

1) Aufgabe (Reader/Writer):

Im Folgenden soll das typische Reader/Writer-Problem behandelt werden, wie dies zum Beispiel bei einem Zugriff auf einen Datensatz bei einer Reisebuchung vorkommen kann. Stellen Sie durch Synchronisation mit Semaphoren sicher, dass nur maximal ein Prozess schreibenden Zugriff hat und hierbei insbesondere kein lesender Prozess erlaubt ist. Die Anzahl der gleichzeitig lesenden Prozesse ist unbeschränkt. Verwenden Sie hierzu die folgenden Semaphore:

1. WRITER: Stellt sicher, dass nur maximal ein Prozess schreibt und keiner liest
2. READER: Ein Zähler der die Anzahl der lesenden Prozesse speichert. Achtung: dieses Semaphor wird nicht zum Blockieren verwendet, sondern nur als globaler Variable/Zähler.
3. MUTEX: Zur Sicherstellung der Atomarität der Synchronisationsaktionen, falls erforderlich.

Gehen Sie hierbei wie folgt schrittweise vor.

- a) Entwickeln Sie den Rahmen mit dem Aufruf von 5 Leseprozessen und 2 Schreibprozesse. Implementieren Sie jeweils die Funktionen der Lese- und Schreibprozesse, die in einer Schleife jeweils dreimal einen Zyklus durchlaufen:
Simulierter Zugriff auf Variable; 1 Sekunde warten; unkritischer Bereich; 1 Sekunde warten.
Schützen Sie den kritischen Bereich nicht mit Semaphoren. Der Zugriff erfolgt unsynchronisiert. Überprüfen Sie den Algorithmus mit geeigneten Ausgaben.
- b) Skizzieren Sie eine Lösung der Synchronisation (auf Papier). Spielen Sie die Lösung an einigen Fallbeispielen durch.
- c) Implementieren Sie die zuvor skizzierte Lösung und überprüfen Sie deren Funktion.

2) Aufgabe (RPC):

Es soll eine Addition, Multiplikation und Quadrierung einer Zahl an einem anderen Rechner mit RPC implementiert werden. Gehen Sie hierzu wie folgt schrittweise vor:

- a) Verwenden Sie den folgenden Inhalt der Datei math.x zur Beschreibung der Basisdefinitionen (Datentyp, RPC-Identifikator, Versionsnummer):

```
/* math.x */

struct intpair {
    int a;
    int b;
};

program MATHPROG {
    version MATHVERS {
        int ADD(intpair) = 1;
        int MULTIPLY(intpair) = 2;
        int CUBE(int) = 3;
    } = 1; /*version*/
} = 0x2001000; /*id userdefined*/
```

Erzeugen Sie mit dem Aufruf

```
rpcgen math.x
```

die Dateien: math.h math_svc.c math_clnt.c math_xdr.c.

- b) Erstellen Sie nun den Code für den Server. Binden Sie hierzu den lokalen Header math.h (`include „math.h“`) und den Systemheader `rpc/rpc.h` ein. Erzeugen Sie nun die dementsprechenden Serverprozeduren. Als Beispiel dient die Signatur der Prozedur zum Addieren:

```
int * add_1_svc (pair, rqstp)
    intpair * pair;
    struct svc_req *rqstp; /*not necessary in this example*/
```

Die Erweiterung `_1_svc` wird von `rpcgen` generiert. Hierbei entspricht die 1 der Version.

Den ersten Parameter `a` können Sie mit `pair->a` ansprechen.

Das Binary erzeugen Sie mit folgender Anweisung:

```
cc -o math_server math_svc.c math_xdr.c server.c -lnsl
```

- c) Erstellen Sie den Code für den Client. Binden Sie hierzu ebenfalls die in Aufgabe b) genannten Header ein. Interpretieren Sie die Argumente der Kommandozeile wie folgt: Das erste Argument ist der Name des Servers, das zweite und dritte Argument jeweils zwei ganze Zahlen, deren Summe berechnet werden soll.

Den Zugriff auf den Server erzielen Sie durch die Verwendung eines Client-Handles `cl` (`CLIENT *`), den Sie mit der Funktion `clnt_create` erzeugen:

```
clnt_create (servername, MATHPROG, MATHVERS, "tcp");
```

Die Prozedur `add_1` auf dem Server rufen sie z.B. mit

```
add_1 (&numbers, cl)
```

auf. Hierbei ist `numbers` von der Sorte `intpair`.

Das Binary erzeugen Sie mit folgender Anweisung:

```
cc -o remo_math math_clnt.c math_xdr.c client.c -lnsl
```

- d) Testen Sie das zuerst auf Ihrem Rechner, indem Sie als Server `localhost` verwenden. Dann Testen Sie die Anwendung auf verschiedenen Rechnern.

Hinweis: Es kann erforderlich sein, dass `rpcbind` zu installieren ist:

```
sudo apt-get install rpcbind
sudo -i service rpcbind stop
sudo -i rpcbind -i -w
sudo -i service rpcbind start
```

Hinweis: Falls Sie Server und Client auf verschiedenen Rechnern installieren wollen, müssen Sie bei der Virtualbox die Option Netzwerkbrücke (`networkbridge`) aktivieren. Somit erhalten Sie eine von außen zugängliche Netzwerkadresse.