

에디로봇아카데미 임베디드 마스터 Lv1 과정

제 5기 2023. 05. 19 최성효

터미널 명령어 정리



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)



어셈블리어

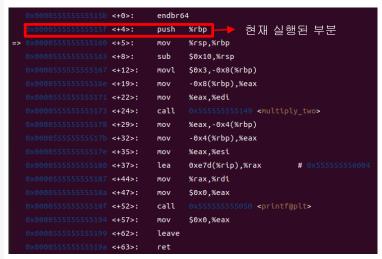
```
endbr64
                                                  다음에 실행될 부분
                           push %rbp
0x00005555555555160 <+5>:
                          mov %rsp,%rbp
0x00005555555555163 <+8>:
                                 $0x10,%rsp
                           movl $0x3,-0x8(%rbp)
0x000055555555516e <+19>:
                                 -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                 %eax,%edi
0x00005555555555173 <+24>:
0x00005555555555178 <+29>:
                                 %eax,-0x4(%rbp)
0x0000555555555517b <+32>:
                                 -0x4(%rbp),%eax
0x0000555555555517e <+35>:
                           mov
                                 %eax,%esi
                                 0xe7d(%rip),%rax
0x00005555555555187 <+44>:
                                 %rax,%rdi
                                 $0x0,%eax
0x0000555555555518f <+52>:
0x00005555555555194 <+57>:
                                 $0x0,%eax
0x00005555555555199 <+62>:
0x0000555555555519a <+63>:
```

rbp	0x1	0x1
гѕр	0x7fffffffde98	0x7fffffffde98

어셈블리어 분석(2)

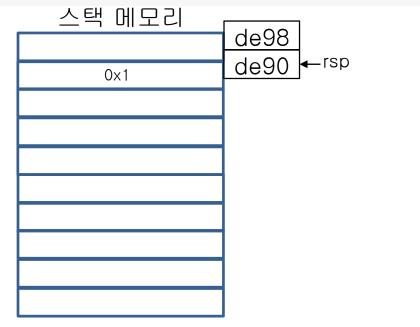


어셈블리어



레지스터

грр	0x1	0x1
rsp	0x7fffffffde90	0x7fffffffde90



Push는 현재 rsp의 한칸 증가시키고 rbp값 저장

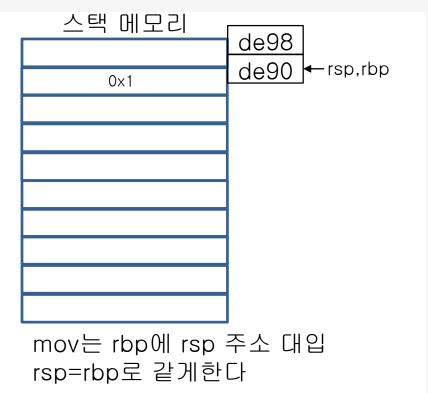
어셈블리어 분석(3)



어셈블리어

```
0x000055555555555555 <+0>:
                               endbr64
  0x00005555555555515f <+4>:
                               push %rbp
    (00005555555555160 <+5>:
                              mov %rsp,%rbp
> 0x00005555555555163 <+8>:
                                     $0x10,%rsp
 0x00005555555555167 <+12>:
                                     $0x3,-0x8(%rbp)
                               movl
 0x0000555555555516e <+19>:
                                      -0x8(%rbp),%eax
                               mov
 0x00005555555555171 <+22>:
                                      %eax,%edi
 0x00005555555555173 <+24>:
                               call
 0x00005555555555178 <+29>:
                                      %eax,-0x4(%rbp)
 0x0000555555555517b <+32>:
                               mov
                                      -0x4(%rbp),%eax
 0x0000555555555517e <+35>:
                                      %eax,%esi
                               mov
 0x00005555555555180 <+37>:
                                      0xe7d(%rip),%rax
                                                              # 0x55555556004
                               lea
                                      %rax,%rdi
 0x000005555555555187 <+44>:
 0x000055555555518a <+47>:
                                      $0x0,%eax
 0x000055555555518f <+52>:
                               call
 0x00005555555555194 <+57>:
                               MOV
                                      S0x0.%eax
 0x00005555555555199 <+62>:
                               leave
 0x0000555555555519a <+63>:
```

