EDDI
Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 5기

2023. 05. 19

최성효

터미널 명령어 정리

EDDI
Electronic Design Development Institute

ls : 현재 디렉토리 파일 확인(list)
cd : change directory
Pwd :print working directory
Mkdir : make directory
rm : remove (rm -rf (reculsive force)/ -r:하위디렉토리까지 삭제 /-f :강제삭제

gcc [소스파일] : 실행파일 a.out
gcc -o [저장할 실행파일이름][소스파일]
gcc -g : 디버깅 옵션 플래그
Ex) gcc -g -o main main.c

gdb 명령어
r: run
b: break point / Ex: b main/ b *메모리주소
Disas : disassembly (어셈블리어 확인
Si : 다음 어셈블리 실행

어셈블리어 분석(1)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x000055555555515b <+0>:     endbr64
=> 0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

다음에 실행될 부분

레지스터

rbp	0x1	0x1
rsp	0x7fffffffde98	0x7fffffffde98

어셈블리어 분석(2)

EDDI
Electronic Design Development Institute

어셈블리어

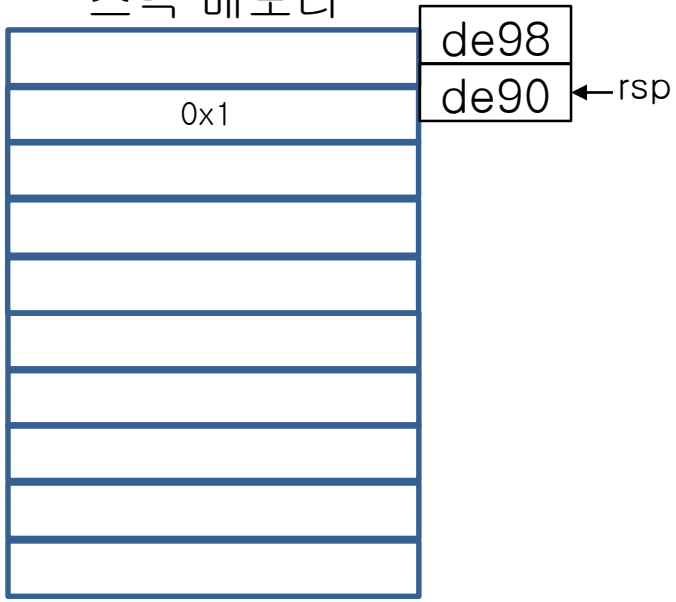
```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
=> 0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

현재 실행된 부분

레지스터

rbp	0x1	0x1
rsp	0x7fffffffde90	0x7fffffffde90

스택 메모리



Push는 현재 rsp의 한칸 증가시키고 rbp값 저장

어셈블리어 분석(3)

EDDI
Electronic Design Development Institute

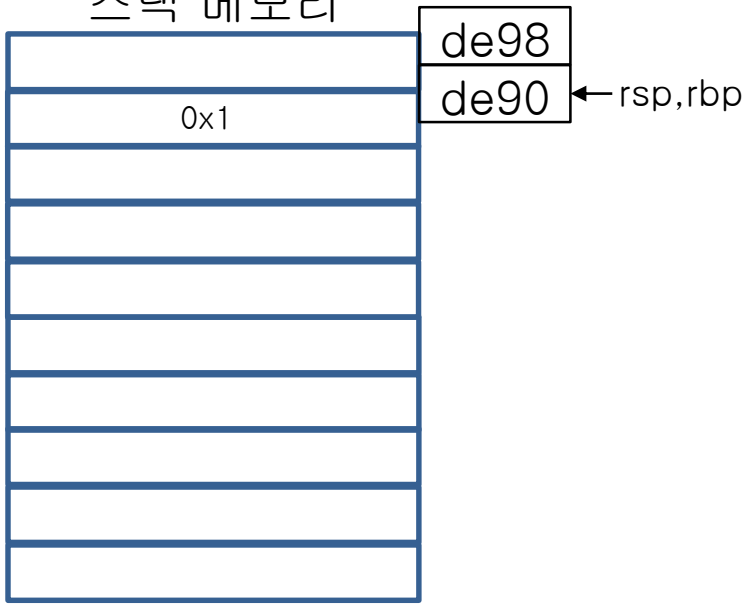
어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
=> 0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde90      0x7fffffffde90
```

스택 메모리



mov는 rbp에 rsp 주소 대입
rsp=rbp로 같게한다

어셈블리어 분석(4)

EDDI
Electronic Design Development Institute

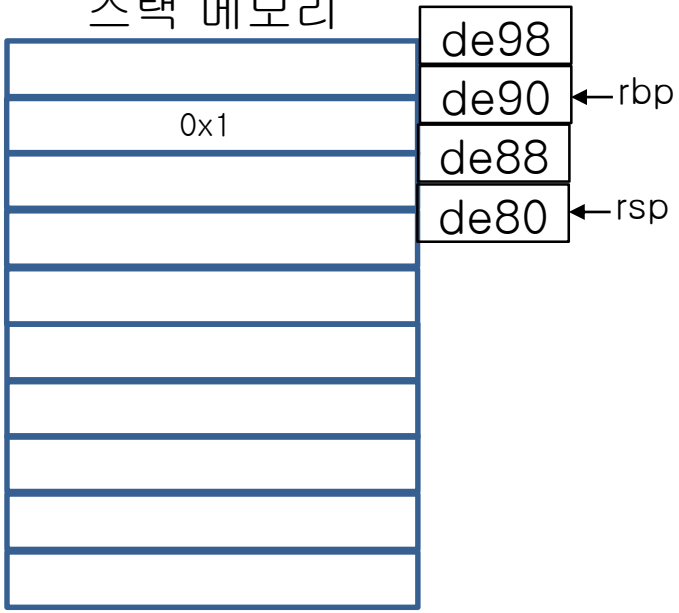
어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
=> 0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

