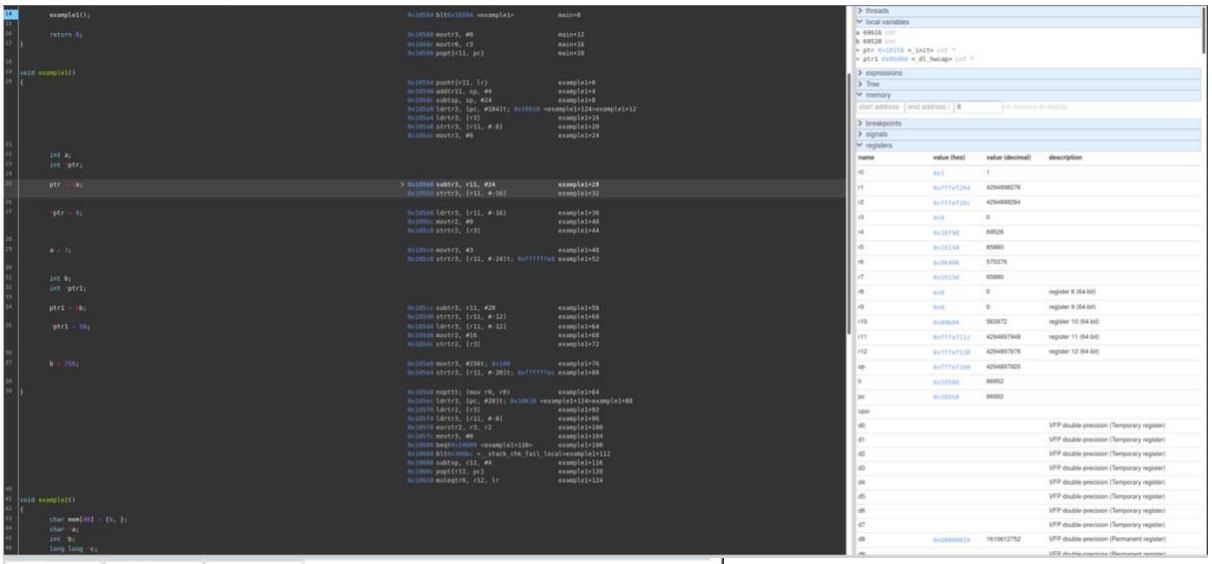
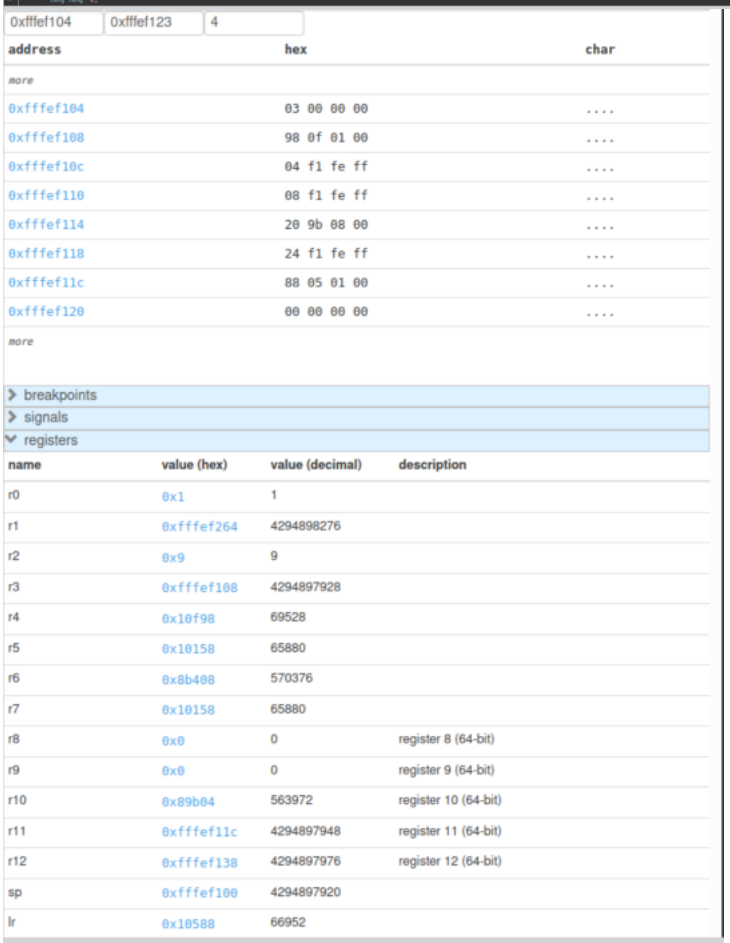
실습 a