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7fffffffde90	0x7fffffffde90

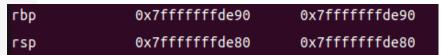


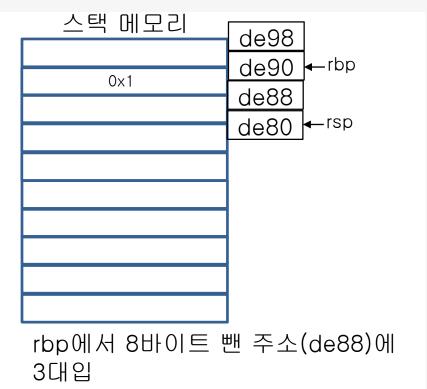
어셈블리어 분석(4)



어셈블리어

```
endbr64
  0x000055555555555515f <+4>:
                               push %rbp
  0x0000555555555160 <+5>:
                                      %rsp,%rbp
      00555555555163 <+8>:
                               sub $0x10,%rsp
=> 0x00005555555555167 <+12>:
                               movl
                                      $0x3,-0x8(%rbp)
  0x0000555555555516e <+19>:
                                       -0x8(%rbp),%eax
                               MOV
  0x00005555555555171 <+22>:
                                       %eax.%edi
                               call
  0x000055555555555173 <+24>:
  0x000055555555555178 <+29>:
                                       %eax,-0x4(%rbp)
                               mov
  0x0000555555555517b <+32>:
                               mov
                                       -0x4(%rbp),%eax
  0x0000555555555517e <+35>:
                                       %eax,%esi
                               MOV
  0x00005555555555180 <+37>:
                                       0xe7d(%rip),%rax
                                                               # 0x55555556004
                               lea
  0x00005555555555187 <+44>:
                                       %rax.%rdi
                               mov
  0x0000555555555518a <+47>:
                                       S0x0.%eax
  0x0000555555555518f <+52>:
                               call
  0x00005555555555194 <+57>:
                                       $0x0,%eax
                               mov
  0x00005555555555199 <+62>:
                               leave
  0x0000555555555519a <+63>:
```





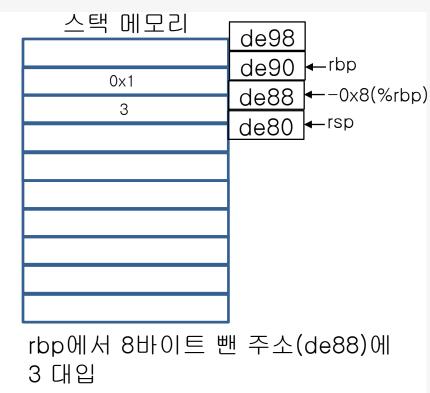
어셈블리어 분석(5)



어셈블리어

```
endbr64
push %rbp
0x00005555555555160 <+5>:
                                  %rsp,%rbp
0x00005555555555163 <+8>:
                                  $0x10,%rsp
x00005555555555167 <+12>:
                           movl $0x3,-0x8(%rbp)
0x0000555555555516e <+19>:
                                  -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                  %eax,%edi
0x00005555555555173 <+24>:
                           call 0x5555555555149 <multiply two>
0x00005555555555178 <+29>:
                                  %eax,-0x4(%rbp)
                           mov
0x0000555555555517b <+32>:
                                  -0x4(%rbp).%eax
0x0000555555555517e <+35>:
                           mov
                                  %eax,%esi
0x00005555555555180 <+37>:
                           lea
                                  0xe7d(%rip),%rax
                                                         # 0x55555556004
0x00005555555555187 <+44>:
                                  %rax,%rdi
0x0000555555555518a <+47>:
                                  $0x0,%eax
                           mov
0x0000555555555518f <+52>:
                           call
0x00005555555555194 <+57>:
                                  $0x0,%eax
0x00005555555555199 <+62>:
                           leave
0x00000555555555519a <+63>:
                           ret
```

