

[숙제 5] lab1 실습 내용

제출할 자료:

**실습 A :** lab1 p33(factorial-recursive)에서 하나의 loop를 아래 설명처럼 line 단위로 실행시켜 레지스터, 메모리 값을 캡처한 후 값이 변하는 부분을 체크해서 첨부하세요.

facnum 초기값은 6이라고 하자. r0을 통해 어셈블리코드에 전달된다. P32 코드 5번 줄에 breakpoint를 설정한다. r0 = 2 가 될 때까지 go한다. r0=2일 때 5번줄부터 single instruction으로 한 line씩 수행하여 loop 코드 내 21번 line이 수행될 때까지 계속 추적한다 또한, 왜 그렇게 변하는지 assembly 명령어를 해석하여 설명하시오.

**실습 B :** 실습 A 에서 8번, 9번 코드 수행 후 10번 코드 또는 14번 코드를 수행하게 되는데 어떤 상황에서 10번 코드를 수행하게 되는 지 이유를 CPSR을 이용해 설명하시오.

화면캡처 및 설명내용을 준비한 뒤 하나의 압축file로 만들어 제출하세요.

**실습 A** : lab1 p33(factorial-recursive)에서 하나의 loop를 아래 설명처럼 line 단위로 실행시켜 레지스터, 메모리 값을 캡처한 후 값이 변하는 부분을 체크해서 첨부하세요.

		memory			registers			
		0xffff050	0xffff097	4	name	value (hex)	value (decimal)	description
1	.text							
2	.global factRecursive							
3								
4	factRecursive:							
5	sub sp, sp, #8							
6	str lr, [sp, #4]							
7	str r0, [sp, #0]							
8	cmp r0, #1							
9	bge loop							
10	mov r0, #1							
11	add sp, sp, #8							
12	mov pc, lr							
13	loop:							
14	sub r0, r0, #1							
15	bl factRecursive							
16	mov r12, r0							
17	ldr r0, [sp, #0]							
18	ldr lr, [sp, #4]							
19	add sp, sp, #8							
20	mul r0, r12, r0							
21	mov pc, lr							
22	.end							
23	(end of file)							
24								

<R0가 2로 바뀐 순간>

		memory			registers			
		0xffff050	0xffff097	4	name	value (hex)	value (decimal)	description
1	.text							
2	.global factRecursive							
3								
4	factRecursive:							
5	sub sp, sp, #8							
6	str lr, [sp, #4]							
7	str r0, [sp, #0]							
8	cmp r0, #1							
9	bge loop							
10	mov r0, #1							
11	add sp, sp, #8							
12	mov pc, lr							
13	loop:							
14	sub r0, r0, #1							
15	bl factRecursive							
16	mov r12, r0							
17	ldr r0, [sp, #0]							
18	ldr lr, [sp, #4]							
19	add sp, sp, #8							
20	mul r0, r12, r0							
21	mov pc, lr							
22	.end							
23	(end of file)							
24								

bl 로 인해 분기하게 되어 pc가 변경되었다.

		memory			registers			
		0xffff050	0xffff097	4	name	value (hex)	value (decimal)	description
1	.text							
2	.global factRecursive							
3								
4	factRecursive:							
5	sub sp, sp, #8							
6	str lr, [sp, #4]							
7	str r0, [sp, #0]							
8	cmp r0, #1							
9	bge loop							
10	mov r0, #1							
11	add sp, sp, #8							
12	mov pc, lr							
13	loop:							
14	sub r0, r0, #1							
15	bl factRecursive							
16	mov r12, r0							
17	ldr r0, [sp, #0]							
18	ldr lr, [sp, #4]							
19	add sp, sp, #8							
20	mul r0, r12, r0							
21	mov pc, lr							
22	.end							
23	(end of file)							
24								

sub에 의해 sp 가 sub연산으로 8 감소 되었다. pc가 변경되었다.

str명령어로 인해 0xffff064번 메모리에 1r값이 저장되었다. pc가 변경되었다.

str명령어로 인해 0xffff060번 메모리에 2가 저장되었다. pc가 변경되었다.

cmp 명령어로 cpsr값이 변경되었지만 표시되지 않는다. pc가 변경되었다.

bge로 pc가 변경되었다.

