2. C 서브루틴 코드 분석

```
.globl _main
            .def _main; .scl 2; .type 32; .endef
main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
                         $-16, %esp
            andl
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
            movl
                         %eax, 4(%esp)
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
                         %eax, (%edx)
            movl
                         $0, %eax
            movl
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
```

LFE2:

main 함수부터 시작



	STACK
-4	
-8	
-12	
-16	
-20	
-24	
-28	
-32	
-36	
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	
	-8 -12 -16 -20 -24 -28 -32 -36 -40 -44 -48 -52 -56 -60 -64 -68 -72 -76 -80 -84 -88 -92 -96

REGISTERS			
eax			
ebx			
есх			
edx			
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

스택 포인터, 베이스 포인터 초기화



	STACK
-4	
-8	
-12	
-16	
-20	
-24	
-28	
-32	
-36	
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	

REGISTERS			
eax			
ebx			
есх			
edx			
eex			

	.globl _maii .def mair	n n; .scl 2; .type 32; .endef	
main: LFB2:		,, 156, 13, pc 5, 16.16.6.	
	.cfi_startprod	2	
	pushl .	%ebp	
	.cfi_def_cfa_d	offset 8	
	.cfi_offset 5,	-8	
	movl	%esp, %ebp	
	.cfi_def_cfa_ı	register 5	
	andl	\$-16, %esp	
	subl	\$32, %esp	
	call	main	
	movl	\$10, 28(%esp)	
	movl	\$10, 24(%esp)	
	movl	\$5, 20(%esp)	
	movl	20(%esp), %eax	
	movl	%eax, 4(%esp)	
	movl	24(%esp), %eax	
	movl	%eax, (%esp)	
	call	_add	
	movl	%eax, 4(%esp)	
	movl	28(%esp), %eax	
	movl call	%eax, (%esp) mul	
	movl	_mui 16(%esp), %edx	
	movl	%eax, (%edx)	
	movl	\$0, %eax	
	leave	30, 70eax	
	.cfi restore 5	_	
	.cfi def cfa 4, 4		
	ret	.,	
	.cfi_endproc		
LFE2:	0		

로컬 변수를 위한 공간 확보

0/ 1		STACK
%ebp	-4	
	-8	
	-12	
	-16	
	-20	
	-24	
	-28	
0/ 000	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax			
ebx			
есх			
edx			
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
            movl
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
            movl
                         %eax, 4(%esp)
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