스택 메모리



rbp에서 8바이트 뺀 주소(de88)에 3대입

어셈블리어 분석(5)

EDDI
Electronic Design Development Institute

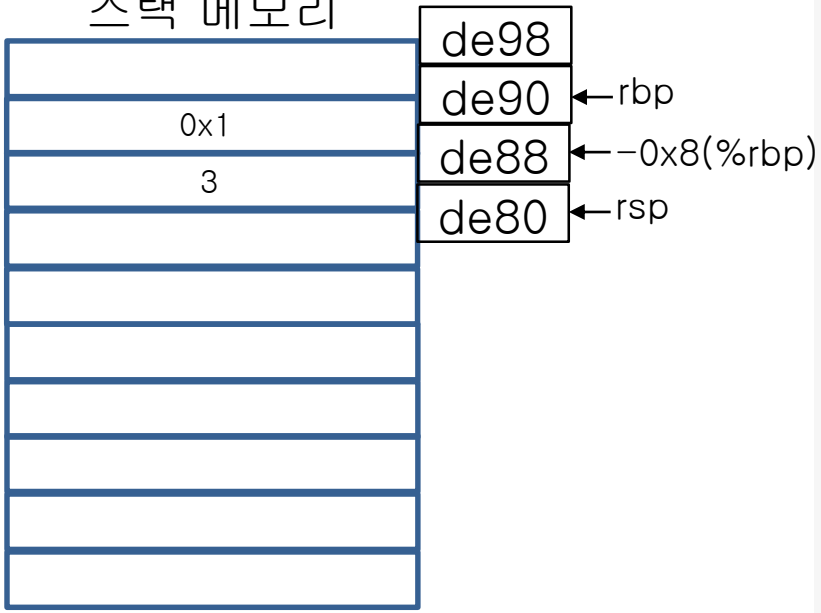
어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
=> 0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

스택 메모리



rbp에서 8바이트 뺀 주소(de88)에
3 대입

어셈블리어 분석(6)

EDDI
Electronic Design Development Institute

어셈블리어

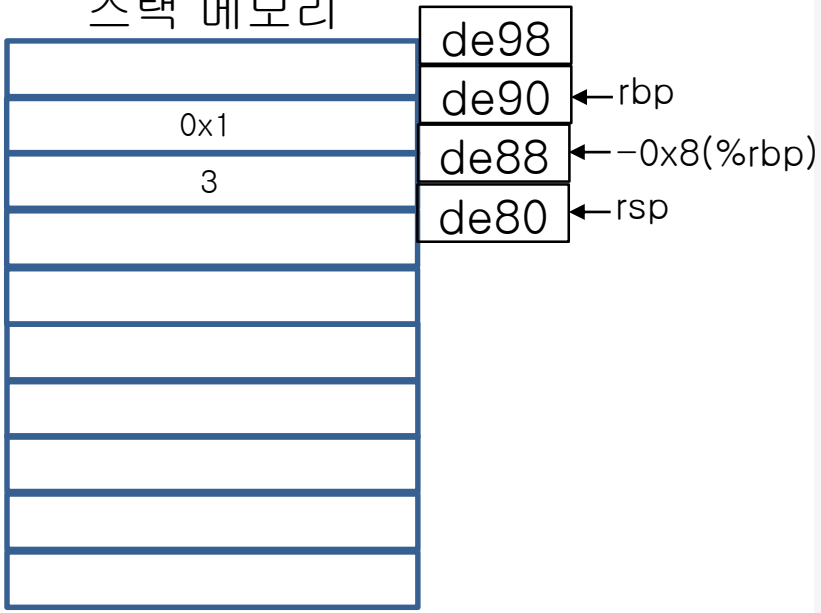
```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
=> 0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

```
(gdb) p $eax
$1 = 3
```

스택 메모리



eax에 3대입

eax는 함수 리턴값 저장에 사용됨

어셈블리어 분석(7)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
=> 0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

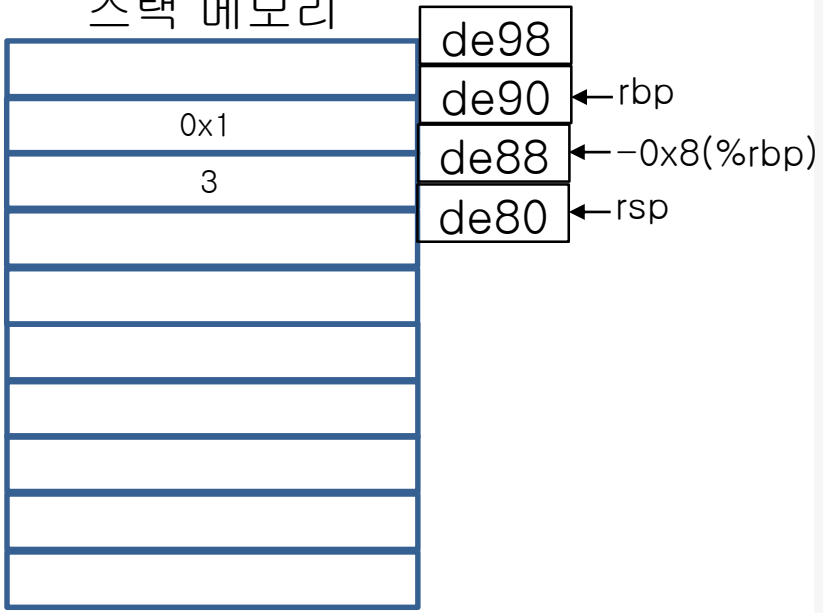
레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

