Assembly Language Lecture Series:

x86-64 Introduction to SASM: Register

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

Copyright Notice

SASM using register I/O macro (output)

PRINT_DEC (decimal output)

Syntax: PRINT_DEC size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:
- Print number **data** **in *signed*** decimal representation

Example:
PRINT_DEC 4, eax

SASM using register I/O macro (**output**)

PRINT_UDEC (decimal output)

Syntax: PRINT_UDEC size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:
- Print number **data** **in *unsigned*** decimal representation

Example:
PRINT_UDEC 4, eax

SASM using register I/O macro (output)

PRINT_HEX
(hexadecimal output)

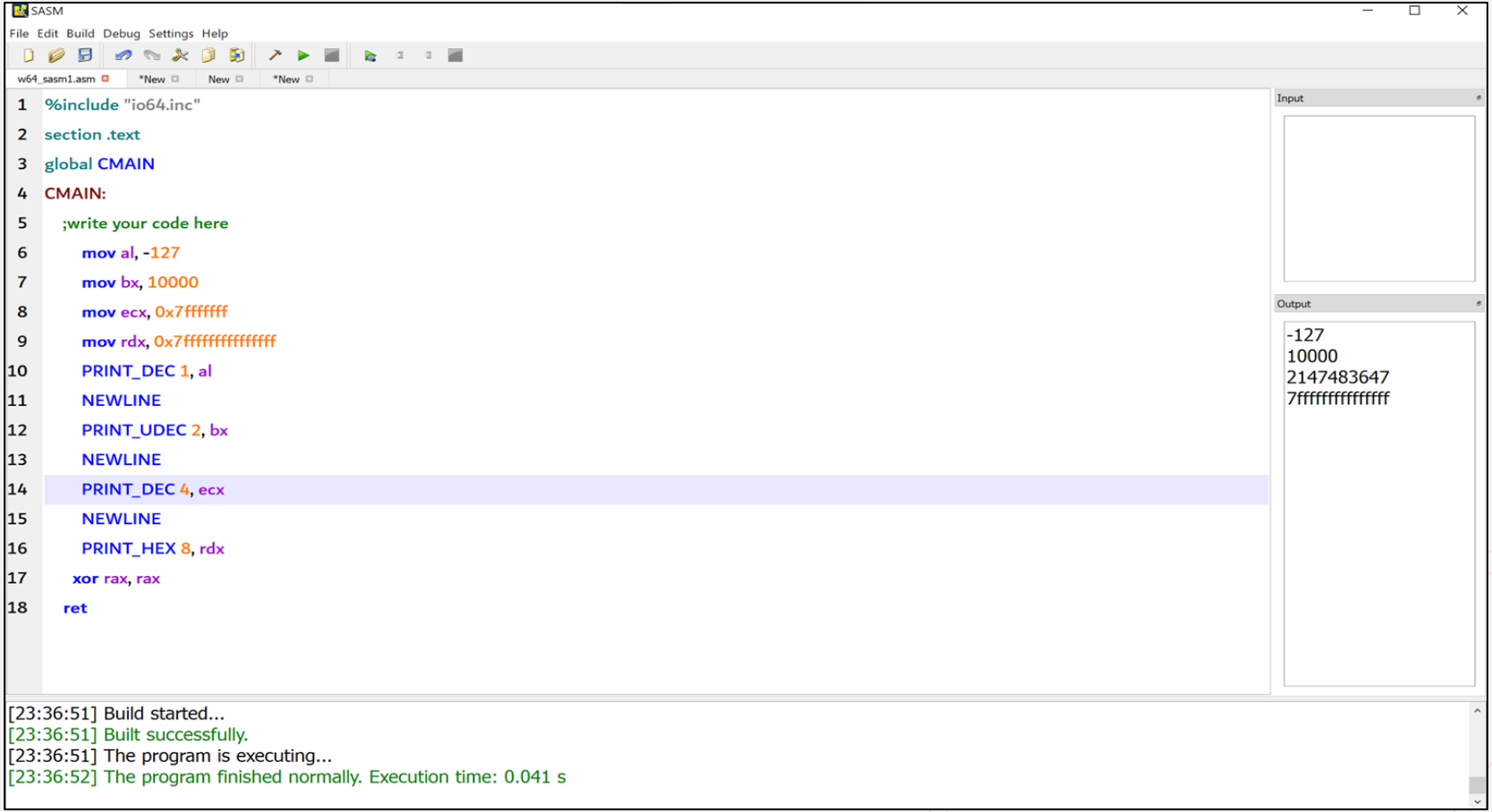
Syntax: PRINT_HEX size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:

- Print number **data** in hexadecimal representation

Example:
PRINT_HEX 4, eax


SASM
File Edit Build Debug Settings Help
w64_sasm1.asm
*New
New
*New
1 %include "io64.inc"
2 section .text
3 global CMAIN
4 CMAIN:
5 ;write your code here
6 mov al, -127
7 mov bx, 10000
8 mov ecx, 0x7fffffff
9 mov rdx, 0x7fffffffffffffff
10 PRINT_DEC 1, al
11 NEWLINE
12 PRINT_UDEC 2, bx
13 NEWLINE
14 PRINT_DEC 4, ecx
15 NEWLINE
16 PRINT_HEX 8, rdx
17 xor rax, rax
18 ret
Input
Output
-127
10000
2147483647
7fffffffffffffff
[23:36:51] Build started...
[23:36:51] Built successfully.
[23:36:51] The program is executing...
[23:36:52] The program finished normally. Execution time: 0.041 s

SASM using register I/O macro (input)

GET_DEC (decimal input)

Syntax: GET_DEC size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:

- Input number **data** **in *signed*** decimal representation from stdin (command line interface) or input window (SASM)

Example:
GET_DEC 4, eax

SASM using register I/O macro (input)

GET_UDEC (decimal input)

Syntax: GET_UDEC size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:

- Input number **data** **in *unsigned*** decimal representation from stdin (command line interface) or input window (SASM)

Example:
GET_UDEC 4, eax

SASM using register I/O macro (input)

GET_HEX (hexadecimal input)

