

Universität Karlsruhe (TH)

Lehrstuhl für Programmierparadigmen

Sprachtechnologie und Compiler WS 2009/2010

Dozent: Prof. Dr.-Ing. G. Snelting

Übungsleiter: Sebastian Buchwald

<http://pp.info.uni-karlsruhe.de/>

snelting@ipd.info.uni-karlsruhe.de

sebastian.buchwald@kit.edu

Übungsblatt 9

Ausgabe: 17.12.2009

Besprechung: 13.1.2010

Aufgabe 1: Aufrufkonventionen

Beschreiben Sie die Aufrufkonventionen (Calling Conventions) von 2 der folgenden Prozessorarchitekturen auf einem Betriebssystem Ihrer Wahl:

1. IA-32
2. Alpha
3. PowerPC
4. AMD-64
5. ARM
6. MIPS

Suchen Sie hierzu im Internet nach den entsprechenden Dokumenten im Internet (z.B. auf den Herstellerseiten).

Geben Sie für jede der von Ihnen gewählten Plattformen an, wie die Argumente beim Aufruf folgender C-Funktion übergeben werden. Wie wird der Rückgabewert zurück gegeben?

```
struct test_struct {
    int x;
    int y;
};

int foobar(int i, char c, double d, int j, struct test_struct s) {
    return 42;
}
```

Aufgabe 2: Activation Records

Gegeben folgendes Programm:

```
int iterations ;

int fib(int x)
{
    int r;

    iterations++;
    if (x == 0) {
        r = 0;
    } else if (x == 1) {
        r = 1;
    } else {
        r = fib(x-1) + fib(x-2);
    }
    return r;
}
```

```

int main(void)
{
    int input;
    scanf("%d", &input);
    printf("Result: %d, needed %d iterations\n", fib(input), iterations );
    return 0;
}

```

Stellen Sie sich einen nicht optimierenden Übersetzer vor, der alle Variablen auf dem Stack ablegt. Außerdem verlangt die Aufrufkonvention dass alle Argumente auf dem Stack übergeben werden. Der Prozessor selbst legt beim betreten einer Funktionen die Rücksprungadresse auf den Stack.

Stellen Sie die Struktur des Stacks unmittelbar vor dem Aufruf der Funktion fib dar und nach dem 2. Rekursionsschritt der Funktion fib.

Aufgabe 3: Array-Adressierung

Geben Sie für ein 4-dimensionales Array $a[-10..10, -20..20, 30..42, 0..100]$ von 32-Bit Ganzzahlen den erzeugten Zwischencode (Bytecode oder Tripelcode) zur Adressberechnung gemäß Hornerschema an. Berechnen Sie zuvor die korrekte virtuelle Anfangsadresse $adr(a[0, 0, 0, 0])$. Sie dürfen annehmen, dass die l_μ Compilezeitkonstanten sind.