```
(gdb) p $eax
$1 = 3
```

```
(gdb) p/x $edi
$5 = 0x3
```

스택 메모리



edi=eax
edi에 3대입

어셈블리어 분석(8)

EDDI
Electronic Design Development Institute

어셈블리어

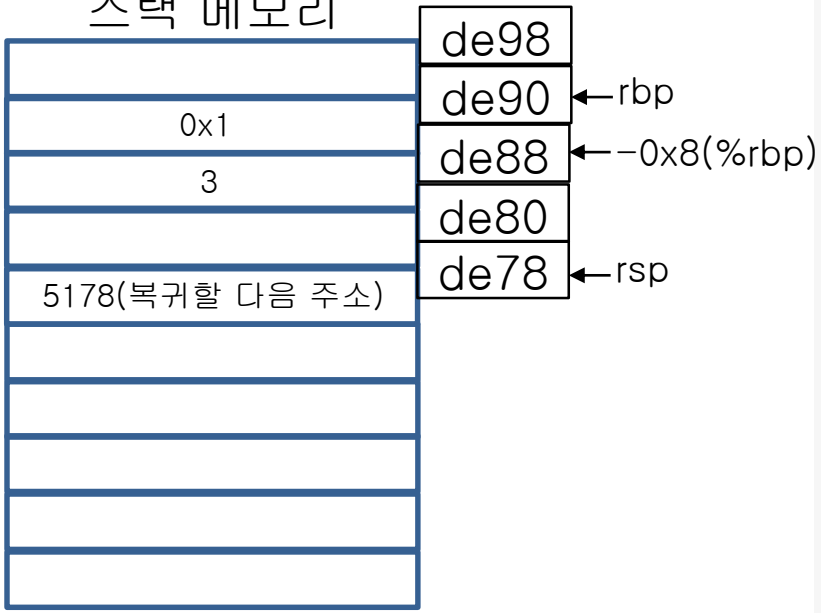
```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
=> 0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
```

```
=> 0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
rbp            0x7fffffffde90     0x7fffffffde90
rsp            0x7fffffffde78     0x7fffffffde78
```

스택 메모리



call= push+jmp
push ->rsp증가 후 복귀주소 저장
jump ->5149로 이동

어셈블리어 분석(9)

EDDI
Electronic Design Development Institute

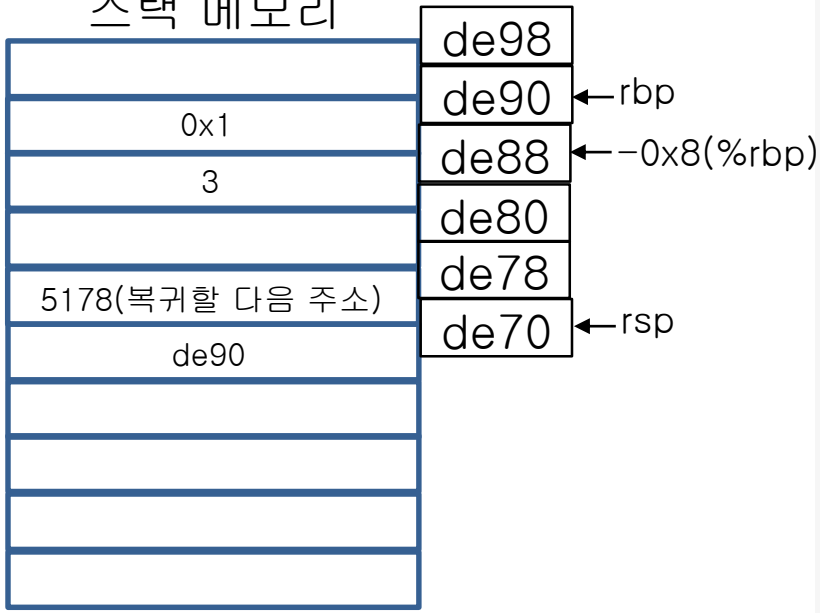
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
=> 0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde70      0x7fffffffde70
```

스택 메모리



push rbp
-> rsp에 rbp주소 대입 후 rsp증가

어셈블리어 분석(10)

