How to run our output.s

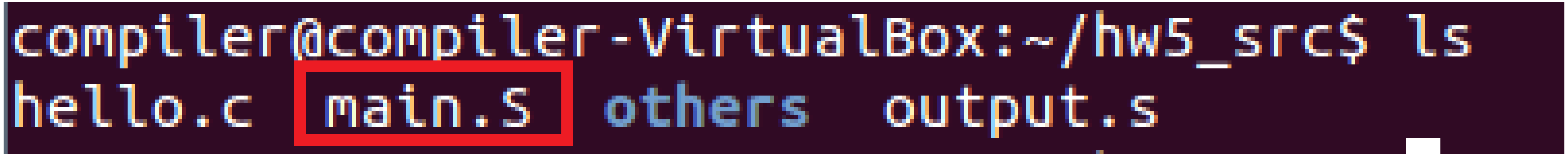
Tools

- arm-linux-gnueabihf-gcc

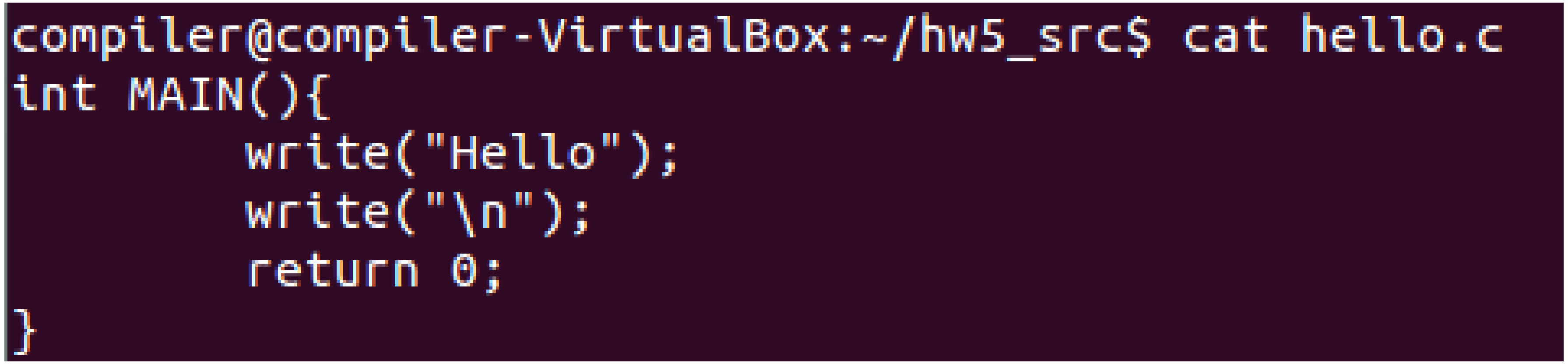
(Ubuntu/Linaro 4.8.2-16ubuntu4) 4.8.2

- arm-linux-gnueabihf-objdump

GNU objdump (GNU Binutils for Ubuntu) 2.24

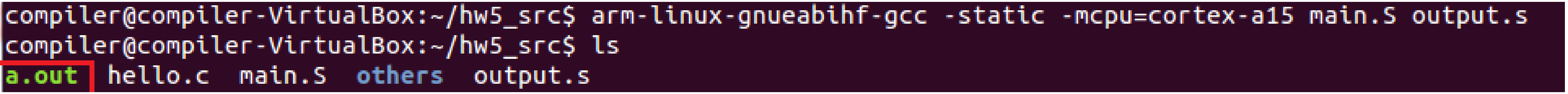


```
compiler@compiler-VirtualBox:~/hw5_src$ ls
hello.c  main.S  others  output.s
```

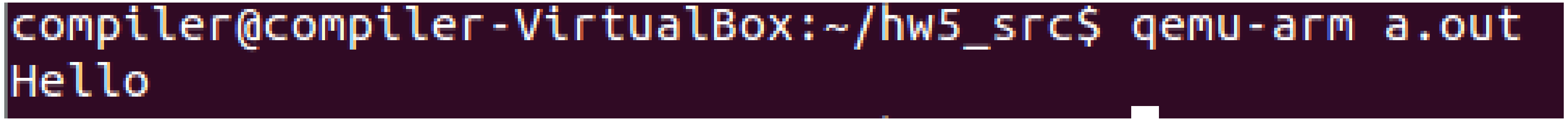


```
compiler@compiler-VirtualBox:~/hw5_src$ cat hello.c
int MAIN(){
        write("Hello");
        write("\n");
        return 0;
}
```

- Don’t modify main.S .
- hello.c is the input file(what we want to cpmpile).
- Run ./parser hello.c. The output.s is the output.



```
compiler@compiler-VirtualBox:~/hw5_src$ arm-linux-gnueabihf-gcc -static -mcpu=cortex-a15 main.S output.s
compiler@compiler-VirtualBox:~/hw5_src$ ls
a.out  hello.c  main.S  others  output.s
```



```
compiler@compiler-VirtualBox:~/hw5_src$ qemu-arm a.out
Hello
```

- Linking: arm-linux-gnueabihf-gcc –static –mcpu=cortex-a15 main.S output.s
- a.out is the arm executable. We run a.out through qemu-arm

You can also use run.sh

- We also provide a script: run.sh to do all these instructions.

Usage:

```
./run.sh [your parser] [ the C-- file]
```

The output is a a.out executable.

Ex:

```
./run.sh .parser ./patern/hello.c
```

How to debug?

