Hochschule Augsburg
Fakultät für Informatik

Sommersemester 2018

Prüfung

Prüfungsfach:	Systemnahe Programmierung
Datum/Uhrzeit:	6. Juli 2018 / 8:30 Uhr
Raum:	J3.19
Prüfer:	Dr. Hubert Högl
Dauer:	**60** Minuten
Hilfsmittel:	keine

Hinweise:

1. Dieses Angabenblatt hat auch eine **Rückseite**! Bitte sofort überprüfen.
2. **Schreiben** Sie bitte **nicht** auf das **Angabenblatt**. Verwenden Sie für Ihre Antworten die separat ausgeteilten Bögen. **Die Angaben dürfen Sie behalten.**
3. Schreiben Sie **nicht mit Bleistift**.

Viel Glück!

Aufgabe 1 (6 Punkte)

Betrachten Sie folgenden Assembler-Befehl:

```
movl    data_items(, %edi, 4), %eax
```

(a) Nennen Sie die einzelnen Bestandteile der verwendeten Adressierungsart.

(b) Welche Adresse wird mit welcher Breite angesprochen, wenn man `data_items` mit `0xa3cd` und `edi` mit 4 annimmt?

(c) Um welchen weiteren Freiheitsgrad könnte man diese Adressierungsart erweitern (Tipp: vor dem Komma)?

Punkte: (a) 3, (b) 2, (c) 1

Aufgabe 2 (4 Punkte)

Fragen zur Endianness

(a) Was bedeutet die „Endianness“ eines Rechners?

(b) Wie kann man herausfinden, welche Endianness ein Rechner hat? Beschreiben Sie das verwendete Prinzip und geben Sie auch die nötigen Assembler-Befehle an.

Punkte: (a) 2, (b) 2

Aufgabe 3 (3 Punkte)

Wozu dienen die drei Bereiche `.data`, `.bss` und `.text` in einem Programm:

```
.section .data
...    # was ist hier?
.section .bss
...    # und hier?
.section .text
...    # und hier?
```

Aufgabe 4 (4 Punkte)

Hier sind einige Fragen zur C Aufrufkonvention:

1. In welcher Reihenfolge werden die Argumente der Funktion `cfun(int a, int b, int c)` auf dem Stack abgelegt?
2. Wie wird der Rückgabewert einer Funktion an den Aufrufer übergeben? Unterscheiden Sie: (a) der Wert ist 32-Bit gross, (b) der Wert ist grösser als 32-Bit.
3. Wer kümmert sich um die Sicherung der Register – der Aufrufer oder der Aufgerufene?
4. Wer korrigiert den Stack, der Aufrufer oder der Aufgerufene?

Aufgabe 5 (5 Punkte)

Der folgende Systemaufruf `ptrace` wird von Debuggern verwendet, um Prozesse zu debuggen. Er hat die Nummer 26. Schreiben Sie in Assembler einen beispielhaften Systemaufruf von `ptrace` hin. Die Schreibweise `void *vp` bezeichnet einen Zeiger `vp` auf einen beliebigen Typ. Das vierte Argument wird in `esi` übergeben.

```
long ptrace(int request, int pid, void *addr, void *data);
```

Aufgabe 6 (4 Punkte)

Schreiben Sie in Assembler eine Funktion „string copy“

```
int strcpy(char *s1, char *s2)
```

die einen Null-terminierten String `s1` an die Adresse `s2` kopiert. Die Schreibweise `char *s1` bedeutet, dass `s1` ein Zeiger auf Character ist. Schreiben Sie die Funktion vollständig mit Prolog und Epilog. Der Rückgabewert soll immer Null sein.

Aufgabe 7 (8 Punkte)

Punkte: a) 2, b) 2, c) 2, d) 2

Im Buch von Bartlett wird im Kapitel 9 die Funktionsweise des Heap erklärt. Beantworten Sie dazu folgende Fragen:

a) Wie heissen die wesentlichen Funktionsaufrufe zur Verwendung des Heap?

b) Was verstehen Sie unter *Unmapped Memory*?

c) Wie kann der Heap-Speicherbereich vergrössert werden?

d) In welchen Datenstrukturen werden die vom Heap angeforderten Speicherblöcke verwaltet? Zeichnen Sie ein Diagramm zur Erläuterung.

Aufgabe 8 (4 Punkte)

Was machen folgende GDB Kommandos?

(a) `(gdb) list main`

(b) `(gdb) br 15`

(c) `(gdb) p/x $edi`

(d) `(gdb) x/4xw $esp+8`