

1. Praktikum

Gruppe 174: - Ulf Gebhardt (rbg: hu56nifa)
- Michael Scholz (rbg: mi48azih)

Aufgabe 2: Ägyptisches Multiplizieren:

1.

➡ Extra Programm (Datei: MUL.java)

2.

➡ Extra Programm (Datei: 2.2.asm). Alle angegebenen Werte wurden getestet.

Aufgabe 3: Variablentausch:

1.

Hauptspeicherzugriffe sind immer langsamer als Registerzugriffe. Aus diesem Grund sollte man von der Lösung mit Verwendung des Hauptspeichers absehen. Addiert man die Takte der drei movl-Befehle, so kommt man auf 28 Takte (Takte aus Übung 1). Die drei xor-Befehle brauchen zusammen nur 9 Takte (Quelle: <http://fredriks.de/8086Assembler/>).

2.

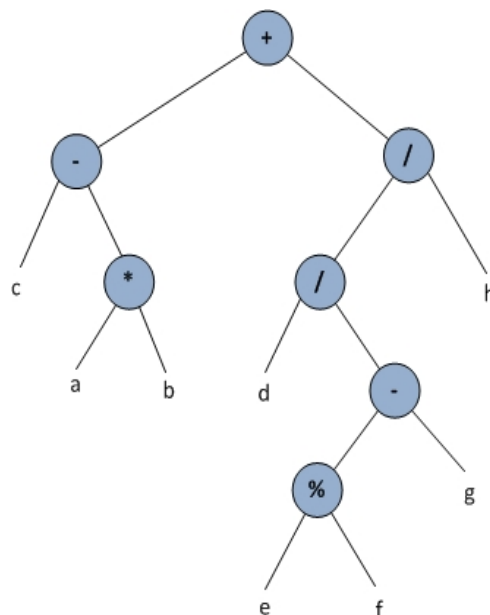
```
xor %eax, %ebx  
xor %ebx, %eax  
xor %eax, %ebx
```

3.

```
xchgl %eax, %ebx # tauscht Registerinhalte von %eax, %ebx
```

Aufgabe 4: Arithmetische Ausdrücke:

1. $(c - (a * b)) + ((d / ((e \% f) - g)) / h)$




2.

LR: $c \downarrow a b * \uparrow - \downarrow d \downarrow e f \% g - \uparrow / h / \uparrow +$

RL: $h \downarrow g \downarrow f e \% \uparrow - d / \uparrow / \downarrow b a * c - \uparrow +$

3.

Beide Traversierungsreihenfolgen benötigen sechs Zugriffe auf den Stack. Demnach ist keine der Beiden besser.

4.  Extra Programm (Datei: 4.asm)

5.

Falls bei einer Division der Divisor null ist, so erhalten wir ein Runtime-Exception "Integer division by zero (Windows)".