Übung 2 - Lösung

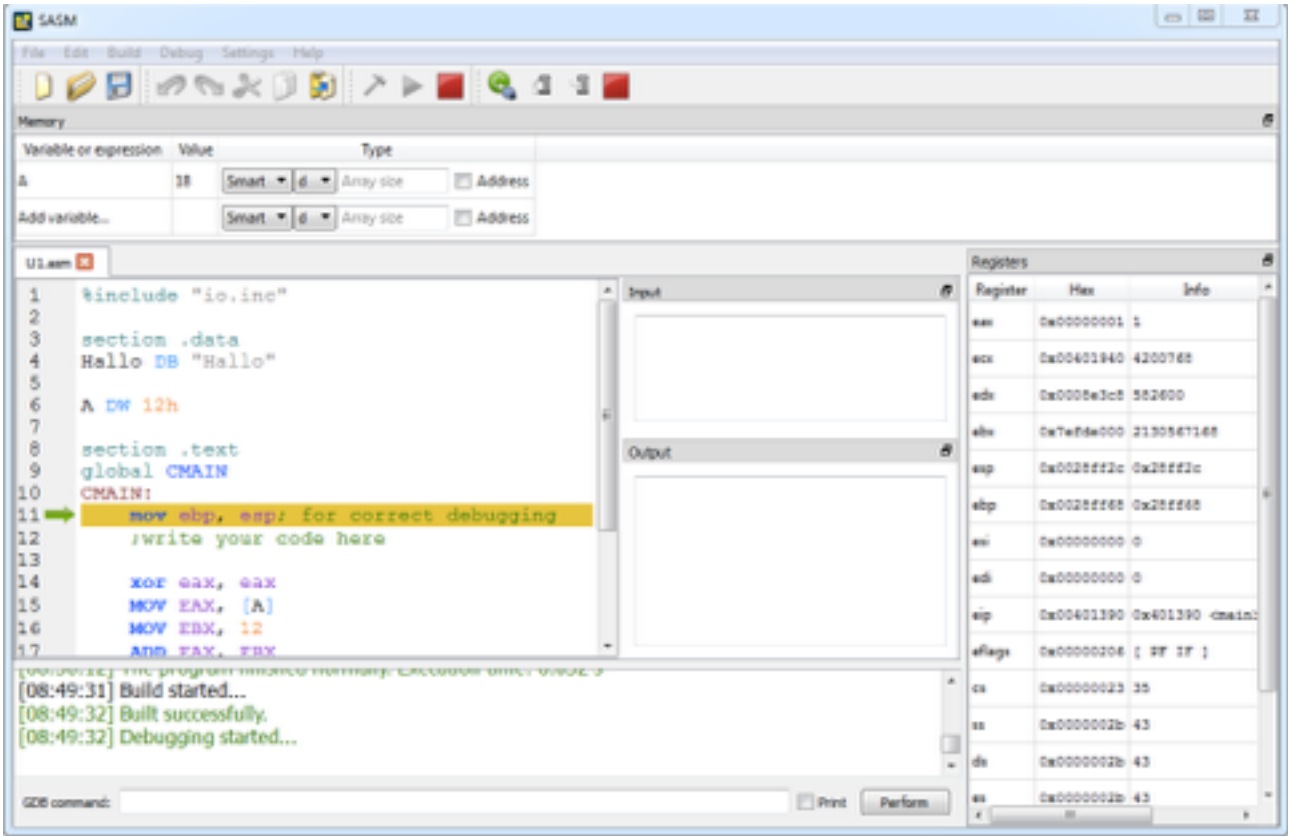
1. SASM IDE

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