Prof. Dr. Stefan Brass
Institut für Informatik
MLU Halle-Wittenberg

11. April 2016

Grundlagen des WWW
— 1. Übungsblatt: Internet —

Dieses Blatt enthält einige Aufgaben für die erste Übung. Es handelt sich dabei um Präsenzübungen, die nicht abgegeben werden müssen. Falls Sie allerdings die Übung verpassen, sollten Sie einen ausreichend großen Teil der Aufgaben bearbeiten und dem Dozenten per EMail schicken (mit "`www16`" in der Betreff-Zeile). Die letzten Aufgaben sind optional. Es empfiehlt sich allerdings, sich wenigstens einen Überblick über die Netzwerk-Programmierung zu verschaffen.

a) Stellen Sie sicher, dass Linux auf Ihrem Pool-Rechner läuft. Falls auf dem Rechner gerade Windows läuft, starten Sie ihn neu und wählen beim Neustart Linux aus. Sollten Sie noch kein Benutzerkonto für die Pools der Informatik besitzen, lassen Sie Ihr Universitäts-Benutzerkonto bei der Pool-Aufsicht (Raum 3.33) oder bei Herrn Trull oder Herrn Ohme (Raum 3.20) freischalten. Es dauert allerdings einige Stunden, bis das Login dann möglich ist. Wenn Sie von zu Hause aus mit ssh/PuTTY auf einem Linux-Rechner der Informatik arbeiten wollen, können Sie sich mit `anubis.informatik.uni-halle.de` verbinden.

b) Öffnen Sie ein Kommandozeilen-Fenster (Terminal, Shell). Machen Sie sich mit dem Befehl `man` vertraut: Er wird "`man Programm`" aufgerufen und zeigt dann eine Hilfe (Handbuch-Seite) zum Programm `Programm` an. Natürlich geht auch "`man man`".

   Durch Drücken von "`q`" beenden Sie die Anzeige, mit der Leertaste kommen Sie eine Seite weiter, mit "Enter/Return" eine Zeile. Die Pfeiltasten funktionieren auch. Mit "`/s`" können Sie in der Ausgabe nach der Zeichenkette `s` suchen. Mit `n` kommen Sie dann zum nächsten Vorkommen der Zeichenkette.

   Das Handbuch gliedert sich in mehrere Abschnitte. Wenn man keinen Abschnitt auswählt, werden alle in einer festgelegten Reihenfolge durchsucht und der erste Treffer geliefert. Wenn man z.B. nach der Bibliotheksfunktion `printf` sucht, muss man den Abschnitt 3 für Bibliotheksfunktionen explizit angeben (weil es "`printf`" auch als Benutzerkommando in Abschnitt 1 gibt). Das geht mit "`man 3 printf`". Systemaufrufe stehen in Abschnitt 2.

   Nach Stichworten im Handbuch kann man suchen mit "`apropos Stichwort`", alternativ mit "`man -k Stichwort`". GNU Software wird häufig durch Info-Seiten besser beschrieben als durch die "Manpage", z.B. "`info mkdir`".

c) Erinnern Sie sich an wichtige Linux-Kommandos wie

   - `ls`, `ls -al`, `ls -d` (ggf. auch `du`, `find`),
   - `cd`, `pwd`,

   - `mkdir`, `rmdir`,
   - `cp`, `mv`,
   - `rm`, `rm -i`, `rm -r` (rekursives Löschen!),
   - `chmod` (z.B. "`chmod go-r Geheim`").
   - `ps`, `ps -ef`, `kill`
   - `fgrep`, `grep`

   Editoren unter Linux sind z.B. `gedit`, `vi/gvim`, `emacs`, `pico/nano`. Bei Bedarf finden Sie weitere Informationen zu Linux z.B. in

   - [http://tille.garrels.be/training/tldp/]
   - [https://help.ubuntu.com/lts/ubuntu-help/index.html]
   - [http://users.informatik.uni-halle.de/~brass/oop10/c1_intro.pdf]

d) Schauen Sie sich mit `man` die Hilfen für die Befehle `whois`, `traceroute`, `ping`, `netstat` und `arp` an. Probieren Sie die `whois`- und `traceroute`-Beispiele aus der Vorlesung aus.

e) Was ist die IP-Adresse des Rechners, an dem Sie arbeiten?

f) Was ist die Ethernet-Adresse des Rechners, an dem Sie arbeiten?

g) Zu welchem Gateway werden IP-Pakete geschickt, die außerhalb des Subnetzes liegen? Geben Sie die IP-Adresse des Gateways an.

h) Finden Sie einen Ansprechpartner für den Rechner `130.75.26.7`. Wenn Sie die `whois`-Abfrage über eine Web-Schnittstelle machen wollen, können Sie z.B. folgende Seiten benutzen:

   - [http://www.whois.com/whois/]
   - [http://www.heise.de/netze/tools/whois/]

i) Geben Sie einige Beispiele für Portnummern auf Ihrem Rechner, die aktuell benutzt werden.

j) Probieren Sie `traceroute` (Windows: `tracert`) aus für folgende Rechner:

   - `www.sis.pitt.edu`,
   - `www.acm.org`,
   - `www.computer.org` (welchen Dienst benutzt die Computer Society?),
   - `www.nico-nle.de`.