EDDI
Electronic Design Development Institute

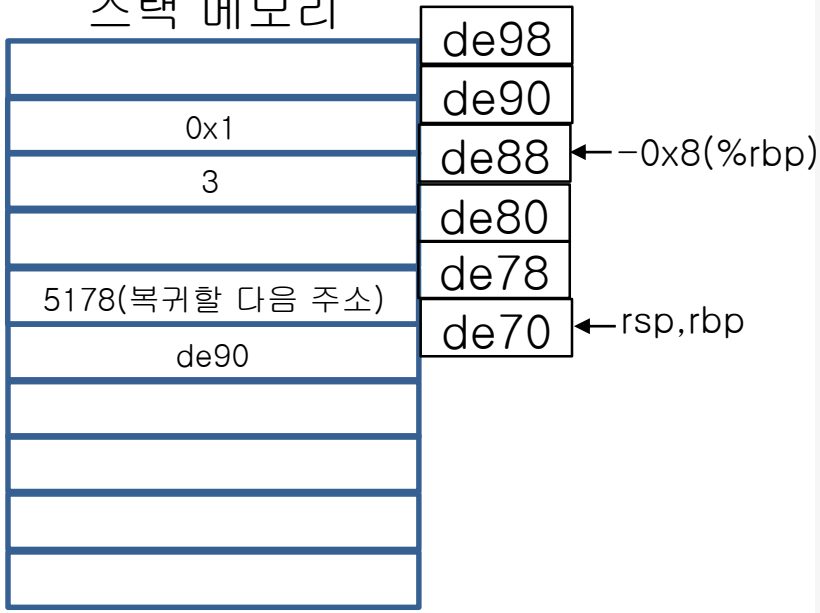
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
=> 0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
rbp            0x7fffffffde70      0x7fffffffde70
rsp            0x7fffffffde70      0x7fffffffde70
```

스택 메모리



rbp에 rsp 대입

어셈블리어 분석(11)

EDDI
Electronic Design Development Institute

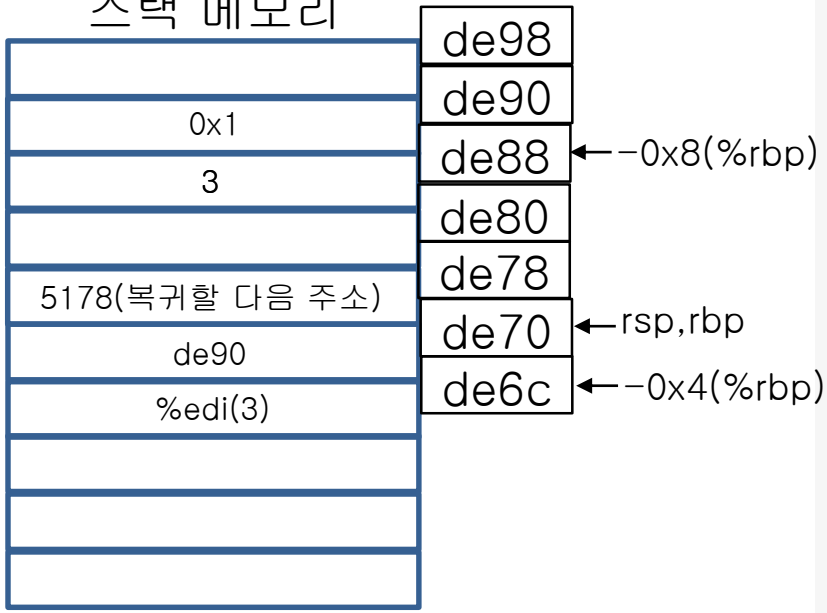
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
=> 0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
rbp            0x7fffffffde70      0x7fffffffde70
rsp            0x7fffffffde70      0x7fffffffde70
```

스택 메모리



rbp에 rsp 대입

어셈블리어 분석(12)

EDDI
Electronic Design Development Institute

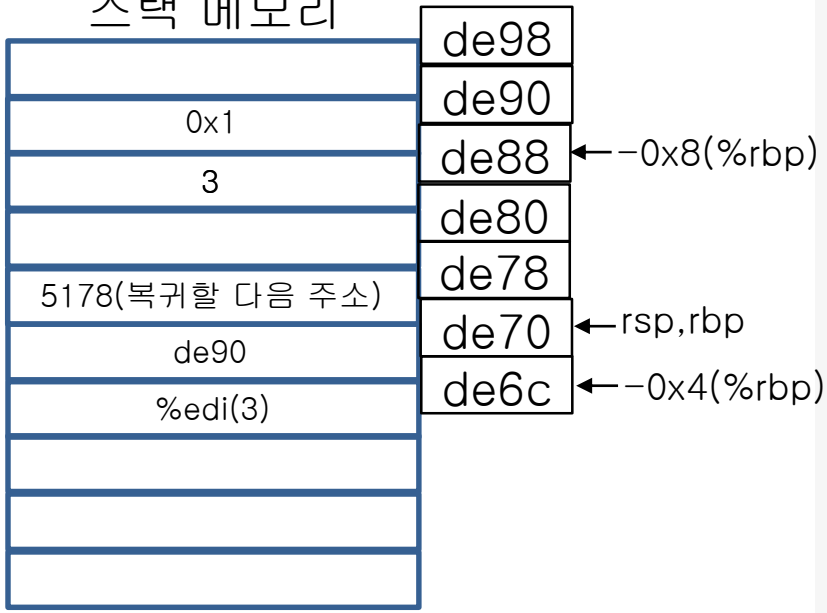
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
=> 0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
(gdb) p $eax
$1 = 3
```

스택 메모리



eax에 0x4(%rbp)=(3) 대입

어셈블리어 분석(13)

EDDI
Electronic Design Development Institute

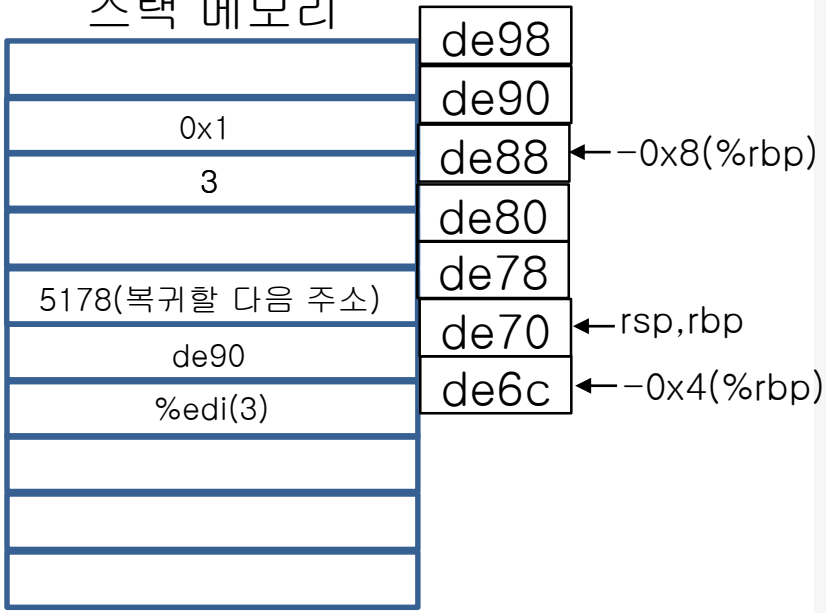
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
=> 0x0000555555555159 <+16>:    pop    %rbp
   0x000055555555515a <+17>:    ret
```

