

Assembly Language Lecture Series:

Introduction to SASM: Constant

Sensei RL Uy, College of Computer Studies,
De La Salle University, Manila, Philippines

Copyright Notice

This lecture contains copyrighted materials and is use solely for instructional purposes only, and not for redistribution.

Do not edit, alter, transform, republish or distribute the contents without obtaining express written permission from the author.

What is SASM?

SASM (SimpleASM) - simple Open Source cross-platform IDE for NASM assembly languages. SASM has syntax highlighting and debugger.

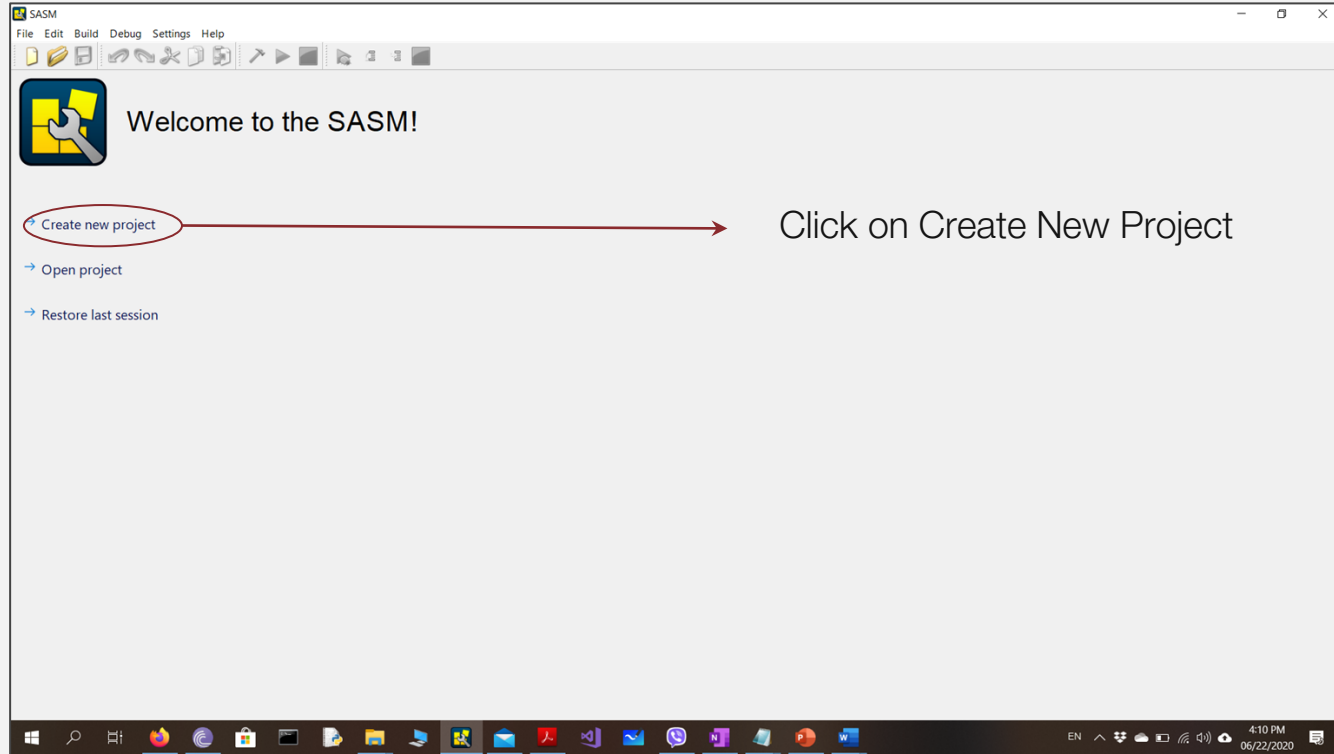
How to Download?

SASM software: <http://dman95.github.io/SASM/english.html>

Installing SASM on Windows:

- Click on the link “Download for Windows” and follow instructions

SASM Interface



Instruction: ○

Type the code below then type in 10 and 20 in the input window. Click the run icon. Your output should be display in the output window

Build

Run

The screenshot shows the SASM IDE interface. The main window contains assembly code for a program that prints 'Hello World'. The code is as follows:

```
1 %include "io64.inc"
2 section .data
3 var1 dq 0x0000_0000_0000_000A
4 var2 dq 0x0000_0000_0000_000B
5 section .text
6 global CMAIN
7 CMAIN:
8 ;write your code here
9 PRINT_STRING "Hello World"
10 NEWLINE
11 GET_DEC 8, rax
12 GET_DEC 8, rbx
```

Below the code is the build log:

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
```

On the right side, there are two windows: the 'Input' window and the 'Output' window. The 'Input' window contains the values 10 and 20. The 'Output' window contains the text 'Hello World', the number 30, and the number -1.

Labels with arrows point to the 'Build' and 'Run' buttons in the toolbar, the 'Main Window' (the code editor), the 'Input Window', the 'Output Window', and the 'Message Window' (the build log).

**Input
Window**

**Output
Window**

**Message
Window**

SASM Options

The screenshot displays the SASM (Simple Assembler) interface. The main window shows an assembly file named `w64_sasm.asm` with the following code:

```
1 %include "io64.inc"
2 section .data
3 var1 dq 0x0000_0000_0000_000A
4 var2 dq 0x0000_0000_0000_000B
5 section .text
6 global CMAIN
7 CMAIN:
8 ;write your code here
9 PRINT_STRING "Hello World"
10 NEWLINE
11 GET_DEC 8, rax
12 GET_DEC 8, rbx
```

A "Settings" dialog box is open, titled "SASM Options". It has three tabs: "Common", "Colors", and "Build". The "Build" tab is selected, showing the following configuration:

- Mode: ☒ x86 ☐ x64
- Assembler: ☒ NASM ☐ GAS ☐ FASM ☐ MASM
- Assembly options: `-f win64 $SOURCE$ -I $LSTOUTPUT$ -o $PROGRAM.OBJ$`
- Linking options: `$PROGRAM.OBJ$ $MACRO.OBJ$ -g -o $PROGRAM$ -m64`
- Assembler path: `C:/Program Files (x86)/SASM/NASM/nasm.exe`
- Linker path: `C:/Program Files (x86)/SASM/MinGW64/bin/gcc.exe`
- Object file name: `program.o`
- Build in current directory: ☐
- Disable linking: ☐

At the bottom of the dialog are buttons for "OK", "Cancel", and "Apply".

On the right side of the SASM window, there are two panels: "Input" and "Output". The "Input" panel shows the values `10` and `20`. The "Output" panel shows the text `Hello World`, the value `30`, and the value `-1`.

At the bottom of the SASM window, a "Build log" is visible:

```
Build log:
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Executing...
```

SASM

File Edit Build **Debug** Settings Help

Continue F5

Step over F10

Step into F11

Toggle breakpoint F8

✓ Show registers Ctrl+R

✓ Show memory Ctrl+M

Stop

Memory

Variable or expression

var1

var2

Add variable...

pe

size Address

size Address

size Address

w64_sasm.asm

5 section .text

6 global CMAIN

7 CMAIN:

8 ;write your code here

9 PRINT_STRING "Hello World"

10 NEWLINE

11 GET_DEC 8, rax

12 GET_DEC 8, rbx

13 add rax, rbx

Build log:

[22:47:30] Build started...

[22:47:30] Built successfully.

[22:47:32] The program is executing...

[22:47:32] The program finished normally. Execution time: 0.045 s

[22:51:39] Build started...

[22:51:39] Built successfully.

[22:51:39] Debugging started

GDB command:

Print Perform

Input

10

20

Output

Registers

Register	Hex
rax	0x7fff0b414d28 1407333822170
rbx	0x1 1
rcx	0x1 1
rdx	0x163f60 1458016
rsi	0x34 52
rdi	0x1622f0 1450736
rbp	0x8 0x8
rsp	0x61fe38 0x61fe38
r8	0x1627c0 1451968
r9	0x7fff0c796950 1407334026714
r10	0x0 0
r11	0x61fc58 6421592
r12	0x163f60 1458016
r13	0x0 0
r14	0x0 0
r15	0x0 0
rip	0x4014e0 0x4014e0 <mai
eflags	0x202 [IF]
cs	0x33 51
ss	0x2b 43
ds	0x0 0
es	0x0 0
fs	0x0 0
gs	0x0 0
st0	0 (raw 0x000000)
st1	0 (raw 0x000000)
st2	0 (raw 0x000000)
st3	0 (raw 0x000000)
st4	0 (raw 0x000000)
st5	0 (raw 0x000000)
st6	0 (raw 0x000000)
st7	0 (raw 0x000000)
fctrl	0x37f 895
fstat	0x0 0
ftag	0x0 0
fiseg	0x0 0
fioff	0x0 0

Debug

SASM

File Edit Build Debug Settings Help

Memory

Variable or expression	Value	Type	Address
var1	0xa	Hex · d · Array size	<input type="checkbox"/> Address
var2	0xb	Hex · d · Array size	<input type="checkbox"/> Address
Add variable...	Smart · d · Array size	<input type="checkbox"/> Address	

Contents of Memory

w64_sasm.asm

```
1 %include "io64.inc"
2 section .data
3 var1 dq 0x0000_0000_0000_000A
4 var2 dq 0x0000_0000_0000_000B
5 section .text
6 global CMAIN
7 CMAIN:
8 ;write your code here
9 PRINT_STRING "Hello World"
```

Contents of Registers

Registers

Register	Hex	Value
rax	0x7fff0b414d28	1407333822170
rbx	0x1	1
rcx	0x1	1
rdx	0x163f60	1458016
rsi	0x34	52
rdi	0x1622f0	1450736
rbp	0x8	0x8
rsp	0x61fe38	0x61fe38
r8	0x1627c0	1451968
r9	0x7fff0c796950	1407334026714
r10	0x0	0
r11	0x61fc58	6421592
r12	0x163f60	1458016
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x4014e0	0x4014e0 <mai
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
st0	0	(raw 0x000000)
st1	0	(raw 0x000000)
st2	0	(raw 0x000000)
st3	0	(raw 0x000000)
st4	0	(raw 0x000000)
st5	0	(raw 0x000000)
st6	0	(raw 0x000000)
st7	0	(raw 0x000000)
fctrl	0x37f	895
fstat	0x0	0
ftag	0x0	0
fiseg	0x0	0
fioff	0x0	0

Build log:

