

## Socketprogrammierung

### Aufgabe 3 Mail

#### 3.1) SMTP über telnet

1-5 / 3-4: Mitschnitt:

```
> 220 asmtpt.htwg-konstanz.de ESMTP Postfix (Debian/GNU)
< AUTH LOGIN
> 334 VXNlcm5hbWU6
< cm5ldGlu
> 334 UGFzc3dvcmQ6
< VWViZW44ZnVlcjhSTg==
> 235 2.7.0 Authentication successful
< MAIL FROM:<sebastian.illges@htwg-konstanz.de>
> 250 2.1.0 Ok
< RCPT TO:<sebastian.illges@htwg-konstanz.de>
> 250 2.1.5 Ok
< DATA
> 354 End adta with <CR><LF>.<CR><LF>
< From:<sebastian.illges@htwg-konstanz.de>
< To:<sebastian.illges@htwg-konstanz.de>
< Subject: Test
< Data
< .
> 250 2.0.0 Ok: queued as E8CCB82A38
```



5.

No.	Time	Source	Destination	Protocol	Length	Info
12	0.720299	141.37.11.129	141.37.194.4	SMTP	111	S: 220 asmtpt.htwg-konstanz.de ESMTP Postfix (Debian/GNU)
14	0.720846	141.37.194.4	141.37.11.129	SMTP	68	C: AUTH LOGIN
16	0.724384	141.37.11.129	141.37.194.4	SMTP	74	S: 334 VXNlcm5hbWU6
18	0.724809	141.37.194.4	141.37.11.129	SMTP	66	C: User: cm5ldGlu
19	0.728286	141.37.11.129	141.37.194.4	SMTP	74	S: 334 UGFzc3dvcmQ6
21	0.728393	141.37.194.4	141.37.11.129	SMTP	78	C: Pass: VWViZW44ZnVlcjhSTg==
22	0.732039	141.37.11.129	141.37.194.4	SMTP	93	S: 235 2.7.0 Authentication successful
24	0.732195	141.37.194.4	141.37.11.129	SMTP	103	C: MAIL FROM:<sebastian.illges@htwg-konstanz.de>
25	0.736065	141.37.11.129	141.37.194.4	SMTP	70	S: 250 2.1.0 Ok
27	0.736440	141.37.194.4	141.37.11.129	SMTP	101	C: RCPT TO:<sebastian.illges@htwg-konstanz.de>
28	0.741008	141.37.11.129	141.37.194.4	SMTP	70	S: 250 2.1.5 Ok
30	0.741062	141.37.194.4	141.37.11.129	SMTP	62	C: DATA
31	0.750535	141.37.11.129	141.37.194.4	SMTP	93	S: 354 End data with <CR><LF>.<CR><LF>
33	0.750680	141.37.194.4	141.37.11.129	SMTP	161	From:<sebastian.illges@htwg-konstanz.de> , To:<sebastian.illges@htwg-konstanz.de> , Subject: Test, Me...
34	0.770219	141.37.11.129	141.37.194.4	SMTP	92	S: 250 2.0.0 Ok: queued as 1E2068527F
36	0.770662	141.37.194.4	141.37.11.129	SMTP	62	C: QUIT
37	0.774238	141.37.11.129	141.37.194.4	SMTP	71	S: 221 2.0.0 Bye

Es werden die gleichen Infos wie in Telnet eingegeben.

6. Scheinbar kann der Inhalt der Mail nicht richtig verarbeitet werden:



3.2) SMTP in Python

#### Aufgabe 4 Rechen-Server

4.1)

1. `socket.send()`
2. `socket.accept()` wartet auf `socket.connect()`  
`socket.recv()` wartet auf `socket.send()`

4.2)

1. Ausgabe mit: `socket.getsockname()`
2. Nach `socket.connect()`.
3. `sock.bind((Server_IP, Server_PORT))`
4. Timeouts werden benötigt, damit das Programm das auf eine Anfrage wartet nicht unnötig lange nichts tut, falls über längere Zeit oder gar keine Anfragen mehr kommen.
5. Wenn der Server sowohl TCP als auch UDP anfragen entgegennehmen kann, z.B. mit der Hilfe von Threads.

#### 5

5.3)

1. TCP: 7, 9, 13, 17, 19  
UDP: 7, 13, 17, 19

## 2. TCP Port 7:

931	109.063930	141.37.168.26	141.37.196.8	TCP	56	7 → 60336 [FIN, ACK] Seq=14 Ack=15 Win=65792 Len=0 TSval=357211 TSecr=2470452208
932	109.063984	141.37.196.8	141.37.168.26	TCP	56	60336 → 7 [ACK] Seq=15 Ack=15 Win=131840 Len=0 TSval=2470452211 TSecr=357211

## TCP Port 8:

tcp.port==8						
No.	Time	Source	Destination	Protocol	Length	Info
1036	113.992469	141.37.196.8	141.37.168.26	TCP	68	60340 → 8 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2915328551 TSecr=0 SACK_PERM
1037	113.998088	141.37.168.26	141.37.196.8	TCP	44	8 → 60340 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

## UDP Port 7:

udp.port==7						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	141.37.196.8	141.37.168.26	ECHO	45	Request
2	0.004665	141.37.168.26	141.37.196.8	ECHO	45	Response
109	33.064574	141.37.196.8	141.37.168.26	ECHO	45	Request
110	33.075274	141.37.168.26	141.37.196.8	ECHO	45	Response

## UDP Port 8:

udp.port==8						
No.	Time	Source	Destination	Protocol	Length	Info
118	52.245765	141.37.196.8	141.37.168.26	UDP	45	55604 → 8 Len=13
119	52.258659	141.37.168.26	141.37.196.8	ICMP	73	Destination unreachable (Port unreachable)

## 3. Funktioniert beides wie erwartet.