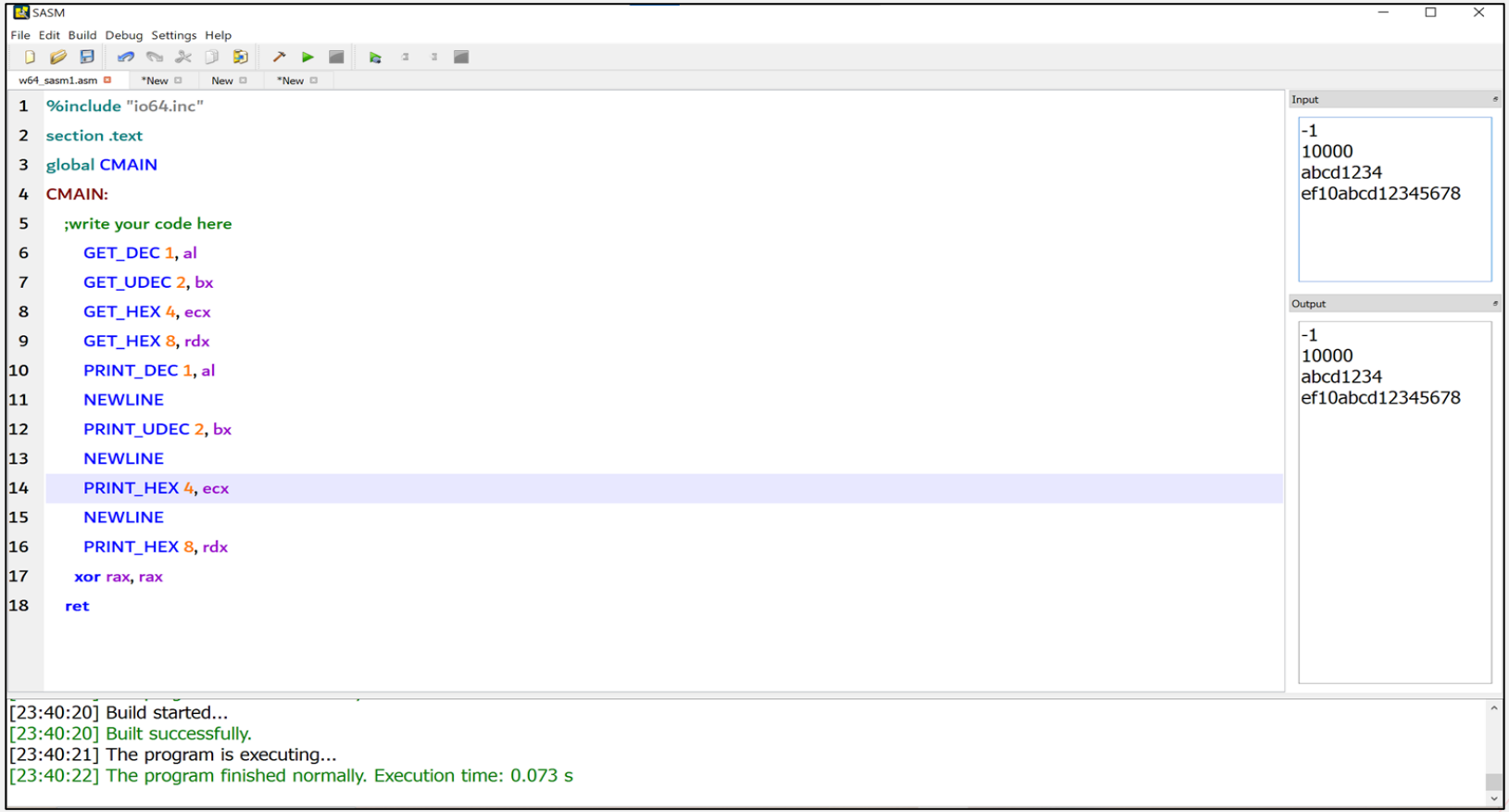
Syntax: GET_HEX size, data

size: size of **data** in bytes
data: number or symbol constant, name of variable, register or address

Note:

- Input number **data** in hexadecimal representation from stdin (command line interface) or input window (SASM)

Example:
GET_UDEC 4, eax


SASM
File Edit Build Debug Settings Help
w64_sasm1.asm
*New
New
*New
1 %include "io64.inc"
2 section .text
3 global CMAIN
4 CMAIN:
5 ;write your code here
6 GET_DEC 1, al
7 GET_UDEC 2, bx
8 GET_HEX 4, ecx
9 GET_HEX 8, rdx
10 PRINT_DEC 1, al
11 NEWLINE
12 PRINT_UDEC 2, bx
13 NEWLINE
14 PRINT_HEX 4, ecx
15 NEWLINE
16 PRINT_HEX 8, rdx
17 xor rax, rax
18 ret
Input
-1
10000
abcd1234
ef10abcd12345678
Output
-1
10000
abcd1234
ef10abcd12345678
[23:40:20] Build started...
[23:40:20] Built successfully.
[23:40:21] The program is executing...
[23:40:22] The program finished normally. Execution time: 0.073 s

SASM using register I/O macro (input)