sub로 인해 r0가 1감소했다. pc가 변경되었다.

b로 pc가 변경되었다.



	memory			registers		
				name	value (hex)	value (decimal)
1 .text	0xfffff050	0xfffff097	4			
2 .global factRecursive						
3						
4 factRecursive:	address	hex				
5 sub sp, sp, #8				r0	0x1	1
6 str lr, [sp, #4]				r1	0x0	0
7 str r0, [sp, #0]				r2	0x0	0
8 cmp r0, #1	0xfffff050	58 01 01 00		r3	0x89b20	564000
9 bge loop	0xfffff054	00 00 00 00		r4	0x10d14	68884
10 mov r0, #1	0xfffff058	00 00 00 00		r5	0x10158	65880
11 add sp, sp, #8	0xfffff05c	04 9b 08 00		r6	0x8b408	570376
12 mov pc, lr	0xfffff060	02 00 00 00		r7	0x10158	65880
13 loop:	0xfffff064	68 05 01 00		r8	0x0	0
14 sub r0, r0, #1	0xfffff068	03 00 00 00		r9	0x0	0
15 bl factRecursive	0xfffff06c	68 05 01 00		r10	0x89b04	563972
16 mov r12, r0	0xfffff070	04 00 00 00		r11	0xfffff094	4294897812
17 ldr r0, [sp, #0]	0xfffff074	68 05 01 00		r12	0x25	37
18 ldr lr, [sp, #4]	0xfffff078	05 00 00 00				
19 add sp, sp, #8	0xfffff07c	68 05 01 00		sp	0xfffff058	4294897752
20 mul r0, r12, r0	0xfffff080	06 00 00 00		lr	0x10568	66920
21 mov pc, lr	0xfffff084	b0 05 01 00		pc	0x10544	66884
22 .end						
23 (end of file)						
24						

sp 가 sub연산으로 8 감소 되었다. pc가 변경되었다.

	memory			registers		
				name	value (hex)	value (decimal)
1 .text	0xfffff050	0xfffff097	4			
2 .global factRecursive						
3						
4 factRecursive:	address	hex				
5 sub sp, sp, #8				r0	0x1	1
6 str lr, [sp, #4]				r1	0x0	0
7 str r0, [sp, #0]				r2	0x0	0
8 cmp r0, #1	0xfffff050	58 01 01 00		r3	0x89b20	564000
9 bge loop	0xfffff054	00 00 00 00		r4	0x10d14	68884
10 mov r0, #1	0xfffff058	00 00 00 00		r5	0x10158	65880
11 add sp, sp, #8	0xfffff05c	68 05 01 00		r6	0x8b408	570376
12 mov pc, lr	0xfffff060	02 00 00 00		r7	0x10158	65880
13 loop:	0xfffff064	68 05 01 00		r8	0x0	0
14 sub r0, r0, #1	0xfffff068	03 00 00 00		r9	0x0	0
15 bl factRecursive	0xfffff06c	68 05 01 00		r10	0x89b04	563972
16 mov r12, r0	0xfffff070	04 00 00 00		r11	0xfffff094	4294897812
17 ldr r0, [sp, #0]	0xfffff074	68 05 01 00		r12	0x25	37
18 ldr lr, [sp, #4]	0xfffff078	05 00 00 00				
19 add sp, sp, #8	0xfffff07c	68 05 01 00		sp	0xfffff058	4294897752
20 mul r0, r12, r0	0xfffff080	06 00 00 00		lr	0x10568	66920
21 mov pc, lr	0xfffff084	b0 05 01 00		pc	0x10548	66888
22 .end						
23 (end of file)						
24						

str명령어로 인해 0xfffff05c번 메모리에 lr값이 저장되었다. pc가 변경되었다.

