Assembly Language Lecture Series:

x86-64 Introduction to GDB

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

Copyright Notice

What is GDB?

01 Acronym:
GNU Debugger

02 Debugger for Assembly Language

03 Also supports C Language

04 See what’s *inside* your program

https://www.gnu.org/software/gdb/

Summary of GDB Command

Syntax	Description
Set disassembly-flavor intel	Intel format display
break main	Set breakpoint
Run (r)	Run program
disassemble/r main	List program; /r means opcode
display/i $pc	Display one line instruction after every execution
next instruction (ni)	Run next instruction

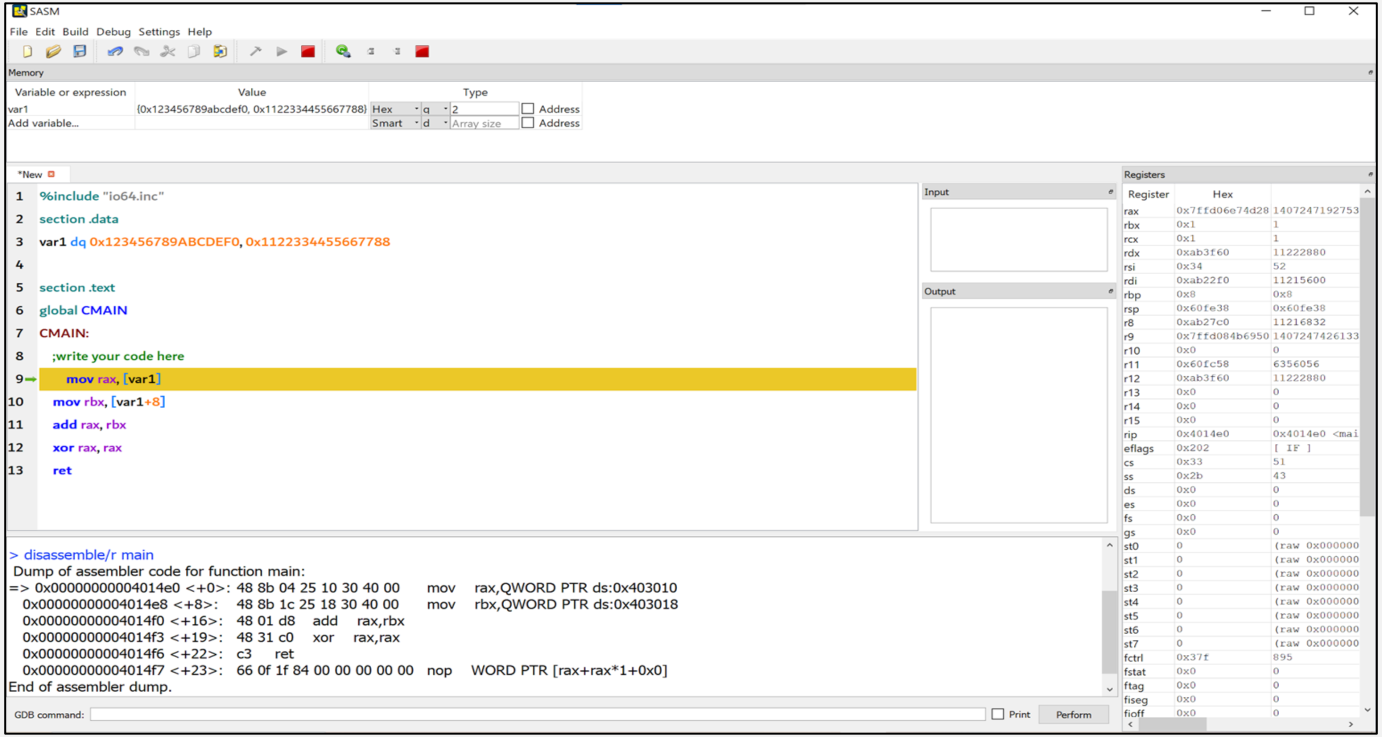
Summary of GDB Command

Syntax	Description
display/fmt $reg Display/nfu <mem_address> Example: display/x $esi display /4fw 0x403030	Display reg or memory value after every execution (n- # of locations; f-format; u-unit) **fmt**: o(octal), x(hex), d(decimal), u(unsigned decimal), t(binary), f(float), a(address), i(instruction), c(char) and s(string) **unit**: b(byte), h(halfword), w(word), g(giant, 8 bytes)
undisplay #	Cancel display request
Print/fmt [$reg | $memvar]	Print value (but preferably register)

