[숙제 5] lab1 실습 내용

제출할 자료:

실습 A : lab1 p33(factorial-recursive)에서 하나의 loop를 아래 설명처럼 line 단위로 실행시켜 레지스터, 메모리 값을 캡쳐한 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.

화면캡쳐 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.

<R0가 2로 바뀐 순간>

bl 로 인해 분기하게 되어 pc가 변경되었다.

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

str명령어로 인해 0xfffef064번 메모리에 lr값이 저장되었다. pc가 변경되었다.

str명령어로 인해 0xfffef060번 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.

```
.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
```

memory 0xfffef050 0xfffef097 4

address	hex
0xfffef050	58 01 01 00
0xfffef054	00 00 00 00
0xfffef058	00 00 00 00
0xfffef05c	04 9b 08 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef058	4294897752	
lr	0x10568	66920	
pc	0x10544	66884	

sp 가 sub연산으로 8 감소 되었다. pc가 변경되었다.

memory 0xfffef050 0xfffef097 4

address	hex
0xfffef050	58 01 01 00
0xfffef054	00 00 00 00
0xfffef058	00 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef058	4294897752	
lr	0x10568	66920	
pc	0x10548	66888	

str명령어로 인해 0xfffef05c번 메모리에 lr값이 저장되었다. pc가 변경되었다.

memory 0xfffef050 0xfffef097 4

address	hex
0xfffef050	58 01 01 00
0xfffef054	00 00 00 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef058	4294897752	
lr	0x10568	66920	
pc	0x1054c	66892	

str명령어로 인해 0xfffef058번 메모리에 1이 저장되었다. pc가 변경되었다.

cmp 명령어로 cpsr값이 변경되었지만 표시되지 않는다. pc가 변경되었다.

bge로 pc가 변경되었다.

sub 연산으로 r0가 1감소하였다. pc가 변경되었다.

bl로 pc가 변경되었다.

sub명령어로 sp값이 8감소하였다. pc가 변경되었다.

str명령어로 인해 0xfffef054번 메모리에 lr이 저장되었다. pc가 변경되었다.

```
.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 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x0	0	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef050	4294897744	
lr	0x10568	66920	
pc	0x1054c	66892	

str명령어로 인해 0xfffef058번 메모리에 0이 저장되었다. pc가 변경되었다.

---

memory 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x0	0	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef050	4294897744	
lr	0x10568	66920	
pc	0x10550	66896	

cmp 명령어로 cpsr값이 변경되었지만 표시되지 않는다. pc가 변경되었다.

---

memory 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x0	0	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x25	37	register 12 (64-bit)
sp	0xfffef050	4294897744	
lr	0x10568	66920	
pc	0x10554	66900	

branch 조건에 맞지 않아 다음 명령어로 이동한다.

mov에 의해 r0 값이 1로 변경되었다. pc가 변경되었다.

add에 의해 sp가 증가했다. pc가 변경되었다.

mov에 의해 pc가 lr에 저장된 값으로 변경된다.

```
.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

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x1	1	register 12 (64-bit)
sp	0xfffef058	4294897752	
lr	0x10568	66920	
pc	0x1056c	66924	
cpsr			

mov에 의해 r12에 r0의 값을 저장한다. pc가 변경된다.

ldr을 통해 r0에 stack의 가장 위의 값이 저장된다. pc가 변경된다. (pc: 0x10570, 66928)

ldr에 의해 lr에 stack에서 pop한 lr이 저장된다. pc값 변경된다. (pc: 0x10574, 66932)

add에 의해 sp가 8만큼 증가한다. pc가 변경되었다.

mov에 의해 r0에 r0*r12가 저장된다. pc가 변경된다.

mov에 의해 pc가 lr에 저장된 값으로 변경된다.

```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x1	1	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x1	1	register 12 (64-bit)
sp	0xfffef060	4294897760	
lr	0x10568	66920	
pc	0x1056c	66924	
cpsr			

mov에 의해 r12에 r0값을 저장한다. pc가 변경된다.


```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x2	2	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x1	1	register 12 (64-bit)
sp	0xfffef060	4294897760	
lr	0x10568	66920	
pc	0x10570	66928	
cpsr			

ldr에 의해 r0에 stack최상단 값을 불러와 저장한다. pc가 변경된다.


```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x2	2	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x1	1	register 12 (64-bit)
sp	0xfffef060	4294897760	
lr	0x10568	66920	
pc	0x10574	66932	
cpsr			

ldr에 의해 lr에 stack에서 pop한 lr값을 저장한다. pc가 변경된다.

add에 의해 sp값에 8을 더한다. pc가 변경된다.

mul에 의해 r0에 r12*r0가 저장된다. pc가 변경된다.

mov에 의해 pc가 lr값으로 변경된다.

