiv bedeutet, das innerhalb der letzten 30 Sekunden ein Kommando übergeben wurde) zu anderen Clients mehr geöffnet hat, beendet sich der Server-Prozess selbst und gibt alle Ressourcen incl. des TCP-Ports frei.
--	--

Jede Nachricht des Servers beginnt mit einem der beiden Schlüsselwörter „OK“ bzw. „ERROR“ gefolgt von einem Leerzeichen als Trennzeichen und weiteren Zeichenketten gemäß folgender Tabelle:

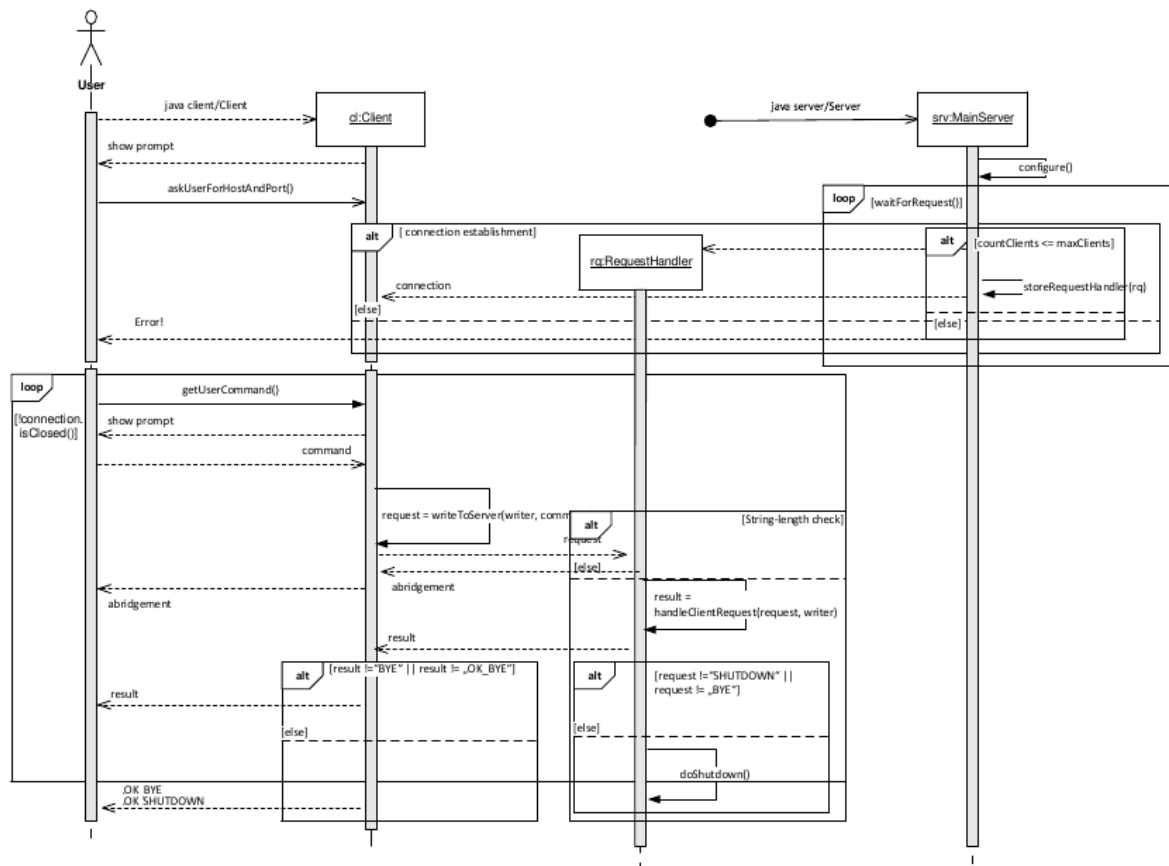
Schlüsselwort Client	Ausgabe Server ohne Fehler	Ausgabe Server bei Fehler
LOWERCASE	OK <string after processing>	ERROR <explanation>
UPPERCASE	OK <string after processing>	ERROR <explanation>
REVERSE	OK <string after processing>	ERROR <explanation>
BYE	OK BYE	ERROR <explanation>
SHUTDOWN	OK SHUTDOWN	ERROR <explanation>

Im Fehlerfall („ERROR“) muss die Zeichenkette <explanation> innerhalb der zur Verfügung stehenden Länge kurz den Grund für den Fehler als Freitext beschreiben, z.B. „ERROR STRING TOO LONG“.

Wird kein gültiges Schlüsselwort erkannt, muss dies mit der Fehlermeldung „ERROR UNKNOWN COMMAND“ vom Server an den Client bekannt gegeben werden. Wird zwar ein gültiges Schlüsselwort erkannt, genügt aber die verbleibende Zeichenkette nicht dem Protokoll, muss dies mit der Fehlermeldung „ERROR SYNTAX ERROR <further explanation>“ näher detailliert werden.

