EDDI
Electronic Design
Development Institute

에디로봇아카데미

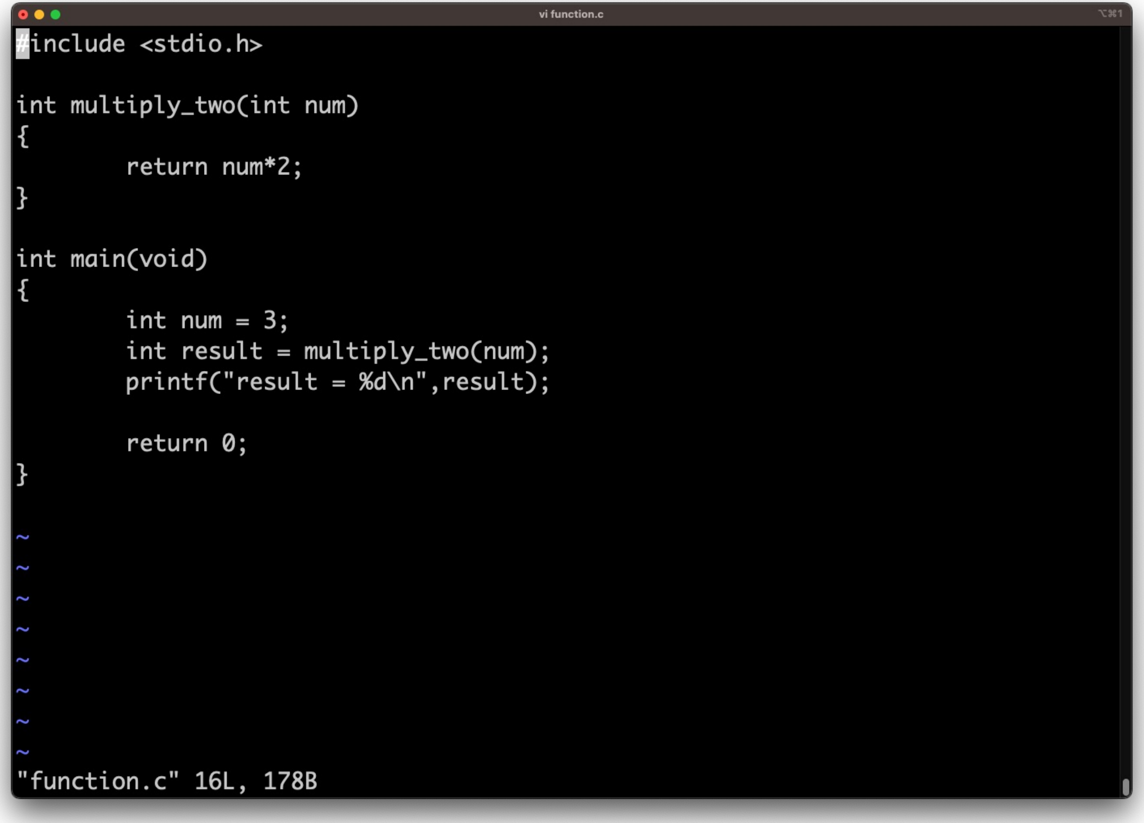
임베디드 마스터 Lv1 과정

제 #5기

2023. 06. 03

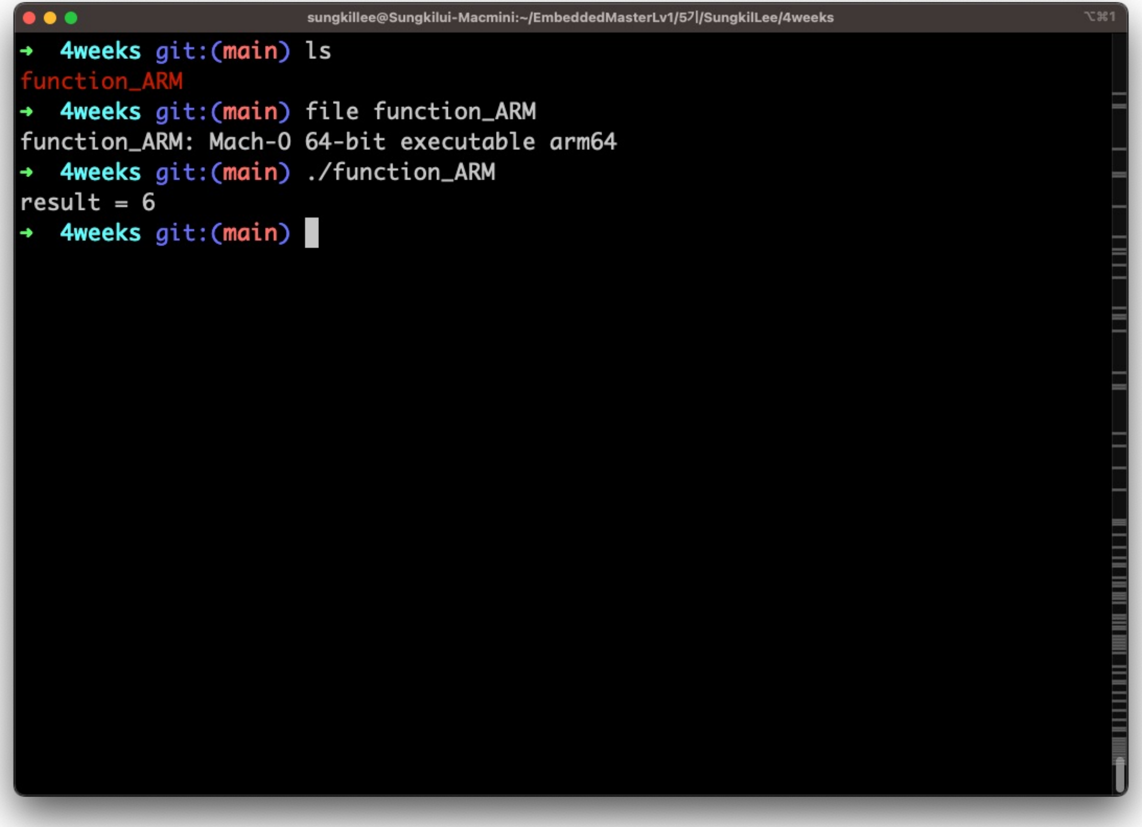
Lee Sungkill

Function.c 파일 생성



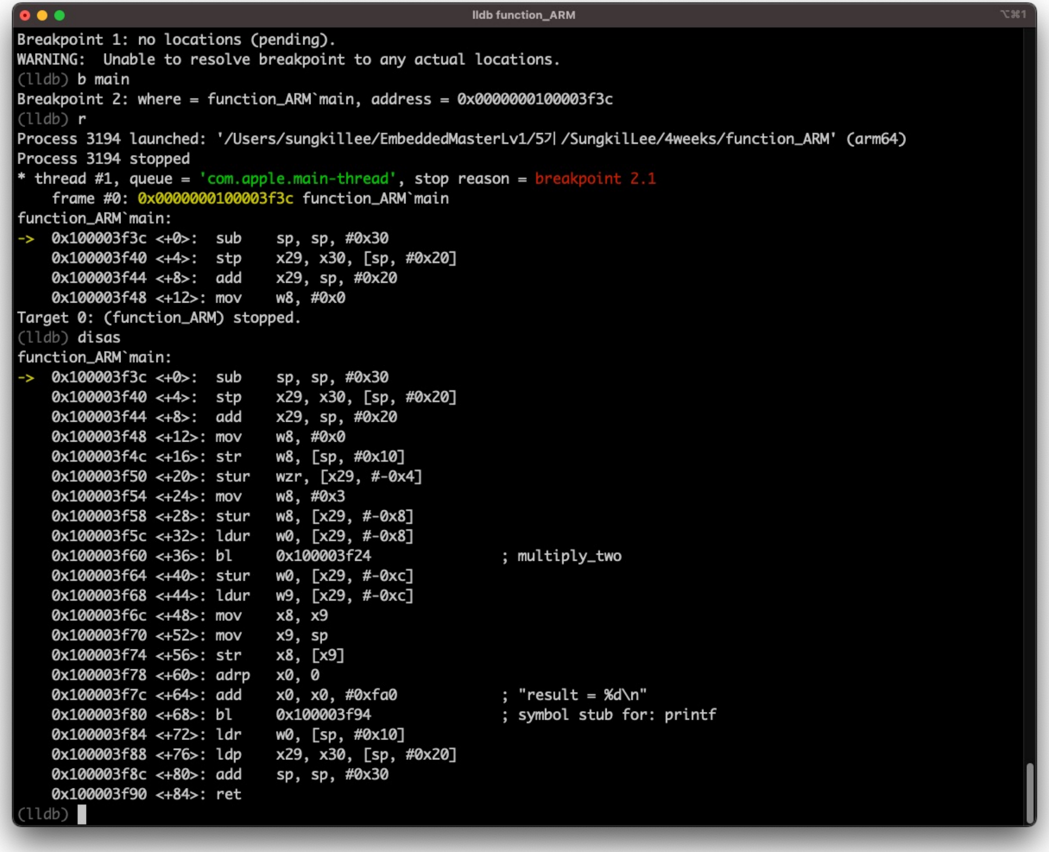
```
#include <stdio.h>

int multiply_two(int num)
{
        return num*2;
}

int main(void)
{
        int num = 3;
        int result = multiply_two(num);
        printf("result = %d\n",result);

        return 0;
}

~
~
~
~
~
~
~
~
"function.c" 16L, 178B
```

Function.c arm64 아키텍쳐 컴파일 실행 및 디버그 파일 생성


sungkillee@Sungkilui-Macmini:~/EmbeddedMasterLv1/5기/SungkilLee/4weeks
⌥⌘1

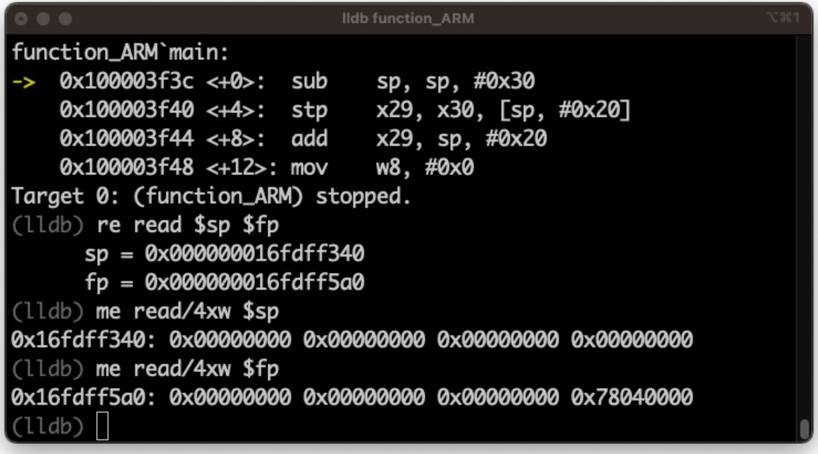
```
➜  4weeks git:(main) ls
function_ARM
➜  4weeks git:(main) file function_ARM
function_ARM: Mach-O 64-bit executable arm64
➜  4weeks git:(main) ./function_ARM
result = 6
➜  4weeks git:(main) 
```