Summary of GDB Command

Syntax	**Description**
Info all	Show all register
Info address <var_name>	Obtain the address of **var_name**
x/nfu <mem_address>	Display address (n- # of locations; f-format; u-unit) unit b(byte), h(halfword), w(word), g(giant, 8 bytes)

GDB in SASM



Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

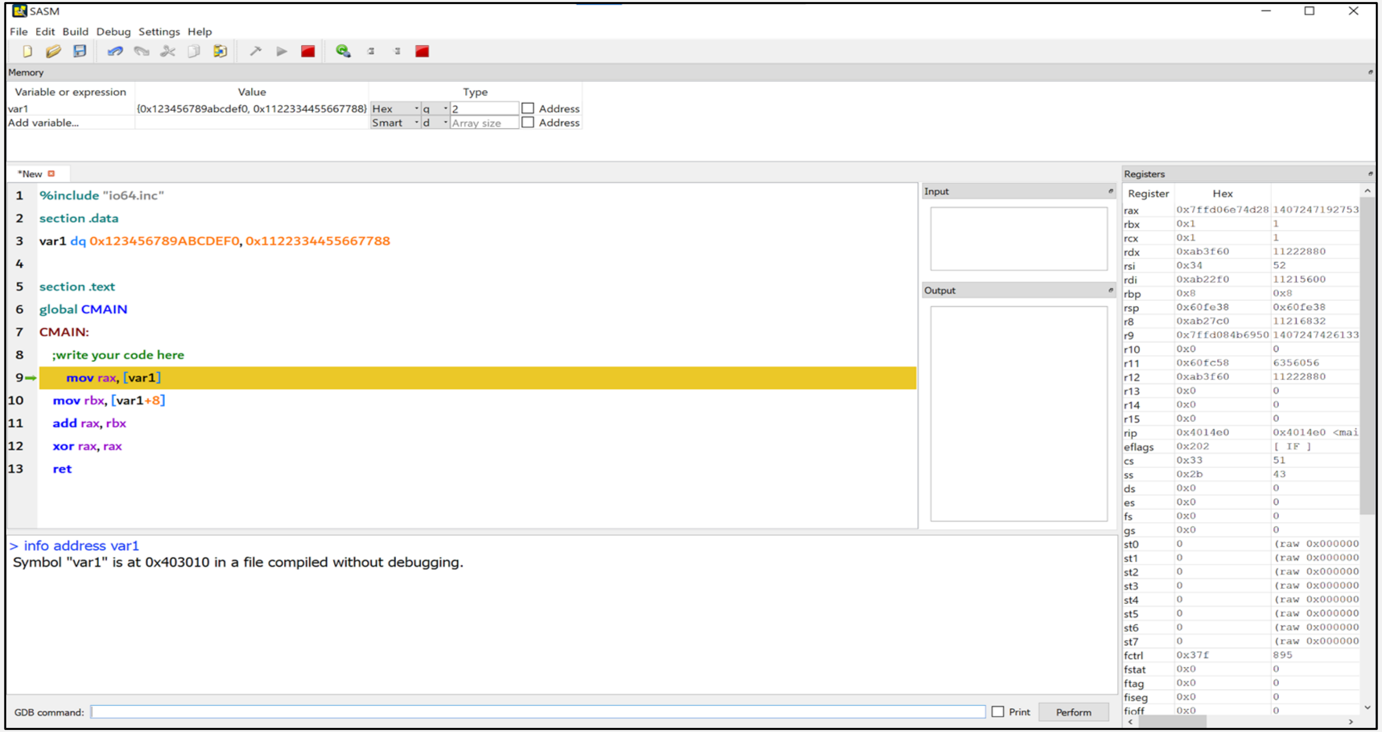
```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
    ;write your code here
    mov rax, [var1]
    mov rbx, [var1+8]
    add rax, rbx
    xor rax, rax
    ret
```

```
> disassemble/r main
 Dump of assembler code for function main:
=> 0x00000000004014e0 <+0>: 48 8b 04 25 10 30 40 00    mov    rax,QWORD PTR ds:0x403010
   0x00000000004014e8 <+8>:  48 8b 1c 25 18 30 40 00    mov    rbx,QWORD PTR ds:0x403018
   0x00000000004014f0 <+16>:  48 01 d8    add    rax,rbx
   0x00000000004014f3 <+19>:  48 31 c0    xor    rax,rax
   0x00000000004014f6 <+22>:  c3    ret
   0x00000000004014f7 <+23>:  66 0f 1f 84 00 00 00 00 00   nop    WORD PTR [rax+rax*1+0x0]
End of assembler dump.
```

