

Использовался онлайн-компилятор OnlineGDB.

1. immediate.s

The screenshot displays the OnlineGDB web interface. On the left is a sidebar with navigation links: IDE, My Projects, Classroom (new), Learn Programming, Programming Questions, Sign Up, and Login. The main area shows the assembly code for 'immediate.s'. The code includes directives like '.intel_syntax noprefix', '.text', and '.globl main', followed by a 'main' function that pushes the frame pointer, moves the stack pointer, moves several immediate values into registers (al, ax, eax, rax), and finally restores the stack pointer and returns.

```
1 # immediate.s
2 # Some instructions to illustrate machine code.
3 .intel_syntax noprefix
4 .text
5 .globl main
6 .type main, @function
7 main:
8     push    rbp           # save caller's frame pointer
9     mov     rbp, rsp      # establish our frame pointer
10
11     mov     al, 0xab      # 8-bit immediate
12     mov     ax, 0xabcd    # 16-bit immediate
13     mov     eax, 0xabcdef12 # 32-bit immediate
14     mov     rax, 0xabcdef12 # to 64-bit reg
15     mov     rax, 0xabcdef0123456789 # 64-bit immediate
16
17     mov     eax, 0        # return 0 to os
18     mov     rsp, rbp     # restore stack pointer
19     pop     rbp          # and frame pointer
20     ret                     # back to caller
21
```

Below the code is the GDB console, which shows the execution of the program. The console output is as follows:

```
(gdb) n
13     mov     eax, 0xabcdef12 # 32-bit immediate
(gdb) n
14     mov     rax, 0xabcdef12 # to 64-bit reg
(gdb) n
15     mov     rax, 0xabcdef0123456789 # 64-bit immediate
(gdb) n
17     mov     eax, 0        # return 0 to os
(gdb) n
18     mov     rsp, rbp     # restore stack pointer
(gdb)
```

On the right side of the interface, there are panels for 'Call Stack', 'Local Variables', and 'Registers'. The 'Registers' panel shows the current state of the CPU registers, including rax, rbx, rcx, rdx, rsi, rdi, rbp, rsp, r8, r9, r10, r11, r12, r13, r14, r15, rip, eflags, cs, ss, ds, es, fs, gs, k0, k1, k2, k3, k4, k5, k6, and k7.

| Register | Value |
|----------|---|
| rax | 0x0 0 |
| rbx | 0x555555555555160 93824992235872 |
| rcx | 0x555555555555160 93824992235872 |
| rdx | 0x7ffffffcd8 140737488350424 |
| rsi | 0x7ffffffcc8 140737488350408 |
| rdi | 0x1 1 |
| rbp | 0x7ffffffbd0 0x7ffffffbd0 |
| rsp | 0x7ffffffbd0 0x7ffffffbd0 |
| r8 | 0x0 0 |
| r9 | 0x7ffff7e0d60 140737354009952 |
| r10 | 0x7 7 |
| r11 | 0x2 2 |
| r12 | 0x555555555555040 93824992235584 |
| r13 | 0x7ffffffcc0 140737488350400 |
| r14 | 0x0 0 |
| r15 | 0x0 0 |
| rip | 0x555555555555151 0x555555555555151 <main+40> |
| eflags | 0x246 [PF ZF IF] |
| cs | 0x33 51 |
| ss | 0x2b 43 |
| ds | 0x0 0 |
| es | 0x0 0 |
| fs | 0x0 0 |
| gs | 0x0 0 |
| k0 | 0x0 0 |
| k1 | 0x0 0 |
| k2 | 0x0 0 |
| k3 | 0x0 0 |
| k4 | 0x0 0 |
| k5 | 0x0 0 |
| k6 | 0x0 0 |
| k7 | 0x0 0 |

