

Übung 9

Subroutinen

Ziele dieser Übung:

Sie sollen in der Lage sein,

- 1) Subroutinen in einem Assembler-Projekt korrekt zu implementieren.

Verwenden Sie für den Test Ihres Codes den Debugger und das Leguan-Board.

Aufgabe 9.1 Verarbeitung von Strings

Eine Subroutine soll die Anzahl 'e' in einem String zählen und zurückgeben. Als C-Funktion hätte die Subroutine folgende Deklaration:

```
int count_e(const char *str);
```

Die Subroutine erwartet als Parameter einen Pointer auf das erste Zeichen des Strings. Der String ist mit '\0' abgeschlossen. Der Rückgabewert ist die Anzahl gefundener Charakter 'e'.

Schreiben Sie die Subroutine "count_e" in Assembler sowie ein kleines Testprogramm. Dieses Testprogramm soll zwei verschiedene Strings anlegen und die Subroutine "count_e" für jeden String aufrufen.

Aufgabe 9.2 Länge eines Strings bestimmen

In "C" gibt es zur Bestimmung der Länge eines Strings die Funktion strlen(). strlen liefert die Länge einer Zeichenkette ohne das abschliessende Zeichen '\0' und hat folgende Deklaration:

```
size_t strlen(const char *str);
```

Schreiben Sie eine Subroutine "asm_strlen", welche die Länge eines Strings bestimmt. Beachten Sie, dass der Parameter "str" in r0 übergeben wird. Die Länge des Strings, d.h. der Rückgabewert der Subroutine, wird ebenfalls in r0 zurückgegeben. Schreiben Sie weiter ein kurzes Testprogramm, welches einen String anlegt und die Subroutine asm_strlen aufruft.

Versuchen Sie, den Code möglichst zu optimieren, d.h. dass im Loop, welcher die einzelnen Zeichen des Strings zählt, möglichst wenige Assembler-Instruktionen verwendet werden.

Aufgabe 9.3 Sortieren eines Arrays (für Fortgeschrittene)

Eine Subroutine soll den Inhalt eines Arrays sortieren, sodass am Schluss beim tiefsten Index der kleinste Wert steht, beim höchsten Index der grösste Wert. Als C-Funktion hätte die Subroutine folgende Deklaration:

```
void sort_array(int array[], int array_size);
```

Der Parameter array gibt die Basisadresse des Arrays an, array_size gibt die Grösse des zu sortierenden Arrays an.

Als Algorithmus wird Bubblesort verwendet. Der Assembler-Code für die Subroutine "sort_array" ist etwas komplizierter, weshalb Sie den Code sowie ein kleines Testprogramm schon auf Moodle finden. Versuchen Sie den Code zu verstehen und beachten Sie insbesondere auch die Kommentare. Es ist ein guter Ansatz, wenn Sie bei komplexeren Subroutinen die Funktionalität in Pseudocode oder einer Hochsprache beschreiben und innerhalb der Assembler-Zeilen darauf verweisen.