

Betriebssysteme und Rechnerarchitektur

Übungsblatt fünf

28.05.2018

(Internationaler Tag der Durchführung)

Wenn Sie sich Ihre bisher erstellten Bausteine zusammen ansehen, ahnen Sie bestimmt schon, was wir dieses Semester basteln - genau, es ist ein Mailserver (was sonst?), von dem man z.B. mit einem gewöhnlichen Mail-Client seine Nachrichten abrufen kann.

Der Mailserver verwaltet für jeden seiner Benutzer eine persönliche Mailbox. Eine Mailbox ist eine Textdatei, deren Name dem Login-Namen des betreffenden Benutzers entspricht. Jeder Benutzer hat genau eine Mailbox-Datei, diese kann beliebig viele Mails für diesen Benutzer enthalten.

Der Inhalt einer **Mailbox**-Datei ist in leicht zu identifizierende Abschnitte eingeteilt, von denen jeder genau eine Mail enthält. Den **Beginn eines Abschnitts** markiert eine Zeile, die mit „From“ startet (genau so geschrieben, einschließlich des Leerzeichens nach dem „From“). Die nachfolgenden Zeilen bilden die Mail-Nachricht, sie wird abgeschlossen mit einer Leerzeile, auf die entweder der nächste Abschnitt oder das Dateiende folgt. Sollte in der eingehenden Mail eine Textzeile vorkommen, die mit „From“ beginnt, wird beim Posteingang automatisch mit ein „>“ davorgehängt - Zeilen, die mit „From“ beginnen, sind also *immer* Anfänge von Mail-Abschnitten.

POP3 sieht einen (Text-)**Dialog** zwischen Client und Ihrem Mailserver vor, der verschiedene **Phasen** hat. Zuerst muss der Benutzer sich mit Usernamen und danach Passwort **authentifizieren**. Nur wenn das geklappt hat, darf er die nachfolgenden Kommandos zum **Abruf** seiner Mails nutzen. Sie kennen das bereits aus der Vorlesung „Rechnernetze“ - trotzdem weiter unten im PDF ein kurzer Überblick über solch einen POP3-Dialog. Eingaben des Clients sind gefettet, Ihre Server-Ausgaben nicht. Die Kommandos selbst sind case-insensitiv, „pass“, „Pass“ oder „PASS“ sind also gleichwertig (das gilt natürlich nicht z.B. für das als Parameter mitgegebene Passwort).

Positive Servermeldungen beginnen immer mit „+OK“, **Fehlermeldungen** mit „-ERR“. Wenn eine Ausgabe **mehrere Zeilen** umfasst, beginnt sie mit „+OK“, dann folgen variabel viele Datenzeilen, den Abschluss bildet eine Zeile, die **nur aus einem Punkt** „.“ besteht. Serverausgabe-Zeilen werden immer mit den zwei Steuerzeichen „Carriage Return / Line Feed“ (**CRLF**), also „\r\n“ abgeschlossen, das ist auch bei Ausgaben von z.B. Nachrichtenlängen an den Client zu berücksichtigen. Wenn ein Textabschnitt auf dem Server, dessen Zeilenende-Konvention z.B. nur „\n“ vorschreibt, incl. Zeilentrenner 100 Bytes umfasst und 10 Zeilen lang ist, wäre die Nachricht *in der Serverausgabe* mit 100+10 = 110 Zeichen (octets) anzukündigen und dem Client ggf. entsprechend auszuliefern.

Erschaffen Sie bitte nun ein Modul pop3.c und beachten Sie folgende Zuleuchtungen:

- Verwenden Sie bitte Ihr „**database**“-Modul vom dritten Übungsblatt, um unter dem key=„joendhard“ mit Eintrags-Kategorie cat=„password“ ein Passwort (value) einzurichten. Mit Hilfe Ihres zugehörigen Kommandozeilentools können Sie Ihre Konfigurationsdatenbank leicht ausgeben und pflegen. Legen Sie bitte

auch einen Eintrag mit key="joendhard" und cat="mailbox" an, deren „value“-Feld einen Dateipfad zur Mailbox-Datei des Users joendhard enthält.

- Schreiben Sie bitte im Modul „pop3.c“ eine Funktion

```
int process_pop3(int infd, int outfd)
```

welche POP3-Kommandos über infd liest und seine Serverausgaben auf dem Ausgabe-Filedescriptor outfd ausgibt. Denken Sie bitte daran, dass Ausgabezeilen gemäß Protokoll immer mit CRLF abzuschließen sind.

- Sie können sich zu Testzwecken eine eigene main() schreiben, welche Ihre process_pop3()-Funktion mit den Filedescriptoren für Standardein- und -ausgabe aufruft. So können sie Ihre POP3-Dialoge leicht über die Konsole testen.
- Wie bereits bemerkt können nicht alle Kommando in jedem Zustand des Dialogs ausgeführt werden. Im **Anfangszustand** gehen nur „user“ und „pass“ so lange, bis eine positive Authentifikation erreicht ist, nach erfolgtem Login sind diese beiden Kommandos aus der Anfangsphase verboten und dafür können mehrere Kommandos zur **Mailboxabfrage** genutzt werden. Nach **Abschluss** mit „quit“ wird (später mal) die Verbindung geschlossen (=neuer Zustand), alle weiteren Kommandos sind also ungültig.