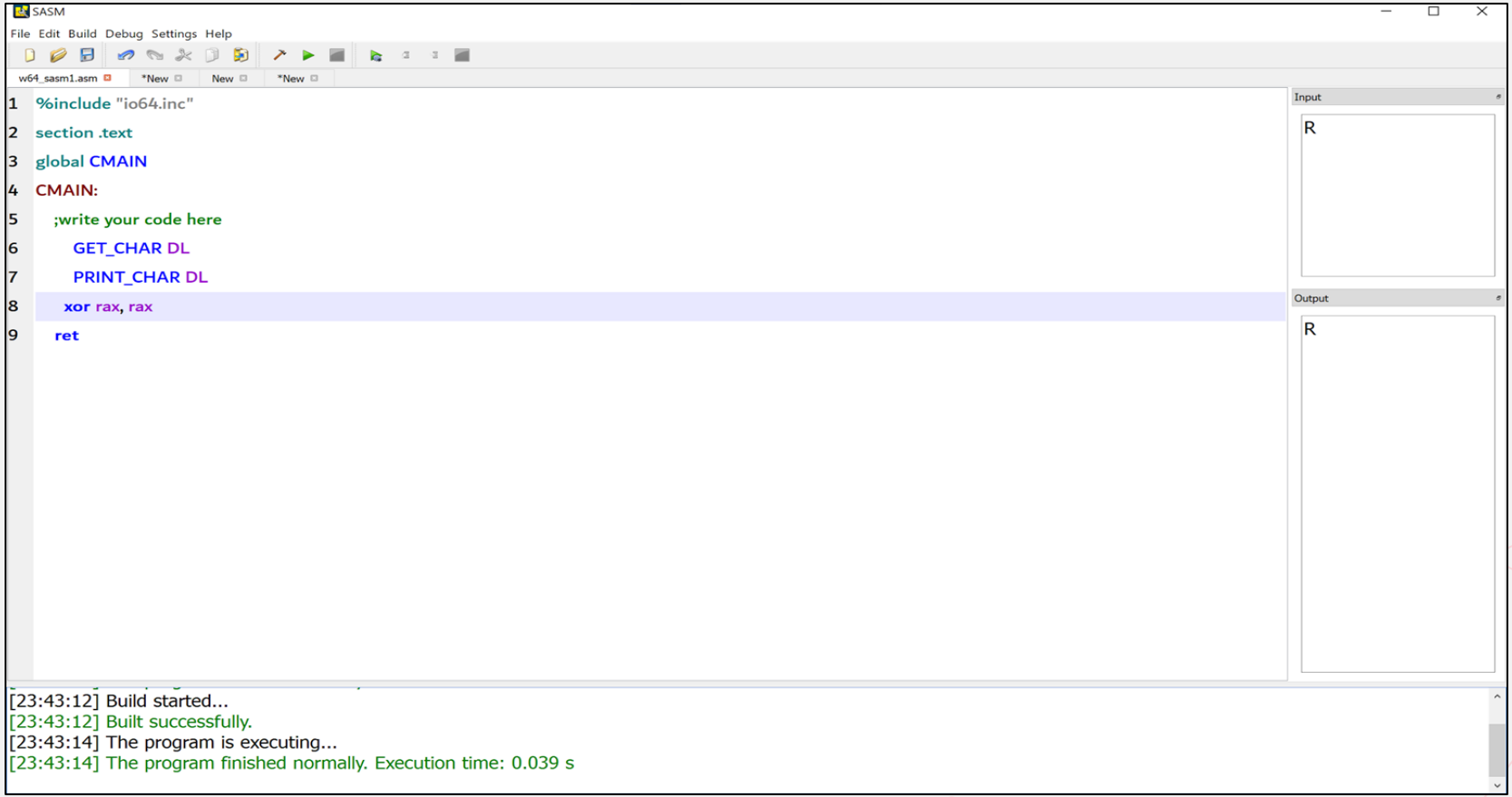
GET_CHAR (character input)

Syntax: GET_CHAR data

data: name of variable, register or address

Note:
- Input 1 character and store to data

Example:
GET_CHAR al


SASM
File Edit Build Debug Settings Help
w64_sasm1.asm
*New
New
*New
1 %include "io64.inc"
2 section .text
3 global CMAIN
4 CMAIN:
5 ;write your code here
6 GET_CHAR DL
7 PRINT_CHAR DL
8 xor rax, rax
9 ret
Input
R
Output
R
[23:43:12] Build started...
[23:43:12] Built successfully.
[23:43:14] The program is executing...
[23:43:14] The program finished normally. Execution time: 0.039 s