

6 Stepping with GDB

How do I start riscv32-unknown-elf-gdb on my ELF, set a breakpoint at main, step, and inspect registers?

Response

- Step-by-Step GDB Debugging (Bare-Metal ELF)

1. Launch GDB on your ELF

```
riscv32-unknown-elf-gdb hello.elf
```

2. Connect to a simulator (e.g. GDB's internal sim)

Inside GDB, type:

```
(gdb) target sim
```

3. Set a Breakpoint at main

```
(gdb) break main
```

4. Run the Program

```
(gdb) run
```

5. Step Through Instructions

Step one C line (may expand into many instructions):

```
(gdb) step
```

Step one assembly instruction:

```
(gdb) stepi
```

6. Inspect Registers

To view general-purpose registers:

```
(gdb) info registers
```

To view a specific register (e.g., a0):

```
(gdb) print $a0
```

Extra Tips

Command	Description
layout asm	Show source + assembly (if supported)
disassemble main	Disassemble main() function
x/4xw \$sp	Examine 4 words at stack pointer
continue or c	Continue execution
quit or q	Exit GDB

```
yogesh@DESKTOP-PB6A9OR: ~ + | - X
t.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=riscv32-unknown-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from hello.elf...
(gdb) target sim
Connected to the simulator.
(gdb) load
Loading section .text, size 0x64 lma 10000
Loading section .rodata, size 0xf lma 10064
Start address 10000
Transfer rate: 920 bits in <1 sec.
(gdb) break main
Breakpoint 1 at 0x10010: file hello.c, line 2.
(gdb) run
Starting program: /home/yogesh/riscv_hello/hello.elf

Breakpoint 1, main () at hello.c:2
2          volatile char* uart = (char*)0x10000000;
(gdb) step
3          const char* msg = "Hello, RISC-V\n";
(gdb) stepi
0x0001001c      3          const char* msg = "Hello, RISC-V\n";
(gdb)
```

```
yogesh@DESKTOP-PB6A9OR: ~ + - X
(gdb) stepi
0x0001001c      3          const char* msg = "Hello, RISC-V\n";
(gdb) info registers
ra      0x0      0x0
sp      0x3fff2a0      0x3fff2a0
gp      0x0      0x0
tp      0x0      0x0
t0      0x0      0
t1      0x0      0
t2      0x0      0
fp      0x3fff2c0      0x3fff2c0
s1      0x0      0
a0      0x1      1
a1      0x0      0
a2      0x0      0
a3      0x0      0
a4      0x0      0
a5      0x10000  65536
a6      0x0      0
a7      0x0      0
s2      0x0      0
s3      0x0      0
s4      0x0      0
s5      0x0      0
s6      0x0      0
s7      0x0      0
s8      0x0      0
s9      0x0      0
s10     0x0      0
s11     0x0      0
t3      0x0      0
t4      0x0      0
t5      0x0      0
```