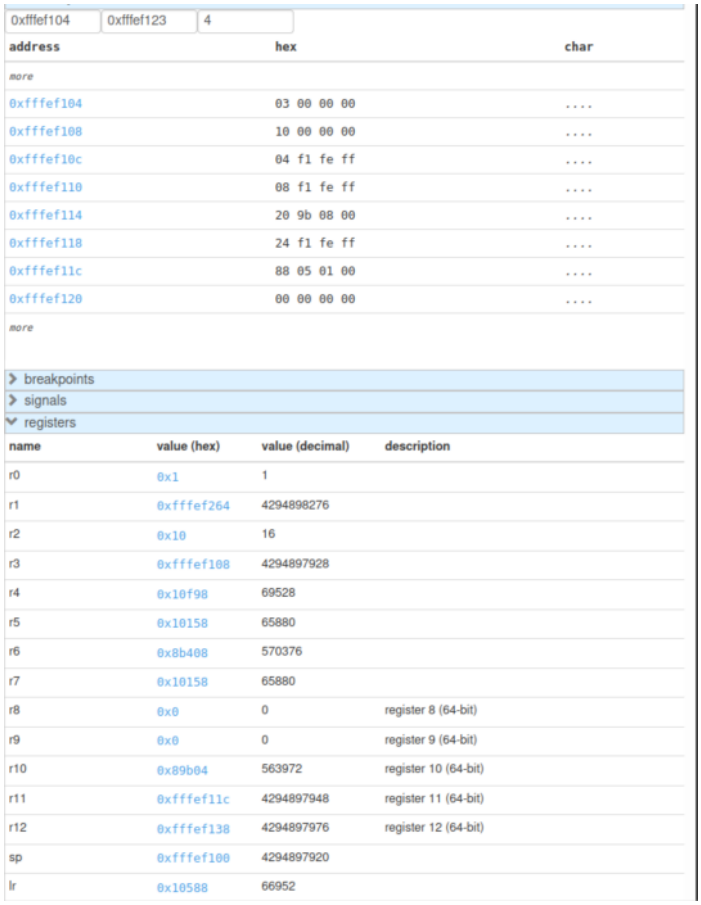


address	hex	char
more		
0xfffef104	03 00 00 00	....
0xfffef108	98 0f 01 00	....
0xfffef10c	04 f1 fe ff	....
0xfffef110	08 f1 fe ff	....
0xfffef114	20 9b 08 00	....
0xfffef118	24 f1 fe ff	....
0xfffef11c	88 05 01 00	....
0xfffef120	00 00 00 00	....
more		

breakpoints
signals
registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
r1	0xfffef264	4294898276	
r2	0x9	9	
r3	0xfffef108	4294897928	
r4	0x10f98	69528	
r5	0x10158	65880	
r6	0x8b408	570376	
r7	0x10158	65880	
r8	0x0	0	register 8 (64-bit)
r9	0x0	0	register 9 (64-bit)
r10	0x89b04	563972	register 10 (64-bit)
r11	0xfffef11c	4294897948	register 11 (64-bit)
r12	0xfffef138	4294897976	register 12 (64-bit)
sp	0xfffef100	4294897920	
lr	0x10588	66952	

B는 0xfffef108에 ptr1은 0xfffef110에 초기화 되어있다.