레지스터

```
(gdb) p $eax
$2 = 6
```

스택 메모리



eax +=eax

eax=6

어셈블리어 분석(14)

어셈블리어

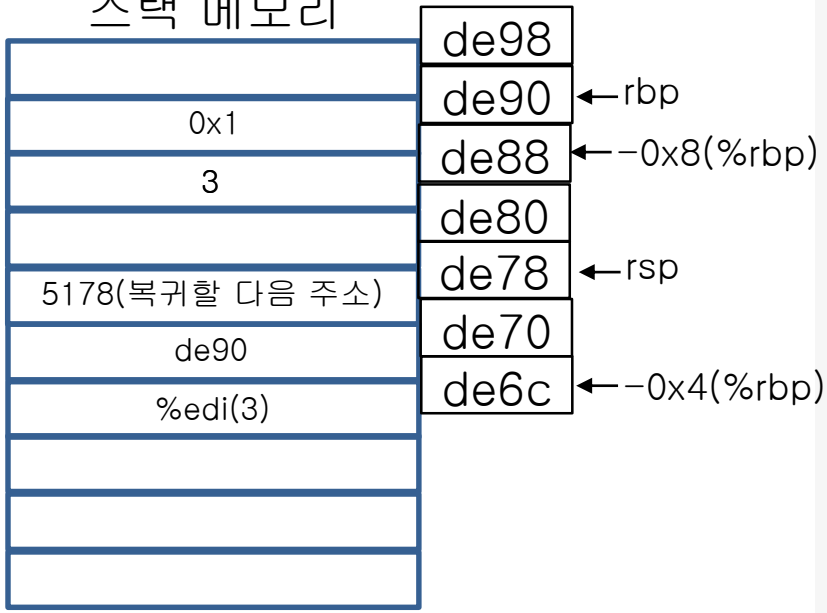
```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
=> 0x000055555555515a <+17>:    ret
```

```
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde78      0x7fffffffde78
```

스택 메모리



pop
->rsp값을 rbp에 대입후 rsp를 한칸 올린다.

어셈블리어 분석(15)

EDDI
Electronic Design Development Institute

어셈블리어

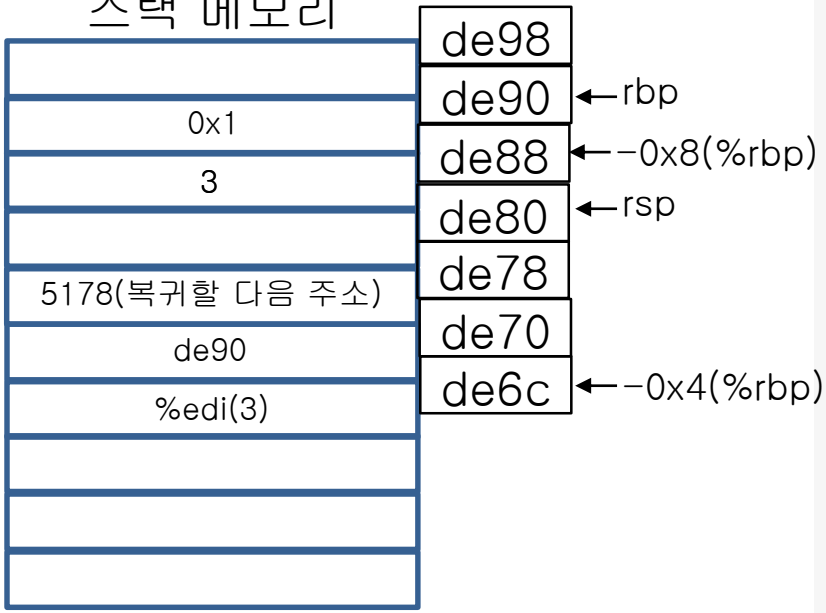
```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     mov    %edi,-0x4(%rbp)
   0x0000555555555154 <+11>:    mov    -0x4(%rbp),%eax
   0x0000555555555157 <+14>:    add    %eax,%eax
   0x0000555555555159 <+16>:    pop    %rbp
=> 0x000055555555515a <+17>:    ret
```

```
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
=> 0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

rbp	0x7fffffffde90	0x7fffffffde90
rsp	0x7fffffffde80	0x7fffffffde80

스택 메모리



ret =pop rip
->rip값에 rsp값(5178(복귀할주소))
전달 후 rsp 한칸 올린다.

어셈블리어 분석(16)

EDDI
Electronic Design Development Institute

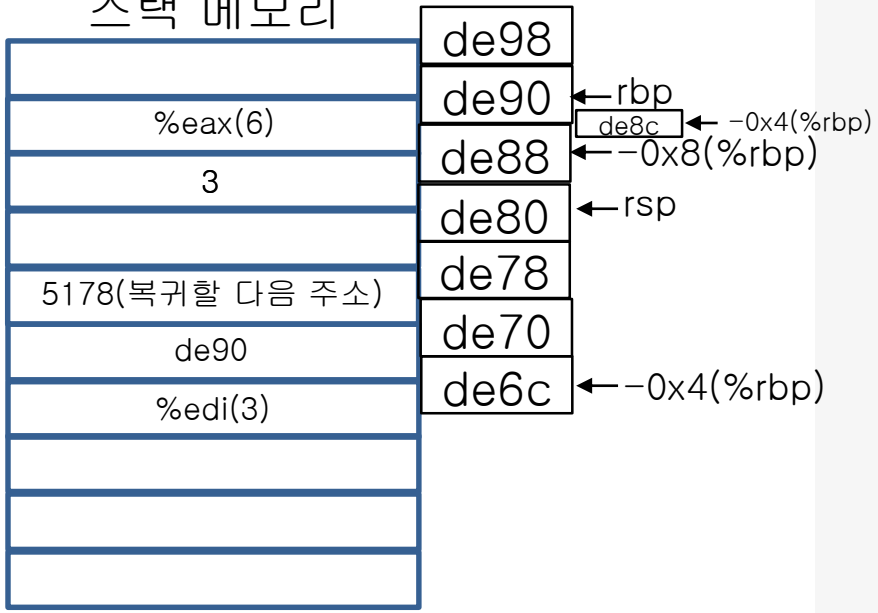
어셈블리어

