

Hardwarenahe Programmierung / Angewandte Informatik

Übungsaufgaben – 4. Dezember 2017

Diese Übung enthält Punkteangaben wie in einer Klausur. Um zu „bestehen“, müssen Sie innerhalb von 80 Minuten unter Verwendung ausschließlich zugelassener Hilfsmittel 13 Punkte (von insgesamt 28) erreichen.

Aufgabe 1: Trickprogrammierung

Wir betrachten das folgende Programm (Datei: [aufgabe-1.c](#)):

```
#include <stdio.h>
#include <stdint.h>

int main (void)
{
    uint64_t x = 4262939000843297096;
    char *s = &x;
    printf ("%s\n", s);
    return 0;
}
```

Das Programm wird kompiliert und auf einem 64-Bit-Little-Endian-Computer ausgeführt:

```
$ gcc -Wall -O aufgabe-1.c -o aufgabe-1
aufgabe-1.c: In function 'main':
aufgabe-1.c:7:13: warning: initialization from incompatible pointer type [...]
$ ./aufgabe-1
Hallo
```

- (a) Erklären Sie die Warnung beim Compilieren. (2 Punkte)
- (b) Erklären Sie die Ausgabe des Programms. (5 Punkte)
- (c) Wie würde die Ausgabe des Programms auf einem 64-Bit-Big-Endian-Computer lauten? (3 Punkte)

Hinweis: Modifizieren Sie das Programm und lassen Sie sich Speicherinhalte ausgeben.

Aufgabe 2: Thermometer-Baustein an I²C-Bus

Eine Firma stellt einen elektronischen Thermometer-Baustein her, den man über die serielle Schnittstelle (RS-232) an einen PC anschließen kann, um die Temperatur auszulesen. Nun wird eine Variante des Thermometer-Bausteins entwickelt, die die Temperatur zusätzlich über einen I²C-Bus bereitstellt.

Um das neue Thermometer zu testen, wird es in ein Gefäß mit heißem Wasser gelegt, das langsam auf Zimmertemperatur abkühlt. Alle 10 Minuten liest ein Programm, das auf dem PC läuft, die gemessene Temperatur über beide Schnittstellen aus und erzeugt daraus die folgende Tabelle:

Zeit / min.	Temperatur per RS-232 / °C	Temperatur per I ² C / °C
0	94	122
10	47	244
20	30	120
30	24	24
40	21	168

- (a) Aus dem Vergleich der Meßdaten läßt sich auf einen Fehler bei der I²C-Übertragung schließen. Um welchen Fehler handelt es sich, und wie ergibt sich dies aus den Meßdaten? (5 Punkte)
- (b) Schreiben Sie eine C-Funktion `uint8_t repair(uint8_t data)`, die eine über den I²C-Bus empfangene fehlerhafte Temperatur `data` korrigiert. (5 Punkte)

Aufgabe 3: XBM-Grafik

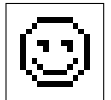
Bei einer XBM-Grafikdatei handelt es sich um ein als C-Quelltext abgespeichertes Array, das die Bildinformationen enthält:

- Jedes Bit entspricht einem Pixel.
- Nullen stehen für Weiß, Einsen für Schwarz.
- LSB first.
- Jede Zeile des Bildes wird auf ganze Bytes aufgefüllt.
- Breite und Höhe des Bildes sind als Konstantendefinitionen (**#define**) in der Datei enthalten.

Sie können eine XBM-Datei sowohl mit einem Texteditor als auch mit vielen Grafikprogrammen öffnen und bearbeiten.

Beispiel ([aufgabe-3.xbm](#)):

```
#define aufgabe_3_width 14
#define aufgabe_3_height 14
static unsigned char aufgabe_3_bits[] = {
    0x00, 0x00, 0xf0, 0x03, 0x08, 0x04, 0x04, 0x08, 0x02, 0x10, 0x32, 0x13,
    0x22, 0x12, 0x02, 0x10, 0x0a, 0x14, 0x12, 0x12, 0xe4, 0x09, 0x08, 0x04,
    0xf0, 0x03, 0x00, 0x00 };
```



Ein C-Programm, das eine XBM-Grafik nutzen will, kann die [.xbm](#)-Datei mit **#include** `"..."` direkt einbinden.

Schreiben Sie ein Programm, das die XBM-Datei als ASCII-Grafik ausgibt, z. B.:

```

  * * * * *
 *           *
*           *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*   *   *   *
*           *
 *           *
  * * * * *
```

(8 Punkte)

(Hinweis für die Klausur: Abgabe auf Datenträger ist erlaubt und erwünscht, aber nicht zwingend.)

Viel Erfolg!