Function.c 디버깅



```
Breakpoint 1: no locations (pending).
WARNING:  Unable to resolve breakpoint to any actual locations.
(lldb) b main
Breakpoint 2: where = function_ARM`main, address = 0x0000000100003f3c
(lldb) r
Process 3194 launched: '/Users/sungkillee/EmbeddedMasterLv1/5기/SungkilLee/4weeks/function_ARM' (arm64)
Process 3194 stopped
* thread #1, queue = 'com.apple.main-thread', stop reason = breakpoint 2.1
    frame #0: 0x0000000100003f3c function_ARM`main
function_ARM`main:
->  0x100003f3c <+0>:  sub    sp, sp, #0x30
    0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
    0x100003f44 <+8>:  add    x29, sp, #0x20
    0x100003f48 <+12>: mov    w8, #0x0
Target 0: (function_ARM) stopped.
(lldb) disas
function_ARM`main:
->  0x100003f3c <+0>:  sub    sp, sp, #0x30
    0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
    0x100003f44 <+8>:  add    x29, sp, #0x20
    0x100003f48 <+12>: mov    w8, #0x0
    0x100003f4c <+16>: str    w8, [sp, #0x10]
    0x100003f50 <+20>: stur   wzr, [x29, #-0x4]
    0x100003f54 <+24>: mov    w8, #0x3
    0x100003f58 <+28>: stur   w8, [x29, #-0x8]
    0x100003f5c <+32>: ldur   w0, [x29, #-0x8]
    0x100003f60 <+36>: bl     0x100003f24               ; multiply_two
    0x100003f64 <+40>: stur   w0, [x29, #-0xc]
    0x100003f68 <+44>: ldur   w9, [x29, #-0xc]
    0x100003f6c <+48>: mov    x8, x9
    0x100003f70 <+52>: mov    x9, sp
    0x100003f74 <+56>: str    x8, [x9]
    0x100003f78 <+60>: adrp   x0, 0
    0x100003f7c <+64>: add    x0, x0, #0xfa0            ; "result = %d\n"
    0x100003f80 <+68>: bl     0x100003f94               ; symbol stub for: printf
    0x100003f84 <+72>: ldr    w0, [sp, #0x10]
    0x100003f88 <+76>: ldp    x29, x30, [sp, #0x20]
    0x100003f8c <+80>: add    sp, sp, #0x30
    0x100003f90 <+84>: ret
(lldb) 
```

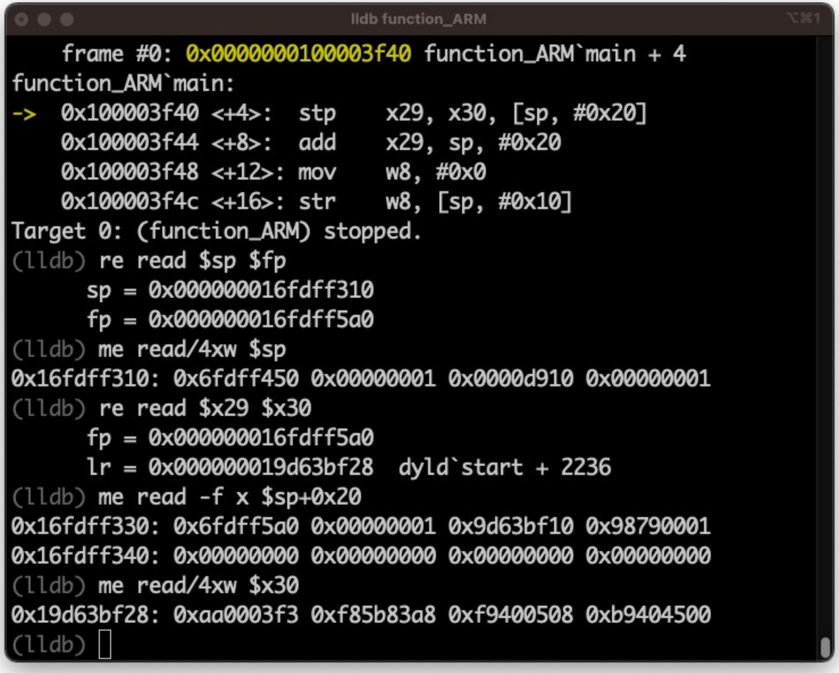
Function.c 디버깅



```
function_ARM`main:
->  0x100003f3c <+0>:  sub    sp, sp, #0x30
    0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
    0x100003f44 <+8>:  add    x29, sp, #0x20
    0x100003f48 <+12>: mov    w8, #0x0
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff340
      fp = 0x000000016fdff5a0
(lldb) me read/4xw $sp
0x16fdff340: 0x00000000 0x00000000 0x00000000 0x00000000
(lldb) me read/4xw $fp
0x16fdff5a0: 0x00000000 0x00000000 0x00000000 0x78040000
(lldb) 
```

fp	0	0	0	78040000	16fdff5a0
sp	0	0	0	0	16fdff340

Function.c 디버깅

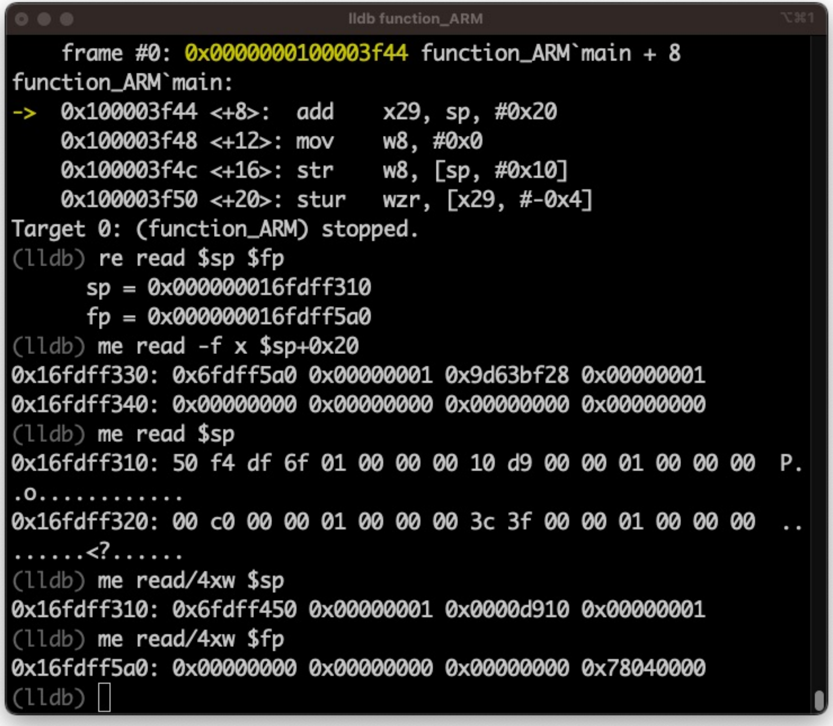


```
    frame #0: 0x0000000100003f40 function_ARM`main + 4
function_ARM`main:
->  0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
    0x100003f44 <+8>:  add    x29, sp, #0x20
    0x100003f48 <+12>: mov    w8, #0x0
    0x100003f4c <+16>: str    w8, [sp, #0x10]
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff310
      fp = 0x000000016fdff5a0
(lldb) me read/4xw $sp
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) re read $x29 $x30
      fp = 0x000000016fdff5a0
      lr = 0x000000019d63bf28  dyld`start + 2236
(lldb) me read -f x $sp+0x20
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf10 0x98790001
0x16fdff340: 0x00000000 0x00000000 0x00000000 0x00000000
(lldb) me read/4xw $x30
0x19d63bf28: 0xaa0003f3 0xf85b83a8 0xf9400508 0xb9404500
(lldb) 
```

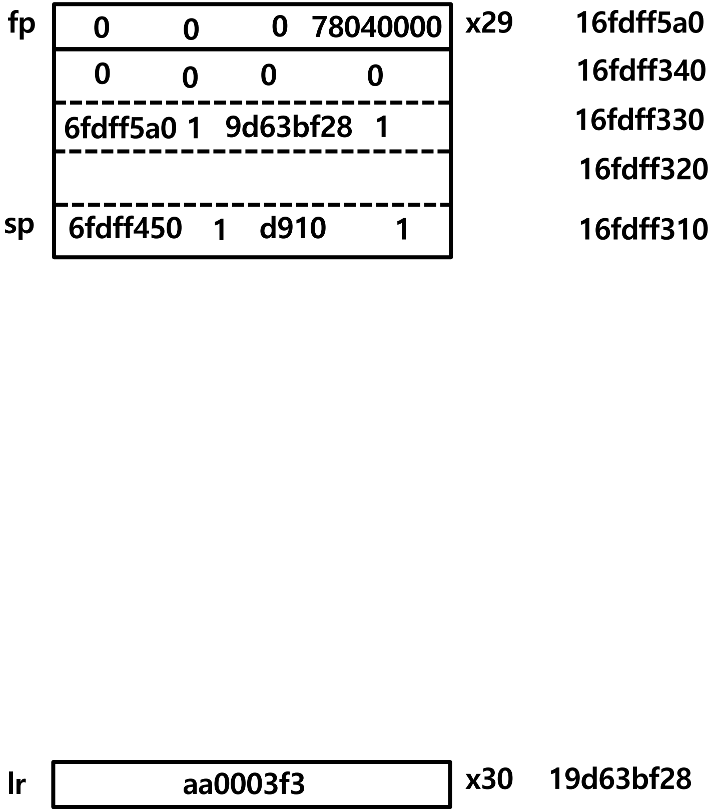
fp	0 0 0 78040000	x29	16fdff5a0
	0 0 0 0		16fdff340
	6fdff5a019d63bf1098790001		16fdff330
			16fdff320
sp	6fdff450 1 d910 1		16fdff310

lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



```
    frame #0: 0x0000000100003f44 function_ARM`main + 8
function_ARM`main:
->  0x100003f44 <+8>:  add    x29, sp, #0x20
    0x100003f48 <+12>: mov    w8, #0x0
    0x100003f4c <+16>: str    w8, [sp, #0x10]
    0x100003f50 <+20>: stur   wzr, [x29, #-0x4]
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff310
      fp = 0x000000016fdff5a0
(lldb) me read -f x $sp+0x20
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf28 0x00000001
0x16fdff340: 0x00000000 0x00000000 0x00000000 0x00000000
(lldb) me read $sp
0x16fdff310: 50 f4 df 6f 01 00 00 00 10 d9 00 00 01 00 00 00  P.
.o...........
0x16fdff320: 00 c0 00 00 01 00 00 00 3c 3f 00 00 01 00 00 00  ..
......<?......
(lldb) me read/4xw $sp
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) me read/4xw $fp
0x16fdff5a0: 0x00000000 0x00000000 0x00000000 0x78040000
(lldb) 
```



