TW-Mailer – Pro Version

Extend the basic TW-Mailer project with some pro features.

- Refactor the first version so that the server works concurrent (not iterative)
  - Use fork() or threads
  - Identify possible synchronization problems and guard the critical sections.
- Extend the project with a LOGIN command and allow the other commands only to authenticated users (except QUIT).
  - The login checks the credentials using the internal LDAP server.
  - For security reasons only 3 login attempts are allowed per user and IP.
  - In case the user fails to login within 3 attempts her/his IP is blacklisted for 1 minute.
  - Persist the blacklist.
  - Remove the possibility in SEND to set a sender manually.
    - Sender in SEND should automatically be set (session information after login)
  - Remove the possibility in LIST/READ/DEL to set the username manually.
    - The username should automatically be set (session information after login)
- Custom features are welcome (please talk to your supervisor in advance)

Hints corresponding to LDAP:

- Use OpenLDAP C-API

  Ubuntu Packet libldap2-dev

  include file <ldap.h>

  gcc Option -lldap -llber
  e.g.: g++ -std=c++14 -Wall -o myldap myldap.c -lldap -llber

- Internal LDAP-Server address

  Host: ldap.technikum.wien.at
  Port: 389
  Search Base: dc=technikum-wien,dc=at

- Further docs: http://www.yolinux.com/TUTORIALS/LinuxTutorialLDAP-SoftwareDevelopment.html

Protocol specification (update)

LOGIN

```
LOGIN\n
<LDAP username>\n
<password in plain text>\n
```

- The server responds with OK\n (and enables all commands for the session) or ERR\n

SEND

```
SEND\n
<Receiver>\n
<Subject (max. 80 chars)>\n
<message (multi-line; no length restrictions) \n>
.\n
```

- The final dot ends the command.
- The server always responds with either “OK\n” or “ERR\n”.

LIST

```
LIST\n
```

- The server responds with

```
<count of messages of the current user (0 if no message or user unknown)>\n
<subject 1>\n
<subject 2>\n
…
<subject N>\n
```

READ

```
READ\n
<Message-Number>\n
```

- The server responds with

```
OK\n
<complete message content (as defined in SEND)>\n
```

- or

```
ERR\n
```

DEL

```
DEL\n
<Message-Number>\n
```

- The server responds with

```
OK\n
```

- or

```
ERR\n
```

Deliverables

- Hand-in
  - The commented code for the client and the server code
  - Makefile for the targets “all” and “clean”
  - Executables
  - Description of (1-2 pages; pdf)
    - the client and server architecture
    - used technologies and libraries
    - development strategy and needed protocol adaptions
    - uses synchronization methods
    - handling of large messages

The second part of the project will be presented in a code review. If there is no presentation at the end of the semester, the submission is automatically rated with 0 points.

Marking System (40 Points)

- 3: submission contains Makefile and protocol
- 5: server is concurrent and synchronized
- 3: persistent and efficient storage of the mail spool directory
- 12: connections and commands
  - 3 SEND (+ Sender from session data)
  - 3 LIST
  - 3 READ
  - 3 DEL
- 5: LOGIN and LDAP connection
- 3: locking clients after 3 attempts
- 4: structure, error-handling, code quality, indentation, comments
- 5: Code understanding during the presentation