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7ffffffffde80	0x7fffffffde80



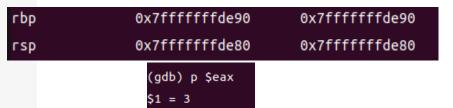
어셈블리어 분석(6)

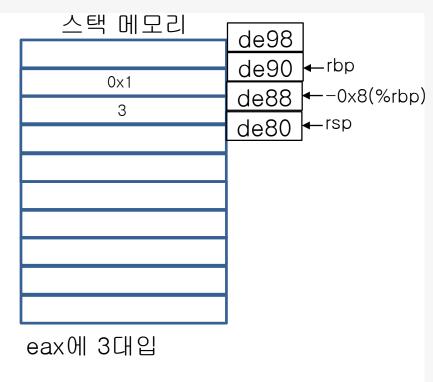


어셈블리어

```
0x000005555555555515b <+0>:
                              endbr64
0x00005555555555515f <+4>:
                              push
                                    %гьр
0x0000555555555160 <+5>:
                                     %rsp,%rbp
0x00005555555555163 <+8>:
                                     $0x10,%rsp
                             movl $0x3,-0x8(%rbp)
0x00005555555555167 <+12>:
                                    -0x8(%rbp),%eax
0x0000555555555516e <+19>:
0x00005555555555171 <+22>:
                                     %eax.%edi
0x000055555555555173 <+24>:
                             call
                                     %eax,-0x4(%rbp)
0x00005555555555178 <+29>:
                                     -0x4(%rbp),%eax
0x0000555555555517b <+32>:
                                     %eax,%esi
0x0000555555555517e <+35>:
0x00005555555555180 <+37>:
                                     0xe7d(%rip),%rax
                                                             # 0x5555555600
0x00005555555555187 <+44>:
                                     %rax,%rdi
0x0000555555555518a <+47>:
                                     $0x0,%eax
0x0000555555555518f <+52>:
                             call
0x00005555555555194 <+57>:
                                     $0x0.%eax
0x00005555555555199 <+62>:
                              leave
0x0000555555555519a <+63>:
                             ret
```

레지스터



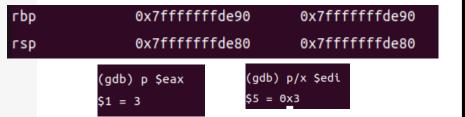


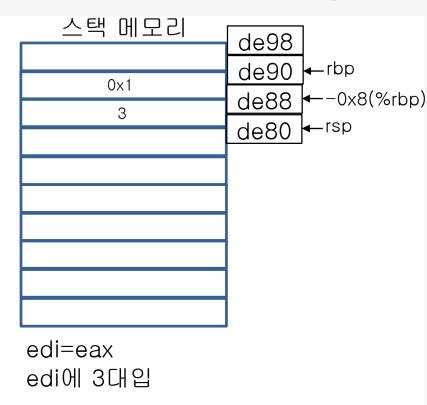
eax는 함수 리턴값 저장에 사용됨

어셈블리어 분석(7)



어셈블리어 0555555555515b <+0>: 0x000005555555555515f <+4>: push %гЬр %rsp,%rbp 0x00005555555555163 <+8>: \$0x10,%rsp \$0x3,-0x8(%rbp) 0x00005555555555167 <+12>: <+19>: -0x8(%rbp),%eax %eax,%edi 0x00005555555555171 <+22>: x0000055555555555173 <+24>: call 0x00005555555555178 <+29>: %eax.-0x4(%rbp) 0x0000555555555517b <+32>: -0x4(%rbp),%eax 0x0000555555555517e <+35>: %eax,%esi 0x00005555555555180 <+37>: lea 0xe7d(%rip),%rax # 0x5555555600 0x00005555555555187 <+44>: %rax.%rdi 0x0000555555555518a <+47>: \$0x0,%eax 0x0000555555555518f <+52>: 0x00005555555555194 <+**57**>: \$0x0,%eax 0x00005555555555199 <+62>: leave 0000555555555519a **<+63>:**



