Linnéuniversitetet
Kalmar Växjö

Report

Assignment 3

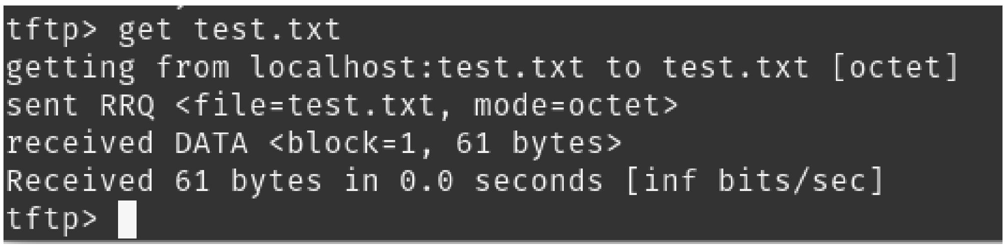
1DV701

Author: Johannes Böök, André Sturesson
Semester: Spring 2022
Email jb224tu@student.lnu.se, as226ng@student.lnu.se

Linnéuniversitetet
Kalmar Växjö

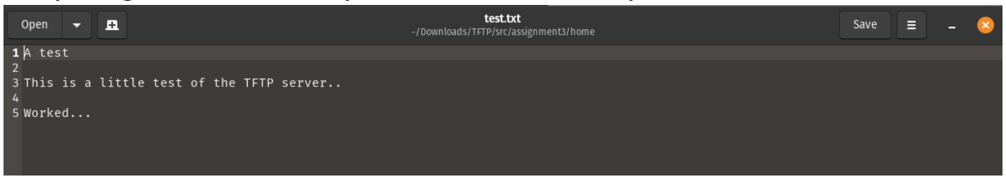
Contents

Discussion

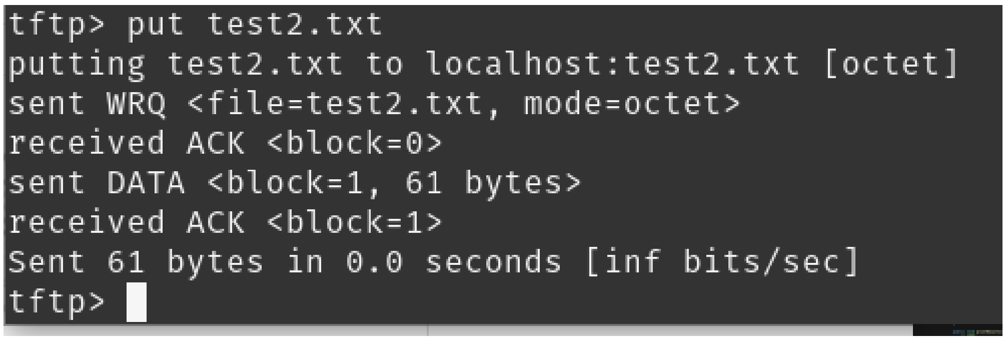


```
tftp> get test.txt
getting from localhost:test.txt to test.txt [octet]
sent RRQ <file=test.txt, mode=octet>
received DATA <block=1, 61 bytes>
Received 61 bytes in 0.0 seconds [inf bits/sec]
tftp>
```

A test of the get (RRQ) which is handled in send_DATA_reveive_ACK and everything works flawlessly for data under 512 bytes.

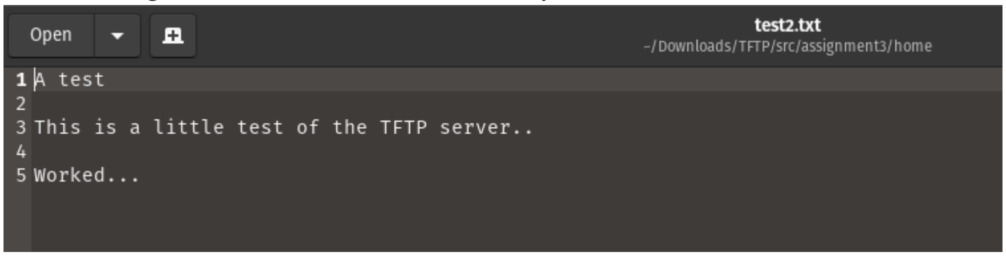


```
A test

This is a little test of the TFTP server..

Worked...
```



```
tftp> put test2.txt
putting test2.txt to localhost:test2.txt [octet]
sent WRQ <file=test2.txt, mode=octet>
received ACK <block=0>
sent DATA <block=1, 61 bytes>
received ACK <block=1>
Sent 61 bytes in 0.0 seconds [inf bits/sec]
tftp>
```

A test of a (WRQ) which is handled in receive_DATA_send_ACK and this also works great, still for data under 512 bytes.



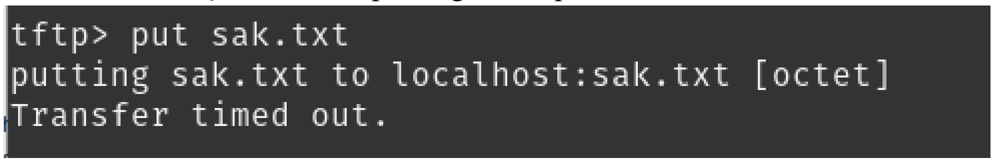
```
A test

This is a little test of the TFTP server..

Worked...
```

Socket and sendSocket

socket handles the connection between server and the client. while the sendSocket which is a DiagramSocket handles the communication over the connection.

Steps for completion

Everything in the beginning was straight forward, the receiveFrom method and the ParseRQ method for parsing the request.



```
tftp> put sak.txt
putting sak.txt to localhost:sak.txt [octet]
Transfer timed out.
```

The thing we struggled the most with was that every get and put was timed out. We later discovered that we had prepared our datagram packets slightly wrong when we looked at the RFC1350 guidelines.

Another problem was the TFTP client in windows wasn't that good, that led us to switching our OS to linux where a TFTP client exists in the terminal which worked much better.

It would also be fairly easy to implement reading and writing for files larger than 512 bytes. In theory it's only to increment the blocksize by one and sort the bigger data into smaller chunks of max 512 bytes each. The smaller packets will then be assembled before finalization.

Work distribution

We agree that the work is distributed as evenly as possible (50%) . Because we were both doing everything together as well as troubleshooting and writing the report together.