	memory			registers		
				name	value (hex)	value (decimal)
1 .text	0xfffff050	0xfffff097	4			
2 .global factRecursive						
3						
4 factRecursive:	address	hex				
5 sub sp, sp, #8				r0	0x1	1
6 str lr, [sp, #4]				r1	0x0	0
7 str r0, [sp, #0]				r2	0x0	0
8 cmp r0, #1	0xfffff050	58 01 01 00		r3	0x89b20	564000
9 bge loop	0xfffff054	00 00 00 00		r4	0x10d14	68884
10 mov r0, #1	0xfffff058	01 00 00 00		r5	0x10158	65880
11 add sp, sp, #8	0xfffff05c	68 05 01 00		r6	0x8b408	570376
12 mov pc, lr	0xfffff060	02 00 00 00		r7	0x10158	65880
13 loop:	0xfffff064	68 05 01 00		r8	0x0	0
14 sub r0, r0, #1	0xfffff068	03 00 00 00		r9	0x0	0
15 bl factRecursive	0xfffff06c	68 05 01 00		r10	0x89b04	563972
16 mov r12, r0	0xfffff070	04 00 00 00		r11	0xfffff094	4294897812
17 ldr r0, [sp, #0]	0xfffff074	68 05 01 00		r12	0x25	37
18 ldr lr, [sp, #4]	0xfffff078	05 00 00 00				
19 add sp, sp, #8	0xfffff07c	68 05 01 00		sp	0xfffff058	4294897752
20 mul r0, r12, r0	0xfffff080	06 00 00 00		lr	0x10568	66920
21 mov pc, lr	0xfffff084	b0 05 01 00		pc	0x1054c	66892
22 .end						
23 (end of file)						
24						

str명령어로 인해 0xfffff058번 메모리에 1이 저장되었다. pc가 변경되었다.

cmp 명령어로 cpsr값이 변경되었지만 표시되지 않는다. pc가 변경되었다.

bge로 pc가 변경되었다.

sub 연산으로 r0가 1감소하였다. pc가 변경되었다.

				registers			
				memory			
1	.text			0xffff050	0xffff097	4	
2	.global factRecursive						
3							
4	factRecursive:						
5	sub	sp, sp, #8		name	value (hex)	value (decimal)	description
6	str	lr, [sp, #4]		r0	0x0	0	
7	str	r0, [sp, #0]		r1	0x0	0	
8	cmp	r0, #1		r2	0x0	0	
9	bge	loop		r3	0x89b20	564000	
10	mov	r0, #1		r4	0x10d14	68884	
11	add	sp, sp, #8		r5	0x10158	65880	
12	mov	pc, lr		r6	0x8b408	570376	
13	loop:			r7	0x10158	65880	
14	sub	r0, r0, #1		r8	0x0	0	register 8 (64-bit)
15	bl	factRecursive		r9	0x0	0	register 9 (64-bit)
16	mov	r12, r0		r10	0x89b04	563972	register 10 (64-bit)
17	ldr	r0, [sp, #0]		r11	0xffff094	4294897812	register 11 (64-bit)
18	ldr	lr, [sp, #4]		r12	0x25	37	register 12 (64-bit)
19	add	sp, sp, #8		sp	0xffff058	4294897752	
20	mul	r0, r12, r0		lr	0x10568	66920	
21	mov	pc, lr		pc	0x10540	66880	
22	.end						
23							
24	(end of file)						

비로 pc가 변경되었다.

				registers			
				memory			
1	.text			0xffff050	0xffff097	4	
2	.global factRecursive						
3							
4	factRecursive:						
5	sub	sp, sp, #8		name	value (hex)	value (decimal)	description
6	str	lr, [sp, #4]		r0	0x0	0	
7	str	r0, [sp, #0]		r1	0x0	0	
8	cmp	r0, #1		r2	0x0	0	
9	bge	loop		r3	0x89b20	564000	
10	mov	r0, #1		r4	0x10d14	68884	
11	add	sp, sp, #8		r5	0x10158	65880	
12	mov	pc, lr		r6	0x8b408	570376	
13	loop:			r7	0x10158	65880	
14	sub	r0, r0, #1		r8	0x0	0	register 8 (64-bit)
15	bl	factRecursive		r9	0x0	0	register 9 (64-bit)
16	mov	r12, r0		r10	0x89b04	563972	register 10 (64-bit)
17	ldr	r0, [sp, #0]		r11	0xffff094	4294897812	register 11 (64-bit)
18	ldr	lr, [sp, #4]		r12	0x25	37	register 12 (64-bit)
19	add	sp, sp, #8		sp	0xffff050	4294897744	
20	mul	r0, r12, r0		lr	0x10568	66920	
21	mov	pc, lr		pc	0x10544	66884	
22	.end						
23							
24	(end of file)						

sub명령어로 sp값이 8감소하였다. pc가 변경되었다.