```
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
=> 0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

스택 메모리



rbp-4(byte)에 eax(6)대입

어셈블리어 분석(16)

EDDI
Electronic Design Development Institute

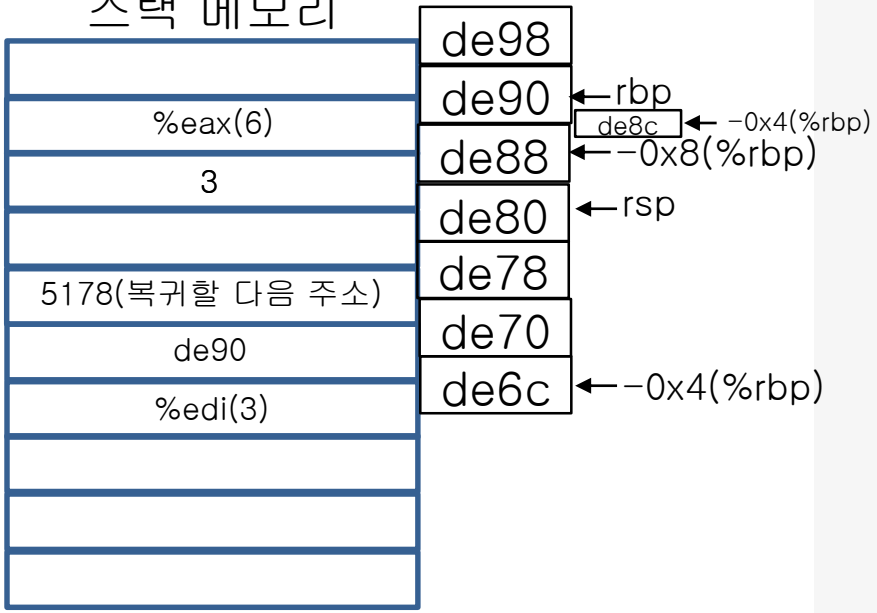
어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
=> 0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
   0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

레지스터

```
rbp            0x7fffffffde90      0x7fffffffde90
rsp            0x7fffffffde80      0x7fffffffde80
```

스택 메모리



eax에 6대입(?)

어셈블리어 분석(17)