fp | 0 0 0 78040000 | x29 | 16fdff5a0

0 0 0 0 | 16fdff340

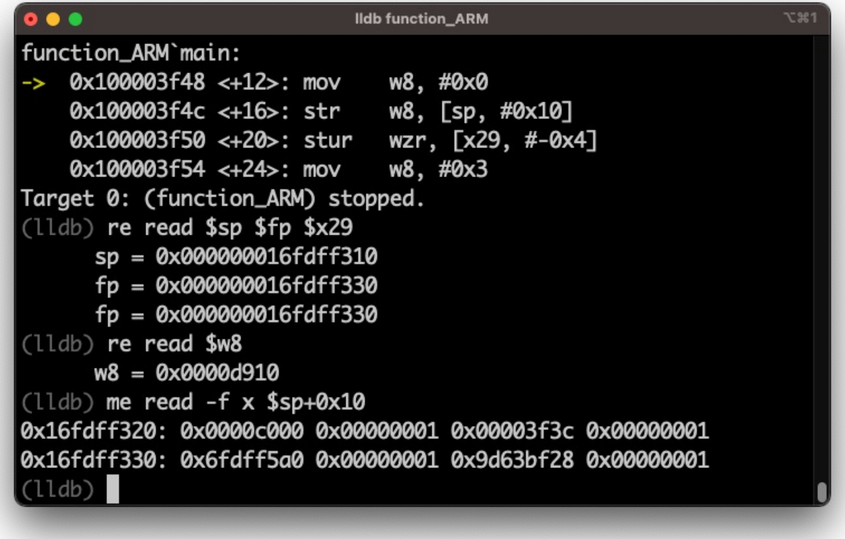
6fdff5a0 1 9d63bf28 1 | 16fdff330

16fdff320

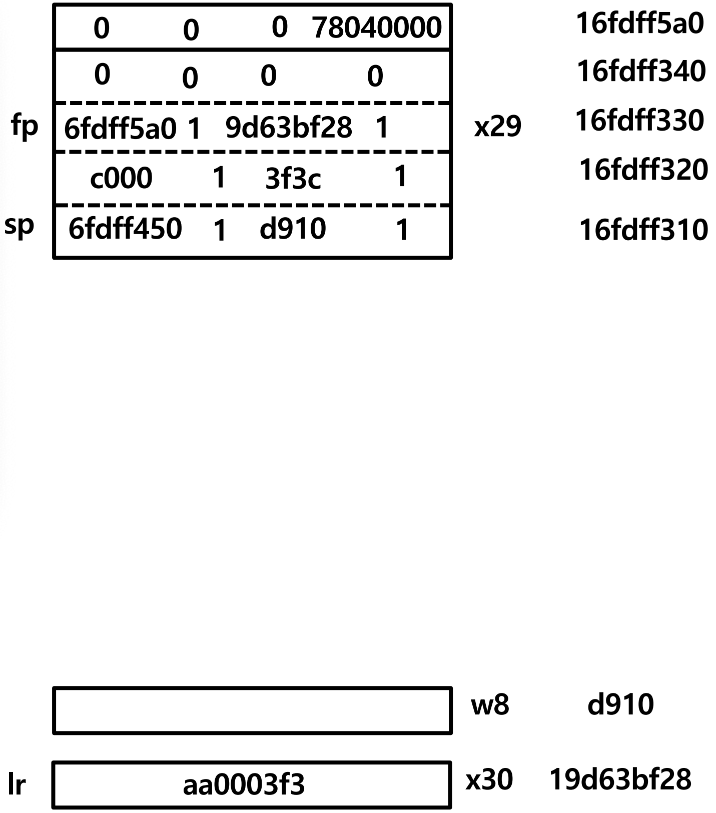
sp | 6fdff450 1 d910 1 | 16fdff310

lr | aa0003f3 | x30 | 19d63bf28

Function.c 디버깅



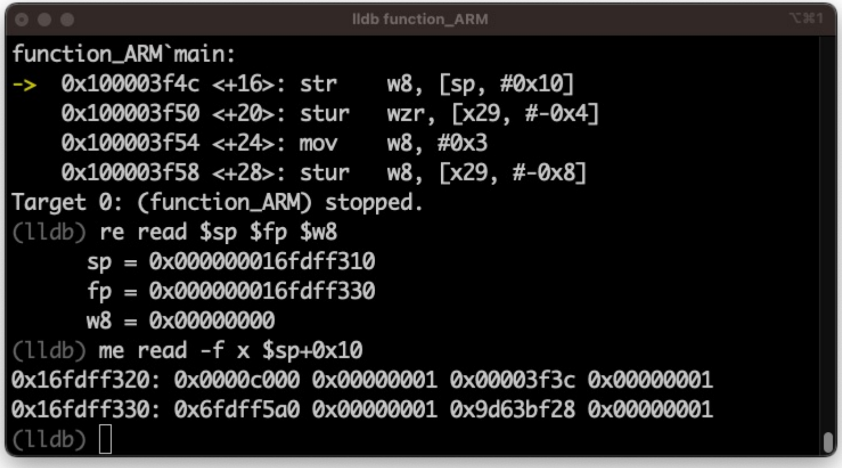
```
function_ARM`main:
->  0x100003f48 <+12>: mov    w8, #0x0
    0x100003f4c <+16>: str    w8, [sp, #0x10]
    0x100003f50 <+20>: stur   wzr, [x29, #-0x4]
    0x100003f54 <+24>: mov    w8, #0x3
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp $x29
      sp = 0x000000016fdff310
      fp = 0x000000016fdff330
      fp = 0x000000016fdff330
(lldb) re read $w8
      w8 = 0x0000d910
(lldb) me read -f x $sp+0x10
0x16fdff320: 0x0000c000 0x00000001 0x00003f3c 0x00000001
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf28 0x00000001
(lldb) 
```



	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	c000	1	3f3c	1		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w8	d910
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅

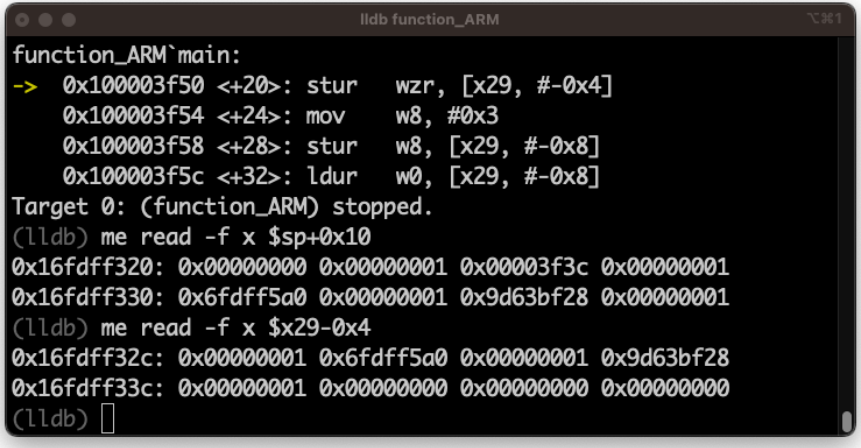


```
function_ARM`main:
->  0x100003f4c <+16>: str    w8, [sp, #0x10]
    0x100003f50 <+20>: stur   wzr, [x29, #-0x4]
    0x100003f54 <+24>: mov    w8, #0x3
    0x100003f58 <+28>: stur   w8, [x29, #-0x8]
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp $w8
      sp = 0x000000016fdff310
      fp = 0x000000016fdff330
      w8 = 0x00000000
(lldb) me read -f x $sp+0x10
0x16fdff320: 0x0000c000 0x00000001 0x00003f3c 0x00000001
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf28 0x00000001
(lldb) 
```

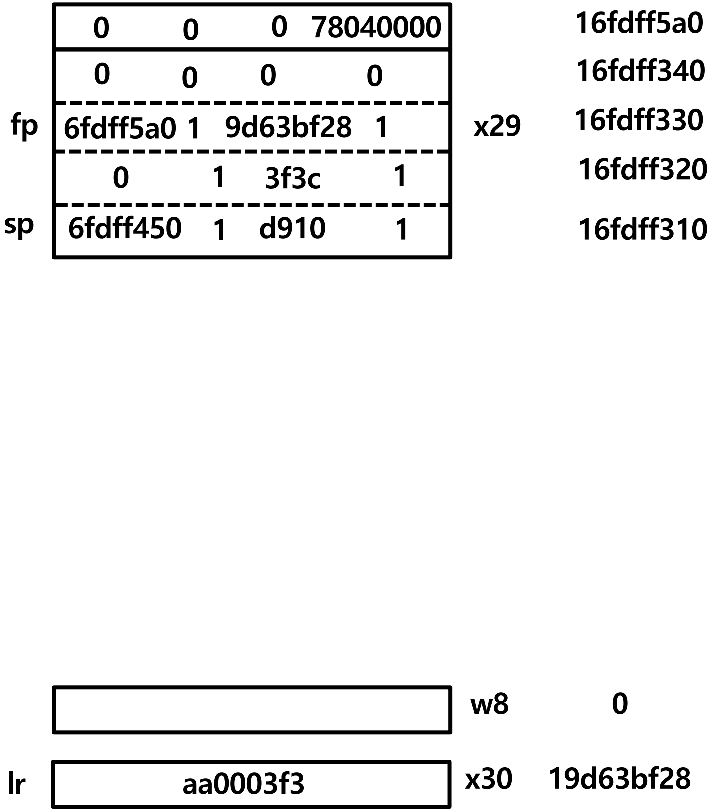
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	c000	1	3f3c	1		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w8	0
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