				registers			
				memory			
1	.text			0xffff050	0xffff097	4	
2	.global factRecursive						
3							
4	factRecursive:						
5	sub	sp, sp, #8		name	value (hex)	value (decimal)	description
6	str	lr, [sp, #4]		r0	0x0	0	
7	str	r0, [sp, #0]		r1	0x0	0	
8	cmp	r0, #1		r2	0x0	0	
9	bge	loop		r3	0x89b20	564000	
10	mov	r0, #1		r4	0x10d14	68884	
11	add	sp, sp, #8		r5	0x10158	65880	
12	mov	pc, lr		r6	0x8b408	570376	
13	loop:			r7	0x10158	65880	
14	sub	r0, r0, #1		r8	0x0	0	register 8 (64-bit)
15	bl	factRecursive		r9	0x0	0	register 9 (64-bit)
16	mov	r12, r0		r10	0x89b04	563972	register 10 (64-bit)
17	ldr	r0, [sp, #0]		r11	0xffff094	4294897812	register 11 (64-bit)
18	ldr	lr, [sp, #4]		r12	0x25	37	register 12 (64-bit)
19	add	sp, sp, #8		sp	0xffff050	4294897744	
20	mul	r0, r12, r0		lr	0x10568	66920	
21	mov	pc, lr		pc	0x10548	66888	
22	.end						
23							
24	(end of file)						

str명령어로 인해 0xffff054번 메모리에 lr이 저장되었다. pc가 변경되었다.



str명령어로 인해 0xffff058번 메모리에 0이 저장되었다. pc가 변경되었다.

cmp 명령어로 cpsr값이 변경되었지만 표시되지 않는다. pc가 변경되었다.

branch 조건에 맞지 않아 다음 명령어로 이동한다.





				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff058	4294897752	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x1056c	66924	
				cpsr			

mov에 의해 r12에 r0의 값을 저장한다. pc가 변경된다.

				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff058	4294897752	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x10570	66928	

ldr을 통해 r0에 stack의 가장 위의 값이 저장된다. pc가 변경된다.

				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff058	4294897752	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x10574	66932	
				cpsr			

ldr에 의해 lr에 stack에서 pop한 lr이 저장된다. pc값 변경된다.

				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff060	4294897760	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x10578	66936	
				cpsr			

add에 의해 sp가 8만큼 증가한다. pc가 변경되었다.

				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff060	4294897760	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x1057c	66940	
				cpsr			

mov에 의해 r0에 r0\*r12가 저장된다. pc가 변경된다.

				registers			
memory							
0xffff050	0xffff097	4		name	value (hex)	value (decimal)	description
address	hex			r0	0x1	1	
more				r1	0x0	0	
0xffff050	00 00 00 00			r2	0x0	0	
0xffff054	68 05 01 00			r3	0x89b20	564000	
0xffff058	01 00 00 00			r4	0x10d14	68884	
0xffff05c	68 05 01 00			r5	0x10158	65880	
0xffff060	02 00 00 00			r6	0x8b408	570376	
0xffff064	68 05 01 00			r7	0x10158	65880	
0xffff068	03 00 00 00			r8	0x0	0	register 8 (64-bit)
0xffff06c	68 05 01 00			r9	0x0	0	register 9 (64-bit)
0xffff070	04 00 00 00			r10	0x89b04	563972	register 10 (64-bit)
0xffff074	68 05 01 00			r11	0xffff094	4294897812	register 11 (64-bit)
0xffff078	05 00 00 00			r12	0x1	1	register 12 (64-bit)
0xffff07c	68 05 01 00			sp	0xffff060	4294897760	
0xffff080	06 00 00 00			lr	0x10568	66920	
0xffff084	b0 05 01 00			pc	0x10568	66920	
				cpsr			

mov에 의해 pc가 lr에 저장된 값으로 변경된다.



mov에 의해 r12에 r0값을 저장한다. pc가 변경된다.

ldr에 의해 r0에 stack최상단 값을 불러와 저장한다. pc가 변경된다.

ldr에 의해 lr에 stack에서 pop한 lr값을 저장한다. pc가 변경된다.



add에 의해 sp값에 8을 더한다. pc가 변경된다.

mul에 의해 r0에  $r12 * r0$ 가 저장된다. pc가 변경된다.

mov에 의해 pc가 lr값으로 변경된다.



add에 의해 sp값에 8을 더한다. pc값이 변경된다.

mul에 의해 r0에  $r0 \cdot r12$ 의 값을 저장한다. pc값이 변경된다.

mov에 의해 pc의 값을 lr에 저장된 값으로 변경한다.