GDB command:

Register	Hex	
rax	0x7ffd06e74d28	1407247192753
rbx	0x1	1
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	1407247426133
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
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

GDB in SASM



Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

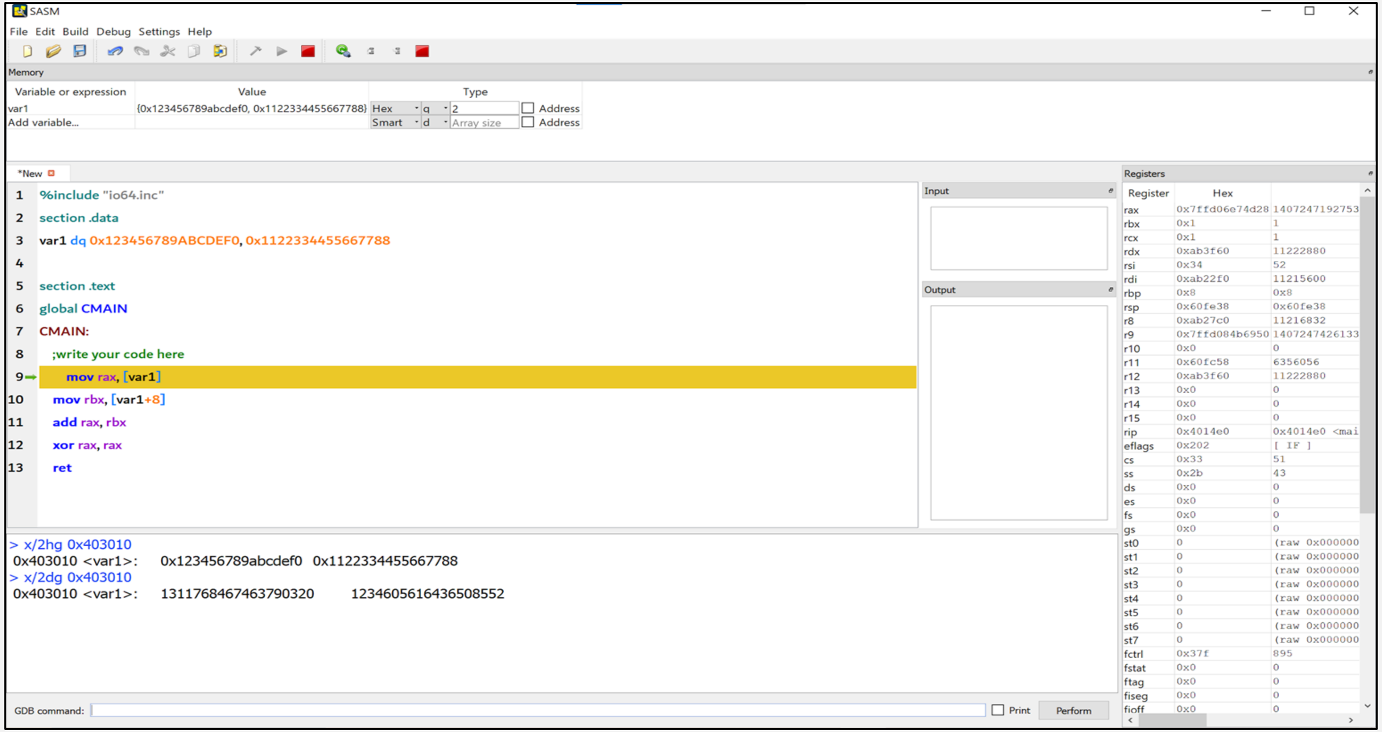
```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
    ;write your code here
     mov rax, [var1]
    mov rbx, [var1+8]
    add rax, rbx
    xor rax, rax
    ret
```

```
> info address var1
Symbol "var1" is at 0x403010 in a file compiled without debugging.
```