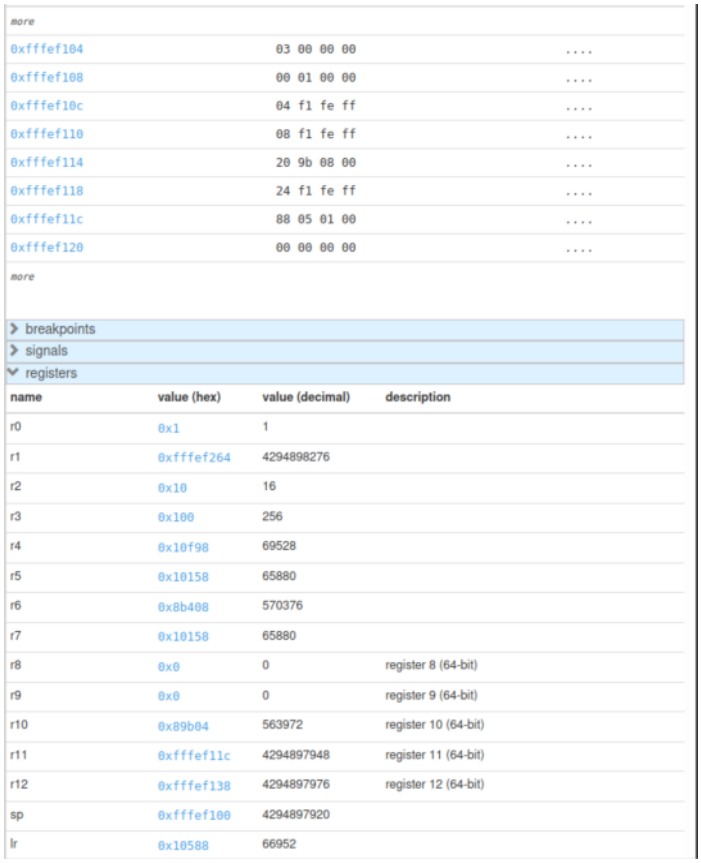
0xfffef104 | 0xfffef123 | 4

address	hex	char
more		
0xfffef104	03 00 00 00	....
0xfffef108	10 00 00 00	....
0xfffef10c	04 f1 fe ff	....
0xfffef110	08 f1 fe ff	....
0xfffef114	20 9b 08 00	....
0xfffef118	24 f1 fe ff	....
0xfffef11c	88 05 01 00	....
0xfffef120	00 00 00 00	....
more		

> breakpoints
> signals
registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
r1	0xfffef264	4294898276	
r2	0x10	16	
r3	0xfffef108	4294897928	
r4	0x10f98	69528	
r5	0x10158	65880	
r6	0x8b408	570376	
r7	0x10158	65880	
r8	0x0	0	register 8 (64-bit)
r9	0x0	0	register 9 (64-bit)
r10	0x89b04	563972	register 10 (64-bit)
r11	0xfffef11c	4294897948	register 11 (64-bit)
r12	0xfffef138	4294897976	register 12 (64-bit)
sp	0xfffef100	4294897920	
lr	0x10588	66952	

*ptr1 = 16에 의해 b =16이 되었다.



more			
0xfffef104	03 00 00 00	....	
0xfffef108	00 01 00 00	....	
0xfffef10c	04 f1 fe ff	....	
0xfffef110	08 f1 fe ff	....	
0xfffef114	20 9b 08 00	....	
0xfffef118	24 f1 fe ff	....	
0xfffef11c	88 05 01 00	....	
0xfffef120	00 00 00 00	....	
more			

> breakpoints
> signals
registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
r1	0xfffef264	4294898276	
r2	0x10	16	
r3	0x100	256	
r4	0x10f98	69528	
r5	0x10158	65880	
r6	0x8b408	570376	
r7	0x10158	65880	
r8	0x0	0	register 8 (64-bit)
r9	0x0	0	register 9 (64-bit)
r10	0x89b04	563972	register 10 (64-bit)
r11	0xfffef11c	4294897948	register 11 (64-bit)
r12	0xfffef138	4294897976	register 12 (64-bit)
sp	0xfffef100	4294897920	
lr	0x10588	66952	

b=256에 의해 b의 값이 256이 되었다.

실습b

```
void example2()
{
    int mem[40] = {0, };
    char *a;

    int i;

    a = (char *)mem;
    for(i = 0; i < 6; i++){
        *(a + i*3) = 10; //3byte를 건너뛰고 char형인 10을 저장한다.
    }
}
```

0xfffef064	0a 00 00 0a 00 00 0a 00
0xfffef06c	00 0a 00 00 0a 00 00 0a
0xfffef074	00 00 00 00 00 00 00 00
0xfffef07c	00 00 00 00 00 00 00 00

실습c

0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00
0xfffef114	34 0a 01 00
0xfffef118	00 00 00 00
0xfffef11c	01 00 00 00
0xfffef120	54 f2 fe ff

r3, r4, r5의 데이터를 sp-12에서부터 pre-indexing으로 4바이트씩 저장한다.

0xfffef0f4	01 00 00 00
0xfffef0f8	0c 0f 01 00
0xfffef0fc	58 01 01 00
0xfffef100	08 b4 08 00
0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00

r3를 1로 하고 pre-indexing하여 sp+0에 r3를 저장하고 sp = sp+0이다.

0xfffef0f4	02 00 00 00
0xfffef0f8	0c 0f 01 00
0xfffef0fc	58 01 01 00
0xfffef100	08 b4 08 00
0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00

r4를 2로 하고 post-indexing하여 sp에 r4를 저장하고 sp = sp+4가 되었다.

0xfffef0f4	02 00 00 00
0xfffef0f8	00 00 00 00
0xfffef0fc	58 01 01 00
0xfffef100	08 b4 08 00
0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00

r4를 0로 하고 post-indexing하여 sp에 r4를 저장하고 sp = sp+0가 되었다.