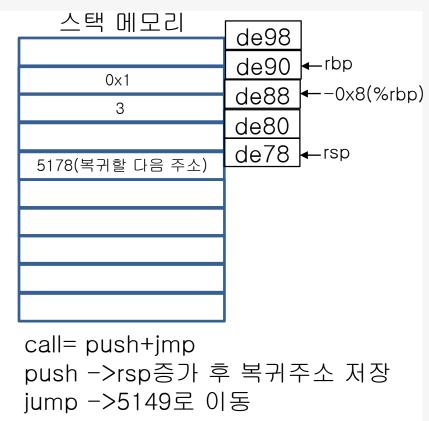


어셈블리어 분석(8)



```
어셈블리어
                  <+0>:
   0x000055555555555515f <+4>:
                          push
                               %rbp
                                %rsp,%rbp
   0x00005555555555163 <+8>:
                                $0x10,%rsp
   0x00005555555555167 <+12>:
                                $0x3,-0x8(%rbp)
   0x0000555555555516e <+19>:
                                -0x8(%rbp),%eax
   0x00005555555555171 <+22>:
                               0x5555555555149 <multiply_two>
   0x000055555555555178 <+29>:
                                %eax.-0x4(%rbp)
=> 0x0000555555555149 <+0>:
                                   endbr64
   0x0000555555555514d <+4>:
                                   push
                                          %rbp
   0x0000555555555514e <+5>:
                                          %rsp,%rbp
                                   MOV
   0x000055555555555151 <+8>:
                                           %edi,-0x4(%rbp)
                                   MOV
   0x000055555555555554 <+11>:
                                   mov
                                           -0x4(%rbp),%eax
   0x000055555555555557 <+14>:
                                           %eax,%eax
                                   add
   pop
                                           %rbp
   0x00005555555555515a <+17>:
                                   ret
```

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7fffffffde78	0x7fffffffde78



어셈블리어 분석(9)

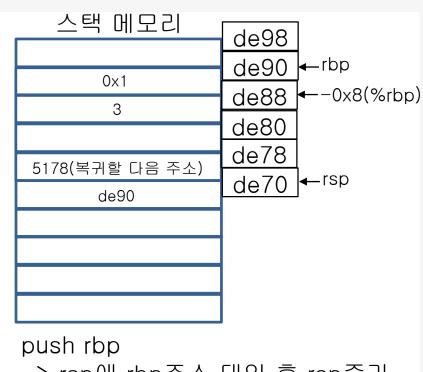


어셈블리어

0x0000555555555149	<+0>:	endbr6	4
0x000055555555514d	<+4>:	push	%гЬр
=> 0x000055555555514e	<+5>:	mov	%rsp,%rbp
0x00005555555555151	<+8>:	mov	%edi,-0x4(%rbp)
0x00005555555555154	<+11>:	mov	-0x4(%rbp),%eax
0x000055555555555	<+14>:	add	%eax,%eax
0x000055555555555	<+16>:	рор	%гЬр
0x000055555555515a	<+17>:	ret	

레지스터

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7fffffffde70	0x7fffffffde70



-> rsp에 rbp주소 대입 후 rsp증가

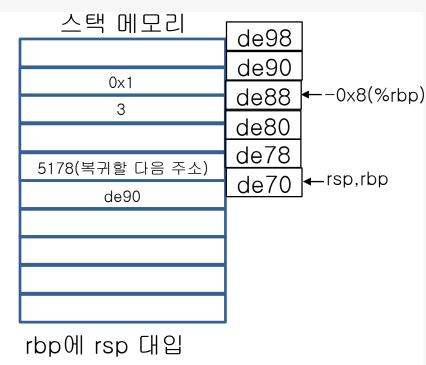
어셈블리어 분석(10)