```
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
[22:51:39] Build started...
[22:51:39] Built successfully.
[22:51:39] Debugging started
```

GDB command:

Debug

SASM

File Edit Build Debug Settings Help

Memory

Variable or expression	Value	Type	Address
var1	0xa	Hex · d · Array size	<input type="checkbox"/> Address
var2	0xb	Hex · d · Array size	<input type="checkbox"/> Address
Add variable...	Smart · d · Array size	<input type="checkbox"/> Address	

Contents of Memory

w64_sasm.asm

```
1 %include "io64.inc"
2 section .data
3 var1 dq 0x0000_0000_0000_000A
4 var2 dq 0x0000_0000_0000_000B
5 section .text
6 global CMAIN
7 CMAIN:
8 ;write your code here
9 PRINT_STRING "Hello World"
```

Contents of Registers

Registers

Register	Hex	Value
rax	0x7fff0b414d28	1407333822170
rbx	0x1	1
rcx	0x1	1
rdx	0x163f60	1458016
rsi	0x34	52
rdi	0x1622f0	1450736
rbp	0x8	0x8
rsp	0x61fe38	0x61fe38
r8	0x1627c0	1451968
r9	0x7fff0c796950	1407334026714
r10	0x0	0
r11	0x61fc58	6421592
r12	0x163f60	1458016
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x4014e0	0x4014e0 <main>
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
st0	0	(raw 0x000000)
st1	0	(raw 0x000000)
st2	0	(raw 0x000000)
st3	0	(raw 0x000000)
st4	0	(raw 0x000000)
st5	0	(raw 0x000000)
st6	0	(raw 0x000000)
st7	0	(raw 0x000000)
fctrl	0x37f	895
fstat	0x0	0
ftag	0x0	0
fiseg	0x0	0
fioff	0x0	0

Build log:

```
[22:47:30] Build started...
[22:47:30] Built successfully.
[22:47:32] The program is executing...
[22:47:32] The program finished normally. Execution time: 0.045 s
[22:51:39] Build started...
[22:51:39] Built successfully.
[22:51:39] Debugging started
```

GDB command:

SASM using constant I/O macro (output)

SASM output macro	Description	Example
PRINT_STRING <i>string</i>	Print null-terminated text <i>string</i>	PRINT_STRING "Hello"
PRINT_CHAR <i>ch</i>	Print symbol <i>ch</i>	PRINT_CHAR "A"
PRINT_DEC <i>size</i> , <i>data</i>	Print number <i>data</i> in signed decimal representation	PRINT_DEC 2, -5
PRINT_UDEC <i>size</i> , <i>data</i>	Print number <i>data</i> in unsigned decimal representation	PRINT_UDEC 2, 10000
PRINT_HEX <i>size</i> , <i>data</i>	Print number <i>data</i> in hexadecimal representation	PRINT_HEX 4, 0x9ABCDEFO

- **ch, data** - a constant, variable name, register or address
- **string** – string constant, variable name or address
- **size** – the size of data in bytes (1,2,4,8)
- **Note: The byte size of "1" is not working correctly in SASM. It is the same as the byte size "2."**
 - Check: PRINT_DEC 1, 32767 [It printed 32767. But what do you think should be the correct output?]

Constant Output using SASM macro

The screenshot displays the SASM (Standalone Assembler) application window. The main editor shows assembly code for a program named `w64_sasm1.asm`. The code includes a macro `%include "io64.inc"` and defines a `.text` section with a `CMAIN` label. The program prints "Hello World", the character 'A', the decimal value -5, the decimal value 32767, and the hexadecimal value 0x123456789ABCDEF0. The status bar at the bottom indicates the program was built successfully and executed normally, taking 0.067 seconds.

Assembly Code:

```
1 %include "io64.inc"
2 section .text
3 global CMAIN
4 CMAIN:
5 ;write your code here
6 PRINT_STRING "Hello World"
7 NEWLINE
8 PRINT_CHAR "A"
9 NEWLINE
10 PRINT_DEC 1, -5
11 NEWLINE
12 PRINT_UDEC 2, 32767
13 NEWLINE
14 PRINT_HEX 8, 0x123456789ABCDEF0
15 xor rax, rax
16 ret
```

Input Window:

Input Window

Output Window:

Output Window

Hello World
A
-5
32767
123456789abcde
f0

[23:09:16] Built successfully.
[23:09:16] The program is executing...
[23:09:17] The program finished normally. Execution time: 0.067 s