

Aufgabe 7: simail (12 Punkte)

Bearbeitung in Zweier-Gruppen

10.05.2011

Schreiben Sie ein SMTP-Client-Programm **simail** (simple mail), das zum Versenden von E-Mails über einen fest definierten Server verwendet werden kann. Das Programm wird wie folgt aufgerufen:

```
simail [-s <subject>] <address>
```

Das Programm versendet dann eine E-Mail mit dem optionalen Betreff `subject` an die Empfängeradresse `address` über den SMTP-Server `lists.informatik.uni-erlangen.de` (Port 25, `getaddrinfo(3)`, `socket(2)`, `connect(2)`). Die Textnachricht (Body-Sektion) der E-Mail wird von der Standardeingabe gelesen (bis EOF).

Die Mail soll die Header-Felder `From`, `To` und optional `Subject` enthalten. Die Header-Sektion ist durch eine Leerzeile ("`\r\n`") vom Body der E-Mail getrennt. Die Kommunikation mit dem Server ist exemplarisch in folgendem Beispieldialog festgehalten (Client fettgedruckt, Server nicht fettgedruckt):

```
220 lists.informatik.uni-erlangen.de ESMTP spoken here
HELO faui48a.informatik.uni-erlangen.de
250 lists.informatik.uni-erlangen.de
MAIL FROM: <mike@faui48a.informatik.uni-erlangen.de>
250 2.1.0 Ok
RCPT TO: <i4sp@informatik.uni-erlangen.de>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>. <CR><LF>
From: Michael Stilkerich <mike@faui48a.informatik.uni-erlangen.de>
To: <i4sp@informatik.uni-erlangen.de>
Subject: Hallo Liste
```

**Dies ist der Body der Mail. Auch hier werden Zeilen mit <CRLF> getrennt.
..Diese Zeile beginnt in der Eingabe mit nur einem Punkt.**

```
.
```

```
250 2.0.0 Ok: queued as 65AC33E9A9
```

QUIT

```
221 2.0.0 Bye
```

Hinweise

- Das Programm soll sowohl IPv4 als auch IPv6 unterstützen. Es muss davon ausgegangen werden, dass der Server DNS-Einträge für beide Protokolle enthält, aber nur über ein Protokoll auf Port 25 erreichbar ist. `lists` ist über IPv4 und IPv6 erreichbar.
 - Durch Rewriting auf den Uni-Mailservern kann es dazu kommen, dass die Absenderadresse einer zugestellten Mail nicht mehr mit der ursprünglich von simail eingetragenen Adresse übereinstimmt.
 - Zur Bestimmung des vollständigen Rechnernamens (FQDN) des Client-Rechners bestimmen Sie zunächst den Kurznamen mit `gethostname(3)` und führen Sie dann einen DNS-Lookup mit `getaddrinfo(3)` mit dem hints-Flag `AI_CANONNAME` durch. Das Feld `ai_canonname` der addrinfo-Struktur enthält dann den FQDN, der im HELO verwendet werden soll.
 - Die Absenderadresse setzt sich zusammen aus dem lokalen Benutzernamen (`getuid(2)`, `getpwuid(3)`) und dem FQDN des Client-Rechners. Im `From`-Header soll außerdem der volle Name des Benutzers enthalten sein. Dieser findet sich im Feld `pw_gecos` der `passwd`-Struktur als Teilstring vor dem ersten Komma.
 - Alle an den Server übertragenen Zeilenumbrüche, müssen von der Form Carriage-Return `<CR>` Linefeed `<LF>` sein ("`\r\n`"). Wenn Zeilenumbrüche auf dem Client-System nur aus einem `<LF>` bestehen, sind diese in der Übertragung um das fehlende `<CR>` zu ergänzen.
 - Das Ende des Body wird durch einen Punkt in einer eigenen Zeile signalisiert, abgetrennt mit "`\r\n`-Zeilenumbrüchen. Zeilen im Body, die mit einem Punkt beginnen, ist ein weiterer Punkt voranzustellen.
 - Jede Antwort des Servers beginnt mit einem dreiziffrigen Statuscode. Prüfen Sie in jedem Schritt, ob die Antwort des Servers den erwarteten Statuscode enthält (entsprechend obigem Beispiel). Weicht der Statuscode von dem erwarteten Wert ab, soll das Programm die letzte Antwort des Servers anzeigen und sich mit einem Fehler beenden.
 - Die Spezifikation des SMTP-Protokolls finden Sie zur Information unter
<http://www.ietf.org/rfc/rfc2821.txt>
- Es soll jedoch nur die in dieser Aufgabenstellung beschriebene Funktionalität implementiert werden; bei Abweichungen vom Standard hat die Aufgabenstellung Vorrang.

Abgabe: bis spätestens Donnerstag, 19.5.2011, 17:30 Uhr