ulm university universität
uulm

Institut für Numerische Mathematik

Dr. Andreas F. Borchert

04. Juni 2019
Blatt 6

Systemnahe Software II (SS 2019)

Abgabe bis zum 11. Juni 2019, 14:00 Uhr

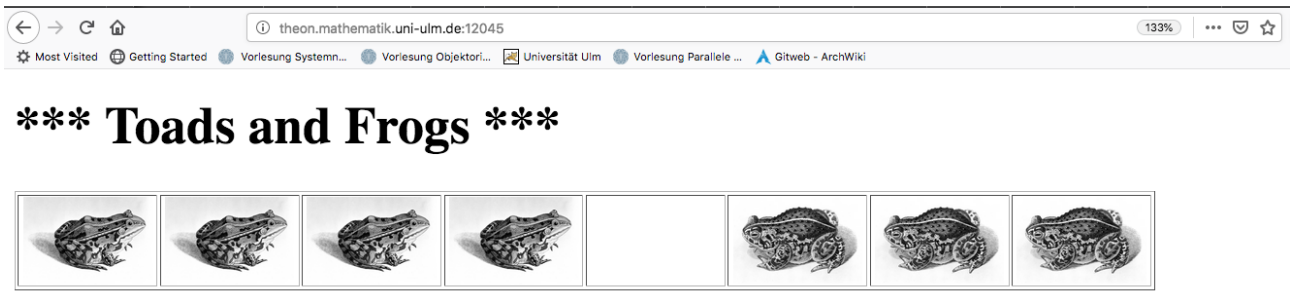
Lernziele:

- Aufsetzen eines einfachen Netzwerkdienstes mit Hilfe von *tcpserver*

Aufgabe 6: Toads and Frogs

In dieser Aufgabe soll eine Variante des Spiels Toads and Frogs implementiert werden. Um die Sache zu vereinfachen, verwenden wir die eindimensionale Puzzle-Variante. Wir haben auf der linken Seite m Frösche und auf der rechten Seite n Kröten, die jeweils ein Spielfeld belegen. Zwischen beiden Seiten ist genau ein freies Feld, so dass das Spielfeld insgesamt aus $m + n + 1$ Feldern besteht. Die Frösche können sich nur nach rechts und die Kröten nur nach links bewegen. Das Ziel des Spieles ist es, alle Kröten auf die linke und alle Frösche auf die rechte Seite zu bringen. Ein Frosch oder eine Kröte kann nur bewegt werden, wenn das nächste Feld in Bewegungsrichtung frei ist oder, wenn es von der anderen Spezies belegt ist und das Feld dahinter in Bewegungsrichtung frei ist.

Entwickeln Sie im Rahmen dieser Übungsaufgabe einen auf *tcpserver* basierten Netzwerkdiensts, der dieses Spiel auf Basis des HTTP-Protokolls anbietet, so dass es mit Hilfe eines Webbrowsers gespielt werden kann:



Dies ist ein Anfangsszenario mit $m = 4$ und $n = 3$. Ein Zug wird durchgeführt, indem ein Frosch oder eine Kröte angeklickt wird, die sich bewegen kann.

Der Zustand des Spiels ist jeweils an die URL anzuhängen. Die Beispiellösung verwendet „F“ für Frösche, „T“ für Kröten und „E“ für das leere Feld. Beispielsweise verweist der linke bewegungsbereite Frosch im oben gezeigten Anfangsszenario auf die URL `http://theon.mathematik.uni-ulm.de:12045/FFEFTTTT`.

In der ersten (durch CR-LF abgeschlossenen) Eingabezeile Ihres Programms finden Sie dann eine der beiden folgenden Varianten, wobei statt *theon.mathematik.uni-ulm.de* natürlich auch irgendein anderer Rechnername stehen kann:

- `GET /FFEFTTTT HTTP/1.1`
- `GET http://theon.mathematik.uni-ulm.de:12045/FFEFTTTT HTTP/1.1`