Das folgende Sequenzdiagramm zeigt das Protokoll noch einmal im Gesamtzusammenhang:

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	



### Vorgaben für die Implementierung:

Besonderer Wert muss bei allen Netzwerk-Anwendungen auf die Stabilität und Fehlertoleranz des Servers gelegt werden. Fehlt diese, kann es keine verlässliche Funktion des Gesamtsystems geben.

Insbesondere muss der Server mit – beabsichtigtem aber auch unbeabsichtigtem – Fehlverhalten der Clients bzw. Problemen im Netzwerk umgehen. Hierzu gehören Verbindungsabbrüche genauso dazu wie Protokollfehler oder Überlast durch zu viele Clients.

Im Einzelnen bedeutet dies für Sie:

- **MAX\_CLIENTS ::= 3**

Es muss eine maximale Anzahl von Clients, die gleichzeitig unterstützt werden können, definiert werden. Der gewählte Wert von drei erlaubt ein einfaches Testen und muss natürlich in der Praxis anders definiert werden.

- **Protokollkonformität:**

Alle Nachrichten müssen komplett geprüft werden (maximale Länge, Zeichensatz, Trennzeichen, Newline-Zeichen am Ende, bekanntes Kommando), bevor diese ausgeführt werden. Nachrichten, die nicht protokollkonform sind, werden durch eine entsprechende Fehlermeldung und der Detaillierung des ersten gefundenen Fehlers quittiert.

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	

Fehlt das NEWLINE-Zeichen als letztes Zeichen einer Nachricht, ist die komplette Nachricht zu verwerfen. Ebenso wenn in einer Nachricht mehrere NEWLINE-Zeichen vorkommen, obwohl die maximale Länge noch nicht erreicht ist.

- **Abfangen von Exceptions:**

Die Prozesse müssen jederzeit mit „unerwarteten“ Ereignissen zurechtkommen und diese entweder durch eine geeignete Fehlermeldung dem Benutzer am Bildschirm zur Kenntnis bringen oder eine sinnvolle weitere Funktion sicherstellen. Keinesfalls darf ein Prozess sich aufgrund eines nicht abgefangenen Fehlers vorzeitig beenden.

Für die Abnahme während des Praktikums müssen Sie nachweisen, dass:

- Ihr Server gleichzeitig eine maximale Anzahl von drei Clients bedienen kann
- Ihr Server bei der erreichten maximalen Anzahl von Clients weitere Clients nicht bedient
- Ihr Server alle Kommandos einwandfrei erkennt und bearbeiten kann, und es hierbei auch bei Umlauten zu keinen Problemen kommt (Achtung: UTF-8 Codierung ist durchgehend nötig)
- Ihr Server nicht protokollkonforme Nachrichten erkennt und mit einem Fehler quittiert
- Ihr Server den Shutdown nur durchführt bei korrektem Passwort
- Ihr Server trotz des Shutdown-Befehls andere verbundene Clients weiterhin bedient, bis auch diese länger als 30 Sekunden keine Nachrichten mehr geschickt haben
- Ihr Server bei erteiltem Shutdown-Befehl keine neuen Verbindungen zu weiteren Clients mehr akzeptiert

Wenn Sie sich auf die Abnahme vorbereiten, müssen Sie die obigen Tests selbst durchgeführt haben. D.h. keine Abnahme ohne die Erklärung, dass Sie sich selbst überzeugt haben, dass die Programme machen, was sie machen sollen!

BAI4-RN	Praktikum Rechnernetze – Aufgabe 1	MDE/KSS
SoSe 2024	Entwicklung eines einfachen Client/Server-Systems	

### **Aufgabe 1b: Analyse von Netzwerkverkehr**

Zeichnen Sie eine Verbindung zwischen Ihrem Client und Ihrem TCP-Server auf, wobei Sie die Einstellungen so treffen, dass keine anderen Endgeräte zugeordneten Verbindungen oder Pakete aufgezeichnet werden. Übergeben Sie zwei oder drei Kommandos mit nicht zu langen Zeichenketten an den Server bevor Sie die Verbindung wieder mit dem Kommando BYE beenden. Der Server bleibt aktiv, es werden keine weiteren Verbindungen aufgebaut.

Verwenden Sie dazu einen Netzwerksniffer wie z.B. WIRESHARK. Analysieren Sie das Anwendungsprotokoll und untersuchen Sie den der Verbindung zugeordneten TCP-Strom mit den Mitteln des Werkzeugs. Machen Sie sich damit vertraut, wie Protokolldaten – insbesondere Header – angezeigt werden und wie Nutzdaten inspiziert werden können. Auch das geht ohne Internet auf dem LOOPBACK-Interface.