

# CN2 Aufgabe 3

Julian Bertol

November 7, 2024

## 1 Aufgabe 1

```
1 import java.io.BufferedReader;
2 import java.io.IOException;
3 import java.io.InputStreamReader;
4 import java.io.OutputStreamWriter;
5 import java.net.ServerSocket;
6 import java.net.Socket;
7 import java.text.ParseException;
8 import java.text.SimpleDateFormat;
9 import java.util.Date;
10
11 public class HTTP_Server {
12     public static void main(String[] args) {
13         try (ServerSocket serverSocket = new ServerSocket(6789)) {
14             System.out.println("Auf eingehende Verbindungen warten...");
15             while (true) {
16                 try (Socket s = serverSocket.accept()) {
17                     System.out.println("Verbunden mit: " + s.getRemoteSocketAddress());
18                     ;
19                     String request = readText(s);
20                     Date lastAccessTimestamp = parseCookie(request);
21                     String response = generateResponse(lastAccessTimestamp);
22                     sendText(s, response);
23                 } catch (IOException e) {
24                     System.out.println("Fehler bei der Verbindung.");
25                     e.printStackTrace();
26                 }
27             } catch (IOException e) {
28                 System.out.println("Fehler beim Erstellen des Sockets");
29                 e.printStackTrace();
30             }
31         }
32
33         private static String readText(Socket s) throws IOException {
34             BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));
35             StringBuilder sb = new StringBuilder();
36             String line;
37             while ((line = in.readLine()) != null && !line.isEmpty()) {
38                 sb.append(line).append("\r\n");
39             }
40             return sb.toString();
41         }
42
43         private static void sendText(Socket s, String response) {
44             System.out.println("Durch Socket bertragen ...");
45             try (OutputStreamWriter osw = new OutputStreamWriter(s.getOutputStream())) {
46                 osw.write(response);
47                 osw.flush();
48             } catch (IOException e) {
49                 System.out.println("Fehler beim bertragen ber Socket");
50                 e.printStackTrace();
51             }
52         }
53     }
54 }
```

```

53
54 private static Date parseCookie(String request) {
55     SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy, HH:mm:ss");
56     String[] lines = request.split("\r\n");
57     for (String line : lines) {
58         if (line.startsWith("Cookie:")) {
59             String[] cookies = line.substring(7).split(";");
60             for (String cookie : cookies) {
61                 String[] parts = cookie.trim().split("=");
62                 if (parts.length == 2 && "lastAccess".equals(parts[0])) {
63                     try {
64                         return dateFormat.parse(parts[1]);
65                     } catch (ParseException e) {
66                         System.out.println("Fehler beim Parsen des Cookie-Datums")
67                             ;
68                     }
69                 }
70             }
71         }
72     }
73     return null; // kein g ltiges Zugriffsdatum gefunden
74 }
75
76 private static String generateResponse(Date lastAccessTimestamp) {
77     SimpleDateFormat dateFormat = new SimpleDateFormat("dd.MM.yyyy, HH:mm:ss");
78     Date currentDate = new Date();
79
80     StringBuilder response = new StringBuilder();
81
82     if (lastAccessTimestamp == null) {
83         response.append("Set-Cookie: lastAccess=" + dateFormat.format(currentDate)
84             + "\r\n");
85         response.append("\r\n");
86         response.append("Hallo! Es ist der " + dateFormat.format(currentDate) + "
87             . Sie sind zum ersten Mal auf diesem Server gelandet.");
88     } else {
89         response.append("Set-Cookie: lastAccess=" + dateFormat.format(currentDate)
90             + "\r\n");
91         response.append("\r\n");
92         response.append("Welcome back! Es ist der " + dateFormat.format(
93             currentDate) + ". Ihr letzter Zugriff war am " + dateFormat.format(
94                 lastAccessTimestamp));
95     }
96     return response.toString();
97 }
98 }

```

## 1.1 1. mal verbinden

```

Set-Cookie: lastAccess=07.11.2024, 10:47:01
Hallo! Es ist der 07.11.2024, 10:47:01. Sie sind zum ersten Mal auf diesem Server gelandet.
Verbindung zu Host verloren.

```

## 1.2 verbinden mit cookie

```

Set-Cookie: lastAccess=07.11.2024, 10:48:36
Welcome back! Es ist der 07.11.2024, 10:48:36. Ihr letzter Zugriff war am 01.01.2023, 10:00:00
Verbindung zu Host verloren.