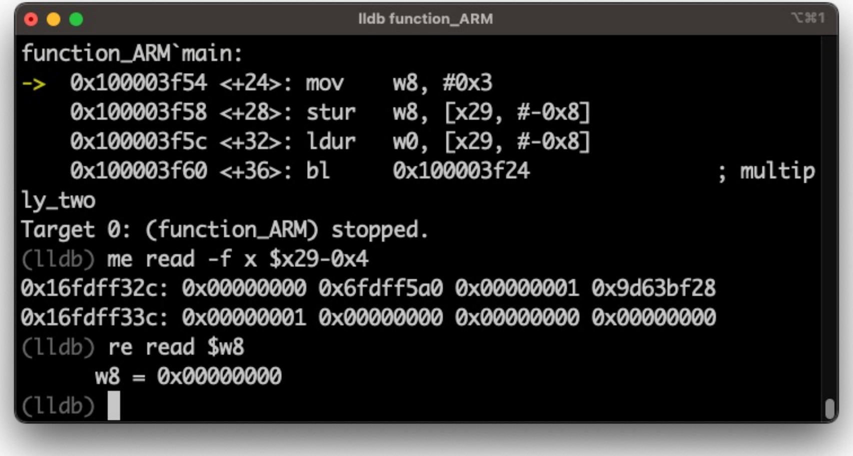
```
function_ARM`main:
->  0x100003f50 <+20>: stur   wzr, [x29, #-0x4]
    0x100003f54 <+24>: mov    w8, #0x3
    0x100003f58 <+28>: stur   w8, [x29, #-0x8]
    0x100003f5c <+32>: ldur   w0, [x29, #-0x8]
Target 0: (function_ARM) stopped.
(lldb) me read -f x $sp+0x10
0x16fdff320: 0x00000000 0x00000001 0x00003f3c 0x00000001
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf28 0x00000001
(lldb) me read -f x $x29-0x4
0x16fdff32c: 0x00000001 0x6fdff5a0 0x00000001 0x9d63bf28
0x16fdff33c: 0x00000001 0x00000000 0x00000000 0x00000000
(lldb) 
```



	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3f3c	1		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w8	0
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅

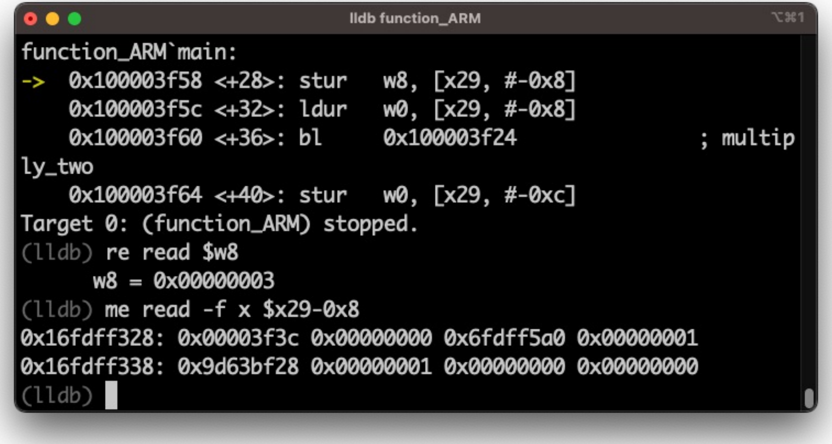


```
function_ARM`main:
->  0x100003f54 <+24>: mov    w8, #0x3
    0x100003f58 <+28>: stur   w8, [x29, #-0x8]
    0x100003f5c <+32>: ldur   w0, [x29, #-0x8]
    0x100003f60 <+36>: bl     0x100003f24               ; multip
ly_two
Target 0: (function_ARM) stopped.
(lldb) me read -f x $x29-0x4
0x16fdff32c: 0x00000000 0x6fdff5a0 0x00000001 0x9d63bf28
0x16fdff33c: 0x00000001 0x00000000 0x00000000 0x00000000
(lldb) re read $w8
      w8 = 0x00000000
(lldb) 
```

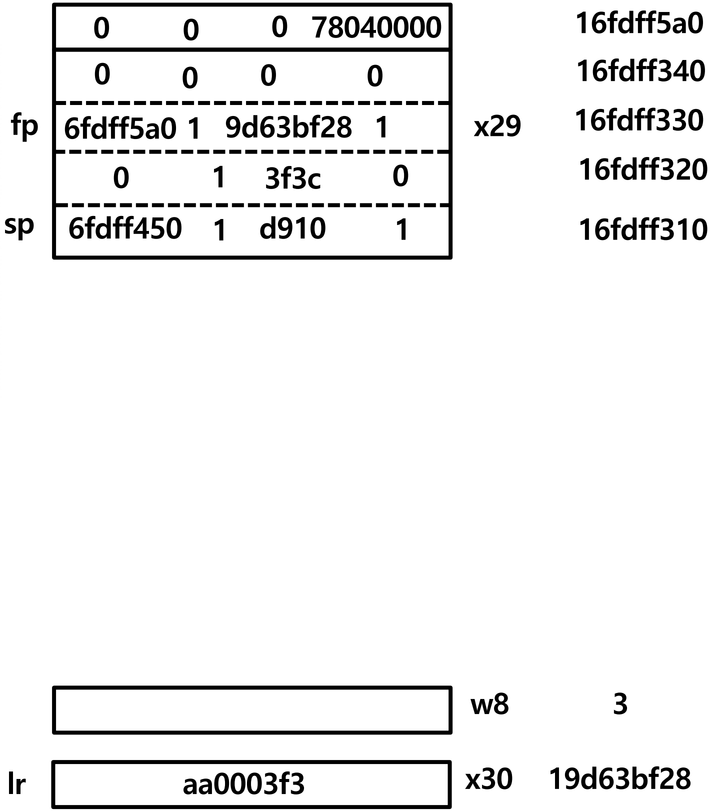
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3f3c	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w8	0
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



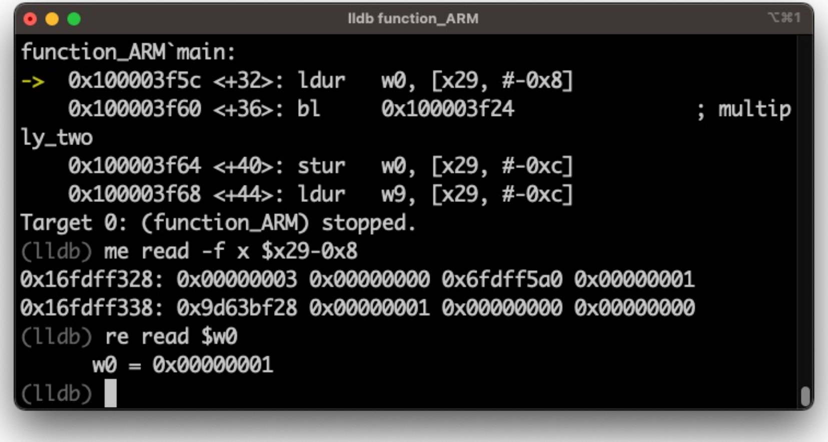
```
function_ARM`main:
->  0x100003f58 <+28>: stur   w8, [x29, #-0x8]
    0x100003f5c <+32>: ldur   w0, [x29, #-0x8]
    0x100003f60 <+36>: bl     0x100003f24               ; multip
ly_two
    0x100003f64 <+40>: stur   w0, [x29, #-0xc]
Target 0: (function_ARM) stopped.
(lldb) re read $w8
      w8 = 0x00000003
(lldb) me read -f x $x29-0x8
0x16fdff328: 0x00003f3c 0x00000000 0x6fdff5a0 0x00000001
0x16fdff338: 0x9d63bf28 0x00000001 0x00000000 0x00000000
(lldb)
```



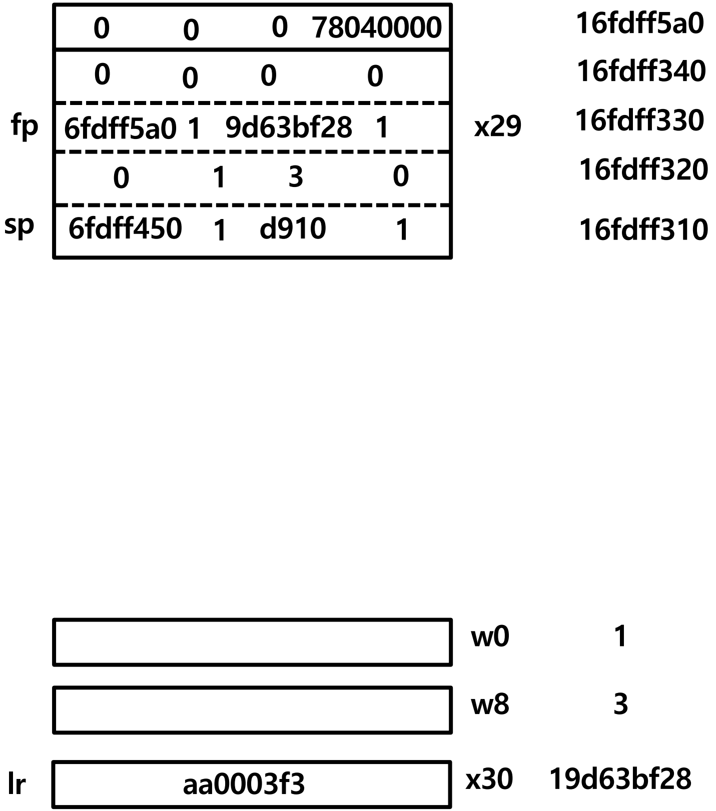
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3f3c	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w8	3
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



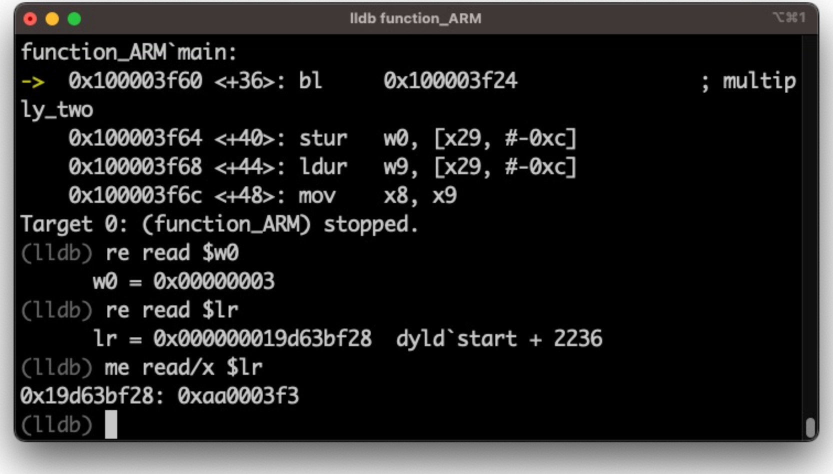
```
function_ARM`main:
->  0x100003f5c <+32>: ldur   w0, [x29, #-0x8]
    0x100003f60 <+36>: bl     0x100003f24               ; multip
ly_two
    0x100003f64 <+40>: stur   w0, [x29, #-0xc]
    0x100003f68 <+44>: ldur   w9, [x29, #-0xc]
Target 0: (function_ARM) stopped.
(lldb) me read -f x $x29-0x8
0x16fdff328: 0x00000003 0x00000000 0x6fdff5a0 0x00000001
0x16fdff338: 0x9d63bf28 0x00000001 0x00000000 0x00000000
(lldb) re read $w0
      w0 = 0x00000001
(lldb) 
```