GDB command: □ Print

Register	Hex	
rax	0x7ffd06e74d28	1407247192753
rbx	0x1	1
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	1407247426133
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
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

GDB in SASM



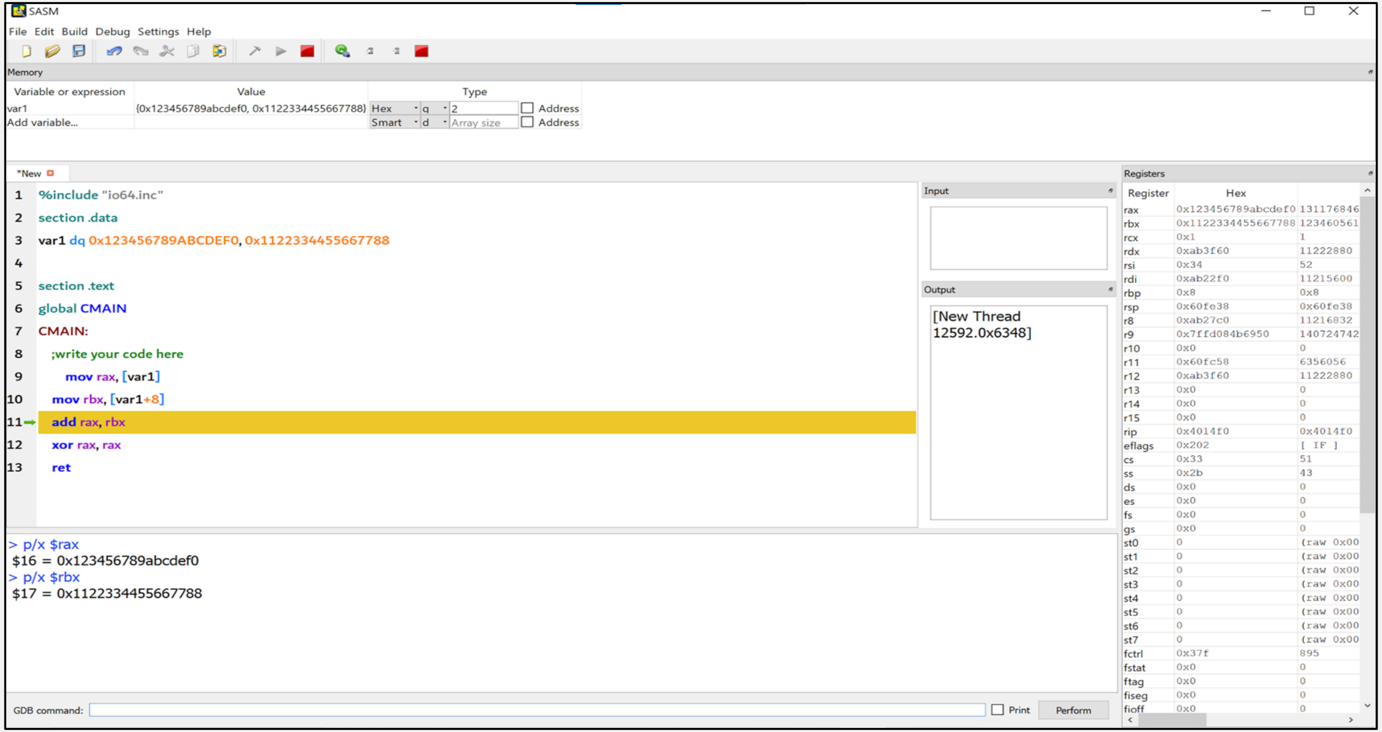
Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
 ;write your code here
    mov rax, [var1]
  mov rbx, [var1+8]
  add rax, rbx
  xor rax, rax
  ret
```