Falls auf Ihrem Rechner kein `traceroute` installiert ist, können Sie auch eine der folgenden Webseiten benutzen:

- [http://network-tools.com/]
- [http://www.traceroute.org/]
- [https://www.noc.dfn.de/lg/]

k) Besorgen Sie sich den RFC 1180. Wer ist Autor dieses RFC? RFCs gibt es auf vielen Webseiten, u.a.:

- [http://www.rfc-editor.org/]
- [http://www.rfc-archive.org/]

l) Laden Sie das Programm unter der Adresse

[http://www.informatik.uni-halle.de/˜brass/www16/dt_client.c]

herunter und übersetzen Sie es mit dem Befehl

```
gcc -o dt_client dt_client.c
```

Das erzeugte Programm `dt_client` ist ein Client des "Day-Time-Servers", es funktioniert aber auch mit dem Server aus Aufgabe j). Führen Sie das Programm aus. Sie können Rechnernamen und Port-Nummer wie bei dem Programm telnet explizit angeben. Die Syntax ist: "`dt_client host port`". Das Programm kann auch unter Windows ubersetzt werden, wenn man zu Beginn das Präprozessorsymbol `OS_WINDOWS` definiert. Was tut dieses Programm und was ist dessen Ausgabe (mit und ohne Parameter)?

Eine Einführung in C++ für Java-Programmierer finden Sie z.B. in

[http://www.informatik.uni-halle.de/˜brass/dbi15/cpp.pdf]

Das obige Programm verwendet allerdings nur die Sprache "C" (kleine, nicht objektorientierte Teilmenge von C++). Eine Einführung in die Socket-Verwendung in Java finden Sie z.B. in

- [http://docs.oracle.com/javase/tutorial/networking/sockets/]
- [http://javabeginners.de/Netzwerk/Socketverbindung.php]

m) Beschaffen Sie sich nun das Programm unter folgender Adresse:

[http://www.informatik.uni-halle.de/˜brass/www16/server1.c]

Finden Sie in dem Programm die eingebaute Port-Nummer des Servers. Übersetzen Sie es wieder und starten Sie es mit "`./server1`". Greifen Sie dann mit `telnet` oder `dt_client` auf den Server zu. Sie müssen dazu Rechnernamen und Portnummer wie oben erläutert als Kommandozeilen-Parameter angeben. Was gibt der Server aus? (Hinweis: Sie können den Server danach mit Strg-C wieder beenden.)

n*) Betrachten Sie das Programm unter

[http://www.informatik.uni-halle.de/˜brass/www16/server2.c]

Finden Sie heraus, was es tut. In der vorliegenden Version wird die Anfrage vom Klienten erst ausgegeben, sobald sie komplett empfangen wurde (also nach der Leerzeile) oder der Puffer voll ist. Ändern Sie das Programm so, dass die einzelnen Zeichen unmittelbar nach dem Empfang ausgegeben werden. Lassen Sie sich nicht davon verwirren, dass aufgrund der Pufferung der Zeichenströme die Zeichen tatsächlich nur zeilenweise verschickt und entsprechend vom Server ausgegeben werden.

o*) Ändern Sie das Programm nun so ab, dass die Verbindung zum Klienten nach etwa 10 Sekunden abgebrochen wird. Sie können dies erreichen, indem sie die Verbindung mit

```
fcntl(socket, F SETFL, O_NONBLOCK);
```

in den nicht-blockierenden Modus versetzen und mit `usleep` immer eine Zehntelsekunde warten, bevor Sie erneut Daten abfragen. Im nicht-blockierenden Modus gibt `read` den Wert `-1` zurück und setzt `errno` auf `EAGAIN`, falls noch keine neuen Daten angekommen sind. Mit `time(NULL)` können Sie die aktuelle Systemzeit in Sekunden ermitteln. Sie können sich z.B. eine Funktion schreiben, die wie `read` im blockierenden Modus auf ein eingehendes Zeichen wartet, aber nur bis maximal zehn Sekunden nach Entgegennahme der Anfrage.