2. register.s

OnlineGDB beta
online compiler and debugger for C/C++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Run Debug Stop Share Save Beautify

main.S

```
1 # register.s
2 # Some instructions to illustrate machine code.
3 .intel_syntax noprefix
4 .text
5 .globl main
6 .type main, @function
7
8 main:
9     push    rbp        # save caller's frame pointer
10    mov     rbp, rsp    # establish our frame pointer
11
12    mov     eax, ecx    # 32 bits, low reg codes
13    mov     edi, esi    # highest reg codes
14    mov     ax, cx      # 16 bits
15    mov     al, cl      # 8 bits
16    mov     eax, r8d    # 32 bits, 64-bit register
17    mov     rax, rcx    # 64 bits
18
19    mov     eax, 0      # return 0 to os
20    mov     rsp, rbp    # restore stack pointer
21    pop     rbp        # and frame pointer
22    ret
```

input

(gdb) n
13 mov ax, cx # 16 bits
(gdb) n
14 mov al, cl # 8 bits
(gdb) n
15 mov eax, r8d # 32 bits, 64-bit register
(gdb) n
16 mov rax, rcx # 64 bits
(gdb) n
17 mov eax, 0 # return 0 to os
(gdb)

Language Assembly

Call Stack

| # | Function | File:Line |
|---|----------|-----------|
| 0 | main | main.S:18 |

Local Variables

| Variable | Value |
|----------|-------|
|----------|-------|

Registers

| Register | Value |
|----------|---|
| rax | 0x55555555150 93824992235856 |
| rbx | 0x55555555150 93824992235856 |
| rcx | 0x55555555150 93824992235856 |
| rdx | 0x7ffffffce8 140737488350440 |
| rsi | 0x7ffffffcd8 140737488350424 |
| rdi | 0xffffcd8 4294962392 |
| rbp | 0x7ffffffbe0 0x7ffffffbe0 |
| rsp | 0x7ffffffbe0 0x7ffffffbe0 |
| r8 | 0x0 0 |
| r9 | 0x7ffff7fe0d60 140737354009952 |
| r10 | 0x7 7 |
| r11 | 0x2 2 |
| r12 | 0x555555555040 9382499223584 |
| r13 | 0x7ffffffcd0 140737488350416 |
| r14 | 0x0 0 |
| r15 | 0x0 0 |
| rip | 0x55555555513c 0x55555555513c <main+19> |
| eflags | 0x246 [PF ZF IF] |
| cs | 0x33 51 |
| ss | 0x2b 43 |
| ds | 0x0 0 |
| es | 0x0 0 |
| fs | 0x0 0 |
| gs | 0x0 0 |
| k0 | 0x0 0 |
| k1 | 0x0 0 |
| k2 | 0x0 0 |
| k3 | 0x0 0 |
| k4 | 0x0 0 |
| k5 | 0x0 0 |
| k6 | 0x0 0 |
| k7 | 0x0 0 |

Display Expressions

3. memory.s

OnlineGDB beta
online compiler and debugger for C/C++
code, compile, run, debug, share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Run Debug Stop Share Save Beautify

main.S

```
1 # memory.s
2 # Some instructions to illustrate machine code.
3 .intel_syntax noprefix
4 .text
5 .globl main
6 .type main, @function
7
8 main:
9     push    rbp        # save caller's frame pointer
10    mov     rbp, rsp    # establish our frame pointer
11    sub     rsp, 48     # local variables
12
13    mov     rcx, 5       # for indexing
14    mov     eax, [rbp]   # indirect
15    mov     eax, -48[rbp] # indirect + offset
16    mov     eax, -48[rbp+rcx] # indirect + offset and index
17    mov     eax, -48[rbp+4*rcx] # and scaled index
18
19    mov     eax, 0       # return 0 to os
20    mov     rsp, rbp    # restore stack pointer
21    pop     rbp        # and frame pointer
22    ret
```

input

start pause continue step over step into step out help

(gdb) n
12 mov rcx, 5 # for indexing
(gdb) n
13 mov eax, [rbp] # indirect
(gdb) n
14 mov eax, -48[rbp] # indirect + offset
(gdb) n
15 mov eax, -48[rbp+rcx] # indirect + offset and index
(gdb) n
16 mov eax, -48[rbp+4*rcx] # and scaled index
(gdb)