1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

```
.text
.global factRecursive

factRecursive:
    sub    sp, sp, #8
    str    lr, [sp, #4]
    str    r0, [sp, #0]
    cmp    r0, #1
    bge    loop
    mov    r0, #1
    add    sp, sp, #8
    mov    pc, lr

loop:
    sub    r0, r0, #1
    bl     factRecursive
    mov    r12, r0
    ldr    r0, [sp, #0]
    ldr    lr, [sp, #4]
    add    sp, sp, #8
    mul    r0, r12, r0
    mov    pc, lr

.end

(end of file)
```

memory

0xffff0050

0xffff097

4

address

hex

more

0xffff0050

00 00 00 00

0xffff0054

68 05 01 00

0xffff0058

01 00 00 00

0xffff005c

68 05 01 00

0xffff0060

02 00 00 00

0xffff0064

68 05 01 00

0xffff0068

03 00 00 00

0xffff006c

68 05 01 00

0xffff0070

04 00 00 00

0xffff0074

68 05 01 00

0xffff0078

05 00 00 00

0xffff007c

68 05 01 00

0xffff0080

06 00 00 00

0xffff0084

b0 05 01 00

registers

name

value (hex)

value (decimal)

description

r0

0x6

6

r1

0x0

0

r2

0x0

0

r3

0x89b20

564000

r4

0x10d14

68884

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

0xffff094

4294897812

register 11 (64-bit)

r12

0x6

6

register 12 (64-bit)

sp

0xffff070

4294897776

lr

0x10568

66920

pc

0x1056c

66924

cpsr

mov에 의해 r12에 r0의 값을 저장한다. pc값이 변경되었다.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
.text
.global factRecursive

factRecursive:
    sub    sp, sp, #8
    str    lr, [sp, #4]
    str    r0, [sp, #0]
    cmp    r0, #1
    bge    loop
    mov    r0, #1
    add    sp, sp, #8
    mov    pc, lr

loop:
    sub    r0, r0, #1
    bl     factRecursive
    mov    r12, r0
    ldr    r0, [sp, #0]
    ldr    lr, [sp, #4]
    add    sp, sp, #8
    mul    r0, r12, r0
    mov    pc, lr

.end

(end of file)
```

memory

0xffff0050

0xffff097

4

address

hex

more

0xffff0050

00 00 00 00

0xffff0054

68 05 01 00

0xffff0058

01 00 00 00

0xffff005c

68 05 01 00

0xffff0060

02 00 00 00

0xffff0064

68 05 01 00

0xffff0068

03 00 00 00

0xffff006c

68 05 01 00

0xffff0070

04 00 00 00

0xffff0074

68 05 01 00

0xffff0078

05 00 00 00

0xffff007c

68 05 01 00

0xffff0080

06 00 00 00

0xffff0084

b0 05 01 00

registers

name

value (hex)

value (decimal)

description

r0

0x4

4

r1

0x0

0

r2

0x0

0

r3

0x89b20

564000

r4

0x10d14

68884

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

0xffff094

4294897812

register 11 (64-bit)

r12

0x6

6

register 12 (64-bit)

sp

0xffff070

4294897776

lr

0x10568

66920

pc

0x10570

66928

cpsr

ldr로 stack 최상단 값을 pop해 r0에 저장한다. pc가 변경된다.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
.text
.global factRecursive

factRecursive:

    sub    sp, sp, #8
    str    lr, [sp, #4]
    str    r0, [sp, #0]
    cmp    r0, #1
    bge    loop
    mov    r0, #1
    add    sp, sp, #8
    mov    pc, lr

loop:
    sub    r0, r0, #1
    bl     factRecursive
    mov    r12, r0
    ldr    r0, [sp, #0]
    ldr    lr, [sp, #4]
    add    sp, sp, #8
    mul    r0, r12, r0
    mov    pc, lr

.end

(end of file)
```