__main 은 비어있음

0/ 1		STACK
%ebp	-4	
	-8	
	-12	
	-16	
	-20	
	-24	
	-28	
0/ 050	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax			
ebx			
есх			
edx			
eex			

```
.globl _main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                           main
                         $10, 28(%esp)
            movl
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
                         20(%esp), %eax
            movl
                         %eax, 4(%esp)
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

int a = 10;

		STACK	
%ebp	-4	1	0
	-8		
	-12		
	-16		
	-20		
	-24		
	-28		
0/	-32		
%esp	-36		
	-40		
	-44		
	-48		
	-52		
	-56		
	-60		
	-64		
	-68		
	-72		
	-76		
	-80		
	-84		
	-88		
	-92		
	-96		
	-100		

REGISTERS			
eax			
ebx			
есх			
edx			
eex			

.globl _main .def _main; .scl 2; .type 32; .en _main: LFB2:	def
.cfi_startproc	
pushl %ebp	
.cfi def cfa offset 8	
.cfi offset 5, -8	
movl %esp, %ebp	
.cfi_def_cfa_register 5	
andl \$-16, %esp	
subl \$32, %esp	
callmain	
movl \$10, 28(%esp)	
movl \$10, 24(%esp)	
movl \$5, 20(%esp)	
movl 20(%esp), %eax	
movl %eax, 4(%esp)	
movl 24(%esp), %eax movl %eax, (%esp)	
call add	
movl %eax, 4(%esp)	
movl 28(%esp), %eax	
movl %eax, (%esp)	
call mul	
movl 16(%esp), %edx	
movl %eax, (%edx)	
movl \$0, %eax	
leave	
.cfi_restore 5	
.cfi_def_cfa 4, 4	
ret	
.cfi_endproc	
LFE2:	

int b = 10;

0/ alan		STACK	
%ebp	-4	10)
	-8	10)
	-12		
	-16		
	-20		
	-24		
	-28		
0/ 000	-32		
%esp	-36		
	-40		
	-44		
	-48		
	-52		
	-56		
	-60		
	-64		
	-68		
	-72		
	-76		
	-80		
	-84		
	-88		
	-92		
	-96		
	-100		

REGISTERS		
eax		
ebx		
есх		
edx		
eex		

.globl _main .def _main; .scl 2; .type 32; .endef
_main: LFB2:
.cfi_startproc
pushl %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
andl \$-16, %esp
subl \$32, %esp
callmain
movl \$10, 28(%esp)
movl \$10, 24(%esp)
movl \$5, 20(%esp) movl 20(%esp), %eax
movl 20(%esp), %eax movl %eax, 4(%esp)
movl 24(%esp), %eax
movl %eax, (%esp)
call add
movl %eax, 4(%esp)
movl 28(%esp), %eax
movl %eax, (%esp)
call _mul _
movl 16(%esp), %edx
movl %eax, (%edx)
movl \$0, %eax
leave
.cfi_restore 5
.cfi_def_cfa 4, 4
ret
.cfi_endproc LFE2:

int c = 5;

0/ -1		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	
0/ 0.00	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax		
ebx		
есх		
edx		
eex		

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
                         20(%esp), %eax
            movl
                         %eax, 4(%esp)
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
                         %eax, (%esp)
            movl
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

0/ alan		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	
0/ 050	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	5		
ebx			
есх			
edx			
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
            movl
                         $10, 24(%esp)
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
                         %eax, (%esp)
            movl
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

0/ 0 0 0		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/ 050	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	5		
ebx			
есх			
edx			
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
            movl
                         $5, 20(%esp)
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
            movl
                         %eax, (%esp)
            call
                         add
            movl
                         %eax, 4(%esp)
            movl
                         28(%esp), %eax
                         %eax, (%esp)
            movl
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

0/ 0 0 0		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/ 050	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	10		
ebx			
есх			
edx			
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                        %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
            movl
                         $5, 20(%esp)
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
                         add
            call
            movl
                         %eax, 4(%esp)
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

0/ alan		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/ 050	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS				
eax	10			
ebx				
есх				
edx				
eex				

.globl _main
.def _main; .scl 2; .type 32; .endef main:
LFB2:
.cfi_startproc
pushl %ebp
.cfi def cfa offset 8
.cfi_offset 5, -8
movl %esp, %ebp
.cfi_def_cfa_register 5
andl \$-16, %esp
subl \$32, %esp
callmain
movl \$10, 28(%esp)
movl \$10, 24(%esp)
movl \$5, 20(%esp)
movl 20(%esp), %eax movl %eax, 4(%esp)
movl %eax, 4(%esp) movl 24(%esp), %eax
movl %eax, (%esp)
call add
movl %eax, 4(%esp)
movl 28(%esp), %eax
movl %eax, (%esp)
call _mul
movl 16(%esp), %edx
movl %eax, (%edx)
movl \$0, %eax
leave .cfi restore 5
.cfi def cfa 4, 4
ret
.cfi_endproc
LFE2:

add(b, c) 수행 시작

% obn		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS				
eax	10			
ebx				
есх				
edx				
eex				

add(b, c) 수행 시작

0/ alas		STACK
%ebp	-4	
	-8	
	-12	
	-16	
	-20	
	-24	
	-28	
9/ osp	-32	
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	

-100

10 10 5

5

10

REGISTERS			
eax			10
ebx			
есх			
edx			
eex			

.globl	_add
.def	add; .scl 2; .type 32; .endef

_add: LFB0:

> .cfi_startproc pushl

%ebp .cfi_def_cfa_offset 8

.cfi_offset 5, -8

movl %esp, %ebp

.cfi_def_cfa_register 5

subl \$16, %esp

8(%ebp), %edx movl 12(%ebp), %eax movl addl %edx, %eax

%eax, -4(%ebp) movl -4(%ebp), %eax

movl

leave

.cfi_restore 5 .cfi_def_cfa 4, 4

ret

.cfi_endproc

LFE0:

```
.globl _add
            .def _add; .scl 2; .type 32; .endef
_add:
LFB0:
            .cfi startproc
                         %ebp
            pushl
            .cfi_def_cfa_offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            subl
                         $16, %esp
                         8(%ebp), %edx
            movl
                         12(%ebp), %eax
            movl
                         %edx, %eax
            addl
                         %eax, -4(%ebp)
            movl
            movl
                         -4(%ebp), %eax
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE0:
```

이전 base pointer 위치 저장

0/ alan		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
9/ ocn	-36	(former base pointer)
%esp	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS				
eax			10	
ebx				
есх				
edx				
eex				

