

## Hardwarenahe Programmierung Gruppe 6 (Stefan)

*In dieser Übung wenden Sie Ihre gesammelten Assembler-Fähigkeiten an und verknüpfen Assembler- und C-Programme. Die Aufgaben 1, 2 und 3 werden vollständig in Assembler gelöst. In Aufgaben 4 bzw. 5 müssen Sie (auch) ein C-Programm schreiben.*

**Wichtig:** Beachten Sie die Richtlinien für Assemblerprogrammierung aus dem ILIAS-Lernmodul!

- **Bearbeiten Sie bitte nur Aufgabe 5**, wenn Sie bereits eine Prüfung in “Informatik 2” bzw. “Einführung in die technische Informatik” inklusive deren Assembler-Teil bestanden haben und daher an den Praktikumstagen 2 und 3 nicht teilnehmen mussten. Sie haben bis zum Ende der heutigen Übung Zeit, die Aufgabe zu lösen.
- **Bearbeiten Sie bitte nur Aufgaben 1 bis 4**, falls der im vorherigen Punkt beschriebene Sonderfall nicht auf Sie zutrifft! Wenn Sie nicht innerhalb der heutigen Übung fertig werden, haben Sie bis morgen Zeit (insg. 24h ab Start der heutigen Übung), Ihre Abgaben in ILIAS hochzuladen. Sie müssen alle Aufgaben schaffen!

### Aufgabe 1 Potenz

Wir betrachten zwei Algorithmen zur Berechnung von  $a^b$  für zwei natürliche Zahlen  $a$  und  $b$ . Alle Funktionen, die Sie in dieser Aufgabe schreiben müssen den C-Aufrufkonventionen entsprechen. Insbesondere müssen also die Parameter über den Stack übergeben werden und die Rückgabe in EAX erfolgen.

- (a) Die iterative Variante des Algorithmus lautet in Pseudocode:

```
potenz_it(a, b):
    int x = 1
    while (b != 0):
        x = x * a
        b = b - 1
    return x
```

Implementieren Sie diese in Assembler.

- (b) Schreiben Sie ein Hauptprogramm, das zwei Zahlen **a** und **b** von der Konsole einliest, die iterative Funktion aufruft und das Ergebnis ausgibt.

(c) Die rekursive Variante des Algorithmus lautet in Pseudocode:

```
potenz_rek(a, b):  
    if (b == 0):  
        return 1  
    else:  
        return potenz_rek(a, b-1) * a
```

Implementieren Sie diese in Assembler.

(d) Schreiben Sie ein Hauptprogramm, das zwei Zahlen **a** und **b** von der Konsole einliest, die rekursive Funktion aufruft und das Ergebnis ausgibt.

*Hinweis: Fehlerhafte Eingaben müssen Sie nicht abfangen.*

## Aufgabe 2 Stackentwicklung

Gegeben das Assemblerprogramm *stack03.asm*. Einige Teile dieser Aufgabe müssen Sie schriftlich (also in einer normalen Textdatei) einreichen.

- (a) Geben Sie mittels einer mathematischen Vorschrift an, was die Funktion `calc` berechnet.
- (b) Welchen Fehler macht der Programmierer bei der Stackverwaltung? Erklären und korrigieren Sie diesen.
- (c) Skizzieren Sie für jede der im Quellcode markierten Stellen, wie der Stack zu diesem Zeitpunkt aussieht. Orientieren Sie sich dafür an der folgenden Skizze. Gehen Sie davon aus, dass der Stack vor Aufruf der Funktion `calc` wie folgt aussieht:

+-----+	
Rücksprungadresse (zu C)	
+-----+	
gesicherter EBP	<- aktueller EBP
+-----+	
gesicherte Register	
+-----+	
3 (push eax)	<- aktueller ESP
+-----+	

## Aufgabe 3 C-Funktionen

Schreiben Sie ein Assembler-Programm, welches die C-Funktion `puts()` aus Assembler heraus mit einer beliebigen Eingabe aufruft und das Ergebnis auf der Konsole ausgibt.

#### Aufgabe 4 *C und Assembler*

Gegeben ist die Assemblerdatei *secret.asm*.

- (a) Passen Sie die Datei so an, dass die Funktion `secret_func` von einem C-Programm aufgerufen werden kann. Als Hilfestellung wurden die Abschnitte markiert, in denen noch etwas fehlt.
- (b) Schreiben Sie ein C-Programm, welches die Funktion `secret_func` aus *secret.asm* mit einem ganzzahligen Übergabeparameter aufruft.
- (c) Geben Sie den ganzzahligen Rückgabewert auf der Konsole aus und erläutern Sie, welche Berechnung die Funktion ausführt.

#### Aufgabe 5 *C und Assembler*

**Falls Sie an den Assemblertagen nicht teilnehmen mussten, bearbeiten Sie bitte nur diese Aufgabe!**

Gegeben ist die Objektdatei *secret.o*.

- (a) Schreiben Sie ein C-Programm, welches die Funktion `secret_func` aus *secret.o* mit einem ganzzahligen Übergabeparameter aufruft.
- (a) Geben Sie den ganzzahligen Rückgabewert auf der Konsole aus und erläutern Sie, welche Berechnung die Funktion ausführt.