Language Assembly

Call Stack

| # | Function | File:Line |
|---|----------|-----------|
| 0 | main | main.S:16 |

Local Variables

| Variable | Value |
|----------|-------|
|----------|-------|

Registers

| Register | Value |
|----------|---|
| rax | 0x5000007f 1342177407 |
| rbx | 0x5555555555150 93824992235856 |
| rcx | 0x5 5 |
| rdx | 0x7fffffecc8 140737488350424 |
| rsi | 0x7fffffecc8 140737488350408 |
| rdi | 0x1 1 |
| rbp | 0x7fffffebd0 0x7fffffebd0 |
| rsp | 0x7fffffeba0 0x7fffffeba0 |
| r8 | 0x0 0 |
| r9 | 0x7fffffe0d60 140737354009952 |
| r10 | 0x7 7 |
| r11 | 0x2 2 |
| r12 | 0x5555555555040 93824992235584 |
| r13 | 0x7fffffecc0 140737488350400 |
| r14 | 0x0 0 |
| r15 | 0x0 0 |
| rip | 0x5555555555142 0x5555555555142 <main+25> |
| eflags | 0x206 [PF IF] |
| cs | 0x33 51 |
| ss | 0x2b 43 |
| ds | 0x0 0 |
| es | 0x0 0 |
| fs | 0x0 0 |
| gs | 0x0 0 |
| k0 | 0x0 0 |
| k1 | 0x0 0 |
| k2 | 0x0 0 |
| k3 | 0x0 0 |
| k4 | 0x0 0 |
| k5 | 0x0 0 |
| k6 | 0x0 0 |
| k7 | 0x0 0 |

Display Expressions

4. constToMemory.s

OnlineGDB beta
online compiler and debugger for C/C++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Run Debug Stop Share Save Beautify

main.S

```
1 # constToMemory.s
2 # Some instructions to illustrate machine code.
3 .intel_syntax noprefix
4 .text
5 .globl main
6 .type main, @function
7 main:
8     push    rbp           # save caller's frame pointer
9     mov     rbp, rsp      # establish our frame pointer
10    sub     rsp, 48        # local variables
11
12    mov     rcx, 5          # for indexing
13    mov     eax, [rbp]      # indirect
14    mov     dword ptr -48[rbp], 0x12000034 # indirect + offset
15    mov     dword ptr 48[rbp+rcx], 0x56000078 # indirect + offset and index
16    mov     dword ptr -48[rbp+4*rcx], 0x91000023 # and scaled index
17
18    mov     eax, 0          # return 0 to os
19    mov     rsp, rbp       # restore stack pointer
20    pop     rbp            # and frame pointer
21    ret
22
```

input

Starting program: /home/a.out

Breakpoint 1, main () at main.S:13
13 mov eax, [rbp] # indirect
(gdb) n
14 mov dword ptr -48[rbp], 0x12000034 # indirect + offset
(gdb) n
15 mov dword ptr 48[rbp+rcx], 0x56000078 # indirect + offset and index
(gdb) n
16 mov dword ptr -48[rbp+4*rcx], 0x91000023 # and scaled index
(gdb)

Language Assembly

Call Stack

| # | Function | File:Line |
|---|----------|-----------|
| 0 | main | main.S:16 |