```
.globl _add
            .def _add; .scl 2; .type 32; .endef
_add:
LFB0:
            .cfi_startproc
                         %ebp
            pushl
            .cfi_def_cfa_offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            subl
                         $16, %esp
                         8(%ebp), %edx
            movl
                         12(%ebp), %eax
            movl
                         %edx, %eax
            addl
                         %eax, -4(%ebp)
            movl
            movl
                         -4(%ebp), %eax
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE0:
```

Base pointer 끌어오기

	STACK
-4	10
-8	10
-12	5
-16	
-20	
-24	
-28	5
-32	10
-36	(former base pointer)
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	

REGISTERS			
eax	10		
ebx			
есх			
edx			
eex			

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi def cfa register 5 subl \$16, %esp 8(%ebp), %edx movl movl 12(%ebp), %eax %edx, %eax addl %eax, -4(%ebp) movl -4(%ebp), %eax movl leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

함수 내 로컬 변수 공간 확보

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
20	-36	(former base pointer)
ob	-40	
	-44	
	-48	
-n	-52	
sp	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

R	EGISTERS
eax	10
ebx	
есх	
edx	
eex	

첫 번째 인자 가져오기

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
0/ ohn	-36	(former base pointer)
%ebp	-40	
	-44	
	-48	
%esp	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

F	REGISTERS
eax	10
ebx	
есх	
edx	10
eex	

_add:	.globl _add .def _add;	.scl 2; .type 32; .endef
LFB0:	.cfi_startproc	
	pushl	%ebp
	.cfi_def_cfa_c .cfi_offset_5,	
	movl	%esp, %ebp
	.cfi_def_cfa_r	egister 5
	subl	\$16, %esp
	movl	8(%ebp), %edx
	movl addl	12(%ebp), %eax %edx, %eax
	movl	%eax, -4(%ebp)
	movl	-4(%ebp), %eax
	leave	
	.cfi_restore 5	
	.cfi_def_cfa 4 ret	t, 4
	.cfi_endproc	
LFE0:	·	

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl addl %edx, %eax %eax, -4(%ebp) movl -4(%ebp), %eax movl leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

두 번째 인자 가져오기

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
2/ obp	-36	(former base pointer)
%ebp	-40	
	-44	
	-48	
2/ocn	-52	
%esp	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	5		
ebx			
есх			
edx	10		
eex			

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl addl %edx, %eax movl %eax, -4(%ebp) movl -4(%ebp), %eax leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

두 인자의 값 더하기

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
0/ obn	-36	(former base pointer)
%ebp	-40	
	-44	
% osp	-48	
	-52	
%esp	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

R	EGISTERS
eax	15
ebx	
есх	
edx	10
eex	

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl %edx, %eax addl %eax, -4(%ebp) movl movl -4(%ebp), %eax leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

계산 결과 스택에 가져오기

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
0/ alau	-36	(former base pointer)
%ebp	-40	
	-44	15
	-48	
2/ 0510	-52	
%esp	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS				
eax		15		
ebx				
есх				
edx		10		
eex				

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl %edx, %eax addl movl %eax, -4(%ebp) -4(%ebp), %eax movl leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

반환 값 레지스터에 저장하기

		STACK
	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
	-32	10
0/ alau	-36	(former base pointer)
%ebp	-40	
	-44	15
	-48	
2/ 0510	-52	
%esp	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

	RE	GISTERS	
eax			15
ebx			
есх			
edx			10
eex			

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 %esp, %ebp movl .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl %edx, %eax addl %eax, -4(%ebp) movl movl -4(%ebp), %eax leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

베이스 포인터 회수

		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/ 2010	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

R	EGISTERS
eax	15
ebx	
есх	
edx	10
eex	

.globl _add .def _add; .scl 2; .type 32; .endef _add: LFB0: .cfi_startproc %ebp pushl .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 subl \$16, %esp 8(%ebp), %edx movl 12(%ebp), %eax movl %edx, %eax addl %eax, -4(%ebp) movl movl -4(%ebp), %eax leave .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE0:

함수 실행 종료

0/ 1		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	5
0/ 000	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	15		
ebx			
есх			
edx	10		
eex			

cfi_startproc pushl %ebp cfi_def_cfa_offset 8 cfi_offset 5, -8 movl %esp, %ebp cfi_def_cfa_register 5 andl \$-16, %esp subl \$32, %esp callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl \$0(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 andl \$-16, %esp subl \$32, %esp callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl \$0(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
cfi_def_cfa_offset 8 cfi_offset 5, -8 movl %esp, %ebp cfi_def_cfa_register 5 andl \$-16, %esp subl \$32, %esp callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl \$0(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
.cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 andl \$-16, %esp subl \$32, %esp call main movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
movl %esp, %ebp .cfi_def_cfa_register 5 andl andl \$-16, %esp subl \$32, %esp call main movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
