Computer networks - 1DV701 Assignment 3

Author: Michael Johansson & Jakob

Heyder

Semester: VT 2017

Contents

1	Assignment summary	1
2	Problem one	2
3	Problem two	3
	3.1 VG.1 HTTP status code implementation	4
	3.2 VG.2 POST vs PUT	5
4	Problem three	6

1 Assignment summary

- Michael Work percentage 50%
- **Jakob** Work percentage 50%

2 Problem one

In figure 2.1 we show a successful RRQ (Read-Request) to the implemented tftp server retrieving a small test-text file. We use the original socket as a kind of server socket accepting the requests on port 4970 (Usually 69 in FTP) and then after receiving a request, either write or read, we open a new thread for the specific client connection. In this thread we use the sending socket which acquires a free port (zero port indicates to get a random free port e.g. 5800) and then send the new specified port to the client in the ACK (WRQ) or DATA (RRQ) packet. This is defined in the TFTP-Specification that the initialization will be on a predefined port and then the server communicates the used port to the client. This way we can also handle multiple client requests since the original socket is freed after handling the client in a separate thread.

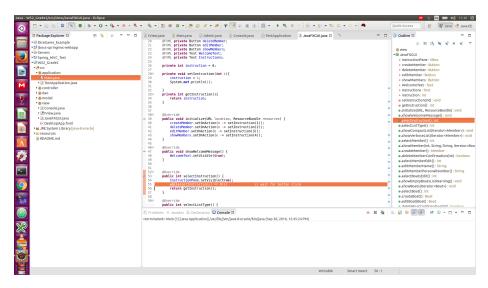


Figure 2.1: Successful READ-Request. We can see the tftp-client in the terminal window, eclipse tftp server and the home directory where the retrieved file is saved to.

3 Problem two

List of all implemented server responses

• 200: OK

• 201: Created

• 204: No Content

• 302: Found

• 400: Bad request

• 403: Access Denied

• 404: Not Found

• 411: Length Required

• 414: URI Too Long

• 415: Unsupported Media Type

• 500: Internal Server Error

• 501: Not implemented

• 505: HTTP Version Not Supported

3.1 VG.1 HTTP status code implementation

For some images below we just show the error.html response but we send the status code as well for every error.

3.2 VG.2 POST vs PUT

We use POST when the user uploads a new resource that our sever handles where it should be placed. In our POST implementation the user can upload an image through our upload page. The server then creates that images with a hash as the file name and under the images folder.

If instead the user has the exact request-URI then we can use PUT to create or overwrite the file on that exact URI.

So in short the POST is used for when the client don't need to know the exact URI and just upload the file where we want it. The PUT when the client want to Create or replace a file on an exact URI.

4 Problem three