Prof. Franz Schweiggert | Abteilung Angewandte Informationsverarbeitung | 4. Juli 2006
Christian Ehrhardt | Blatt 10

Systemnahe Software (SS 2006)

Abgabetermin 11.Juli 2006

POP3 oder Das richtige Leben (10 Punkte)

POP3 (Post Office Protocol Version 3) ist ein sehr einfaches Protokoll, das eingesetzt wird, um Mails von einem Server herunterzuladen. Praktisch alle Mail-Clients sind in der Lage, mit Hilfe dieses Protokolls Nachrichten von einem POP3-Server herunterzuladen und anzuzeigen.
Eure Aufgabe in diesem Blatt ist es, einen POP3-Server zu schreiben, der eine Test-Mailbox so zur Verfügung stellt, daß ein von Euch frei wählbarer Mail-Client den Inhalt dieser Mailbox über POP3 abrufen und bearbeiten kann. Die Funktionsweise des POP3-Protokolls wird weiter unten beschrieben.
Der Server sollte so implementiert sein, daß er mehrere Verbindungen gleichzeitig abhandeln kann, es empfiehlt sich also ein "concurrent-server".

Test-Mailbox

Als Test-Mailbox wird ein (hart im Programm angegebenes) Verzeichnis verwendet, das pro Mail in der Mailbox eine einzelne Datei enthält. Eine solche Test-Mailbox kann als tar.gz-Archiv (entpacken mit `tar -xvzf mailbox.tar.gz`) auf der Homepage der Vorlesung heruntergeladen werden. Außerdem liefern die Dateien `mailbox.h` und `mailbox.c` eine Funktion, die den Inhalt des Verzeichnisses in geeigneten Datenstrukturen zur Verfügung stellt.
Wer die Test-Mailbox mit eigenen Dateien füllen möchte sollte auf zwei Dinge achten:

- Die Mails sollten aussehen wie normale Mails mit From und Subject, weil sonst der Mail-Client möglicherweise Probleme bei der Darstellung der Mails hat.
- Die Mails sollten der Einfachheit halber keine Zeilen enthalten, die mit einem Punkt beginnen. Warum das sinnvoll ist, wird aus der Beschreibung des POP3-Protokolls klar werden.

Ein möglicher Client

Wie oben bereits erwähnt steht Euch die Wahl des verwendeten Mail-Clients frei. Dennoch hier der Hinweis auf einen Client, der auf allen unseren Suns installiert ist und das POP3-Protokoll beherrscht: Durch den Aufruf

```
mutt -F/dev/null -f pop://user:pass@server:port
```

wird der Mail-Client `mutt` gestartet und der durch "server" und "port" angegebene POP3-Server kontaktiert. Wir werden die Authentifizierung über Benutzername (user) und Passwort (pass) für dieses Blatt weitgehend ignorieren, dennoch sollte für beiden Werte ein nicht leerer Text angegeben werden.

Das POP3-Protokoll

Kommandos

Ähnlich wie beim MasterMind-Server im vorletzten Blatt ist das POP3-Protokoll zeilenorientiert wobei jede Zeile durch \r\n beendet wird. Kommandos können Argumente haben, gelegentlich sind diese Argumente optional.

Antworten

Der Server beantwortet jedes Kommando mit einer Zeile, die im Erfolgsfall mit `+OK` und im Fehlerfall mit `-ERR` beginnt. Anschließend folgt ein Leerzeichen und Text. Je nach Kommando folgen auf eine erfolgreiche Antwortzeile noch weitere Daten. Diese Daten werden dann durch eine Zeile, die nur einen Punkt enthält beendet.

Varianten und unbekannte Kommandos

Es gibt verschiedene Varianten und Erweiterungen, von POP3. Ein Client testet ob der Server eine Erweiterung unterstützt, indem er einfach das Kommando abschickt. Ein Server, der das Kommando nicht unterstützt wird mit einer Fehlermeldung antworten, es ist daher wichtig, daß unbekannte Kommandos mit einer Fehlermeldung beantwortet werden.