Wie gut, dass wir vom ersten Übungsblatt noch das **Modul „dialog“** haben, das uns das die Verarbeitung von Kommando/Parameter-Paaren unter Berücksichtigung von Dialogzuständen erleichtert. Verwenden Sie es bitte hier zur Implementierung.

- Bitte implementieren Sie vorerst diese POP3-Kommandos (jeweilige Bedeutung im Beispiel, siehe PDF zu diesem Übungsblatt)
 - user *benutzername* - *Benutzername für Login*
 - pass *passwort* - *zugehöriges Passwort*
 - stat - Mailbox-Status (Anzahl/Gesamtgröße)
 - list *nachrichtenummer* - Nummer/Größe einer Nachricht
 - list - Liste Nummer/Größe aller Nachrichten
 - retr *nachrichtenummer* - Inhalt dieser Nachricht abrufen
- Sie sehen, dass Sie **Informationen zu Anzahl und Länge** der in der Mailbox-Datei enthaltenen Nachrichten benötigen, ebenfalls wollen Sie zügig auf eine konkrete Nachricht (gemäß Nachrichtennummer) zugreifen, um Sie z.B. mit „retr“ abzurufen. Da ist es ein willkommener Zufall, dass Sie sich beim letzten Mal ein „linebuffer“- bzw. darauf aufbauend ein „**fileindex**“-Modul geschrieben haben. Diese sollten Ihnen wesentliche Teile der Arbeit abnehmen.
- Falls Sie dazu Erweiterungen / Anpassungen Ihrer selbstgeschriebenen Hilfsmodule aus früheren Aufgaben vornehmen möchten, können Sie dies gerne tun, solange diese nicht sinnentstellend für die jeweiligen Module sind.
- Verwenden Sie bitte die im read.MI bereitgestellten **Musterdateien**, um Ihre Implementierung (insb. die Längenangaben) zu testen:
 - Mailbox-Datei „joendhard.mbox“
 - POP3-Beispieldialog „joendhard-pop3-dialog.txt“ mit Ausgabe genau zu dieser Mailbox-Datei.
- Die Beispieldatei enthält Informationen zu Jöndhards Privatleben, die *nicht* für die Öffentlichkeit außerhalb des Betriebssysteme-Kurses bestimmt sind.
- Weitere POP3-Kommandos ziehen wir später nach. Die vollständige Spezifikation des Protokolls finden Sie im RFC1939 (<https://tools.ietf.org/html/rfc1939>)

+OK meintollerpopserver ready.	Server gibt Begrüßung aus.
user joendhard	„user“-Kommando mit Login-Namen
+OK	Server bestätigt Kommando (heißt nicht, dass es den User tatsächlich gibt!)
pass biffel	Übermittlung des Passworts
+OK Logged in.	User/Passwort-Kombi geprüft, Login erfolgreich
stat	stat-Abfrage für Mailbox „joendhard“
+OK 7 8258	Mailbox hat 7 Nachrichten, 8258 Octets lang. entspricht der Summe der Längen in list (s.u.)
list	ohne Parameter Liste aller Nachrichten
+OK 7 messages:	verstanden, 7 Nachrichten gibt's, die folgen nun
1 993 2 867 3 1012 4 1106 5 1885 6 923 7 1472	Ab 1 durchnummerierte Paare mit laufender Nummer und Größe der Nachricht (incl. CRLF-Zeilende-Zählkorrektur, siehe oben)
.	Einsamer Punkt = „list“-Output abgeschlossen
list 2	list mit Parameter: nur für Nachricht 2
+OK 2 867	kein Mehrzeiler - nur OK mit Daten wie oben
retr 1	retrieve (Abrufen) Nachricht Nr. 1
+OK 993 octets Return-Path: <tupel@gruenemuetze.xy> To: <joendhard@biffel.xy> Message-Id: <20180527160357.518C149EAA@mail.biffel.xy> Date: Sun, 27 May 2018 18:03:57 +0200 (CEST) From: tupel@gruenemuetze.xy Sehr geehrter Joendhard, Zeit für eine neue, warme Kopfbedeckung? In Mr. Tupel's Zipfelmützenparadies finden Sie (usw, Spam halt)	OK-Zeile kündigt Nachrichtenlänge an (entspricht der in list, s.o.), danach folgen die Inhaltszeilen aus der Mailbox für diese Nachricht, abgeschlossen mit einem einsamen Punkt. Sollte in der Nachricht eine Zeile mit einem Punkt beginnen, setzt der Server (also Sie) automatisch bei der Ausgabe einen weiteren davor. Diese zählen bei Octet-Längenangaben nicht mit und werden vom Mail-Client automatisch wieder entfernt.
.	Einsamer Punkt = retr-Ausgabe abgeschlossen
quit	Ende, ausloggen
+OK Logging out.	wird bestätigt (und Verbindung würde vom Server geschlossen)