



<i>BAI4-RNP</i>	<i>Praktikum Rechnernetze</i>	<i>KSS/SLZ</i>
<i>WS14/15</i>	Aufgabe 1: Netzwerktools & -programmierung	Seite 1 von 3

Vorbemerkungen

Ziele dieser Aufgabe sind:

- der Einsatz von Werkzeugen zur Arbeit mit Netzwerken. Machen Sie sich hierzu vor dem Praktikum mit den Programmen in [diesem Dokument](#) ( / ) vertraut, ggf. unter zusätzlicher Verwendung der einschlägigen Manual Pages
- Netzwerkprogrammierung: Entwicklung einer einfachen vernetzten Client-/Serveranwendung

Teil 1: Linux-Werkzeuge zur Arbeit mit Netzwerken

Für die nachfolgenden Punkte ist ein Protokoll anzufertigen und zur Abnahme vorzulegen.

- (1) Welche Informationen erhält man mit dem Programm `/sbin/ifconfig`?
- (2) Messen Sie die Round Trip Time für ein 1000-Datenbyte-Paket nach www.google.de, nach www.haw-hamburg.de und zum Rechner `lab22` jeweils als Durchschnittswerte von 10 Stichproben.
- (3) Finden Sie heraus, welches Programm auf Ihrem Rechner den Port 9400 benutzt. Welches Protokoll setzt dieses Programm auf der Transportschicht ein? In welchem Systemaufruf "hängt" das Programm? (Für letzteres brauchen Sie noch das Programm `strace` mit der Option `-p <pid>`)
- (4) Stellen Sie eine TCP-Socketverbindung zwischen der Shell (Standardein-/ausgabe) und dem Server www.bsh.de auf dessen Port 80 her. Senden Sie die Zeichenkette „GET“ und protokollieren Sie die Antwort des Servers. Um welches Protokoll handelt es sich hier?
- (5) Loggen Sie sich von Ihrem Arbeitsplatzrechner aus auf dem zugehörigen Gegenstellenrechner (siehe [Tabelle](#)) ein und lassen Sie sich dort mit dem Dateimanager (Befehl `konqueror` von der Kommandozeile) den Inhalt des dortigen `/tmp`-Verzeichnisses zeigen.

Teil 2: Netzwerkprogrammierung

Es ist eine einfache Client-/Serveranwendung auf TCP-Basis zu implementieren. Die Software ist vor dem Praktikum zu entwerfen; das Entwurfsdokument muss zum Praktikum vorliegen.

Der **Client** nimmt von Benutzer eine beliebige Zeichenkette (aber begrenzter Länge, vgl. unten) entgegen sowie einen Operationsbefehl (Umwandeln in Grossbuchstaben, in Kleinbuchstaben, Spiegeln, d.h. „aBcd“ wird „dcBa“, Sitzung beenden oder Server herunterfahren). Die Antwort des Servers (Ergebnis bzw. Fehlermeldung) gibt er aus.

Der **Server** empfängt von einem Client einen Operationsbefehl sowie die zu bearbeitende Zeichenkette und sendet dem Client als Antwort die bearbeitete Zeichenkette. Der Server soll von

<i>BAI4-RNP</i>	<i>Praktikum Rechnernetze</i>	<i>KSS/SLZ</i>
<i>WS14/15</i>	Aufgabe 1: Netzwerktools & -programmierung	Seite 2 von 3

mehreren Clients gleichzeitig genutzt werden können. Das Herunterfahren soll über einen speziellen Befehl und Passwort veranlasst werden.

Protokollspezifikation:

Jede der nachfolgend dargestellten Nachrichten ist als Bytefolge gemäss UTF-8 Codierung zu senden, die mit Newline (`\n` bzw. ASCII #10) abgeschlossen ist. Zwischen den Schlüsselwörtern (in GROSSBUCHSTABEN) und den darauf folgenden Zeichenketten muss ein Blank als Trenner stehen. Eine Nachricht darf maximal 255 Bytes lang sein.

Operation	Client	Server
Umwandeln in Kleinbuchstaben	LOWERCASE <string>	-----> <----- OK <string>
Umwandeln in Grossbuchstaben	UPPERCASE <string>	-----> <----- OK <string>
Spiegeln	REVERSE <string>	-----> <----- OK <string>
Sitzungsende (Danach ist die Verbindung zu schliessen.)	BYE	-----> <----- BYE
Herunterfahren des Servers (siehe Hinweis unten. Danach ist die Verbindung zu schliessen)	SHUTDOWN <password>	-----> <----- OK_BYE oder ERROR <message>
Ungültiger Befehl	<ungültiger befehl>	-----> <----- ERROR <message>

Allgemeine Randbedingungen und Hinweise

- Nach dem SHUTDOWN-Kommando nimmt der Server keine neuen Sitzungen mehr an. Sobald alle noch laufenden Sitzungen zum Ende gekommen sind (Sitzungsende-Dialog), fährt der Server herunter.
- Besonderer Wert wird auf Fehlertoleranz und Stabilität des Servers gelegt. Er soll auch bei Fehlverhalten von Clients (z.B. Verbindungsabbrüche, Protokollfehler oder Überlast durch zu viele Clients) weiterlaufen. Nach dem Shutdown-Befehl soll ein geordnetes Herunterfahren aller Komponenten gewährleistet sein.
- Der Listening-Port muss zur Laufzeit einstellbar sein (Parameter). Denken Sie beim Testen

<i>BAI4-RNP</i>	<i>Praktikum Rechnernetze</i>	<i>KSS/SLZ</i>
<i>WS14/15</i>	Aufgabe 1: Netzwerktools & -programmierung	Seite 3 von 3

daran, dass Ports unterhalb von 50000 vom Betriebssystem reserviert sein können.

- Die Realisierung kann in Java, C oder C++ erfolgen. Eine GUI ist nicht zwingend erforderlich, es genügt auch eine Bedienung über die Kommandozeile.
- Die Programme sollen sowohl in der IDE (sofern verwendet) als auch auf der Kommandozeile starten können.
- Zum Testen empfiehlt sich das Programm *socat*.

Downloads und Links:

- Dokument zu den Programmen für Teil 1:
<http://users.informatik.haw-hamburg.de/~scotty/pub/Rechnernetze/tools/README-DE.pdf>
- Zuordnungstabelle der Gegenstellenrechner:
<http://users.informatik.haw-hamburg.de/~scotty/pub/Rechnernetze/Gegenstellenzuordnung.html>
- weitere nützliche Dokumente und Downloads zum Praktikum RNP:
<http://users.informatik.haw-hamburg.de/~scotty/pub/Rechnernetze/> ,
<http://users.informatik.haw-hamburg.de/~scotty/pub/Rechnernetze/tools/>.