```
.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 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x2	2	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef068	4294897768	
lr	0x10568	66920	
pc	0x1056c	66924	
cpsr			

mov에 의해 r12에 r0의 값을 저장한다. pc값이 변경된다.

registers

name	value (hex)	value (decimal)	description
r0	0x3	3	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef068	4294897768	
lr	0x10568	66920	
pc	0x10570	66928	

ldr에 의해 r0에 stack 최상단 값을 저장한다. pc값이 변경된다.

registers

name	value (hex)	value (decimal)	description
r0	0x3	3	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef068	4294897768	
lr	0x10568	66920	
pc	0x10574	66932	
cpsr			

ldr에 의해 lr에 stack에서 pop한 lr값을 저장한다. pc가 변경된다.

```
.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
```

memory — 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x3	3	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef070	4294897776	
lr	0x10568	66920	
pc	0x10578	66936	
cpsr			

add에 의해 sp값에 8을 더한다. pc값이 변경된다.

```
.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
```

memory — 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef070	4294897776	
lr	0x10568	66920	
pc	0x1057c	66940	

mul에 의해 r0에 r0*r12의 값을 저장한다. pc값이 변경된다.

```
.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
```

memory — 0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x2	2	register 12 (64-bit)
sp	0xfffef070	4294897776	
lr	0x10568	66920	
pc	0x10568	66920	
cpsr			

mov에 의해 pc의 값을 lr에 저장된 값으로 변경한다.

mov에 의해 r12에 r0의 값을 저장한다. pc값이 변경되었다.

ldr로 stack 최상단 값을 pop해 r0에 저장한다. pc가 변경된다.

ldr로 stack에서 값을 pop해 lr에 저장한다. pc가 변경된다.

```
.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
0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x6	6	register 12 (64-bit)
sp	0xfffef078	4294897784	
lr	0x10568	66920	
pc	0x10578	66936	
cpsr			

add로 sp의 값을 8만큼 증가시킨다. pc가 변경되었다.

```
.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
0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x18	24	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x6	6	register 12 (64-bit)
sp	0xfffef078	4294897784	
lr	0x10568	66920	
pc	0x1057c	66940	
cpsr			

mul에 의해 r0에 r0*r12값이 저장된다. pc가 변경되었다.

```
.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
0xfffef050 0xfffef097 4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x18	24	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x6	6	register 12 (64-bit)
sp	0xfffef078	4294897784	
lr	0x10568	66920	
pc	0x10568	66920	
cpsr			

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.

```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x5	5	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x18	24	register 12 (64-bit)
sp	0xfffef080	4294897792	
lr	0x10568	66920	
pc	0x10578	66936	
cpsr			

add에 의해 sp값이 8만큼 증가한다. pc가 변경되었다.

```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x78	120	
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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x18	24	register 12 (64-bit)
sp	0xfffef080	4294897792	
lr	0x10568	66920	
pc	0x1057c	66940	

mul에 의해 r0에 r0*r12의 값이 저장된다. pc가 변경되었다.

```
.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

0xfffef050	0xfffef097	4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	b0 05 01 00

registers

name	value (hex)	value (decimal)	description
r0	0x78	120	
r1	0x0	0	
r2	0x8	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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x18	24	register 12 (64-bit)
sp	0xfffef080	4294897792	
lr	0x10568	66920	
pc	0x10568	66920	
cpsr			

mov에 의해 pc에 lr값이 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.

```
.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

0xfffef050   0xfffef097   4

address	hex
more	
0xfffef050	00 00 00 00
0xfffef054	68 05 01 00
0xfffef058	01 00 00 00
0xfffef05c	68 05 01 00
0xfffef060	02 00 00 00
0xfffef064	68 05 01 00
0xfffef068	03 00 00 00
0xfffef06c	68 05 01 00
0xfffef070	04 00 00 00
0xfffef074	68 05 01 00
0xfffef078	05 00 00 00
0xfffef07c	68 05 01 00
0xfffef080	06 00 00 00
0xfffef084	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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x78	120	register 12 (64-bit)
sp	0xfffef088	4294897800	
lr	0x105b0	66992	
pc	0x10578	66936	
cpsr			

add에 의해 sp를 8만큼 증가시킨다. pc가 변경되었다.

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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x78	120	register 12 (64-bit)
sp	0xfffef088	4294897800	
lr	0x105b0	66992	
pc	0x1057c	66940	
cpsr			

mul에 의해 r0에 r0*r12가 저장된다. pc가 변경되었다.

```
#include <stdio.h>

extern int factorial(int num);
extern int factRecursive(int num);

int main(void){

        int facnum = 6;
        int result = 0;

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

        result = factRecursive(facnum);

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

        return 0;
}

(end of file)
```

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	0xfffef094	4294897812	register 11 (64-bit)
r12	0x78	120	register 12 (64-bit)
sp	0xfffef088	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 #1이다. cmp명령어는 sub연산을 통해 operand간의 대소를 비교하고 뺄셈 결과 자체는 저장하지 않고, cpsr 내 Condition bit의 N, Z, C, V값을 갱신시킨다. 갱신된 Condition bit값은 9번 bge loop에 전달된다. bge는 앞의 cmp의 인수들의 비교로 분기하게 되는데 cmp의 1번 인수가 2번 인수보다 크거나 작으면 분기한다. 각 변수가 unsigned가 아닌 signed이므로 cpsr값 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.