



---

**Übungen zur Vorlesung  
“Database Systems and Modern CPU Architecture”**

Sommersemester 2009

Jan Rittinger ([jan.rittinger@uni-tuebingen.de](mailto:jan.rittinger@uni-tuebingen.de))

**2. Übungsblatt**

Ausgabe: 6. Mai 2009 · Besprechung: 13. Mai 2009

Bitte schicken Sie die Lösung der Aufgaben, die mit „**(Abgabe)**“ markiert sind (bis spätestens 15 Minuten vor der Übung) an [jan.rittinger@uni-tuebingen.de](mailto:jan.rittinger@uni-tuebingen.de). Die Abgabe dient als Einschätzung für die „aktive“ Teilnahme an den Übungen und als bessere Diskussionsgrundlage für die Übungsbesprechung.

**Aufgabe 1: Instruction Set Architectures**

Die verschiedenen Instruction Set-Architekturen von Vorlesungsfolie 11 (Stack, Accumulator, Register (register-memory), Register (load-store)) benötigen verschiedene Assembler-Sequenzen, um gleiche Programme zu implementieren.

**Schreiben Sie für die vier Instruction Set-Architekturen die Code-Sequenzen, die die Auswertung des Ausdrucks  $(A * B) - (B * C) - (A * D)$  (Folie 12) implementieren.**

Hinweis: Nehmen Sie an, dass es analog zu `Add` die Anweisungen `Mul` für Multiplikation und `Sub` für Subtraktion gibt.

**Aufgabe 2: Byte Ordering**

**(Abgabe)**

Wie in der Vorlesung (Folie 15) angesprochen, verwenden verschiedene Rechnerarchitekturen unterschiedliche Speicher-Layouts um Multi-Byte-Objekte im Hauptspeicher darzustellen. Man bezeichnet diese üblicherweise als „Little Endian“ und „Big Endian“.

**Schreiben Sie ein Programm, das die „Endianness“ Ihres Systems ermittelt, indem es „big“ bzw. „little“ ausgibt.**

**Aufgabe 3: Memory Alignment**

**(Abgabe)**

Aus Effizienzgründen verlangt die CPU-Speicheranbindung üblicherweise, dass auf den Speicher „aligned“ zugegriffen wird, d.h., ausschließlich auf *passende* Speicheradressen (Folie 16).

**Wie reagiert Ihr System auf Speicherzugriffe, die nicht „aligned“ sind? Schreiben Sie ein Programm, das diesen Effekt demonstriert, indem es die benötigte Zeit für (Sequenzen von) „aligned“ bzw. „unaligned“ Speicherzugriffe misst.**