TechGI IV – Praktische Übungen

Hinweise zu Aufgabe 3 – Kausal geordnetes MQTT

1 Einleitung

Im diesem praktischen Versuch werdet Ihr Euch mit dem MQTT Protokoll und Vektoruhren beschäftigen. In den folgenden Kapiteln wird zuerst eine kleine Anleitung gegeben, wie Ihr den von uns bereitgestellten Broker startet und den Client kompiliert. Danach folgt eine kurze Einführung in MQTT. Wir raten natürlich auch eigenständig nach zusätzlichen Informationen zu suchen. Wichtige Quellen sind dabei natürlich das WWW (z.B. `http://mqtt.org/`) und für die Programmierung die man-Pages. Wir wünschen Euch viel Erfolg bei der Bearbeitung!

2 Grundgerüst in C

Wir haben Euch für diese Aufgabe ein Grundgerüst für den Client in C und eine Broker-Binary vorgegeben. Wir verwenden den mosquitto MQTT Broker und die dazugehörige Client Library in Version 1.1.3 (`http://mosquitto.org/`). Der Broker liegt als Binary für Linux vor. Die Bibliothek wird als 32 und 64bit Version für Linux bereitgestellt und ist dazu gedacht, statisch gelinkt zu werden. Das Grundgerüst könnt ihr mit dem folgen Aufruf kompilieren:

```
gcc -o client -Iinclude/ -Llib/32bit -Llib/64bit *.c -lmosquitto -lpthread
```

Im include-Pfad befindet sich dabei die Header-Datei der libmosquitto, in der die Funktionsrümpfe deklariert sind. In den beiden Unterverzeichnissen in lib finden sich die kompilierten Bibliotheken. Der Linker sucht sich die passende Bibliothek, je nachdem, ob für 32 oder 64bit kompiliert wird. Der folgende Hinweis kann dabei ignoriert werden. Er deutet nur darauf hin, dass eine der beiden Bibliotheken nicht anwendbar war:

```
/usr/bin/ld: skipping incompatible lib/32bit/libmosquitto.a when
searching for -lmosquitto
```

Der Client sollte schon kompilierbar und ausführbar sein, aber noch nichts tun.

Ihr solltet eure Fortschritte mit dem beiliegenden Broker testen. Ihr startet Ihn mit:

```
./mosquitto -p 4711
```

Dabei ist 4711 der Port, auf dem der Broker auf eingehende Verbindungen wartet. Ihr solltet hier gerade auf Uni-Systemen mit potentiell mehreren Nutzern darauf achten, dass Ihr euch selbst einen möglichst zufälligen Port aussucht. Auch hier nochmals der Hinweis: In unseren Broker ist eine Verzögerung eingebaut, die bewirkt, dass sich Nachrichten überholen können!

Der Client kann dann z.B. wie folgt gestartet werden:

```
./client localhost 4711 1
```

Dabei ist das erste Argument der Hostname des Brokers, das zweite der Port und das dritte eine ClientID, welche irgendeine Zahl ist und bei jedem gestarteten Client unterschiedlich sein sollte. Ihr bekommt diese ID in einer Funktion übergeben und müsst sie dann verwerten. In MQTT muss sich jeder Client mit einem eindeutigen String beim Broker anmelden. Wir benutzen in unserer Musterlösung die IDs 0 bis 3 für 4 Clients.

Das sollte genügen, um die Aufgabe lösen zu können. Bei Problemen könnt Ihr euch natürlich immer an die Tutoren wenden oder die Frage im Forum stellen. Wir wünschen euch viel Erfolg bei der Bearbeitung der Aufgabe!

Wer tief greifendere Änderungen vornehmen möchte, findet in der Datei include/mosquitto.h die Funktionen der mosquitto Client library mit entsprechenden Kommentaren. Das sollte aber für das Lösen der Aufgabe nicht nötig sein. Wer sich näher dafür interessiert, kann natürlich die Library auch selbst kompilieren und findet hier auch ein kleines Beispiel zur Nutzung der mosquitto lib:

- `http://mosquitto.org/man/libmosquitto-3.html`

3 MQTT

MQTT steht für Message Queuing Telemetry Transport und ist ein Publish/Subscribe Messaging-Protokoll. Es wurde vor allem für Geräte mit geringer Rechenleistung konzipiert, welche nur über eine unzuverlässige Netzanbindung mit niedriger Bandbreite und hoher Latenz verfügen. Durch diese Eigenschaften eignet sich das Protokoll besonders für das aufkommende „Internet der Dinge“ und für mobile Anwendungen, bei denen genutzte Bandbreite und Stromverbrauch gering sein müssen. Daher wird es sowohl in der Forschung, als auch in Unternehmen vermehrt eingesetzt. Ein Beispiel für eine Anwendung von MQTT ist der Facebook-Messanger auf Mobiltelefonen. MQTT wird zur Zeit von der Standardisierungsorganisation OASIS zu einem offenen Standard weiterentwickelt.

Das MQTT Protokoll basiert auf einige grundlegende Konzepte, wobei hier nur die wichtigsten angeschnitten werden. Mehr Informationen gibt es z.B. hier, sind aber zum Lösen der Aufgabe nicht nötig:

- `http://mqtt.org/`
- `http://mosquitto.org/man/mqtt-7.html`
- `http://www.redbooks.ibm.com/abstracts/sg248054.html`

3.1 Publish/Subscribe Mechanismus

MQTT basiert auf dem Publish/Subscribe-Modell mit Themen (topics) und Abonnements (subscriptions). Dabei können Nachrichten an andere Benutzer verschickt werden, wobei der Sender die Nachricht nicht direkt an andere Benutzer schickt, sondern die Nachricht unter einem bestimmten Thema (topic) über einen Broker veröffentlicht. Auf der anderen Seite können Benutzer, die sich für eine bestimmtes Thema interessieren, dieses beim Broker abonnieren (subscribe) und erhalten dann die Nachrichten, die unter diesem Thema geschickt wurden. Abonnements können explizit sein, d.h. es wird ein bestimmtes Thema angegeben, welches abonniert wird. Zusätzlich können aber auch Wildcards verwendet werden, wie die Raute (#), um Nachrichten für eine Vielzahl von Themen zu erhalten (Probiert das ruhig einmal aus). Der Sender weiß bei diesem Mechanismus grundsätzlich nicht, wie viele Empfänger seine Nachricht erhalten.