```
> x/2hg 0x403010
 0x403010 <var1>:    0x123456789abcdef0  0x1122334455667788
> x/2dg 0x403010
 0x403010 <var1>:    1311768467463790320      1234605616436508552
```

Register	Hex	
rax	0x7ffd06e74d28	1407247192753
rbx	0x1	1
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	1407247426133
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
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

GDB in SASM



Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
    ;write your code here
       mov rax, [var1]
    mov rbx, [var1+8]
    add rax, rbx
    xor rax, rax
    ret
```

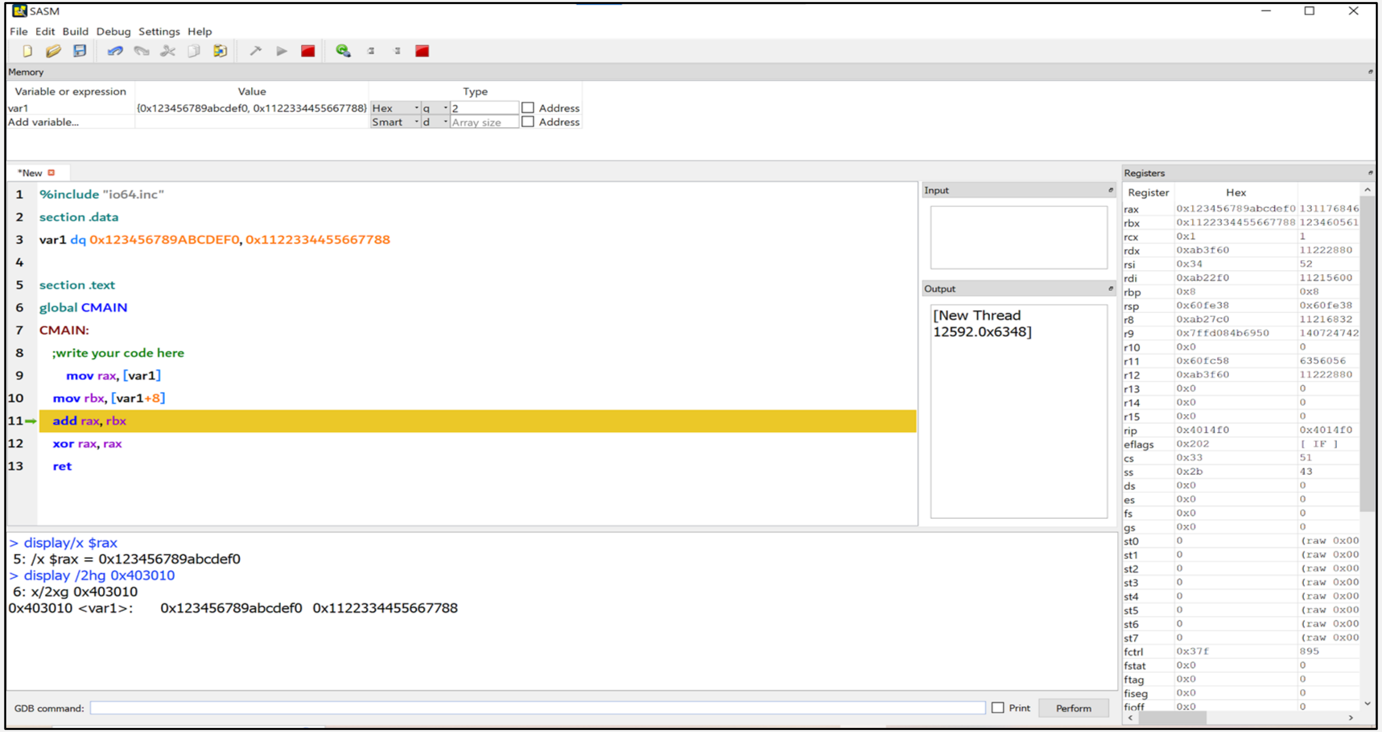
```
[New Thread
12592.0x6348]
```

```
> p/x $rax
 $16 = 0x123456789abcdef0
> p/x $rbx
 $17 = 0x1122334455667788
```

