

Cross 컴파일 및 gdb 실행

1. arm-linux-gnueabi-gcc <file.c name>

```
kiott01a@kiott01a-Z20NH-AS51B5U:~/final_project/IntegrationProject/asm/donghyeok/1$ arm-linux-gnueabi-gcc arm_func.c
```

크로스 컴파일

cf) file <실행파일 이름> : 파일 정보 확인

2. qemu-arm-static -g <접속 포트 숫자> -L /usr/arm-linux-gnueabi <실행파일 이름>

arm 실행 파일 실행(x86으로 명령어 전환해서 실행) -g

```
kiott01a@kiott01a-Z20NH-AS51B5U:~/final_project/IntegrationProject/asm/donghyeok/1$ qemu-arm-static -g 1234 -L /usr/arm-linux-gnueabi ./a.out
```

3. 새로운 창을 띄워서 gdb-multiarch 한 다음 file <실행파일 이름>

```
kiott01a@kiott01a-Z20NH-AS51B5U: ~/final_project/IntegrationProject/asm/donghyeok/
(gdb) s
Single stepping until exit from function main,
which has no line number information.
[Inferior 1 (Remote target) exited normally]
(gdb) q
kiott01a@kiott01a-Z20NH-AS51B5U:~/final_project/IntegrationProject/asm/donghyeok/1$ gdb-multiarch
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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".
(gdb) file a.out
Reading symbols from a.out...(no debugging symbols found)...done.
(gdb)
```

4. multiarch 서버 접속

```
(gdb) target remote localhost:1234
Remote debugging using localhost:1234
warning: remote target does not support file transfer, attempting to access file
s from local filesystem.
warning: Unable to find dynamic linker breakpoint function.
GDB will be unable to debug shared library initializers
and track explicitly loaded dynamic code.
0xf67ceb00 in ?? ()
```

5. main break 한 다음 c , disas해서 어셈블리어 확인

```
kiott01a@kiott01a-Z20NH-AS51B5U: ~/final_project/IntegrationProject/asm/donghyeok/
GDB will be unable to debug shared library initializers
and track explicitly loaded dynamic code.
0xf67ceb00 in ?? ()
(gdb) b main
Breakpoint 1 at 0x1046c
(gdb) c
Continuing.
warning: Could not load shared library symbols for 2 libraries, e.g. /lib/libc.s
o.6.
Use the "info sharedlibrary" command to see the complete listing.
Do you need "set solib-search-path" or "set sysroot"?

Breakpoint 1, 0x0001046c in main ()
(gdb) disas
Dump of assembler code for function main:
   0x00010460 <+0>:    push    {r11, lr}
   0x00010464 <+4>:    add     r11, sp, #4
   0x00010468 <+8>:    sub     sp, sp, #8
=> 0x0001046c <+12>:   mov     r3, #3
   0x00010470 <+16>:   str     r3, [r11, #-12]
   0x00010474 <+20>:   ldr     r0, [r11, #-12]
   0x00010478 <+24>:   bl      0x10438 <mult>
   0x0001047c <+28>:   str     r0, [r11, #-8]
   0x00010480 <+32>:   ldr     r1, [r11, #-8]
```