

KSP - Tutorium

Test 2 Africa's Geek



KSP

Test für die Klausurvorbereitung – KSP

Nachname:	
Vorname:	
Betreuer:	Donchi Fofack Donald

	Max. Punktzahl	erreicht
Aufgabe 1	21	
Aufgabe 2	9	
Aufgabe 3	22	
Aufgabe 4	8	
Aufgabe 5	25	
Wissensfragen	15	
Gesamt	100	

Aufgabe 1: (21 Punkte)

Gegeben ist folgender Code:

```
int main(int argc, char const *argv[])
{
    long z[] = {2, 19, 40, 12, 0};
    int a[5] = {5, 4, 3, 2, 1};
    int *b = a + 2;
    signed int c=0;

    char str[] = "Hallo Welt";
    char q = str[b[-1]];
    printf("z[z[z[0]]] = %ld\n", z[z[z[4]]] );
    printf("*b = %d\n", *b);
    printf("*(str+3) = %c\n", *(str + 3));
    printf("*str+3 = %c\n", *str +3);
    printf("q = %c\n", q);
    printf("sizeof(str) = %ld\n", sizeof(str));
    printf("sizeof(z) = %ld\n", sizeof(z));
    return 0;
}
```

a) Ergänzen Sie die umgangssprachlichen Erklärungen zu obenstehenden Deklarationen. Zu Vorbelegungen brauchen Sie nichts schreiben! (benutzen Sie dabei deutsche Begriffe): (7 Punkte)

z ist ...

a ist ...

b ist ...

c ist ...

str ist ...

q ist ...

main ist ...

b) welche Ausgabe erscheint bei der Ausführung des Programms?

NB: Der Architektur des Zielsystems X86_64 (14 punkte)

.....

Aufgabe 2: (9 Punkte)

Wir haben zwei c-datei: **main.c** und **helper.c**:

Welcher Befehl benötigt man, damit ein ausführbares Programm (**main**) aus den beiden C-Quellcode erstellt wird?

Dabei soll alle Warnungen ausgegeben werden, das Debug-symbol soll im Programm vorhanden sein, das Programm soll nach c99 standart kompiliert werden, das Programm soll pedantisch an Standart halten.

.....

Aufgabe 3: (22 Punkte)

I- Definieren Sie folgende Makros: (6 Punktes)

a) FUNCTION1: a/b

b) FUNCTION2: $(x*y)\%5$

c) FUNCTION3: x^5+x^4+5

II- (16 Punktes)

a) Definieren Sie ein Makro: Set(x,n), welches das n-te Bit in der Zahl x setzt.

b) Definieren Sie ein Makro, neg(x), welches alle Bits in Zahl x negiert.

c) Definieren Sie ein Makro: del(x,n), welches das n-te Bit in der Zahl x löscht(delete).

d) Definieren Sie ein Makro: tog(x,n), welches das n-te Bit in der Zahl x toggelt(toggeln).

e) Angenommen, wir haben foldende Zahl in Hexadeximal: **x = 0x0F2E**,
*negieren Sie bitte alle Bits in Zahl x

...

*setzen Sie bitte das 5-te und 6-te Bit in der Zahl x.

...

*Löschen Sie bitte das 4-te und 10-te Bit in der Zahl x.

...

*Toggeln Sie bitte das 5 1-te und 2-te Bit in der Zahl x.

....

Aufgabe 4: (8 Punktes)

Gegeben Seien die 4 folgenden Header add.h, mul.h, sub.h, mod.h und die Datei main.c

add.h	mul.h
<pre>..... void add(int a, int b); </pre>	<pre>..... void mul(int a, int b); </pre>
main.c	sub.h
<pre>#include <stdio.h> #include <stdlib.h> int main(int argc, char *argv[]){ add(6,9); mul(10,7); sub(20, 2); mod(10, 5); }</pre>	<pre>..... void sub(int a, int b); </pre>
mod.h	Frage: Verhindern Sie bitte die Mehrfachinclusion!!
<pre>..... void mod(int a, int b); </pre>	

Aufgabe 5: (25 Punktes)

Wir haben folgende Berechnungen:

- a) $x = 40 * x + 2 * y + z$
- b) $y = x + y + z$
- c) $z = x - y + 6 * z$
- d) $x = 5 * y - z$
- e) $y = 6 * x + 2 * y + 19$

Initiale Belegung der SDA:

sda[2]	z	5
sda[1]	y	1
sda[0]	x	-2

1- Erstellen Sie für jede Berechnung ein Programm in Ninja-Assembler. (10 Punktes)

2- zeichnen Sie noch bitte für jeden Assembler-code den Stack-Inhalt mit der größten Ausdehnung, dabei geben Sie den sp(stackpointer), pc(programmcounter) und welche Werte nun in der SDA stehen. (15 Punktes)