.cfi_def_cfa_register 5 andl \$-16, %esp subl \$32, %esp call main movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
andl \$-16, %esp subl \$32, %esp callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
subl \$32, %esp callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
callmain movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
movl \$10, 28(%esp) movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
movl \$10, 24(%esp) movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
movl \$5, 20(%esp) movl 20(%esp), %eax movl %eax, 4(%esp) movl 24(%esp), %eax
movl %eax, 4(%esp) movl 24(%esp), %eax
movl 24(%esp), %eax
movl %eax, (%esp)
call _add
movl %eax, 4(%esp)
movl 28(%esp), %eax
movl %eax, (%esp) call mul
movl 16(%esp), %edx
movl %eax, (%edx)
movl \$0, %eax
leave
.cfi restore 5
.cfi_def_cfa 4, 4
ret
.cfi_endproc
LFE2:

함수 수행 결과 가져오기

% obn		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	15		
ebx			
есх			
edx	10		
eex			

_main:	.globl _mai .def _mair	n n; .scl 2; .type 32; .endef
LFB2:		
	.cfi_startpro	С
	pushl	%ebp
	.cfi_def_cfa_	offset 8
	.cfi_offset 5,	-8
	movl	%esp, %ebp
	.cfi_def_cfa_	register 5
	andl	\$-16, %esp
	subl	\$32, %esp
	call	main
	movl	\$10, 28(%esp)
	movl	\$10, 24(%esp)
	movl	\$5, 20(%esp)
	movl	20(%esp), %eax
	movl	%eax, 4(%esp)
	movl	24(%esp), %eax
	movl	%eax, (%esp)
	call	_add
	movl	%eax, 4(%esp)
	movl	28(%esp), %eax
	movl call	%eax, (%esp) mul
	movl	_mui 16(%esp), %edx
	movl	%eax, (%edx)
	movl	\$0, %eax
	leave	30, 70eax
	.cfi_restore !	_
	.cfi def cfa	
	ret	','
	.cfi_endproc	
LFE2:	cap.oc	

mul 실행을 위한 인자 세팅

0/ alan		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 050	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	10		
ebx			
есх			
edx	10		
eex			

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
            movl
                         $5, 20(%esp)
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
                         %eax, (%esp)
            movl
            call
                         mul
            movl
                         16(%esp), %edx
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi_restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

mul 실행을 위한 인자 세팅

0/ -1		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 0.515	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	10		
ebx			
есх			
edx	10		
eex			

main:	.globl _mai .def _mair	n n; .scl 2; .type 32; .endef
LFB2:		
	.cfi_startpro	С
	pushl '	%ebp
	.cfi def cfa	offset 8
	.cfi_offset 5,	-8
	movl	%esp, %ebp
	.cfi_def_cfa_	register 5
	andl	\$-16, %esp
	subl	\$32, %esp
	call	main
	movl	\$10, 28(%esp)
	movl	\$10, 24(%esp)
	movl	\$5, 20(%esp)
	movl	20(%esp), %eax
	movl	%eax, 4(%esp)
	movl	24(%esp), %eax
	movl	%eax, (%esp)
	call	_add
	movl	%eax, 4(%esp)
	movl	28(%esp), %eax
	<u>movl</u> call	%eax, (%esp) mul
	movl	_mui 16(%esp), %edx
	movl	%eax, (%edx)
	movl	\$0, %eax
	leave	90, 70Cax
	.cfi restore !	
	.cfi def cfa	
	ret	.,
	.cfi_endproc	
LFE2:		

mul 수행 시작

0/ alan		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 050	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

R	EGISTERS
eax	10
ebx	
есх	
edx	10
eex	

mul 수행 시작

0/ -		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 0.515	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

R	EGISTERS
eax	10
ebx	
есх	
edx	10
eex	

.globl .def	_mul mul;	.scl
2;	.type	32;
.endef		

_mul: LFB1:

.cfi_startproc

pushl %ebp .cfi_def_cfa_offset 8

.cfi_offset 5, -8

movl %esp, %ebp

.cfi_def_cfa_register 5

movl 8(%ebp), %eax imull 12(%ebp), %eax

popl %ebp

.cfi_restore 5 .cfi_def_cfa 4, 4

ret

.cfi_endproc

LFE1:

.globl .def 2; _mul; _mul; .type .scl 32; .endef

_mul: LFB1:

LFE1:

.cfi_startproc

ten_startprot	
pushl	%ebp
.cfi_def_cfa_c .cfi_offset 5,	
movl	%esp, %ebp
.cfi_def_cfa_r	egister 5
movl	8(%ebp), %eax
imull	12(%ebp), %eax
popl	%ebp
.cfi_restore 5	
.cfi_def_cfa 4	·, 4
ret	
.cfi_endproc	

이전 base pointer 위치 저장

0/ alau		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
	-32	10
0/ ocn	-36	(former base pointer)
%esp	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS			
eax	10		
ebx			
есх			
edx	10		
eex			

.globl _mul .def _mul; .scl 32; .type .endef _mul: LFB1: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi offset 5, -8 %esp, %ebp movl .cfi_def_cfa_register 5 8(%ebp), %eax movl imull 12(%ebp), %eax popl %ebp .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE1:

Base pointer 끌어오기

%ebp

	STACK
-4	10
-8	10
-12	5
-16	
-20	
-24	
-28	15
-32	10
-36	(former base pointer)
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	

REGISTERS		
eax	10	0
ebx		
есх		
edx	10	0
eex		

.globl _mul .def .scl 32; _mul; .type .endef _mul: LFB1: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 8(%ebp), %eax movl 12(%ebp), %eax imull popl %ebp .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE1:

함수 인자 가져오기

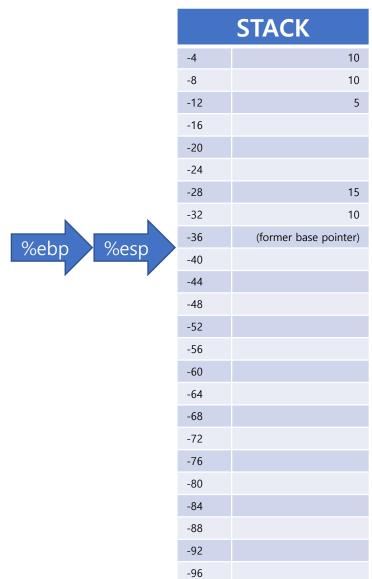
%ebp

	STACK
-4	10
-8	10
-12	5
-16	
-20	
-24	
-28	15
-32	10
-36	(former base pointer)
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	

REGISTERS		
eax	10	
ebx		
есх		
edx	10	
eex		

.globl _mul .def _mul; .scl 32; .type .endef _mul: LFB1: .cfi_startproc pushl %ebp .cfi_def_cfa_offset 8 .cfi_offset 5, -8 movl %esp, %ebp .cfi_def_cfa_register 5 8(%ebp), %eax movl imull 12(%ebp), %eax popl %ebp .cfi_restore 5 .cfi_def_cfa 4, 4 ret .cfi_endproc LFE1:

곱 연산 수행



-100

REGISTERS		
eax		120
ebx		
есх		
edx		10
eex		