Die Rückantwort, die *tf-server* auf der Standardausgabe ausgibt, besteht dann aus zwei Teilen, dem Header und dem eigentlichen Inhalt. Folgendes Beispiel zeigt dies. Damit die überlangen Zeilen hier noch gezeigt werden können, wurden die Links auf die Bilder durch „...“ ersetzt.

```
HTTP/1.0 200 Ok
Content-Type:text/html

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
   "http://www.w3.org/TR/html4/loose.dtd">
<html><head>
<title>Toads and Frogs</title><meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head><body>
<h1>*** Toads and Frogs ***</h1>
<table border="1" style="table-layout:fixed"><tr>
   <td align=center style="width:12%"><img alt="frog" width="100" src="..."></td>
   <td align=center style="width:12%"><a href="/FEFFTTTT"><img alt="frog" width="100" src="..."></a></td>
   <td align=center style="width:12%"> </td>
   <td align=center style="width:12%"><img alt="frog" width="100" src="..."></td>
   <td align=center style="width:12%"><a href="/FFTFETTT"><img alt="toad" width="100" src="..."></a></td>
   <td align=center style="width:12%"><img alt="toad" width="100" src="..."></td>
   <td align=center style="width:12%"><img alt="toad" width="100" src="..."></td>
   <td align=center style="width:12%"><img alt="toad" width="100" src="..."></td>
</tr></table>
</body></html>
```

Zur grafischen Darstellung können die beiden folgenden Zeichnungen des Naturforschers und Künstlers August Johann Rösel von Rosenhof (1705–1759) verwendet werden (zuerst der Frosch, nach rechts schauend, dann die Kröte, nach links ausgerichtet):

- `https://upload.wikimedia.org/wikipedia/commons/f/f1/Frog_from_Roesel_von_Rosenhof%3B_1758_Wellcome_L0001704.jpg`
- `https://upload.wikimedia.org/wikipedia/commons/9/98/Frog_from_Roesel_von_Rosenhof%3B_1758_Wellcome_L0001710.jpg`

Ihr Dienst sollte von der Standardeingabe die Anfrage entsprechend des HTTP-Protokolls einlesen und auf der Standardausgabe die Antwort entsprechend des HTTP-Protokolls erzeugen. Dann kann Ihr Dienst mit folgendem Kommando gestartet werden:

```
theon$ tcpserver localhost 12045 tf-server &
```

Danach können Sie mit dem Webbrowser darauf zugreifen unter der Verwendung der URL `http://localhost:12045`.

Die URL spezifiziert dabei den Rechnernamen (hier *localhost*, was nur klappt, wenn der Dienst und der Webbrowser auf dem gleichen Rechner laufen) und die Portnummer 12045 ist. Wenn Sie auf einem anderen Rechner arbeiten, können Sie den Dienst auch so starten, dass von außen darauf zugegriffen werden kann:

```
theon$ tcpserver 0 12045 tf-server &
```

Im Fehlerfalle ist auch eine entsprechende Seite zu generieren, wobei im Textteil eine beliebige Fehlermeldung untergebracht werden kann:

```
HTTP/1.0 500 Failed

Out of memory.
```

Hinweise:

Sie dürfen die Ein- und Ausgabe gerne so organisieren, wie es Ihnen günstig erscheint. Es steht Ihnen insbesondere auch frei, hierfür die Vorlesungsbibliothek zu verwenden, die auf Basis von inbuf auch inbuf_scan anbietet. Diese Funktion erlaubt es, mit Hilfe regulärer Ausdrücke einzulesen. Dies kann die Analyse der HTTP-Anfrage deutlich erleichtern. Wenn Sie sich dazu entschließen, sind folgende Bibliotheken zusätzlich beim Zusammenbau in Ihrem *Makefile* zu nennen: *LDLIBS := -lafb -lowfat -lpcre*. Mit *-lafb* kommt die Vorlesungsbibliothek hinzu, mit *-lowfat* u.a. die *stralloc*-Bibliothek und mit *-lpcre* eine Bibliothek für reguläre Ausdrücke, die sich an denen von Perl orientieren.

Es geht aber auch ohne die Vorlesungsbibliothek. Sie können einfach die gesamte Eingabe mit *read* einlesen, diese in einem dynamisch wachsenden Puffer ablegen und dort in der ersten Zeile nach dem letzten „/“ suchen. Dahinter finden Sie dann entweder den Zeilentrenner (Anfangssituation) oder einen expliziten Spielstand.

Ihre Lösung können Sie wiederum mit Hilfe von *tar* verpacken und dann mit *submit* einreichen:

```
theon$ tar cvf tf-server.tar *.[ch] [mM]akefile
theon$ submit ss2 6 tf-server.tar
```

Viel Erfolg!