Local Variables

| Variable | Value |
|----------|-------|
|----------|-------|

Registers

| Register | Value |
|----------|---|
| rax | 0x0 0 |
| rbx | 0x5555555555160 93824992235872 |
| rcx | 0x5 5 |
| rdx | 0x7ffffffcd8 140737488350424 |
| rsi | 0x7ffffffcc8 140737488350408 |
| rdi | 0x1 1 |
| rbp | 0x7ffffffebd0 0x7ffffffebd0 |
| rsp | 0x7ffffffeba0 0x7ffffffeba0 |
| r8 | 0x0 0 |
| r9 | 0x7ffff7fe0d60 140737354009952 |
| r10 | 0x7 7 |
| r11 | 0x2 2 |
| r12 | 0x5555555555040 93824992235584 |
| r13 | 0x7ffffffcc0 140737488350400 |
| r14 | 0x0 0 |
| r15 | 0x0 0 |
| rip | 0x555555555514a 0x555555555514a <main+33> |
| eflags | 0x206 [PF IF] |
| cs | 0x33 51 |
| ss | 0x2b 43 |
| ds | 0x0 0 |
| es | 0x0 0 |
| fs | 0x0 0 |
| gs | 0x0 0 |
| k0 | 0x0 0 |
| k1 | 0x0 0 |
| k2 | 0x0 0 |
| k3 | 0x0 0 |
| k4 | 0x0 0 |
| k5 | 0x0 0 |
| k6 | 0x0 0 |
| k7 | 0x0 0 |

Display Expressions

5. jumps.s

OnlineGDB beta
online compiler and debugger for C/C++
code. compile. run. debug. share.
IDE
My Projects
Classroom new
Learn Programming
Programming Questions
Sign Up
Login

About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy
© 2016 - 2022 GDB Online

Run Debug Stop Share Save Beautify

main.S

```
1 # jumps.s
2 # Some instructions to illustrate machine code.
3 .intel_syntax noprefix
4 .text
5 .globl main
6 .type main, @function
7 main:
8     push    rbp        # save caller's frame pointer
9     mov     rbp, rsp    # establish our frame pointer
10
11     xor     rax, rbx    # sets status flags
12     jne     forward    # test ZF
13
14 back:
15     mov     r8, r9      # stuff to jump over
16     mov     rbx, rcx
17
18 forward:
19     xor     rax, rbx    # sets status flags
20     je      back        # test ZF
21
22     mov     eax, 0      # return 0 to os
23     mov     rsp, rbp    # restore stack pointer
24     pop     rbp        # restore caller's frame pointer
25     ret             # back to caller
```

input

start pause continue step over step into step out help

(gdb) step
forward () at main.S:17
(gdb) n
17 xor rax, rbx # sets status flags
(gdb) n
18 je back # test ZF
(gdb) n
20 mov eax, 0 # return 0 to os
(gdb) n
21 mov rsp, rbp # restore stack pointer
(gdb) |

Language Assembly

Call Stack

| # | Function | File:Line |
|---|----------|-----------|
| 0 | forward | main.S:21 |

Local Variables

| Variable | Value |
|----------|-------|
|----------|-------|

Registers

| Register | Value |
|----------|--|
| rax | 0x0 |
| rbx | 0x55555555150 93824992235856 |
| rcx | 0x55555555150 93824992235856 |
| rdx | 0x7ffffffcd8 140737488350424 |
| rsi | 0x7ffffffcc8 140737488350408 |
| rdi | 0x1 |
| rbp | 0x7ffffffbd0 0x7ffffffbd0 |
| rsp | 0x7ffffffbd0 0x7ffffffbd0 |
| r8 | 0x0 |
| r9 | 0x7ffff7e0d60 140737354009952 |
| r10 | 0x7 |
| r11 | 0x2 |
| r12 | 0x555555555040 9382499223584 |
| r13 | 0x7ffffffcc0 140737488350400 |
| r14 | 0x0 |
| r15 | 0x0 |
| rip | 0x55555555142 0x55555555142 <forward+10> |
| eflags | 0x202 [IF] |
| cs | 0x33 51 |
| ss | 0x2b 43 |
| ds | 0x0 |
| es | 0x0 |
| fs | 0x0 |
| gs | 0x0 |
| k0 | 0x0 |
| k1 | 0x0 |
| k2 | 0x0 |
| k3 | 0x0 |
| k4 | 0x0 |
| k5 | 0x0 |
| k6 | 0x0 |
| k7 | 0x0 |

Display Expressions