```

## 2 Aufgabe 2

- 2.1 Auf dem Computer Ihrer Großeltern ist ein alter Browser installiert, der lediglich HTTP/1.0 Verbindungen ermöglicht. Wie lange dauert hiermit die Übertragung der gesamten Webseite?

$$1 \text{ Byte} = 8 \text{ Bit}$$

$$1 \text{ KByte} = 1000 \text{ Byte} \Rightarrow 1 \text{ KByte} = 8000 \text{ Bit}$$

$$\text{Gesamtgröße der Webseite} = 3,5 + 7 + 21 + 14 = 45,5 \text{ KB}$$

$$\text{Gesamtgröße in Bit} = 45,5 \times 8000 = 364.000 \text{ Bit}$$

1. Berechnung der reinen Übertragungszeit:

$$\begin{aligned} \text{Übertragungszeit} &= \frac{\text{Gesamtgröße in Bit}}{\text{Modemgeschwindigkeit}} \\ &= \frac{364.000 \text{ Bit}}{56.000 \text{ Bit/s}} \approx 6,5 \text{ Sekunden} \end{aligned}$$

2. RTT für jede Komponente:

Für jede Komponente wird eine separate Verbindung benötigt, die jeweils eine RTT von 250 ms benötigt, um die Anfrage zu senden und die Antwort zu empfangen.

$$\text{Gesamt-RTT-Zeit für alle vier Verbindungen} = 4 \times 0,25 \text{ s} = 1 \text{ Sekunde}$$

3. Gesamtdauer:

$$\begin{aligned} \text{Gesamtdauer} &= \text{Übertragungszeit} + \text{RTT-Zeit} \\ &= 6,5 \text{ s} + 1 \text{ s} = 7,5 \text{ Sekunden} \end{aligned}$$

- 2.2 Sie sich daran erinnert, dass Sie ja Ihren Laptop dabei haben. Damit kann HTTP/1.1 genutzt werden. Wie lange dauert hiermit die Übertragung der Webseite?

**Annahme für HTTP/1.1:** Da HTTP/1.1 Verbindungspersistenz erlaubt, können alle Komponenten in einer einzigen Verbindung übertragen werden, was bedeutet, dass nur eine RTT benötigt wird, um die Verbindung herzustellen.

1. RTT-Zeit für eine Verbindung:

Nur eine RTT (250 ms) ist erforderlich.

2. Gesamtdauer:

$$\begin{aligned} \text{Gesamtdauer mit HTTP/1.1} &= \text{Übertragungszeit} + \text{RTT} \\ &= 6,5 \text{ s} + 0,25 \text{ s} = 6,75 \text{ Sekunden} \end{aligned}$$

- 2.3 Angenommen ein HTTP-Proxy wäre im Netzwerk Ihrer Großeltern installiert, der über das Modem mit dem Internet verbunden ist. Hätte das an dieser Aufgabe etwas verändert? Wie und unter welchen Voraussetzungen würde ein HTTP-Proxy die Übertragungszeit einer Webseite im Netz Ihrer Großeltern verringern?

- Ein Proxy kann anfragen im Cache zwischenspeichern und somit die erneute Anfrage des Clients schneller beantworten.

### 3 In dieser Aufgabe beschäftigen wir uns mit DNS.

#### 3.1 Wieso denken Sie, wird normalerweise UDP bei der Namensauflösung genutzt?

- Mit TCP würde ein Austausch der Daten etwas länger dauern.
- Mit UDP geht das deutlich schneller.
- DNS-Anfragen und -Antworten passen typischerweise klein in ein UDP-Paket (maximal 512 Bytes).

#### 3.2 Benutzen Sie nslookup oder dig, um einen der Rootserver nach der Adresse von `www.hs-furtwangen.de` zu fragen. Zeigen Sie die Antwort auf der Konsole und interpretieren Sie die Ausgabe.

```
Server:  dns.google
Address:  8.8.8.8

Nicht autorisierende Antwort:
Name:     webs.alpha.hs-furtwangen.de
Address:  185.44.133.90
Aliases:  www.hs-furtwangen.de
```

- Name = Ist die Webserver-Adresse
- Address = Ist die öffentliche IP Adresse des Server.
- Aliases: Ist die Domain die auf die IP-Adresse verweist.

**3.3 Die Ausgabe aus Antwort b) bringt Sie Ihrem Ziel etwas näher. Nutzen Sie das Gelernte, um iterativ an die IP-Adresse von `hs-furtwangen.de` zu gelangen. Geben Sie hierbei an, was Sie Schritt für Schritt eingegeben und was Sie als Ergebnis zurückbekommen. Eventuell sind nicht alle Namensserver erreichbar.**

- Um alle Nameserver zu erreichen muss folgendes in der CMD unter verwendung von NSLOOKUP eingegeben werden:

– `nslookup www.hs-furtwangen.de a.root-servers.net`

1. Zunächst wird ein Rootserver nach `www.hs-furtwangen.de` gefragt. Der Rootserver verweist auf einen `.de`-Nameserver.
2. Der `.de`-Nameserver wird dann nach `hs-furtwangen.de` gefragt und gibt einen autoritativen Nameserver für die Domain `hs-furtwangen.de` zurück.
3. Der autoritative Nameserver für `hs-furtwangen.de` wird schließlich nach `www.hs-furtwangen.de` gefragt und liefert die IP-Adresse der gesuchten Domain.

**3.4 Unter welchem Namen ist `www.hs-furtwangen.de` noch bekannt?**

- Unter der IP: 185.44.133.90
- `webs.alpha.hs-furtwangen.de`