

# Übung 2 - Lösung

## 1. SASM IDE

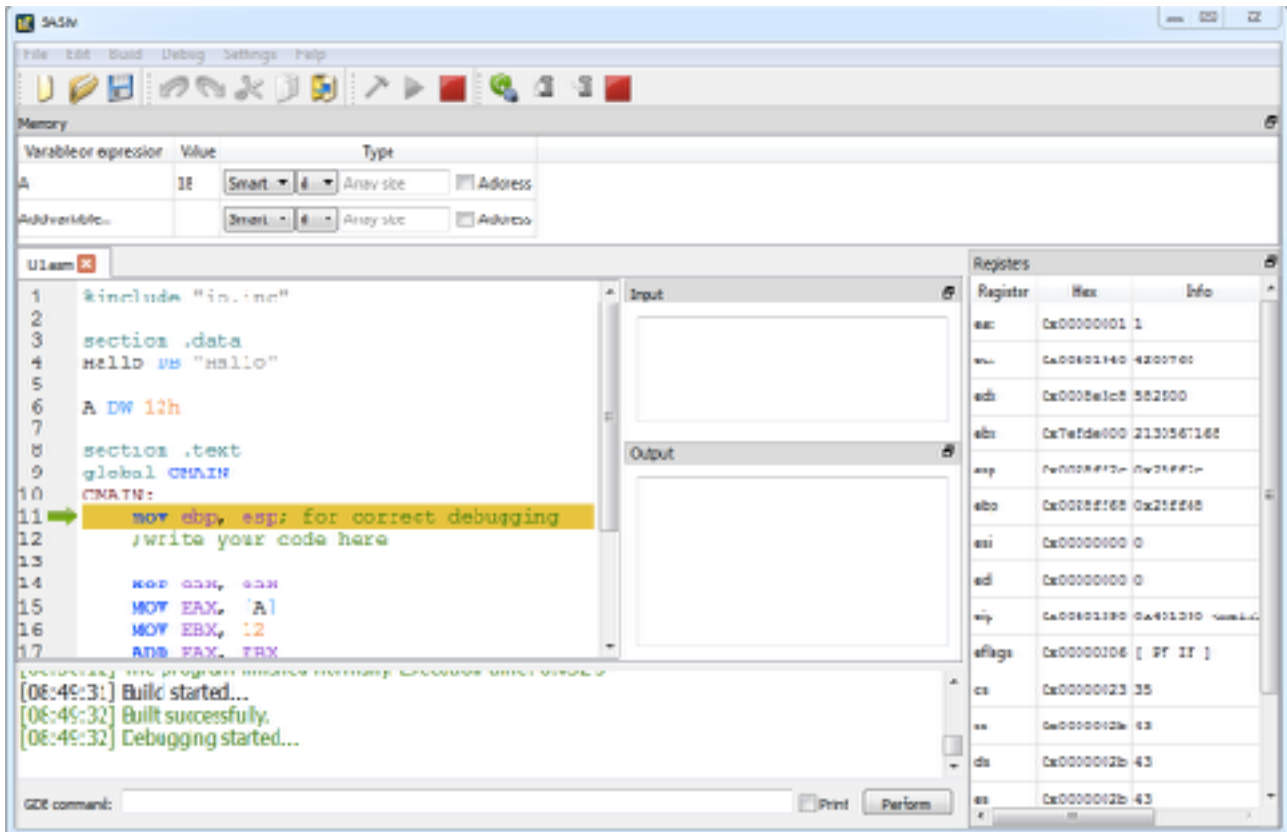
Laden Sie die SASM IDE von folgender Seite als .zip und entpacken Sie die Datei: <http://dman95.github.io/SASM/english.html>

Geben Sie folgenden Code in den Editor ein und beschreiben Sie anschließend die Codeabschnitte:

Code	Beschreibung
<code>%include "io.inc"</code>	Makro Datei wird integriert
<code>section .data</code> <code>Hallo DB "Hallo"</code>	Daten Sektion beginnt hier Variable Hallo mit String "Hallo"
<code>A DD 12h</code>	Variable A mit 12 als Hex Wert
<code>section .text</code> <code>global CMAIN</code> <code>CMAIN:</code> <code>    mov ebp, esp ; for correct debugging</code> <code>    ;write your code here</code>	Hier beginnt die Code Sektion in CMAIN steht der globale Code Abschnitt CMAIN beginnt hier Code von SASM fürs Debugging Kommentar
<code>    xor eax, eax</code> <code>    MOV EAX, [A]</code> <code>    MOV EBX, 12</code> <code>    ADD EAX, EBX</code>	Mittels XOR wird EAX auf 0 gesetzt Der Inhalt von A wird in EAX geschrieben 12 (Dezimal) wird in EBX geschrieben EBX wird zu EAX addiert und in EAX gespeichert
<code>    PRINT_STRING Hallo</code> <code>    PRINT_DEC 4, EAX</code>	Der String aus Hallo wird ausgegeben Die Zahl aus EAX wird ausgegeben (in Dezimal)
<code>    xor eax, eax</code> <code>    ret</code>	Mittels XOR wird EAX auf 0 gesetzt Ende Programm (ret als Rücksprung)

Gehen Sie den Code per Debug (F5) zeilenweise durch und machen Sie sich mit den Registern und deren Inhalt vertraut.

Um Variablen zu überwachen, geben Sie oben unter Memory (nur im Debug Modus sichtbar) den Namen ein:



## 2. Variablen und Ausgabe in HEX

Erweitern Sie das Programm so, dass eine dritte Zahl addiert wird, welche in einer Variable B eingegeben wird. Das Ergebnis der Rechnung soll zusätzlich in einer weiteren Variable ERG abgespeichert und in Hexadezimal ausgegeben werden.

## 3. MUL und DIV

Machen Sie sich mit den MUL und DIV Befehlen vertraut und analysieren Sie welche Register verwendet werden (siehe Vorlesung).

```
%include "io.inc"
```

```
section .data  
Hallo DB "Hallo"
```

```
A DD 12h  
B DD 8h
```

```
section .text  
global CMAIN  
CMAIN:  
    mov ebp, esp; for correct debugging  
    ;write your code here
```

```
    xor eax, eax  
    MOV EAX, [A]  
    MOV EBX, 12  
    ADD EAX, EBX  
    ADD EAX, [B]
```

```
    PRINT_STRING Hallo  
    PRINT_HEX 4, EAX
```

```
    xor eax, eax  
    ret
```