

## Aufgabe 1

### Begriffe

Beschreiben Sie jeden der folgenden Begriffe mit maximal einem Satz. Instruction Set, Process, File, Syscall, Interrupt, Microkernel, monolithic Kernel.

### Textaufgabe

Angenommen Sie sind in der Industrie bei einem Autohersteller tätig. Sie sollen ein Betriebssystem für den Bordcomputer des neuesten Modells entwerfen. Welche Wahl treffen Sie in Punkto I/O Verwaltung, wenn es um die Abarbeitung externer Geräte wie Navigationssystem, ABS oder Air Bag geht?

## Aufgabe 2

### Produzierendes System

In dieser Veranstaltung wird der Umgang mit der Programmiersprache C gelehrt. Da C eine plattformabhängige Sprache ist, sind einige Richtlinien nötig, um die korrekte Funktionalität ihrer Programme auf den Referenzsystemen sicher zu stellen. Es empfiehlt sich ein solches produzierendes System minimal als Virtuelle Maschine einzurichten, oder an den Poolrechnern zu arbeiten. Probleme, die auf die Nichteinhaltung der Anforderungen zurückzuführen sind, können nicht verfolgt werden und führen zu fehlerhafter Abgabe. Anforderungen an ihr produzierendes System sind:

1. Linux 64 Bit (z.B. Linux Mint, Ubuntu, ...)
2. GNU C Compiler (GCC) mit Standardbibliotheken
3. Shell (z.B. Zsh, Bash, ...)
4. Editor (z.B. Atom, Emacs, Vim, ...)

Ihr Programm muss mit folgenden Compilerflags ohne Warnungen und / oder Fehler compilieren:

```
$ gcc -std=c11 -o <program.out> -Wall -Wextra -pedantic <program.c>
```

### C-Aufgabe

Erstellen Sie ein Programm, dass  $\langle n \rangle$  Kommandozeilenparameter bekommt, die ersten  $\langle n-1 \rangle$  Kommandozeilenparameter addiert und prüft ob diese Summe dem  $\langle n \rangle$ ten Kommandozeilenparameter entspricht. Ist dies der Fall gibt es den char-Wert '1' aus, sonst den char-Wert '0'. Führen Sie Fehlerbehandlung durch und erläutern Sie im Quellcode (als Kommentar) warum Sie sich für genau diese Fehlerbehandlung entschieden haben. Beachten Sie den Typ der übergebenen Parameter und nutzen Sie die inttypes.h-Bibliothek.

Beispielausgabe:

```
$ ./program 1 2 3 1 1 8
$ 1
$ ./program 1 2 3 1 1 9
$ 0
```