

Aufgabe 1: snail (12.0 Punkte)

Schreiben Sie ein SMTP-Client-Programm **snail** (**s**imple **n**atty **m**ail), das zum Versenden von E-Mails über einen fest definierten Server verwendet werden kann. Das Programm wird wie folgt aufgerufen:

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

Das Programm versendet dann eine E-Mail mit dem optionalen Betreff *subject* an die Empfängeradresse *address* über den SMTP-Server *faii40.cs.fau.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. Gibt der Benutzer keinen Betreff an, so soll das *Subject*-Feld im Header entfallen. 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 faii40.informatik.uni-erlangen.de ESMTP spoken here
HELO faii06j.informatik.uni-erlangen.de
250 faii40.informatik.uni-erlangen.de
MAIL FROM: <be15piel@faii06j.informatik.uni-erlangen.de>
250 2.1.0 Ok
RCPT TO: <hansdampf@example.org>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: Max Mustermann <be15piel@faii06j.informatik.uni-erlangen.de>
To: <hansdampf@example.org>
Subject: Hallo Hans
```

Dies ist der Body der Mail. Auch hier werden Zeilen mit `\r\n` 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
```

TODO TODO TODO: Makefile schreiben lassen TODO TODO TODO

Hinweise zur Aufgabe:

- Zur Verbesserung der Lesbarkeit Ihres Programmes verwenden Sie für die Kommunikation mit dem Server die Funktionen **fprintf(3)**, **fscanf(3)** und **getc(3)**. Die genannten Funktionen benötigen `FILE*`. Mit Hilfe der Funktion **fdopen(3)** können Sie zu einen Filedeskriptor einen korrespondierenden `FILE*` erzeugen.
- Überlegen Sie außerdem, welche Funktionalität in Funktionen ausgelagert werden kann, um die Lesbarkeit Ihres Programmes zu verbessern und doppelten Quellcode zu vermeiden.
- 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.
- Zur Bestimmung des vollständigen Rechnernamens (FQDN) des Client-Rechners bestimmen Sie zunächst den Kurznamen mit **gethostname(3)** und führen 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.
- Durch Rewriting auf den Uni-Mailservern kann es dazu kommen, dass die Absenderadresse einer zugestellten Mail nicht mehr mit der ursprünglich von **snail** eingetragenen Adresse übereinstimmt.
- Alle an den Server übertragenen Zeilenumbrüche müssen im DOS-Format (`\r\n`) sein. Wenn Zeilenumbrüche auf dem Client-System nur aus einem `\n` bestehen, sind diese in der Übertragung um das fehlende `\r` zu ergänzen.
- Das Ende des Bodys wird dem Server 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.
- Zum Debuggen können Sie sich die Kommunikation mit dem Server wie im obigen Beispieldialog ausgeben lassen. Im fertigen Programm sollen diese Kommunikationsdetails jedoch vor dem Benutzer verborgen bleiben.
- 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 ausgeben und sich mit einem Fehler beenden. Sie dürfen davon ausgehen, dass die Antwortzeile maximal 512 Zeichen lang ist.

Hinweise zur Abgabe:

Erforderliche Dateien: `snail.c`

Bearbeitung: Einzel

Bearbeitungszeit: 10 Werktage

Abgabezeit: 17:30 Uhr