□ Print

Register	Hex	
rax	0x123456789abcdef0	131176846
rbx	0x1122334455667788	123460561
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	140724742
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x4014f0	0x4014f0
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
st0	0	(raw 0x00
st1	0	(raw 0x00
st2	0	(raw 0x00
st3	0	(raw 0x00
st4	0	(raw 0x00
st5	0	(raw 0x00
st6	0	(raw 0x00
st7	0	(raw 0x00
fctrl	0x37f	895
fstat	0x0	0
ftag	0x0	0
fiseg	0x0	0
fioff	0x0	0

GDB in SASM



Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

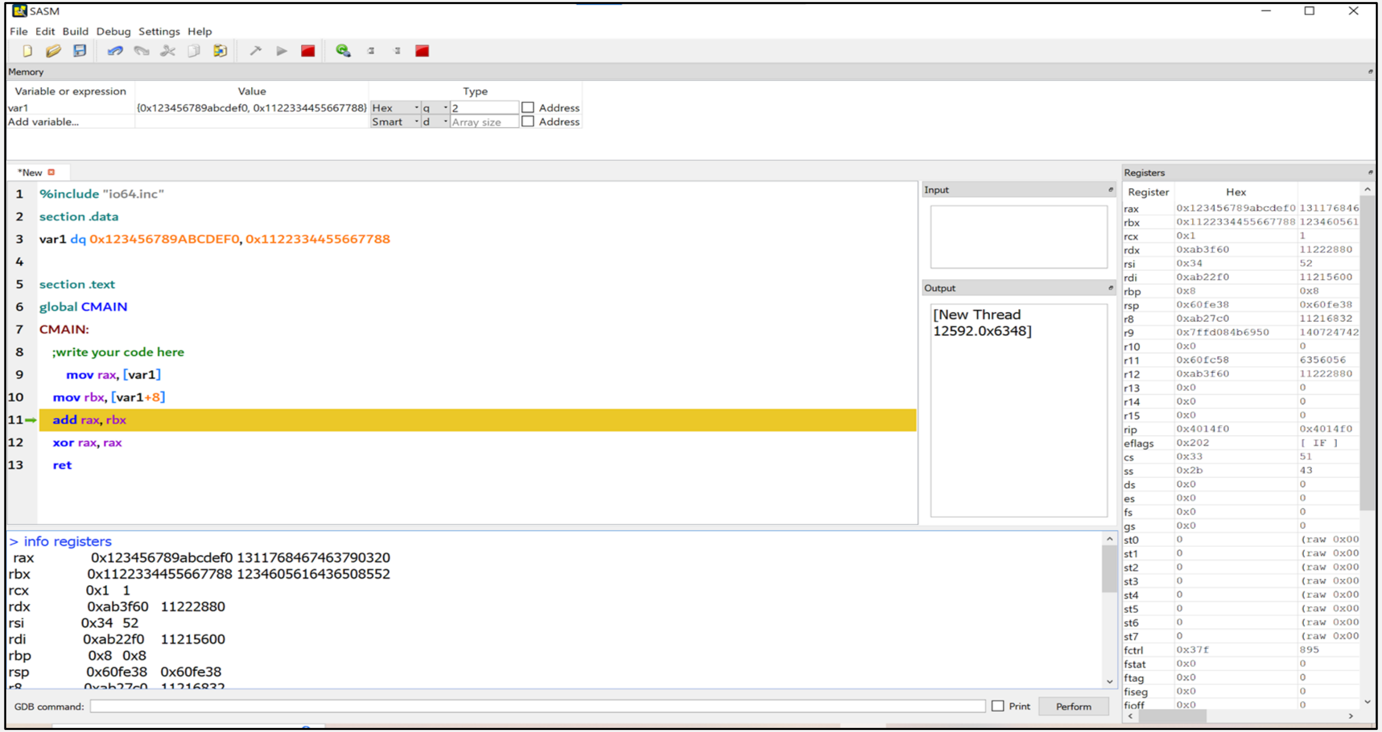
```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
    ;write your code here
      mov rax, [var1]
    mov rbx, [var1+8]
    add rax, rbx
    xor rax, rax
    ret
```

```
[New Thread
12592.0x6348]
```

```
> display/x $rax
 5: /x $rax = 0x123456789abcdef0
> display /2hg 0x403010
 6: x/2xg 0x403010
0x403010 <var1>:     0x123456789abcdef0  0x1122334455667788
```