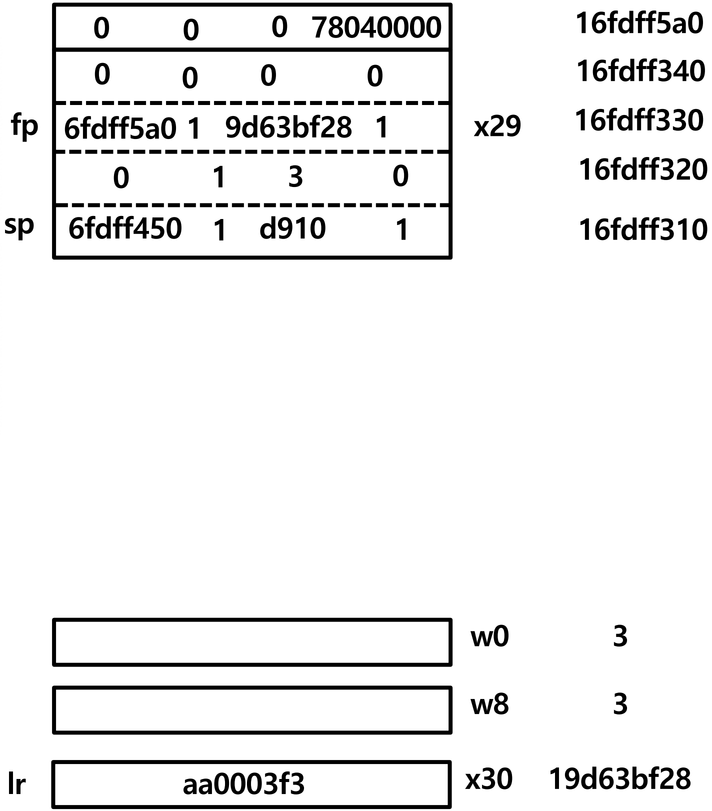
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w0	1
		w8	3
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



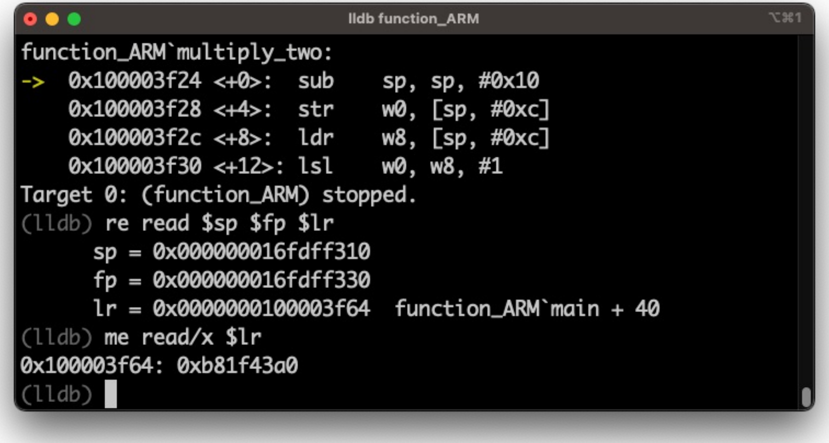
```
function_ARM`main:
->  0x100003f60 <+36>: bl     0x100003f24               ; multip
ly_two
    0x100003f64 <+40>: stur   w0, [x29, #-0xc]
    0x100003f68 <+44>: ldur   w9, [x29, #-0xc]
    0x100003f6c <+48>: mov    x8, x9
Target 0: (function_ARM) stopped.
(lldb) re read $w0
      w0 = 0x00000003
(lldb) re read $lr
      lr = 0x000000019d63bf28  dyld`start + 2236
(lldb) me read/x $lr
0x19d63bf28: 0xaa0003f3
(lldb) 
```



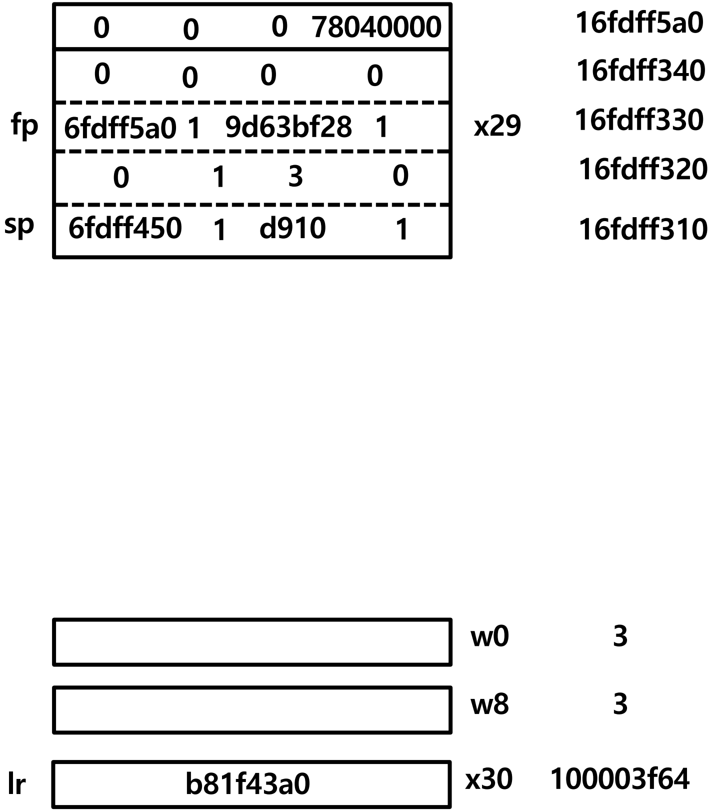
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w0	3
		w8	3
lr	aa0003f3	x30	19d63bf28

Function.c 디버깅



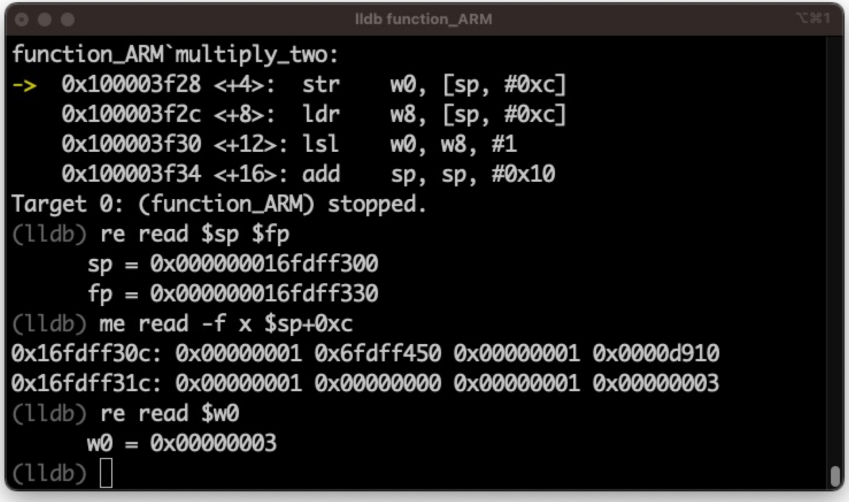
```
function_ARM`multiply_two:
->  0x100003f24 <+0>:  sub    sp, sp, #0x10
    0x100003f28 <+4>:  str    w0, [sp, #0xc]
    0x100003f2c <+8>:  ldr    w8, [sp, #0xc]
    0x100003f30 <+12>: lsl    w0, w8, #1
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp $lr
      sp = 0x000000016fdff310
      fp = 0x000000016fdff330
      lr = 0x0000000100003f64  function_ARM`main + 40
(lldb) me read/x $lr
0x100003f64: 0xb81f43a0
(lldb)
```



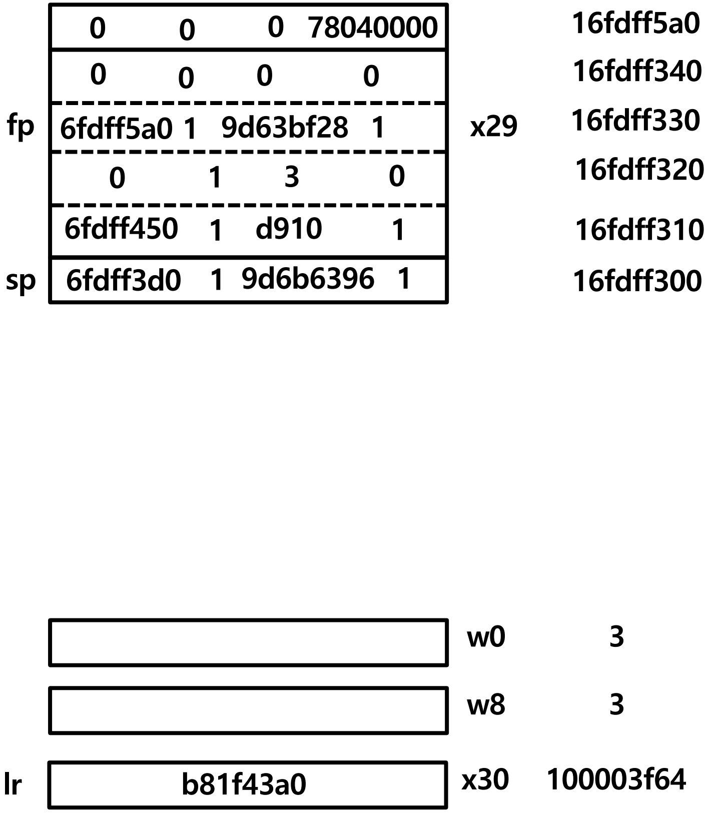
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310

		w0	3
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅



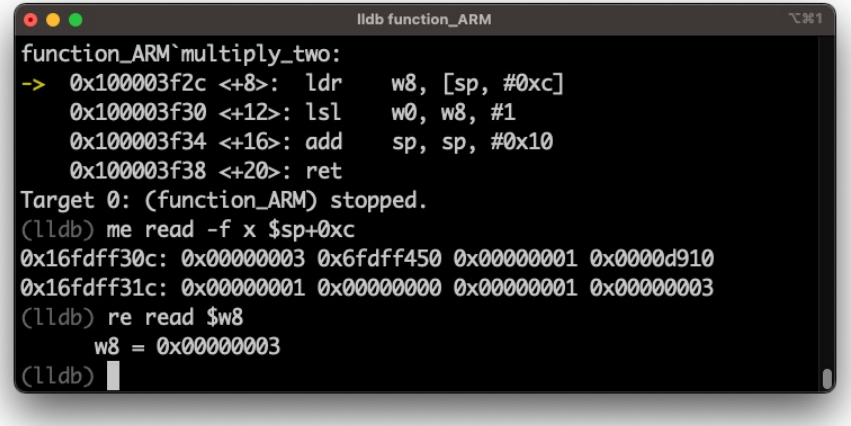
```
function_ARM`multiply_two:
->  0x100003f28 <+4>:  str    w0, [sp, #0xc]
    0x100003f2c <+8>:  ldr    w8, [sp, #0xc]
    0x100003f30 <+12>: lsl    w0, w8, #1
    0x100003f34 <+16>: add    sp, sp, #0x10
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff300
      fp = 0x000000016fdff330
(lldb) me read -f x $sp+0xc
0x16fdff30c: 0x00000001 0x6fdff450 0x00000001 0x0000d910
0x16fdff31c: 0x00000001 0x00000000 0x00000001 0x00000003
(lldb) re read $w0
      w0 = 0x00000003
(lldb)
```