EDDI
Electronic Design Development Institute

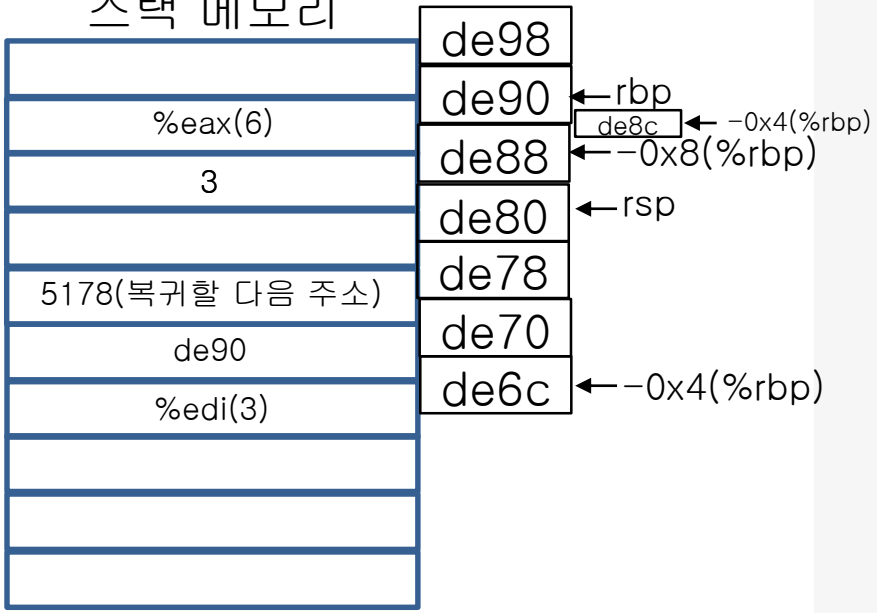
어셈블리어

```
0x000055555555515b <+0>:     endbr64
0x000055555555515f <+4>:     push   %rbp
0x0000555555555160 <+5>:     mov    %rsp,%rbp
0x0000555555555163 <+8>:     sub    $0x10,%rsp
0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
0x0000555555555171 <+22>:    mov    %eax,%edi
0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
0x000055555555517e <+35>:    mov    %eax,%esi
=> 0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
0x0000555555555187 <+44>:    mov    %rax,%rdi
0x000055555555518a <+47>:    mov    $0x0,%eax
0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
0x0000555555555194 <+57>:    mov    $0x0,%eax
0x0000555555555199 <+62>:    leave
0x000055555555519a <+63>:    ret
```

레지스터

```
(gdb) p $esi
$7 = 6
```

스택 메모리



esi=eax=6

어셈블리어 분석(18)

EDDI
Electronic Design Development Institute

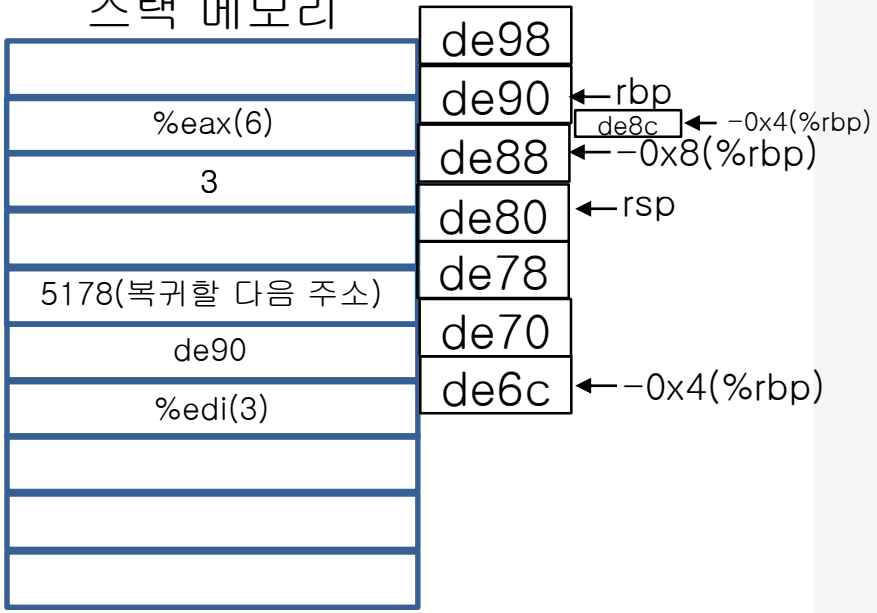
어셈블리어

```
0x000055555555515b <+0>:     endbr64
0x000055555555515f <+4>:     push   %rbp
0x0000555555555160 <+5>:     mov    %rsp,%rbp
0x0000555555555163 <+8>:     sub    $0x10,%rsp
0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
0x0000555555555171 <+22>:    mov    %eax,%edi
0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
0x000055555555517e <+35>:    mov    %eax,%esi
=> 0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
0x0000555555555187 <+44>:    mov    %rax,%rdi
0x000055555555518a <+47>:    mov    $0x0,%eax
0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
0x0000555555555194 <+57>:    mov    $0x0,%eax
0x0000555555555199 <+62>:    leave
0x000055555555519a <+63>:    ret
```

레지스터

```
rax            0x555555556004      93824992239620
```

스택 메모리



lea는 주소를 대입하기 위해 사용
rax에 6004대입

어셈블리어 분석(19)

EDDI
Electronic Design Development Institute

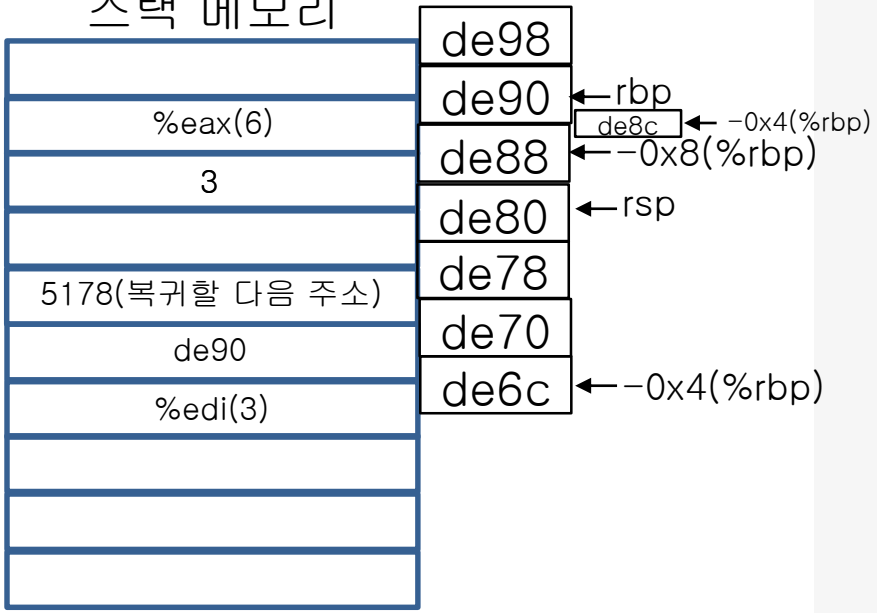
어셈블리어

```
0x000055555555515b <+0>:     endbr64
0x000055555555515f <+4>:     push   %rbp
0x0000555555555160 <+5>:     mov    %rsp,%rbp
0x0000555555555163 <+8>:     sub    $0x10,%rsp
0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
0x0000555555555171 <+22>:    mov    %eax,%edi
0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
0x000055555555517e <+35>:    mov    %eax,%esi
0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
0x0000555555555187 <+44>:    mov    %rax,%rdi
0x000055555555518a <+47>:    mov    $0x0,%eax
0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
0x0000555555555194 <+57>:    mov    $0x0,%eax
0x0000555555555199 <+62>:    leave
0x000055555555519a <+63>:    ret
```