Das Mailbox-Modell

Aus Sicht des POP3-Protokolls besteht eine Mailbox aus einer Reihe von einzelnen Mails. Beim Öffnen der Mailbox wird jeder Mail eine laufende Nummer beginnend bei 1 zugewiesen. Außerdem hat jede Mail eine ID, die sich über mehrerer Sitzungen nicht ändern darf. Am einfachsten wird dafür der Dateiname im Mailbox-Verzeichnis verwendet. Gelöschte Mails werden zunächst nur für die laufende Sitzung als gelöscht markiert, erst beim Beenden der Verbindung mit `QUIT` werden sie auf dem Server tatsächlich gelöscht.

Die Kommandos im einzelnen

Im folgenden werden die einzelnen Kommandos beschrieben. Vielfach ist die beschriebenen Implementierung so vereinfacht, daß eine Kommunikation dennoch funktionieren sollte. In der Regel wird nur die erfolgreiche Antwort beschrieben, tritt tatsächlich ein Fehler auf, ist natürlich auch eine Fehlerantwort immer zulässig.

Begrüßung Nach dem Verbindungsaufbau durch einen Client schickt der Server eine erfolgreiche Antwort als Begrüßung, ohne daß der Client ein Kommando zu schicken braucht.

USER, PASS und NOOP Diese Kommandos werden hier der Einfachheit halber immer erfolgreich beantwortet. Die ersten beiden dienen sonst der Authentifizierung.

STAT Dieses Kommando liefert immer eine erfolgreiche Antwort. Auf `+OK` folgen zwei Zahlen, die Anzahl der Mails in der Mailbox und die Gesamtgröße aller Mails.

LIST *num* Wird als Argument eine Zahl angegeben, dann wird diese Zahl als Nummer einer Mail interpretiert. Ist die Nummer gültig und die zugehörige Mail noch nicht als gelöscht markiert, dann folgt auf das `+OK` der erfolgreichen Antwort in der gleichen Zeile die Nummer der Nachricht und deren Größe durch Leerzeichen getrennt.

LIST Wird keine Zahl angegeben, dann liefert der Server zunächst eine erfolgreiche Antwortzeile. Auf diese folgt eine Liste aller Mails, eine Mail pro Zeile. Jede Zeile enthält die Nummer einer Mail und ihre Größe. Die Liste wird durch einen Zeile, die nur einen Punkt enthält, beendet. Bereits als gelöscht markierte Mails sind in dieser Liste nicht mehr enthalten.

RETR *num* und TOP *num num2* Diese beiden Kommandos bekommen als Argument eine Zahl (TOP bekommt eine zweite, die wir ignorieren). Ist diese Zahl eine gültige Nummer einer noch nicht gelöschten Mail, dann wird nach der Zeile für eine erfolgreiche Antwort der vollständige Inhalt der Mail unverändert gesendet. Die Mail wird durch \r\n.\r\n beendet. Das zusätzliche \r\n vor dem Punkt ist hier notwendig, weil eine Mail nicht unbedingt mit einem \n enden muß.
Anmerkung: Eigentlich sollte TOP nur die ersten Zeilen der Mail liefern, wir machen es uns hier aber einfach.

DELE *num* Diese Kommando markiert die Mail mit der angegebenen Nummer als gelöscht. Diese Nummer ist anschließend nicht mehr gültig.

RSET Dieses Kommando markiert alle als gelöscht markierten Mails wieder als ungelöscht. Sie erhalten dann ihre alten Nummern zurück.

UIDL Diese Kommando funktioniert ganz ähnlich wie LIST, es wird lediglich statt der Größe der Mail ihre eindeutige ID angegeben. Wie oben beschrieben, kann dafür der Dateiname der Mail verwendet werden. Mit Hilfe dieses Kommandos können Clients eine Mail über mehrere Sitzungen hinweg identifizieren, auch wenn sich ihre Nummer (zum Beispiel durch Löschen von Mails) ändern sollte.
Genau wie bei LIST kann optional die Nummer einer Mail als Argument angegeben werden.

QUIT Dieses Kommando beendet die Verbindung. Der Server löscht jetzt alle als gelöscht markierten Mails tatsächlich und beantwortet dann das Kommando mit einer erfolgreichen Antwort.

Viel Erfolg