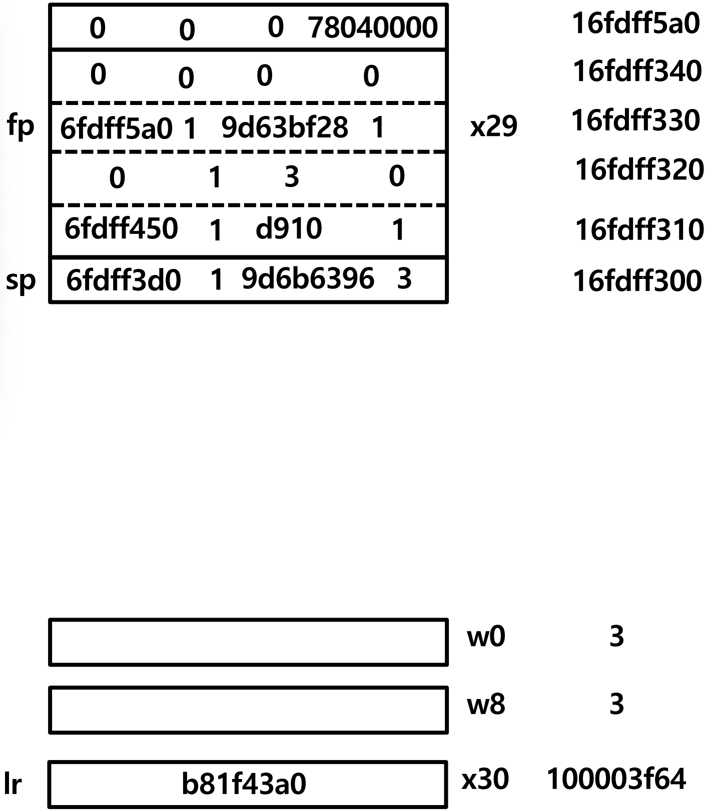
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
	6fdff450	1	d910	1		16fdff310
sp	6fdff3d0	1	9d6b6396	1		16fdff300

		w0	3
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅



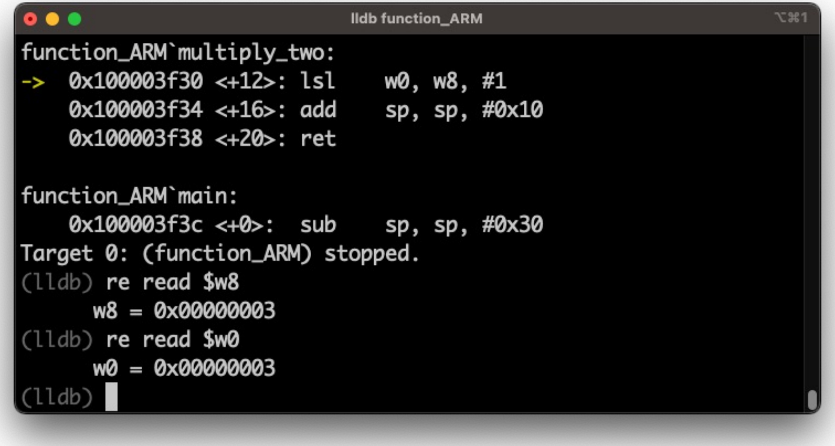
```
function_ARM`multiply_two:
->  0x100003f2c <+8>:  ldr    w8, [sp, #0xc]
    0x100003f30 <+12>: lsl    w0, w8, #1
    0x100003f34 <+16>: add    sp, sp, #0x10
    0x100003f38 <+20>: ret
Target 0: (function_ARM) stopped.
(lldb) me read -f x $sp+0xc
0x16fdff30c: 0x00000003 0x6fdff450 0x00000001 0x0000d910
0x16fdff31c: 0x00000001 0x00000000 0x00000001 0x00000003
(lldb) re read $w8
      w8 = 0x00000003
(lldb) 
```



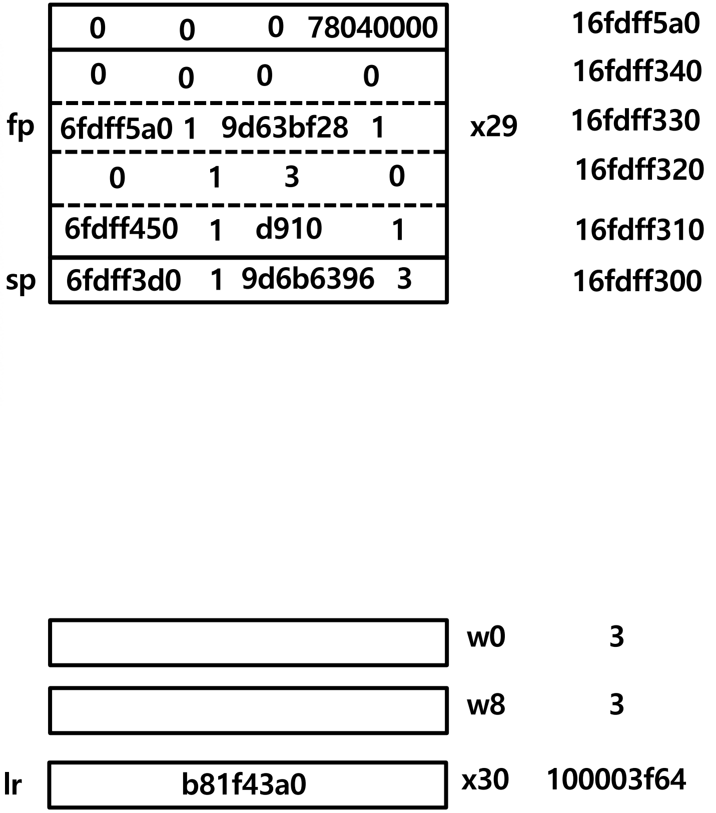
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
	6fdff450	1	d910	1		16fdff310
sp	6fdff3d0	1	9d6b6396	3		16fdff300

		w0	3
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅



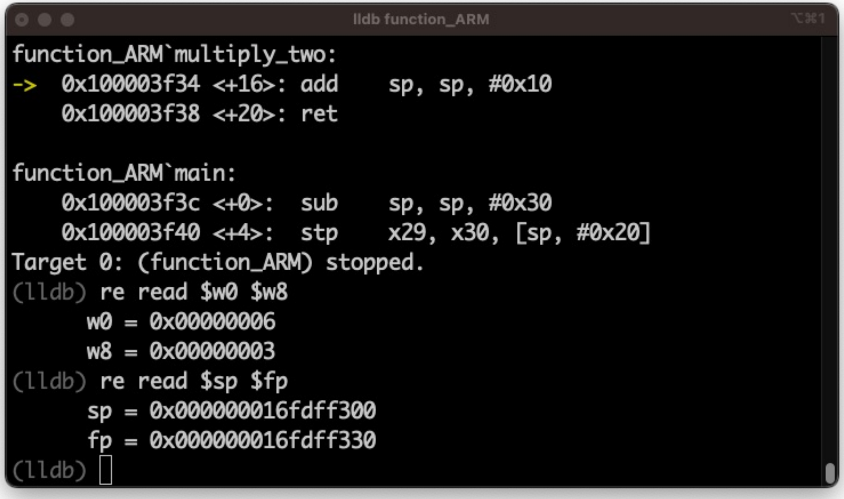
```
function_ARM`multiply_two:
->  0x100003f30 <+12>: lsl    w0, w8, #1
    0x100003f34 <+16>: add    sp, sp, #0x10
    0x100003f38 <+20>: ret

function_ARM`main:
    0x100003f3c <+0>:  sub    sp, sp, #0x30
Target 0: (function_ARM) stopped.
(lldb) re read $w8
      w8 = 0x00000003
(lldb) re read $w0
      w0 = 0x00000003
(lldb) 
```



	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
	6fdff450	1	d910	1		16fdff310
sp	6fdff3d0	1	9d6b6396	3		16fdff300

		w0	3
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅

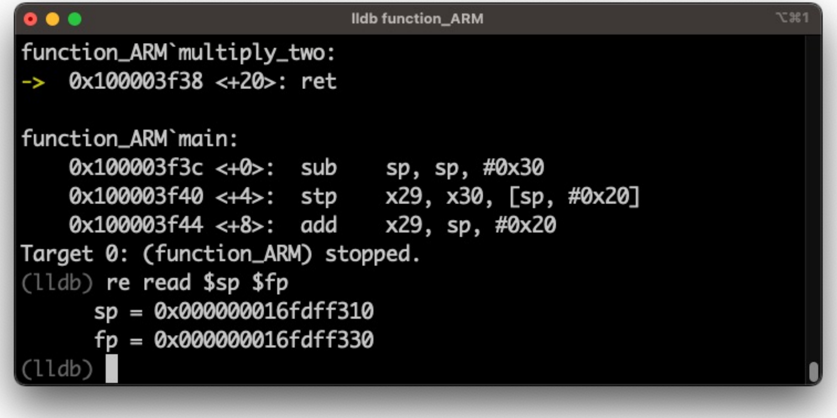


```
function_ARM`multiply_two:
->  0x100003f34 <+16>: add    sp, sp, #0x10
    0x100003f38 <+20>: ret

function_ARM`main:
    0x100003f3c <+0>:  sub    sp, sp, #0x30
    0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
Target 0: (function_ARM) stopped.
(lldb) re read $w0 $w8
      w0 = 0x00000006
      w8 = 0x00000003
(lldb) re read $sp $fp
      sp = 0x000000016fdff300
      fp = 0x000000016fdff330
(lldb) 
```