Using GDB and QEMU

- We can run command: arm-linux-gnueabihf-objdump -d a.out >tmp.s to dump assembly to tmp.s
- Run: qemu-arm-static -g 1234 ./a.out
- create another terminal and type the following command
- gdb-multiarch

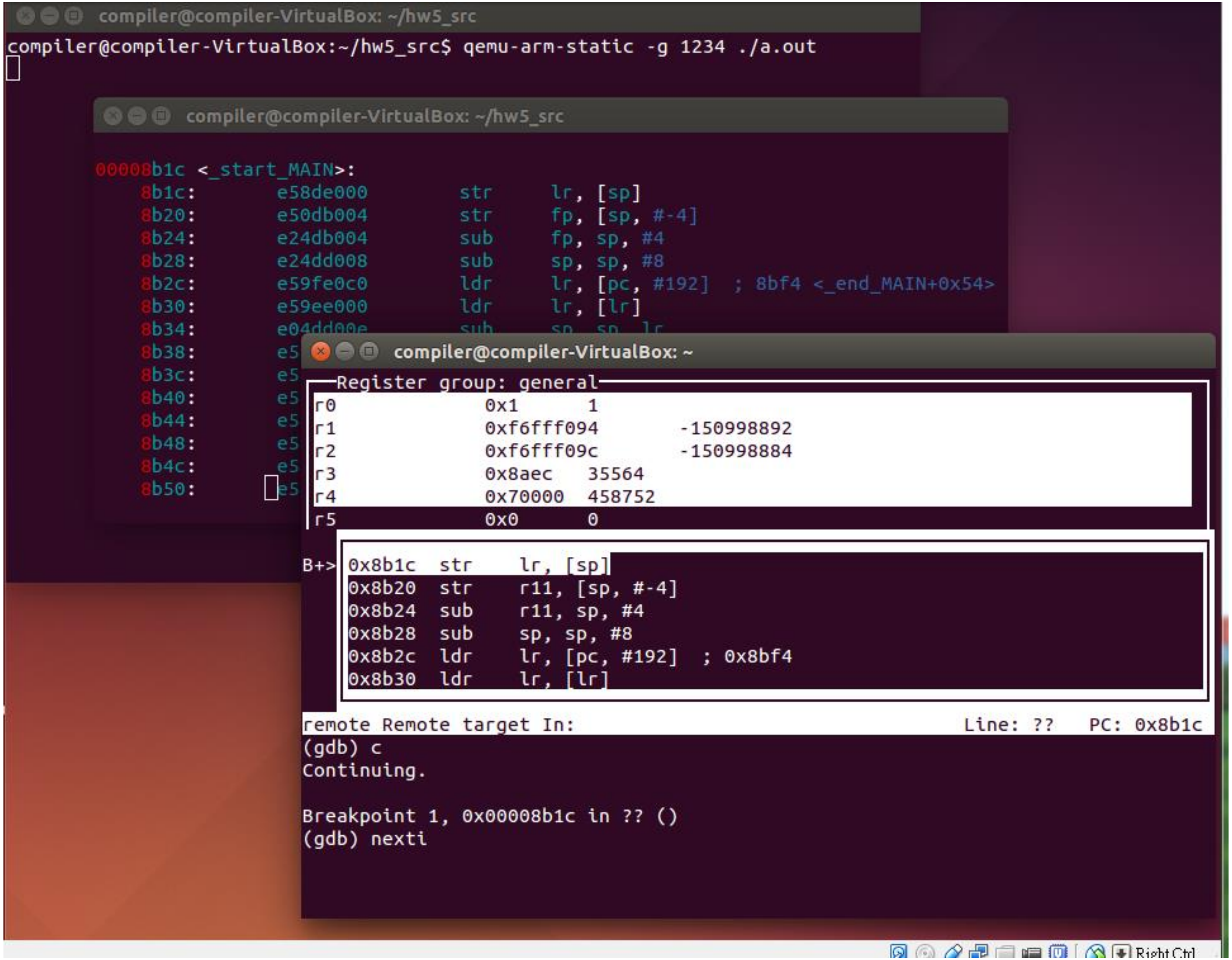
(enter gdb)

- set architecture arm
- target remote localhost:1234

Some useful command for gdb

- b *address
- layout asm
- layout regs
- c
- nexti

Can get more information by typing help or google gdb.



compiler@compiler-VirtualBox: ~/hw5_src

```
compiler@compiler-VirtualBox:~/hw5_src$ qemu-arm-static -g 1234 ./a.out
```

compiler@compiler-VirtualBox: ~/hw5_src

```
00008b1c <_start_MAIN>:
    8b1c:       e58de000        str     lr, [sp]
    8b20:       e50db004        str     fp, [sp, #-4]
    8b24:       e24db004        sub     fp, sp, #4
    8b28:       e24dd008        sub     sp, sp, #8
    8b2c:       e59fe0c0        ldr     lr, [pc, #192]  ; 8bf4 <_end_MAIN+0x54>
    8b30:       e59ee000        ldr     lr, [lr]
    8b34:       e04dd00e        sub     sp, sp, lr
    8b38:       e5
    8b3c:       e5
    8b40:       e5
    8b44:       e5
    8b48:       e5
    8b4c:       e5
    8b50:       e5
```

compiler@compiler-VirtualBox: ~

```
┌─Register group: general─
│r0             0x1      1
│r1             0xf6fff094       -150998892
│r2             0xf6fff09c       -150998884
│r3             0x8aec   35564
│r4             0x70000  458752
│r5             0x0      0

B+>│0x8b1c  str    lr, [sp]
   │0x8b20  str    r11, [sp, #-4]
   │0x8b24  sub    r11, sp, #4
   │0x8b28  sub    sp, sp, #8
   │0x8b2c  ldr    lr, [pc, #192]  ; 0x8bf4
   │0x8b30  ldr    lr, [lr]

remote Remote target In:                              Line: ??   PC: 0x8b1c
(gdb) c
Continuing.

Breakpoint 1, 0x00008b1c in ?? ()
(gdb) nexti
```