Register	Hex	
rax	0x123456789abcdef0	131176846
rbx	0x1122334455667788	123460561
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	140724742
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x4014f0	0x4014f0
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
st0	0	(raw 0x00
st1	0	(raw 0x00
st2	0	(raw 0x00
st3	0	(raw 0x00
st4	0	(raw 0x00
st5	0	(raw 0x00
st6	0	(raw 0x00
st7	0	(raw 0x00
fctrl	0x37f	895
fstat	0x0	0
ftag	0x0	0
fiseg	0x0	0
fioff	0x0	0

GDB in SASM



Variable or expression	Value	Type			
var1	{0x123456789abcdef0, 0x1122334455667788}	Hex	q	2	□ Address
Add variable...		Smart	d	Array size	□ Address

```
%include "io64.inc"
section .data
var1 dq 0x123456789ABCDEF0, 0x1122334455667788

section .text
global CMAIN
CMAIN:
  ;write your code here
     mov rax, [var1]
  mov rbx, [var1+8]
  add rax, rbx
  xor rax, rax
  ret
```

```
[New Thread
12592.0x6348]
```

```
> info registers
 rax          0x123456789abcdef0 1311768467463790320
rbx           0x1122334455667788 1234605616436508552
rcx           0x1   1
rdx           0xab3f60   11222880
rsi          0x34  52
rdi          0xab22f0   11215600
rbp           0x8   0x8
rsp           0x60fe38   0x60fe38
r8           0xab27c0   11216832
```

□ Print

Register	Hex	
rax	0x123456789abcdef0	131176846
rbx	0x1122334455667788	123460561
rcx	0x1	1
rdx	0xab3f60	11222880
rsi	0x34	52
rdi	0xab22f0	11215600
rbp	0x8	0x8
rsp	0x60fe38	0x60fe38
r8	0xab27c0	11216832
r9	0x7ffd084b6950	140724742
r10	0x0	0
r11	0x60fc58	6356056
r12	0xab3f60	11222880
r13	0x0	0
r14	0x0	0
r15	0x0	0
rip	0x4014f0	0x4014f0
eflags	0x202	[IF]
cs	0x33	51
ss	0x2b	43
ds	0x0	0
es	0x0	0
fs	0x0	0
gs	0x0	0
st0	0	(raw 0x00
st1	0	(raw 0x00
st2	0	(raw 0x00
st3	0	(raw 0x00
st4	0	(raw 0x00
st5	0	(raw 0x00
st6	0	(raw 0x00
st7	0	(raw 0x00
fctrl	0x37f	895
fstat	0x0	0
ftag	0x0	0
fiseg	0x0	0
fioff	0x0	0