memory

0xffff0050	0xffff097	4
address	hex	
more		
0xffff0050	00 00 00 00	
0xffff0054	68 05 01 00	
0xffff0058	01 00 00 00	
0xffff005c	68 05 01 00	
0xffff0060	02 00 00 00	
0xffff0064	68 05 01 00	
0xffff0068	03 00 00 00	
0xffff006c	68 05 01 00	
0xffff0070	04 00 00 00	
0xffff0074	68 05 01 00	
0xffff0078	05 00 00 00	
0xffff007c	68 05 01 00	
0xffff0080	06 00 00 00	
0xffff0084	b0 05 01 00	

registers

name	value (hex)	value (decimal)	description
r0	0x4	4	
r1	0x0	0	
r2	0x0	0	
r3	0x89b20	564000	
r4	0x10d14	68884	
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	0xffff094	4294897812	register 11 (64-bit)
r12	0x6	6	register 12 (64-bit)
sp	0xffff070	4294897776	
lr	0x10568	66920	
pc	0x10574	66932	
cpsr			

ldr로 stack에서 값을 pop해 lr에 저장한다. pc가 변경된다.



add로 sp의 값을 8만큼 증가시킨다. pc가 변경되었다.

mul에 의해 r0에  $r0 \cdot r12$ 값이 저장된다. pc가 변경되었다.

mov에 의해 pc에 lr이 저장된다. pc가 변경되었다.

mov에 의해 r0값이 r12에 저장된다. pc가 변경된다.

ldr에 의해 r0에 stack에서 pop된 값이 저장된다. pc가 변경된다.

ldr에 의해 lr에 stack에서 pop된 값이 저장된다. pc가 변경된다.

add에 의해 sp값이 8만큼 증가한다. pc가 변경되었다.

mul에 의해 r0에  $r0 \times r12$ 의 값이 저장된다. pc가 변경되었다.

mov에 의해 pc에 1r값이 저장된다. pc가 변경되었다.



mov에 의해 r12에 r0의 값을 저장한다. pc가 변경된다.

ldr에 의해 r0에 stack에서 pop한 값이 저장된다. pc가 변경되었다.

ldr에 의해 lr에 stack에서 pop한 값이 저장된다. pc가 변경되었다.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
.text
.global factRecursive

factRecursive:
    sub    sp, sp, #8
    str    lr, [sp, #4]
    str    r0, [sp, #0]
    cmp    r0, #1
    bge    loop
    mov     r0, #1
    add     sp, sp, #8
    mov     pc, lr

loop:
    sub     r0, r0, #1
    bl      factRecursive
    mov     r12, r0
    ldr     r0, [sp, #0]
    ldr     lr, [sp, #4]
    add     sp, sp, #8
    mul     r0, r0, r12, r0
    mov     pc, lr

.end

(end of file)
```

memory

0xfffff050	0xfffff097	4
address	hex	
more		
0xfffff050	00 00 00 00	
0xfffff054	68 05 01 00	
0xfffff058	01 00 00 00	
0xfffff05c	68 05 01 00	
0xfffff060	02 00 00 00	
0xfffff064	68 05 01 00	
0xfffff068	03 00 00 00	
0xfffff06c	68 05 01 00	
0xfffff070	04 00 00 00	
0xfffff074	68 05 01 00	
0xfffff078	05 00 00 00	
0xfffff07c	68 05 01 00	
0xfffff080	06 00 00 00	
0xfffff084	b0 05 01 00	

registers

name	value (hex)	value (decimal)	description
r0	0x6	6	
r1	0x0	0	
r2	0x0	0	
r3	0x89b20	564000	
r4	0x10d14	68884	
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	0xfffff094	4294897812	register 11 (64-bit)
r12	0x78	120	register 12 (64-bit)
sp	0xfffff088	4294897800	
lr	0x105b0	66992	
pc	0x10578	66936	
cpsr			

add에 의해 sp를 8만큼 증가시킨다. pc가 변경되었다.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24