레지스터

rax	0x555555556004	93824992239620
rbx	0x0	0
rcx	0x555555557dc0	93824992247232
rdx	0x7fffffffdfb8	140737488347064
rsi	0x6	6
rdi	0x555555556004	93824992239620

스택 메모리



rdi에 rax 대입

어셈블리어 분석(20)

EDDI
Electronic Design Development Institute

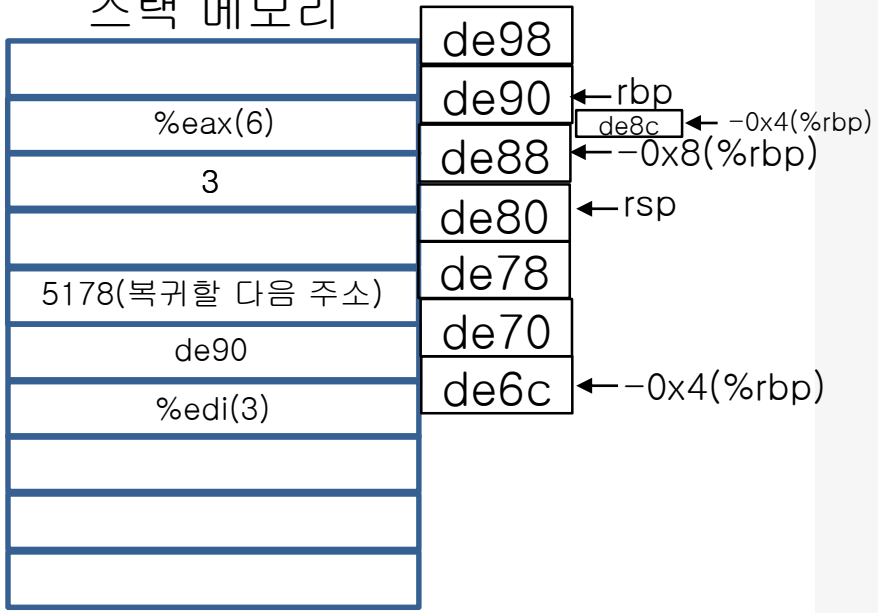
어셈블리어

```
   0x000055555555515b <+0>:     endbr64
   0x000055555555515f <+4>:     push   %rbp
   0x0000555555555160 <+5>:     mov    %rsp,%rbp
   0x0000555555555163 <+8>:     sub    $0x10,%rsp
   0x0000555555555167 <+12>:    movl   $0x3,-0x8(%rbp)
   0x000055555555516e <+19>:    mov    -0x8(%rbp),%eax
   0x0000555555555171 <+22>:    mov    %eax,%edi
   0x0000555555555173 <+24>:    call   0x555555555149 <multiply_two>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rax        # 0x555555556004
   0x0000555555555187 <+44>:    mov    %rax,%rdi
   0x000055555555518a <+47>:    mov    $0x0,%eax
   0x000055555555518f <+52>:    call   0x555555555050 <printf@plt>
=> 0x0000555555555194 <+57>:    mov    $0x0,%eax
   0x0000555555555199 <+62>:    leave
   0x000055555555519a <+63>:    ret
```

```
rax            0xc
rbx            0x0
rcx            0x1
rdx            0x0
rsi            0x5555555592a0
rdi            0x7fffffffd920
rbp            0x7fffffffde90
rsp            0x7fffffffde80
r8             0x0
r9             0x7fffffffdd57
r10            0x0
r11            0x246
r12            0x7fffffffdfa8
r13            0x555555555515b
r14            0x555555557dc0
r15            0x7ffff7ffd040
rip            0x5555555555194
```

```
rax            0x0
rbx            0x0
rcx            0x1
rdx            0x0
rsi            0x5555555592a0
rdi            0x7fffffffd920
rbp            0x1
rsp            0x7fffffffde98
r8             0x0
r9             0x7fffffffdd57
r10            0x0
r11            0x246
r12            0x7fffffffdfa8
r13            0x555555555515b
r14            0x555555557dc0
r15            0x7ffff7ffd040
rip            0x555555555519a
eflags         0x206
```

스택 메모리



- printf 함수호출
- eax에 0대입 (return 0 부분)
- leave :레지스터값 초기화 (스택 메모리 해제)