

Programmieraufgaben im Modul EZMR

Elektro- und Informationstechnik
für Prozessautomation und eingebettete Systeme

Moritz Thümmeler
moritz.thuemmler@htwk-leipzig.de

1 Hinweise

1. Als Programmiersprache für beide Aufgaben sollte verwandt werden.
2. Nutzen Sie die Man-Pages des Systems (Linux).
3. Beispiele für nützliche Man-Pages: `man shmctl socket msgctl semctl`
4. Hinweise zum Erstellen einer Virtual Machine (VM) können der Praktikumsanleitung entnommen werden.

2 Sockets/Netcode

In dieser Aufgabe geht es um die Erstellung von Sockets und deren Verwendung für die Kommunikation von Programmen. Diese Art der Kommunikation ist die Grundlage heutiger EDV-Dienste. Zu erstellen sind mehrere Programme, welche folgende Anforderungen erfüllen:

Teilaufgabe a)

- Erstellen einer Netzwerkverbindung (Port/IP) zur Entgegennahme von Anfragen.
- Das Programm kann ggf. auf Anfragen antworten und tut dies auch. (Protokoll/PDUs)

Teilaufgabe b)

- Eine oder mehrere Anwendungen sind zu erzeugen, welche Kontakt mit a) aufnehmen und Daten senden oder empfangen können. (Tests: Netcat)

Teilaufgabe c)

- Anwendung a) stellt eine Liste zur Verfügung, welche von b)-Anwendungen abgefragt und manipuliert werden kann.
- b)-Anwendungen können anhand dieser Liste auf Nachrichtenverläufe zugreifen (lesen/schreiben).

Teilaufgabe d)

- Es soll keine Unterscheidung mehr zwischen a)- und b)-Anwendungen existieren. (Vorsicht: Das kann aus dem Ruder laufen, wenn man nicht vorsichtig ist.)

3 IPC

Ausgehend von den Grundlagen aus der Vorlesung machen Sie sich mit den Funktionsweisen von IPC-Varianten vertraut und entscheiden Sie sich für eine Methode für die Durchführung der Aufgaben im Folgenden.

Teilaufgabe a)

- Erstellen einer IPC-Infrastruktur zur Entgegennahme von Anfragen.
- Das Programm kann ggf. auf Anfragen antworten und tut dies auch. (Protokoll/PDUs)

Teilaufgabe b)

- Eine oder mehrere Anwendungen sind zu erzeugen, welche Kontakt mit a) aufnehmen und Daten senden oder empfangen können. (Tests: ipcs ipcrm)

Teilaufgabe c)

- Anwendung a) stellt eine Liste zur Verfügung, welche von b)-Anwendungen abgefragt und manipuliert werden kann.
- b)-Anwendungen können anhand dieser Liste auf Nachrichtenverläufe zugreifen (lesen/schreiben).

Teilaufgabe d)

- Es soll keine Unterscheidung mehr zwischen a)- und b)-Anwendungen existieren. (Vorsicht: Das kann aus dem Ruder laufen, wenn man nicht vorsichtig ist.)

Bemerkung: Damit solche Dienste oder Programme im Hintergrund laufen können und als Systemdienste(Windows) bzw. Dämonen(Unix/Linux) fungieren können, müssen noch weitere Voraussetzungen erfüllt werden. Bei Interesse: <https://lloydrochester.com/post/c/unix-daemon-example/>