

# Rechnernetze Aufgabe 1 Konzept

Triebe, Marian

`marian.triebe@haw-hamburg.de`

Kirstein, Katja

`katja.kirstein@haw-hamburg.de`

October 21, 2014

## 1 Aufgabenbeschreibung

Es soll ein Server-Programm entwickelt werden, welches wie ein Multiplexer Bilder weiterleitet. Diese Bilder stammen von einer Kamera. Die einzelnen Bilder bestehen aus 320 x 240 Pixeln. Für jeden Pixel werden 3 Byte übertragen (RGB). Es soll möglich sein mindestens 20 Bilder die Sekunde zu übertragen, desweiteren soll das Programm die empfangenen Bilder der Kamera ausgeben. Die Kommunikation soll mit dem TCP/IP Protokoll abgehandelt werden. Mögliche Angriffsvektoren sind abzusichern. Der Server soll nach fehlerhaften Verhalten der Clienten möglichst ohne Störung weiterarbeiten.

## 2 Aufbau des Programms

Die Anwendung soll aus 4 Hauptklassen bestehen. Jeder dieser Hauptklassen implementiert unseren pthread Wrapper (`thread.t.hpp`). Insgesamt wird es also in der Anwendung mindestens 4 Threads geben (inklusive Hauptthread). Jede aktive Verbindung zu einem Client wird jedoch durch einen zusätzlichen Thread abgearbeitet. Folgende Aufgaben sind auf diese 4 Hauptklassen aufzuteilen:

1. Abholen der Bild Daten (`fetch_impl.hpp`)
2. Warten auf Verbindungsanfragen von Clients (`server_impl.hpp`)
3. Senden der Bild Daten zu Clients (`worker_impl.hpp`)
4. Ausgabe der Bilder am lokalen Rechner (`client_impl.hpp`)

## 3 Buffer Konzept

Die Bufferung soll mittels eines Ringbuffers gehandhabt werden. Für die Größe des Ringbuffers wollen wir erst einmal 100 Bilder verwenden. Der Ringbuffer soll eine Funktion `'wait_on_picture()'` anbieten, welche den Aufrufer blockieren lässt, bis ein neues Bild im Buffer zur Verfügung steht. Der Ringbuffer hat immer eine konstante Größe und kann somit nicht von langsam sendenden workern (bzw. langsam empfangenden Clients) aufgeblasen werden. Nachteil hierbei, es können Bilder verloren gehen.

## 4 Angriffsvektoren

- Überflutung des Servers mit connect() anfragen
- Clients lesen nur sehr kleine Datensätze
- Clients warten lange zwischen dem empfangen der einzelnen Bilder