```
.globl
                          _mul
             .def
                          _mul;
                                       .scl
32;
                          .type
             .endef
_mul:
LFB1:
             .cfi_startproc
             pushl
                          %ebp
             .cfi_def_cfa_offset 8
             .cfi_offset 5, -8
             movl
                          %esp, %ebp
             .cfi_def_cfa_register 5
             movl
                          8(%ebp), %eax
             imull
                          12(%ebp), %eax
                          %ebp
             popl
             .cti_restore 5
             .cfi_def_cfa 4, 4
             ret
             .cfi_endproc
LFE1:
```

Base pointer 회수

0/ obr		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 2010	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax	120	
ebx		
есх		
edx	10	
eex		

함수 실행 종료

.globl

.endef

pushl

movl

movl

imull

popl

ret

.cfi_restore 5 .cfi_def_cfa 4, 4

.cfi_endproc

.cfi_startproc

.cfi_def_cfa_offset 8 .cfi_offset 5, -8

.cfi_def_cfa_register 5

_mul: LFB1:

LFE1:

.def

_mul

_mul;

.type

%ebp

%esp, %ebp

12(%ebp), %eax %ebp

8(%ebp), %eax

.scl 32;

0(STACK
%ebp	-4	10
	-8	10
	-12	<u>!</u>
	-16	
	-20	
	-24	
	-28	15
0/	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax	120	
ebx		
есх		
edx	10	
eex		

	.globl _mai	n
	_	n; .scl 2; .type 32; .endef
main:	.acı _man	1, .5ci 2, .type 52, .eriaei
LFB2:		
	.cfi_startprod	2
	pushl	%ebp
	.cfi_def_cfa_	•
	.cfi_offset 5,	
	movl	%esp, %ebp
	.cfi_def_cfa_	register 5
	andl	\$-16, %esp
	subl	\$32, %esp
	call	main
	movl	\$10, 28(%esp)
	movl	\$10, 24(%esp)
	movl	\$5, 20(%esp)
	movl	20(%esp), %eax
	movl	%eax, 4(%esp)
	movl	24(%esp), %eax
	movl	%eax, (%esp)
	call	_add
	movl	%eax, 4(%esp)
	movl movl	28(%esp), %eax
	call	%eax, (%esp) mul
	movl	16(%esp), %edx
	movl	%eax, (%edx)
	movl	\$0, %eax
	leave	
	.cfi_restore 5	
	.cfi_def_cfa 4	1, 4
	ret	
	.cfi_endproc	
LFE2:		

int *ret 주소 가져오기

0/ -1		STACK
%ebp	-4	10
,	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 050	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax	120	
ebx		
есх		
edx	(address)	
eex		

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
            movl
                         $10, 24(%esp)
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
                         %eax, (%edx)
            movl
            movl
                         $0, %eax
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

함수 수행 결과를 ret의 주소에 저장

0/ ob n		STACK
%ebp	-4	10
	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 0.510	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax	120	
ebx		
есх		
edx	(address)	
eex		