```
.text
.global factRecursive

factRecursive:

    sub    sp, sp, #8
    str    lr, [sp, #4]
    str    r0, [sp, #0]
    cmp    r0, #1
    bge    loop
    mov     r0, #1
    add     sp, sp, #8
    mov     pc, lr

loop:
    sub     r0, r0, #1
    bl     factRecursive
    mov     r12, r0
    ldr     r0, [sp, #0]
    ldr     lr, [sp, #4]
    add     sp, sp, #8
    mul     r0, r12, r0
    mov     pc, lr

.end

(end of file)
```

memory

0xfffff050

0xfffff097

4

address

hex

more

0xfffff050

00 00 00 00

0xfffff054

68 05 01 00

0xfffff058

01 00 00 00

0xfffff05c

68 05 01 00

0xfffff060

02 00 00 00

0xfffff064

68 05 01 00

0xfffff068

03 00 00 00

0xfffff06c

68 05 01 00

0xfffff070

04 00 00 00

0xfffff074

68 05 01 00

0xfffff078

05 00 00 00

0xfffff07c

68 05 01 00

0xfffff080

06 00 00 00

0xfffff084

b0 05 01 00

registers

name

value (hex)

value (decimal)

description

r0

0x2d0

720

r1

0x0

0

r2

0x0

0

r3

0x89b20

564000

r4

0x10d14

68884

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

0xfffff094

4294897812

register 11 (64-bit)

r12

0x78

120

register 12 (64-bit)

sp

0xfffff088

4294897800

lr

0x105b0

66992

pc

0x1057c

66940

cpsr

mul에 의해 r0에 r0\*r12가 저장된다. pc가 변경되었다.

1#include <stdio.h>

2

3extern int factorial(int num);

4extern int factRecursive(int num);

5

6int main(void){

7

8 int facnum = 6;

9 int result = 0;

10

11 printf("Before: %d\n",result);

12

13 result = factRecursive(facnum);

14

15 printf("After: %d\n",result);

16

17 return 0;

18

19}

20

21

(end of file)

memory

0xfffff050	0xfffff097	4
address	hex	
more		
0xfffff050	00 00 00 00	
0xfffff054	68 05 01 00	
0xfffff058	01 00 00 00	
0xfffff05c	68 05 01 00	
0xfffff060	02 00 00 00	
0xfffff064	68 05 01 00	
0xfffff068	03 00 00 00	
0xfffff06c	68 05 01 00	
0xfffff070	04 00 00 00	
0xfffff074	68 05 01 00	
0xfffff078	05 00 00 00	
0xfffff07c	68 05 01 00	
0xfffff080	06 00 00 00	
0xfffff084	b0 05 01 00	

registers

name	value (hex)	value (decimal)	description
r0	0x2d0	720	
r1	0x0	0	
r2	0x0	0	
r3	0x89b20	564000	
r4	0x10d14	68884	
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	0xfffff094	4294897812	register 11 (64-bit)
r12	0x78	120	register 12 (64-bit)
sp	0xfffff088	4294897800	
lr	0x105b0	66992	
pc	0x105b4	66996	
cpsr			

mov에 의해 lr이 pc에 저장되었다. pc는 return한 뒤 다음 명령어의 주소를 가리키고 있다.

**실습 B :** 실습 A 에서 8번, 9번 코드 수행 후 10번 코드 또는 14번 코드를 수행하게 되는데 어떤 상황에서 10번 코드를 수행하게 되는 지 이유를 CPSR을 이용해 설명하시오.

-> 8번 코드는 `cmp r0 #1`이다. `cmp`명령어는 `sub`연산을 통해 operand간의 대소를 비교하고 뺄셈 결과 자체는 저장하지 않고, `cpsr` 내 Condition bit의 N, Z, C, V값을 갱신시킨다. 갱신된 Condition bit값은 9번 `bge loop`에 전달된다. `bge`는 앞의 `cmp`의 인수들의 비교로 분기하게 되는데 `cmp`의 1번 인수가 2번 인수보다 크거나 작으면 분기한다. 각 변수가 `unsigned`가 아닌 `signed`이므로 `cpsr`값 중 N과 Z의 EXOR연산결과가 0일 때 14번 코드로 분기하고, 1일 때 10번 코드로 분기하게 된다. N과 Z의 EXOR연산 결과가 0일 때는 r0의 값이 1미만일 때의 경우고, 1일 때는 r0의 값이 1이상일 때의 경우다.