

Unterprogramme

Beschreiben Sie den Inhalt des Speichers und des Stack-Pointers (%esp) sowie Programmzählers (%eip) während jedes Schrittes der Programmausführung! Wieviel Platz muss für den Stack reserviert werden (i80x86)?

```
00000000: movl $100, %esp
00000001: call 00000010
00000002: call 00000030
00000003: ...

00000010: call 00000020
00000011: call 00000020
00000012: ret

00000020: call 00000030
00000021: ret

00000030: ret
```

Unterprogramme

Unterprogramme

Ein m88100-Prozessor verwendet eines (%r1) der 31 zur Verfügung stehenden General-Purpose-Register (%r1-%r31) zur Speicherung der Rückkehradresse bei Unterprogramm-Aufrufen. Statt einer „ret“-Anweisung wird ein „jmp %r1“ verwendet. (Ehemalige Klausuraufgabe.)

a) In manchen Unterprogrammen muss der Inhalt des Registers %r1 zwischengespeichert werden (meist auf dem Stack), in manchen jedoch auch nicht. Warum?

b) Welche Vorteile, welche Nachteile bietet diese Methode der Speicherung der Rückkehradresse?

Unterprogramme

Antwort zu a):

Antwort zu b):

Unterprogramme

Was kann passieren, wenn einem Unterprogramm versehentlich weniger Werte (auf dem Stack) übergeben werden, als das Unterprogramm eigentlich Parameter erwartet? Was passiert im Falle zuviel übergebener Daten? (Ehemalige Klausuraufgabe.)

Antwort:

Unterprogramm

Wie sieht die Funktion

```
int foo(int a, int b) { return bar(a+b, a-b); }
```

mit und ohne Frame-Pointer in Assembler aus? Wo liegen die Unterschiede?

Unterprogramm

Compilieren Sie:

```
long fac(long int x) {
    long res;

    if (x <= 1) {
        res = 1;
    } else {
        res = fac(x - 1) * x;
    }

    return res;
}
```

Verfolgen Sie den Stack-Inhalt und Stack-Pointer nach Aufruf von `fac` mit dem Wert 3.

Unterprogramm

Unterprogramm

Unterprogramm