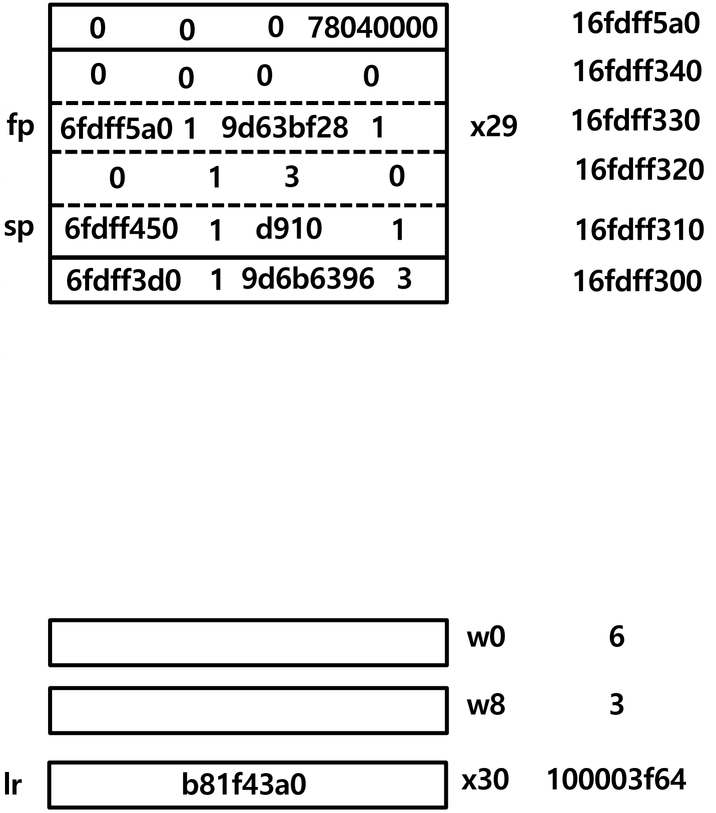
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
	6fdff450	1	d910	1		16fdff310
sp	6fdff3d0	1	9d6b6396	3		16fdff300

		w0	6
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅

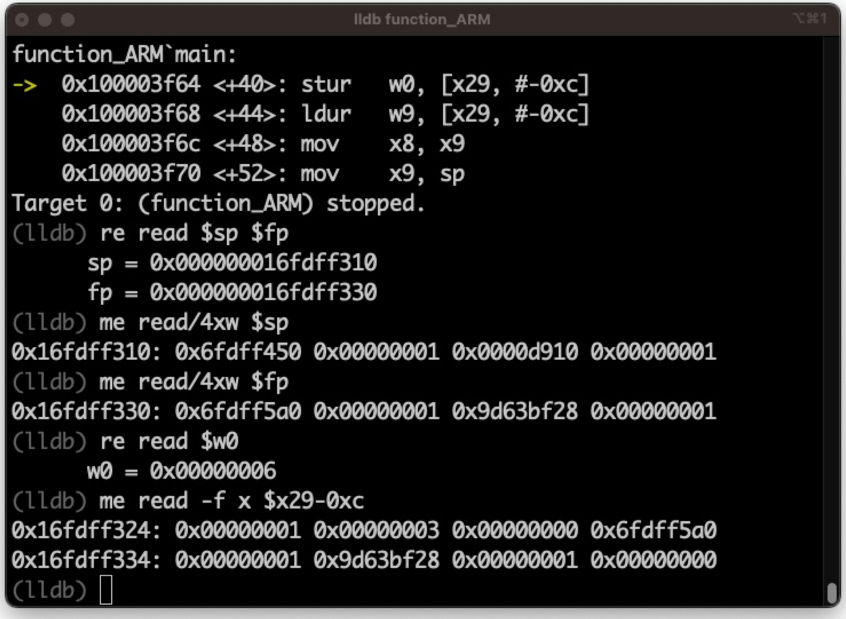


```
function_ARM`multiply_two:
->  0x100003f38 <+20>: ret

function_ARM`main:
    0x100003f3c <+0>:  sub    sp, sp, #0x30
    0x100003f40 <+4>:  stp    x29, x30, [sp, #0x20]
    0x100003f44 <+8>:  add    x29, sp, #0x20
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff310
      fp = 0x000000016fdff330
(lldb)
```



	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310
	6fdff3d0	1	9d6b6396	3		16fdff300
					w0	6
					w8	3
lr	b81f43a0				x30	100003f64

Function.c 디버깅

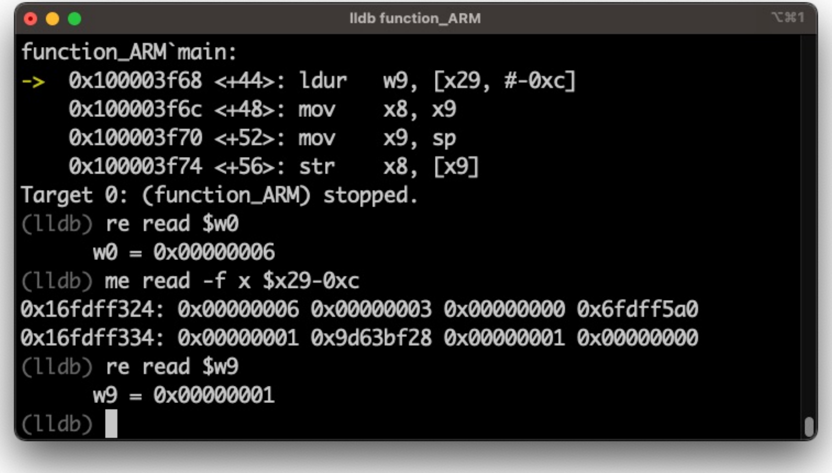


```
function_ARM`main:
->  0x100003f64 <+40>: stur   w0, [x29, #-0xc]
    0x100003f68 <+44>: ldur   w9, [x29, #-0xc]
    0x100003f6c <+48>: mov    x8, x9
    0x100003f70 <+52>: mov    x9, sp
Target 0: (function_ARM) stopped.
(lldb) re read $sp $fp
      sp = 0x000000016fdff310
      fp = 0x000000016fdff330
(lldb) me read/4xw $sp
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) me read/4xw $fp
0x16fdff330: 0x6fdff5a0 0x00000001 0x9d63bf28 0x00000001
(lldb) re read $w0
      w0 = 0x00000006
(lldb) me read -f x $x29-0xc
0x16fdff324: 0x00000001 0x00000003 0x00000000 0x6fdff5a0
0x16fdff334: 0x00000001 0x9d63bf28 0x00000001 0x00000000
(lldb) 
```

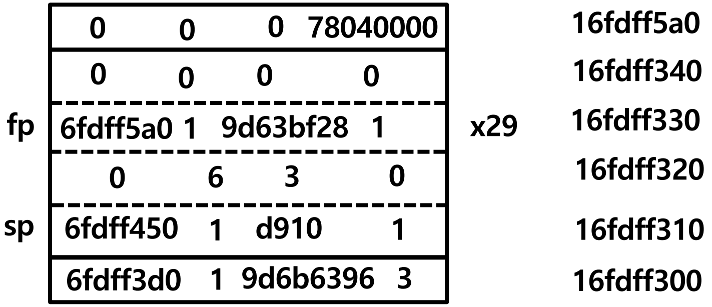
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	1	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310
	6fdff3d0	1	9d6b6396	3		16fdff300

		w0	6
		w8	3
lr	b81f43a0	x30	100003f64

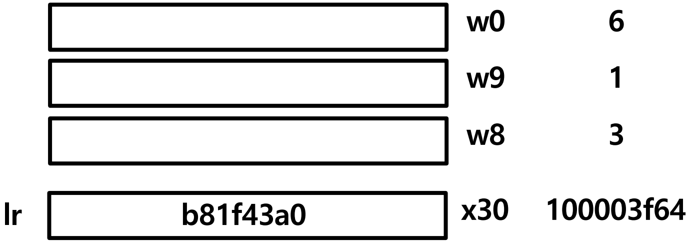
Function.c 디버깅



```
function_ARM`main:
->  0x100003f68 <+44>: ldur   w9, [x29, #-0xc]
    0x100003f6c <+48>: mov    x8, x9
    0x100003f70 <+52>: mov    x9, sp
    0x100003f74 <+56>: str    x8, [x9]
Target 0: (function_ARM) stopped.
(lldb) re read $w0
      w0 = 0x00000006
(lldb) me read -f x $x29-0xc
0x16fdff324: 0x00000006 0x00000003 0x00000000 0x6fdff5a0
0x16fdff334: 0x00000001 0x9d63bf28 0x00000001 0x00000000
(lldb) re read $w9
      w9 = 0x00000001
(lldb) 
```

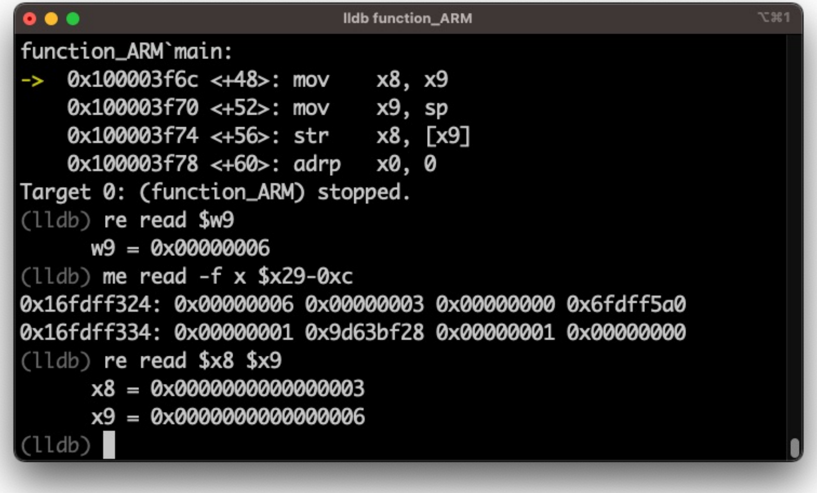


	0 0 0 78040000		16fdff5a0
	0 0 0 0		16fdff340
fp	6fdff5a0 1 9d63bf28 1	x29	16fdff330
	0 6 3 0		16fdff320
sp	6fdff450 1 d910 1		16fdff310
	6fdff3d0 1 9d6b6396 3		16fdff300



		w0	6
		w9	1
		w8	3
lr	b81f43a0	x30	100003f64

Function.c 디버깅

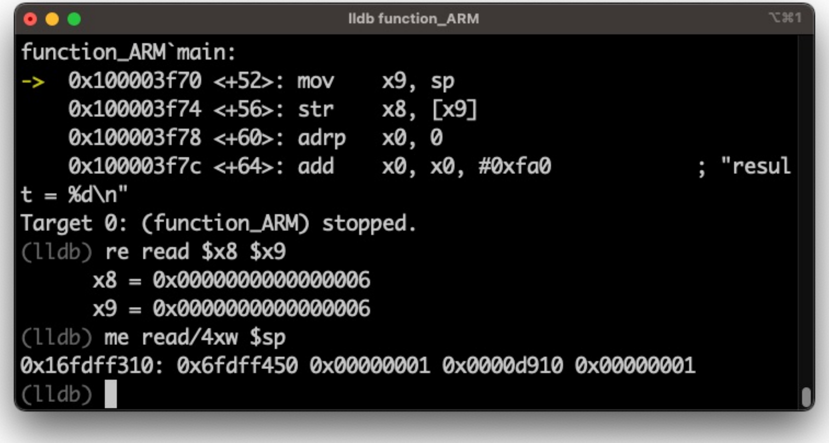


```
function_ARM`main:
->  0x100003f6c <+48>: mov    x8, x9
    0x100003f70 <+52>: mov    x9, sp
    0x100003f74 <+56>: str    x8, [x9]
    0x100003f78 <+60>: adrp   x0, 0
Target 0: (function_ARM) stopped.
(lldb) re read $w9
      w9 = 0x00000006
(lldb) me read -f x $x29-0xc
0x16fdff324: 0x00000006 0x00000003 0x00000000 0x6fdff5a0
0x16fdff334: 0x00000001 0x9d63bf28 0x00000001 0x00000000
(lldb) re read $x8 $x9
      x8 = 0x0000000000000003
      x9 = 0x0000000000000006
(lldb)
```

