

## DS - Übung 04

### U 4.1)

Füllen Sie die untenstehende Tabelle aus und gehen Sie dabei genauso vor wie im Skript Kap. 6, S. 15/16.

Adress	Value	Register	Value
0x1000	0x10	%rax	0x1000
0x1004	0x11	%rcx	0x2
0x1008	0x12	%rdx	0x5
0x100C	0x13		

Operand	Wert (Wert //Operand-Typ und bei Memory-Typ auch Adresse)
%rax	
0x1004	
\$0x1004	
(%rax)	<b>0x10 //Memory (indirect) Adresse 0x1000</b>
4(%rax)	
3(%rax, %rdx)	
0x201(%rcx,%rdx)	<b>Wert an Adresse 0x208 //memory (indexed) Adresse 0x201 + 0x2 + 0x5 = 0x208</b>
0x20(,%rcx,4)	
(%rax,%rdx,8)	

### U 4.2)

Schreiben Sie ein Programm, das die folgende Rechnung durchführt

$$'A' - 0454 + 0x12345678$$

und das Ergebnis mittels **printf()** ausgibt.

- Berechnen Sie den Ausdruck von links nach rechts und wählen Sie dabei jeweils das **kleinstmögliche Register**.
- Vergrößern Sie die Bitanzahl nach Ermittlung des Ergebnisses mittels zero- bzw. sign-Extension auf den nächstgrößeren Datentyp.
- Geben Sie das Ergebnis mit **printf()** folgendermaßen aus:  
'A' - 0454 + 0x12345678 = ? (? = Ihr Ergebnis dezimal)

#### U 4.3)

Kompilieren Sie das folgende Programm mit **as**:

- a) Legen Sie den Wert 3 auf dem Stack ab.
  - Verwenden Sie dafür 8 Byte.
  - Verwenden Sie **sub** und **mov** (anstatt **push**).
- b) Addieren Sie 4 zu diesem Wert und geben Sie das Ergebnis mit dem SYS\_WRITE-Befehl aus.
  - Verwenden Sie auch hier wieder die Befehle aus dem Skript Kap. 6, S. 15/16.
- c) Geben Sie den Stack anschließend wieder frei
  - indem Sie den Stackzeiger unter Verwendung von **add** wieder auf seinen ursprünglichen Wert zurücksetzen.

#### Hinweis:

Beachten Sie, dass das Ergebnis eine Zahl ist, SYS\_WRITE aber nur Zeichen (Größe: Byte) ausgeben kann.

**Geben Sie das Ergebnis und <RETURN> getrennt aus.**