Hochschule Augsburg
Prof. Dr. Hubert Högl
E-mail: Hubert.Hoegl@hs-augsburg.de
WWW: http://hhoegl.informatik.hs-augsburg.de/hhwiki/SysProg

Update: 2013-10-21

Übungen in Systemnaher Programmierung

Übungsblatt 3

1. Aufgabe (Diverses)

1. Zeichnen Sie alle Register der x86 CPU in ein Diagramm. Wie ist der Name für die 8-, 16- und 32-Bit Register?

2. Was bedeuten die folgenden Pseudo-Anweisungen?

   ```
   .data. .bss, .text, .byte, .hword, .long, .int, .ascii, .asciz, .lcomm,
   ```

3. Was bedeutet die folgende Zeile?

   ```
   .global _start, demofunc
   ```

4. Übersetzen Sie die folgende Anweisung in Maschinencode:

   ```
   str1:  .asciz  "SYSPROG"
   ```

5. GNU Assembler Manual

   Installieren Sie unter Linux das Paket `binutils-doc`. Darin ist unter anderem das Handbuch für den GNU Assembler enthalten. Nach der Installation sollten Sie es in elektronischer Form auf dem Rechner lesen können mit `info as`. Das Programm `info` ist der *GNU Info Reader*:

   http://www.gnu.org/software/texinfo

6. Adressierungsarten üben:

   a. Denken Sie die Beispiele in 01-addr-modes/ durch so dass Sie wissen, was bei jedem Befehl passiert.

      Mit dem Makefile können Sie das Programm übersetzen. Im Makefile sind Optionen an den Assembler und Linker enthalten, die sogenannte *Listings* erzeugen. Im Listing `main.lst` finden Sie die erzeugten Maschinenbefehle. Wie lang sind die einzelnen Befehle?

   b. Ist "movl %eax, %ebx" ein gültiger Befehl?

   c. Ist "movl (%eax), (%ebx)" ein gültiger Befehl?

7. Bedingte Sprünge

Schreiben Sie ein kleines Programm, mit dem Sie testen können, ob ein Sprung genommen wurde oder nicht. Der Exit-Code des Programms sollte darüber Auskunft erteilen:

```
movl $..., %eax
cmpl $0, %eax
je exit1

exit0:
   ...

exit1:
   ...
```

Experimentieren Sie mit verschiedenen Werten für

- eax
- die Konstante im `cmpl` Befehl
- den Sprung: je, jne, jg, jge, jl, jle

Nehmen Sie auch negative Werte!

8. Entfernen Sie das `_start` Label bei einem Programm und untersuchen Sie, was dann beim Assemblieren passiert. Verändern Sie in einem zweiten Schritt den Namen dieses Labels.

2. Nochmal: Maximum Suche (zweiter Teil von Kapitel 3)

Wer es noch nicht erledigt hat: *Use the concepts* am Ende von Kapitel 3 (Maximum Suche) machen!

Der Code ist hier:

prog-3-2/maximum.s

Sie sollten nun eine Sitzung mit dem Debugger machen und dabei das Programm Schritt für Schritt (also Anweisung für Anweisung) durchgehen und sich die Funktionsweise klar machen. Die elementarsten GDB Kommandos finden Sie hier:

http://elk.informatik.hs-augsburg.de/sysprog/gdb/gdb.html

Danach sind Sie fit um die Aufgaben "Use the Concepts" am Ende von Kapitel 3 zu machen:

- Modifizieren Sie das Programm so dass es den Wert 3 zurück gibt.
- Modifizieren Sie das Programm so dass es das Minimum statt des Maximum zurück gibt.
- Modifizieren Sie das Programm so dass die Zahl 255 das Array beendet und nicht die Zahl 0.
- Modifizieren Sie das Programm so dass es eine Ende-Adresse für das Array gibt (nicht den abschliessenden Wert 0).
- Modifizieren Sie das Programm so dass es einen Längenzähler für das Array gibt (nicht den abschliessenden Wert 0).
- Was macht `movl _start, %eax`? Was macht hingegen `movl $_start, %eax`?

3. Aufgabe

Lesen Sie das Kapitel 4 ("All About Functions").