```
.globl _main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
                         %eax, (%edx)
            movl
                         $0, %eax
            movl
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

main 함수의 반환값으로 0 설정

0/ -1		STACK
%ebp	-4	10
,	-8	10
	-12	5
	-16	
	-20	
	-24	
	-28	15
0/ 050	-32	10
%esp	-36	
	-40	
	-44	
	-48	
	-52	
	-56	
	-60	
	-64	
	-68	
	-72	
	-76	
	-80	
	-84	
	-88	
	-92	
	-96	
	-100	

REGISTERS		
eax	0	
ebx		
есх		
edx	(address)	
eex		

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
            movl
                         $10, 24(%esp)
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cfi_endproc
LFE2:
```

Base pointer 회수



	STACK
-4	
-8	
-12	
-16	
-20	
-24	
-28	
-32	
-36	
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	

REGISTERS		
eax	0	
ebx		
есх		
edx	(address)	
eex		

```
.globl main
            .def _main; .scl 2; .type 32; .endef
_main:
LFB2:
            .cfi_startproc
            pushl
                         %ebp
            .cfi def cfa offset 8
            .cfi_offset 5, -8
            movl
                         %esp, %ebp
            .cfi_def_cfa_register 5
            andl
                         $-16, %esp
            subl
                         $32, %esp
            call
                         ___main
                         $10, 28(%esp)
            movl
                         $10, 24(%esp)
            movl
                         $5, 20(%esp)
            movl
            movl
                         20(%esp), %eax
                         %eax, 4(%esp)
            movl
                         24(%esp), %eax
            movl
                         %eax, (%esp)
            movl
            call
                         add
                         %eax, 4(%esp)
            movl
            movl
                         28(%esp), %eax
            movl
                         %eax, (%esp)
            call
                         mul
                         16(%esp), %edx
            movl
            movl
                         %eax, (%edx)
                         $0, %eax
            movl
            leave
            .cfi restore 5
            .cfi_def_cfa 4, 4
            ret
            .cti_endproc
LFE2:
```

main 함수 실행 종료



	STACK
-4	
-8	
-12	
-16	
-20	
-24	
-28	
-32	
-36	
-40	
-44	
-48	
-52	
-56	
-60	
-64	
-68	
-72	
-76	
-80	
-84	
-88	
-92	
-96	
-100	
-80 -84 -88 -92 -96	

REGISTERS		
eax	0	
ebx		
есх		
edx	(address)	
eex		