

# Open prepared VirtualBox image

- Please install VirtualBox first.



\*If the Internet is disable, please reconnect the Internet or reboot the OS.

# If internet is disable.

- Please reboot or reconnect.

How to run our output.s

# Tools

- aarch64-linux-gnu-gcc

(Ubuntu/Linaro 4.8.2-13ubuntu1) 4.8.2

- aarch64-linux-gnu-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 compile).
- Run ./parser hello.c. The output.s is the output.

```
compiler@compiler-VirtualBox:~/compiler_2015/hw5_answer$ aarch64-linux-gnu-gcc -static -O0 main.S  
compiler@compiler-VirtualBox:~/compiler_2015/hw5_answer$ qemu-aarch64-static ./a.out
```

- Linking: `aarch64-linux-gnu-gcc -static -O0 main.S`
- `a.out` is the ARM-v8 executable.

We run `a.out` through `qemu-aarch64-static`

# You can also use run.sh

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

Usage:

Put the run.sh and main.S in the same directory.

`./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: `aarch64-linux-gnu-objdump -d a.out >tmp.s` to dump assembly to tmp.s
- Run: `qemu-aarch64-static -g 1234 ./a.out`
- create another terminal and type the following command
- `gdb-multiarch`  
(enter gdb)
- set architecture aarch64
- 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: ~/compiler_2015/hw5_answer
compiler@compiler-VirtualBox:~/compiler_2015/hw5_answer$ qemu-aarch64-static -g
1234 ./a.out

```

```
compiler@compiler-VirtualBox: ~/compiler_2015/hw5_answer
0000000000400e38 <_start_MAIN>:
400e38: f90003fe      str     x30, [sp]
400e3c: f81f83fd      str     x29, [sp,#-8]
400e40: d10023fd      sub     x29, sp, #0x8
400e44: d10043ff      sub     sp, sp, #0x10
400e48: 5800199e      ldr     x30, 401178 <main+0x1c>
400e4c: f94003de      ldr     x30, [x30]
400e50: cb3e43ff      sub     sp, sp, w30, uxtw
400e54: f90007e9      str     x9, [sp,#8]
400e58: f9000bea      str     x10, [sp,#16]
400e5c:
400e60:
400e64:
400e68:
400e6c:
400e70:
400e74:
400e78:
400e7c:
400e80:
400e84:
400e88:

```

compiler@compiler-VirtualBox: ~

Register group: general

x0	0x1	1	
x1	0x40007ffed8	274886295256	
x2	0x40007ffee8	274886295272	
x3	0x40115c	4198748	
x4	0x0	0	
x5	0x4465a026a108bbbe	4928521454997781438	

B+> 0x400e38 str x30, [sp]  
0x400e3c str x29, [sp,#-8]  
0x400e40 sub x29, sp, #0x8  
0x400e44 sub sp, sp, #0x10  
0x400e48 ldr x30, 0x401178  
0x400e4c ldr x30, [x30]

remote Remote target In: Line: ?? PC: 0x400e38  
(gdb) c  
Continuing.  
  
Breakpoint 1, 0x0000000000400e38 in ?? ()  
(gdb)