

Vorbereitung

Machen Sie sich mit den Abschnitten 9.3 und 9.4 des Buchs „Automotive Embedded Systeme“ und den vorgestellten Framework-Klassen für Sockets und der Ihnen als Quelltext zur Verfügung gestellten Implementierung vertraut.

Sichten Sie alle Quelltextdateien, so dass Sie Fragen zur Funktionsweise und Implementierung der Klassen beantworten können.

Aufgabe 1: Kommunikationsaufbau

Ziel ist es, sich mit dem Ablauf des Verbindungsaufbaus auf Programmebene zu beschäftigen und die üblichen Problemfelder der Netzwerkprogrammierung kennen zu lernen.

1. Erzeugen Sie zwei Prozesse per `fork()`. Der Eltern-Prozess soll auf einkommende TCP-Verbindungen warten und Informationen zu den empfangenen Daten ausgeben (z.B. Anzahl der empfangenen Bytes und die Nachricht selbst). Der Kind-Prozess soll kleine Nachrichten (wenige Byte) an den Eltern-Prozess schicken.
2. Was stellen Sie fest, wenn der Kind-Prozess vor dem Eltern-Prozess startet? Warum ist dieser Umstand problematisch? Implementieren Sie eine einfache Methode (ohne das Einstellen von Socket-Optionen), damit der Eltern-Prozess seine Tätigkeit vor dem Kind-Prozess anfängt. Welche Methoden kennen Sie?
3. Nutzen Sie Ihren Code nun, um mit einer anderen Gruppe oder mit Ihrem Laborrechner Nachrichten auszutauschen (ändern Sie hierzu wenn möglich so wenig wie möglich an Ihrem Code).
4. Wie und wann stellen Sie fest, dass der Kind-Prozess die Verbindung erfolgreich geschlossen hat? Was würde passieren, wenn der Kind-Prozess die Verbindung nicht (erfolgreich) schließt?

Tipps:GDB

Wenn Sie in Codeblocks ein Projekt debuggen, welches weitere Prozesse forkt, wird per GDB-StandardEinstellung immer der Elternprozess gefolgt. Wenn Sie jedoch auch den Kind-Prozess debuggen möchten, müssen Sie dies einstellen unter:

Settings → Debugger → GDB/CDB debugger → Debugger initialization commands

Und tragen in das Eingabefeld folgendes Kommando ein:

```
set follow-fork-mode child
```

strace

Mit der Option `-ff` lässt sich die Ausgabe „geforkter“ Prozesse mit erfassen. Kombiniert mit der Option `-o` landet die Ausgabe in verschiedenen Dateien, so dass bei

```
strace -ff -o file myprogram
```

für jeden Prozess eine Datei `file.<pid>` mit der jeweiligen Prozess-ID der Prozesse entsteht.

ifconfig -a

Gibt Ihnen die Konfiguration inklusive IP-Adresse Ihrer Netzwerkschnittstellen aus.

Aufgabe 2: Netzwerkprobleme und Segmentierung

Nun schauen wir uns häufig gemachte Fehler in der Netzwerkprogrammierung an erarbeiten eventuelle Lösungsvorschläge.

1. Nutzen Sie Ihr Programm aus der Aufgabe 1 und vergrößern die zu verschickende Datenmenge auf mindestens 4MB, die Sie als Blob (binary large object) in einem Schritt streamen. Initialisieren Sie den Blob als ein großes Array, welches Sie z.B. mit Zufallszahlen füllen – wichtig ist die zu versendende Datenmenge.
2. Geben Sie in Ihrem Eltern-Prozess (sofern Sie dies nicht schon zuvor getan haben) nun aus, wie viele Byte sie empfangen haben. Streamen Sie lokal und zwischen zwei Rechnern (Raspberry und Laborrechner). Was stellen Sie fest?
3. Analysieren Sie den Weg der Pakete zwischen den beiden Sockets mittels ping. Welches ist der Parameter, der bestimmt, wieviele Pakete beim Empfänger ankommen? Welche Größe besitzt dieser?
4. Korrigieren Sie Ihre Datenübertragung beim Empfänger, so dass Sie in jedem Fall den vollständigen Blob empfangen.
5. Korrigieren Sie den Sender, so dass jedes Paket unverändert beim Sender ankommt.

Tipps:

nc

Mit Hilfe des CLI-Tools *netcat* (oder abgekürzt *nc*) können Sie Socket-Verbindungen aus der Konsole starten und ASCII-Zeichen austauschen:

```
nc -v 127.0.0.1 12345
```

stellt z.B. eine TCP-Verbindung zu Ihrem lokalen Betriebssystem durch den Port 12345 her. Wenn die Verbindung steht, können Sie nun Zeichen eintippen und mit der Enter-Taste abschicken. Mehr Informationen (wie z.B. das Einrichten eines Listeners) entnehmen Sie bitte dem Manual (*man netcat*).

ping

Mittels ping können Sie Daten an die Netzwerkschnittstelle eines anderen Rechners schicken. Mit dem Befehl

```
ping <ip-adresse> -s<size> -M...
```

können Sie Pakete beliebiger Größe verschicken und die „Path MTU Discovery“-Strategie auswählen.