0xfffef0f4	02 00 00 00
0xfffef0f8	00 00 00 00
0xfffef0fc	58 01 01 00
0xfffef100	03 00 00 00
0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00

R5를 3로 하고 auto-indexing하여 sp+8에 r5를 저장하고 sp = sp+8가 되었다고 r5도 #8만큼 업데이트 되었다..

0xfffef0f4	02 00 00 00
0xfffef0f8	00 00 00 00
0xfffef0fc	04 00 00 00
0xfffef100	03 00 00 00
0xfffef104	7c 05 01 00
0xfffef108	0c 0f 01 00
0xfffef10c	58 01 01 00
0xfffef110	00 00 00 00

r0 = 0xffffffff이었는데 이를 r3값을 sp + 4xr0의 주소에 넣는데 이때의 주소가 0xfffef0fc이다.

0xfffef110	00 00 00 00
0xfffef114	34 0a 01 00
0xfffef118	00 00 00 00
0xfffef11c	01 00 00 00
0xfffef120	54 f2 fe ff
0xfffef124	7c 05 01 00
0xfffef128	db eb da 05
0xfffef12c	3b 13 25 fa

Sp = sp+16의 주소에서 pre-indexing방식으로 r3, r4, r5의 값을 다시 불러온다.

실습D

Step1

r0	0x1	1	
r1	0xfffef254	4294898260	
r2	0xfffef25c	4294898268	
r3	0x1057c	66940	
r4	0x10f0c	69388	
r5	0x10158	65880	
r6	0x8b408	570376	
r7	0x10158	65880	
r8	0x0	0	register 8 (64-bit)
r9	0x0	0	register 9 (64-bit)
r10	0x89b04	563972	register 10 (64-bit)
r11	0xfffef10c	4294897932	register 11 (64-bit)
r12	0xfffef128	4294897960	register 12 (64-bit)
sp	0xfffef0e8	4294897896	
lr	0x10588	66952	
pc	0x106cc	67276	

0xfffef0e8	00 00 00 00
0xfffef0ec	ac 0f 01 00
0xfffef0f0	ec b3 08 00
0xfffef0f4	64 0f 01 00
0xfffef0f8	0c 0f 01 00
0xfffef0fc	58 01 01 00
0xfffef100	08 b4 08 00
0xfffef104	58 01 01 00
0xfffef108	14 f1 fe ff
0xfffef10c	88 05 01 00
0xfffef110	00 00 00 00

Lr값이 r11에 저장되어있다. 이값은 sp + 40에 저장되어있다.

Step2

아직 깨지지 않았다.

0xfffef0e8	00 00 00 00 02 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 00 00 00 00
0xfffef100	00 00 00 00 20 9b 08 00
0xfffef108	14 f1 fe ff 88 05 01 00
0xfffef110	00 00 00 00 34 0a 01 00

0xfffef0e8	00 00 00 00 04 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	00 00 00 00 20 9b 08 00
0xfffef108	14 f1 fe ff 88 05 01 00
0xfffef110	00 00 00 00 34 0a 01 00

0xfffef0e8	00 00 00 00 05 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 20 9b 08 00
0xfffef108	14 f1 fe ff 88 05 01 00
0xfffef110	00 00 00 00 34 0a 01 00

여기서부터 깨진다

0xfffef0e8	00 00 00 00 06 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 06 00 00 00
0xfffef108	14 f1 fe ff 88 05 01 00
0xfffef110	00 00 00 00 34 0a 01 00

0xfffef0e8	00 00 00 00 07 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 06 00 00 00
0xfffef108	07 00 00 00 88 05 01 00
0xfffef110	00 00 00 00 34 0a 01 00

0xfffef0e8	00 00 00 00 07 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 06 00 00 00
0xfffef108	07 00 00 00 08 00 00 00
0xfffef110	00 00 00 00 34 0a 01 00

0xfffef0e8	00 00 00 00 09 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 06 00 00 00
0xfffef108	07 00 00 00 08 00 00 00
0xfffef110	09 00 00 00 34 0a 01 00

Step3~5

초기화되어있는 상태이다.

0xfffef0e8	00 00 00 00 ac 0f 01 00
0xfffef0f0	00 00 00 00 00 00 00 00
0xfffef0f8	00 00 00 00 00 00 00 00
0xfffef100	00 00 00 00 20 9b 08 00

5넘어로는 데이터가 깨지지 않았다.

0xfffef0e8	00 00 00 00 05 00 00 00
0xfffef0f0	01 00 00 00 02 00 00 00
0xfffef0f8	03 00 00 00 04 00 00 00
0xfffef100	05 00 00 00 20 9b 08 00

```
✘ juneyong ~/lab2 qemu-arm-static -g 8080 ./lab2

Input buffer value: 12345

Print array: 1 2 3 4 5
```