

### 3. Übung zur Vorlesung „Organisation und Architektur von Rechnern“

Integer Daten II

Abgabe am Montag, 02. Mai – 16:00

Ab diesem Übungsblatt soll die Abgabe in Zweiergruppen stattfinden. Wenn Sie sich im iLearn mit einem anderen Teilnehmer zu einer Gruppe verbinden, muss nur noch einer der beiden Lösungen einreichen. Allerdings müssen beide Teilnehmer in der Lage sein, alle eingereichten Lösungen während der Übungsstunde zu erläutern, ansonsten können Punkte abgezogen werden.

Beachten Sie, dass die Verbindung zu einer Zweiergruppe nur funktioniert, wenn beide Teilnehmer noch keine Lösung zu den aktuell laufenden Aufgaben eingereicht haben.

#### Aufgabe 1 - Puffergröße

4 Punkte

Sie haben die Aufgabe eine Funktion zu schreiben, die den Inhalt einer Datenstruktur in einen Puffer kopiert, aber nur wenn genügend Platz im Puffer verfügbar ist. Sie schreiben folgenden Code:

```
void copy_data(struct data_t *val, void *buf, int maxbytes)
{
    if (maxbytes - sizeof(*val) >= 0) {
        memcpy(buf, val, sizeof(*val));
    }
}
```

Dabei ist `struct data_t` eine beliebige Datenstruktur. Die Funktion `sizeof` bestimmt die Anzahl an Bytes, die ihr Argument im Speicher belegt. Die Funktion `memcpy(void *dest, const void *src, size_t n)` kopiert `n` Bytes von `src` nach `dest` und ist in `string.h` definiert.

Sie testen den Code sorgfältig und stellen fest, dass er nicht funktioniert wie erwartet, weil er *immer* den Inhalt von `val` in den Puffer `buf` kopiert, sogar wenn `maxbytes` zu klein ist.

(a) Erklären Sie warum die Daten immer kopiert werden.

(b) Verändern Sie die Funktion, so dass sie sich korrekt verhält, d.h. sie kopiert den Inhalt von `val` nur dann, wenn `maxbytes` groß genug ist. Schreiben Sie zudem eine `main` Funktion, welche das korrekte Verhalten von `copy_data` demonstriert. Verwenden Sie dazu folgende Datenstruktur.

```
struct data_t {
    char* id;
    int x, y;
};
```