



OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG



FAKULTÄT FÜR
INFORMATIK

Institut für Intelligente Kooperierende Systeme (IKS)
Prof. Dr. David Hausheer

Technische Informatik 2 - SS 2018 - Praktische Übung 2

Abgabe: 29.04.2018

1 Prozesse und Threads

Copyright Ariane Keller, Markus Happe
Communication Systems Group
ETH Zurich

Inhaltsverzeichnis

1	Einleitung	1
2	Aufgaben	1
2.1	Threads erstellen	1
2.2	Hilzer's Barbershop-Problem	1

1 Einleitung

Ziel dieser Übung ist es, dass Sie sich mit POSIX Semaphoren und Threads vertraut machen. Nutzen Sie dazu die Manpages (Sektion 3) Ihres Systems. Für Threads sind diese unter anderem `pthread_create(3)`, `pthread_join(3)` und `pthread_kill(3)`, für Semaphoren `sem_init(3)`, `sem_post(3)`, `sem_wait(3)` und `sem_destroy(3)`.

2 Aufgaben

2.1 Threads erstellen

Schreiben Sie ein kleines C-Programm, welches POSIX-Threads erzeugt (`pthread_create(3)`). Dabei soll die Anzahl der zu erzeugenden Threads per Kommandozeile übergeben werden. Jeder Thread soll dann in einer Endlosschleife den Text Thread `<x>`: Hello World! ausgeben, wobei `<x>` die Nummer des erzeugten Threads ist. Nach jeder Textausgabe soll sich der Thread für eine Sekunde schlafen legen (`sleep(1);`). (Das Programm beendet sich erst durch Drücken von CTRL+C.) Beispiel:

```
$ ./a.out 3
Thread 1: Hello World!
Thread 2: Hello World!
Thread 1: Hello World!
Thread 3: Hello World!
Thread 1: Hello World!
Thread 2: Hello World!
...
```

2.2 Hilzer's Barbershop-Problem

Hilzer's Barbershop-Problem ist eine Variante des bekannten Barbershop-Problems von Dijkstra. In diesem Szenario arbeiten drei Friseure gleichzeitig in einem Laden. Es gibt drei Friseurstühle, ein Sofa mit Platz für vier Kunden, einen Warteraum für weitere Kunden und eine Kasse, welche jedoch gleichzeitig von jeweils nur einem Friseur bedient werden kann. Des weiteren dürfen sich aus versicherungstechnischen Gründen insgesamt maximal zwanzig Kunden im Laden aufhalten; weitere Kunden werden umgehend abgewiesen.

Betritt ein Kunde den Laden, so nimmt er auf dem Sofa platz und wartet, bis ein Friseur Zeit für ihn hat. Ist das Sofa schon besetzt, wartet er im Warteraum des Ladens. Sobald ein Friseur frei wird, kommt ein auf dem Sofa wartender Kunde an die Reihe und ein Kunde aus dem Warteraum nimmt auf dem Sofa Platz. Nach erfolgreichem Haarschnitt begeben sich Kunde und Friseur zur Kasse, wo der Kunde seine Rechnung begleicht und im Gegenzug eine Quittung erhält.

Vereinfacht sieht der Ablauf für einen Kunden wie folgt aus:

1. Falls weniger als 20 Kunden, so betrete den Laden

2. Warte bis es Platz auf dem Sofa hat
3. Warte bis ein Friseur frei ist und lasse die Haare schneiden
4. Warte bis ein Friseur Zeit hat um zu bezahlen
5. Verlasse den Laden

Analog gilt für den Friseur:

1. Warte bis ein Kunde bereit ist
2. Scheide ihm die Haare
3. Warte bis die Kasse frei ist
4. Warte bis der Kunde bezahlt hat

In unserem Beispiel gibt es keine strikte Bindung von Kunden zu ihrem Friseur. Es kann also sein, dass ein Kunde von Friseur A die Haare geschnitten bekommt, aber bei Friseur B bezahlt.

Programmieren Sie, basierend auf dem bereitgestellten Template `barbershop.c`, eine korrekte Umsetzung des beschriebenen Barbershop-Problems **mit Hilfe von Semaphoren**. Lösen Sie dazu die unten aufgeführten Teilaufgaben und überprüfen Sie deren Korrektheit anhand der Textausgabe.

Hinweise zur Implementierung:

1. Stellen Sie sicher, dass sich maximal zwanzig Kunden im Laden befinden (Tipp: Critical section)
2. Fügen Sie das Sofa hinzu (Tipp: Auf vier initialisierte Semaphore)
3. Implementieren Sie das Haarschneiden (Tipp: Rendez-Vous)
4. Fügen Sie den Bezahlvorgang hinzu (Tipp: Critical section, Rendez-Vous)
5. **Optional:** Falls sich bereits zwanzig Leute im Laden befinden, verschwinden neue Kunden nicht auf Nimmerwiedersehen, sondern gehen ein Getränk Ihrer Wahl trinken und versuchen es anschliessend nochmals. Das Trinken des Getränks kann mit der Funktion `sleep()` modelliert werden.
6. **Optional:** Erstellen Sie eine ansprechende GUI z.B. in C, welche den Laden, das Mobiliar, die Friseure und die Kunden entsprechend des Programmablaufs darstellt. Ein sehr gutes Ergebnis dieser Teilaufgabe wird mit einem Zusatztestat belohnt.

Grundsätzlich gilt die korrekte Umsetzung der Teilaufgaben eins bis vier als zu erreichendes Lernziel. Entscheidend ist dabei, dass die jeweiligen Synchronisations-Probleme erkannt und ein adäquater Lösungsansatz verfolgt wurde. Desweiteren empfehlen wir auch das Lösen der ersten optionalen Aufgabe.