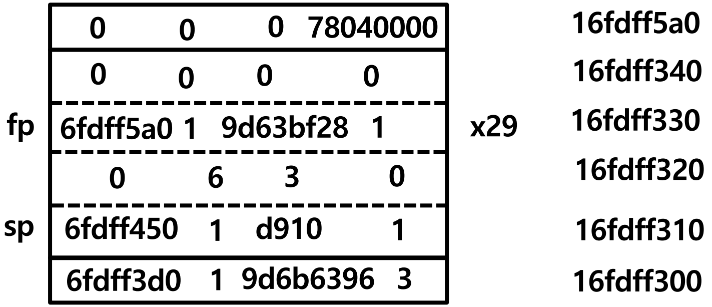
	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	6	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310
	6fdff3d0	1	9d6b6396	3		16fdff300

		w0	6
x9		w9	6
x8		w8	3
lr	b81f43a0	x30	100003f64

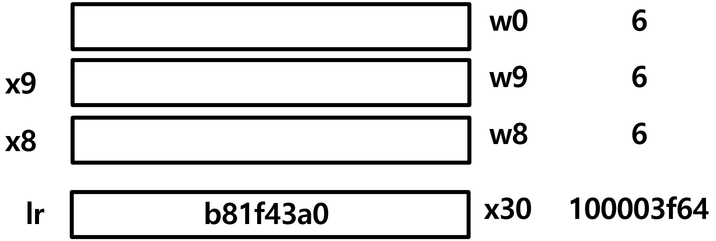
Function.c 디버깅



```
function_ARM`main:
->  0x100003f70 <+52>: mov    x9, sp
    0x100003f74 <+56>: str    x8, [x9]
    0x100003f78 <+60>: adrp   x0, 0
    0x100003f7c <+64>: add    x0, x0, #0xfa0           ; "resul
t = %d\n"
Target 0: (function_ARM) stopped.
(lldb) re read $x8 $x9
      x8 = 0x0000000000000006
      x9 = 0x0000000000000006
(lldb) me read/4xw $sp
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) 
```

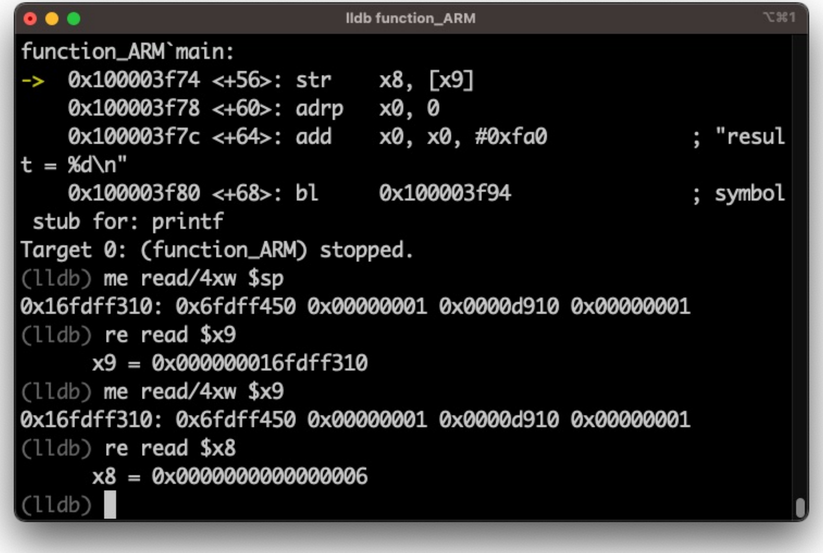


	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	6	3	0		16fdff320
sp	6fdff450	1	d910	1		16fdff310
	6fdff3d0	1	9d6b6396	3		16fdff300

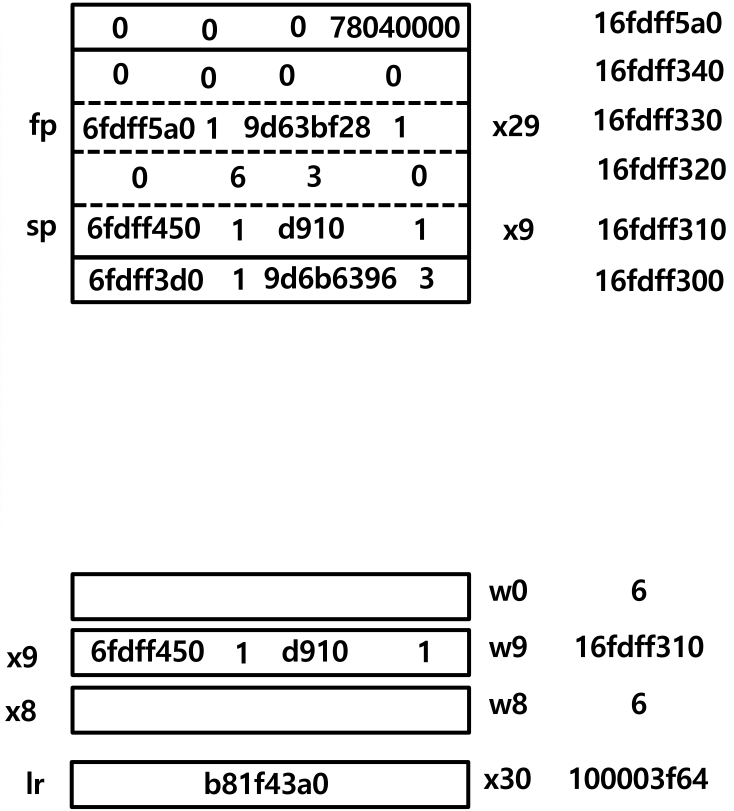


		w0	6
x9		w9	6
x8		w8	6
lr	b81f43a0	x30	100003f64

Function.c 디버깅



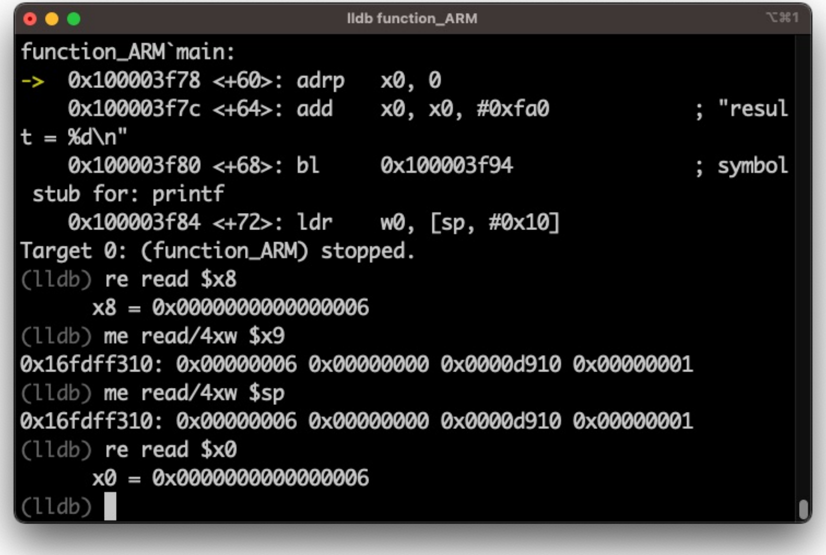
```
function_ARM`main:
->  0x100003f74 <+56>: str    x8, [x9]
    0x100003f78 <+60>: adrp   x0, 0
    0x100003f7c <+64>: add    x0, x0, #0xfa0            ; "resul
t = %d\n"
    0x100003f80 <+68>: bl     0x100003f94               ; symbol
 stub for: printf
Target 0: (function_ARM) stopped.
(lldb) me read/4xw $sp
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) re read $x9
      x9 = 0x000000016fdff310
(lldb) me read/4xw $x9
0x16fdff310: 0x6fdff450 0x00000001 0x0000d910 0x00000001
(lldb) re read $x8
      x8 = 0x0000000000000006
(lldb)
```



	0 0 0 78040000		16fdff5a0
	0 0 0 0		16fdff340
fp	6fdff5a0 1 9d63bf28 1	x29	16fdff330
	0 6 3 0		16fdff320
sp	6fdff450 1 d910 1	x9	16fdff310
	6fdff3d0 1 9d6b6396 3		16fdff300

		w0	6
x9	6fdff450 1 d910 1	w9	16fdff310
x8		w8	6
lr	b81f43a0	x30	100003f64

Function.c 디버깅



```
function_ARM`main:
->  0x100003f78 <+60>: adrp   x0, 0
    0x100003f7c <+64>: add    x0, x0, #0xfa0            ; "resul
t = %d\n"
    0x100003f80 <+68>: bl     0x100003f94               ; symbol
 stub for: printf
    0x100003f84 <+72>: ldr    w0, [sp, #0x10]
Target 0: (function_ARM) stopped.
(lldb) re read $x8
      x8 = 0x0000000000000006
(lldb) me read/4xw $x9
0x16fdff310: 0x00000006 0x00000000 0x0000d910 0x00000001
(lldb) me read/4xw $sp
0x16fdff310: 0x00000006 0x00000000 0x0000d910 0x00000001
(lldb) re read $x0
      x0 = 0x0000000000000006
(lldb) 
```

	0	0	0	78040000		16fdff5a0
	0	0	0	0		16fdff340
fp	6fdff5a0	1	9d63bf28	1	x29	16fdff330
	0	6	3	0		16fdff320
sp	6	0	d910	1	x9	16fdff310
	6fdff3d0	1	9d6b6396	3		16fdff300

x0		w0	6
x9	6 0 d910 1	w9	16fdff310
x8		w8	6
lr	b81f43a0	x30	100003f64