

---

## Klausur

Prüfungsfach: Systemnahe Programmierung (Bachelor)  
Datum/Uhrzeit: 2. Februar 2012 / 12:30 Uhr  
Raum: J401  
Prüfer: Dr. Hubert Högl  
Dauer: **60** Minuten  
Hilfsmittel: keine

### Hinweise:

1. Dieses Angabenblatt umfasst 5 Seiten.
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.

---

### Aufgabe 1 (4 Punkte)

Worin unterscheiden sich Programmiersprachen die sich zur maschinennahen (= systemnahen) Programmierung eignen von Sprachen, die man „high-level“ Sprachen nennt? Nennen Sie jeweils zwei Sprachen aus jeder Gattung.

### Aufgabe 2 (4 Punkte)

Beantworten Sie bitte folgende Fragen zum Stack:

1. Wie heißen die Assemblerbefehle um auf auf den Stack zuzugreifen?
2. Wie implementieren Sie diese Befehle aus elementaren `mov`, `sub` und `add` Befehlen?

### 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 (2 Punkte)

Statten Sie die folgende Funktion `add(a, b)` mit einem Prolog und einem Epilog aus. Die Funktion gibt die Summe der beiden Argumenten zurück.

```
add:
    movl 8(%esp), %eax
    addl 4(%esp), %eax
    ret
```

### Aufgabe 5 (4 Punkte)

Sehen Sie sich das folgende Assemblerprogramm an und beantworten Sie die unten folgenden Fragen dazu:

```
.section .data

array:
    .byte 1, 2, 3, 4, 5, 6, 7, 8, 0

.section .bss
    .lcomm speicher, 8 * 8

.section .text

.globl _start

_start:
    movl $array, %eax
    movl $speicher, %ebx
loop:
    cmpb $0, (%eax)
    je exit
    movb (%eax), %cl
    movb %cl, speicher(%ebx)
    addl $1, %eax
    addl $1, %ebx
    jmp loop
exit:
    movl $1, %eax
    movl $0, %ebx
    int $0x80
```

1. (4 Punkte) Stellen Sie sich die Variable `speicher` als 2-dimensionale Speichertabelle vor mit 64 Einträgen (siehe die folgende Tabelle). Welche Werte stehen in der Tabelle wenn das Programm am Label `exit` angelangt ist?
2. (4 Punkte) Wie müssen Sie das Programm verändern, so dass die Werte aus `array` in die **Diagonale** der Speichertabelle geschrieben werden. Sie haben dazu mit `edx` noch ein Register frei. Schreiben Sie die dazu nötige Änderung in Ihr Lösungsblatt.

Speichertabelle:

<speicher>:	0x00							
<speicher+8>:	0x00							
<speicher+16>:	0x00							
<speicher+24>:	0x00							
<speicher+32>:	0x00							
<speicher+40>:	0x00							
<speicher+48>:	0x00							
<speicher+56>:	0x00							

### Aufgabe 6 (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 7** (4 Punkte)

Obwohl Sie den Systemaufruf `poll()` wahrscheinlich nicht kennen, können Sie seinen Aufruf in Assembler sicher skizzieren. Dieser Aufruf hat die Nummer 168.

```
int poll(struct pollfd *fds, nfd_t nfd, int timeout);
```

In Ihrer Lösung nennen Sie die Parameter einfach `fds`, `nfd` und `timeout`. Wohin müssen diese Parameter übergeben werden?

**Aufgabe 8** (4 Punkte)

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

1. Wie heissen die wesentlichen Funktionsaufrufe zur Verwendung des Heap?
2. Was verstehen Sie unter *Unmapped Memory*?
3. Wie kann der Heap-Speicherbereich vergrössert werden?
4. In welchen Datenstrukturen werden die vom Heap angeforderten Speicherblöcke verwaltet? Verwenden Sie ein Diagramm zur Erläuterung.

---

Ende der Klausur

---