어셈블리어

		<+0>:	endbr6	4
		<+4>:	push	%гЬр
	0x000055555555514e	<+5>:	MOV	%rsp,%rbp
=>		<+8>:	MOV	%edi,-0х4(%гbp)
		<+11>:	MOV	-0x4(%rbp),%eax
		<+14>:	add	%eax,%eax
		<+16>:	рор	%гЬр
		<+17>:	ret	

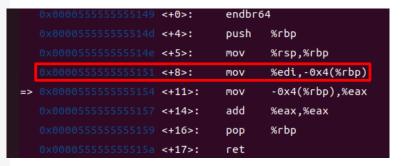
гЬр	0x7fffffffde70	0x7fffffffde70
гѕр	0x7fffffffde70	0x7fffffffde70



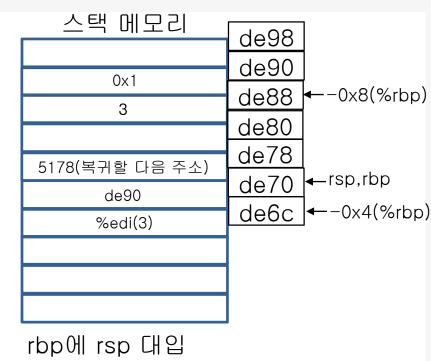
어셈블리어 분석(11)



어셈블리어



гЬр	0x7fffffffde70	0x7fffffffde70
rsp	0x7fffffffde70	0x7fffffffde70



어셈블리어 분석(12)

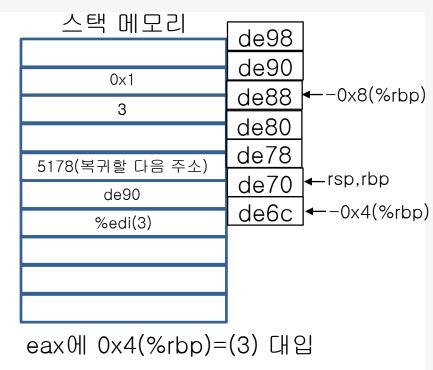


어셈블리어

```
endbr64
0x000005555555555149 <+0>:
0x0000555555555514d <+4>:
                         push
                               %гьр
0x000055555555514e <+5>:
                         MOV
                               %rsp,%rbp
0x0000055555555555151 <+8>:
                               %edi,-0x4(%rbp)
                         mov
0x00005555555555554 <+11>:
                               -0x4(%rbp),%eax
                         mov
add
                               %eax,%eax
%rbp
                         pop
0x00005555555555515a <+17>:
                         ret
```

레지스터

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



어셈블리어 분석(13)

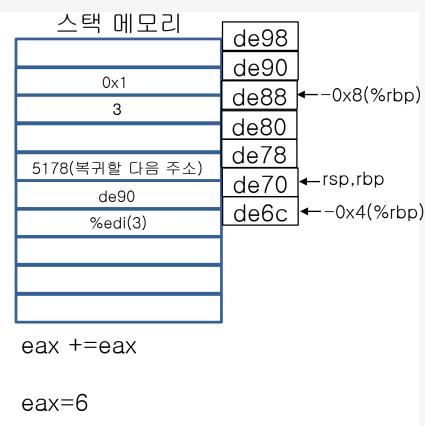


어셈블리어

```
<+0>:
                             endbr64
  0x0000555555555514d <+4>:
                             push
                                    %rbp
  0x000055555555514e <+5>:
                             mov
                                    %rsp,%rbp
                                    %edi,-0x4(%rbp)
  0x000055555555555151 <+8>:
                             mov
  0x00005555555555554 <+11>:
                                    -0x4(%rbp),%eax
                             mov
  0x00005555555555557 <+14>:
                             add
                                    %eax,%eax
%гЬр
                             DOD
  0x00005555555555515a <+17>:
                             ret
```

레지스터

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



어셈블리어 분석(14)



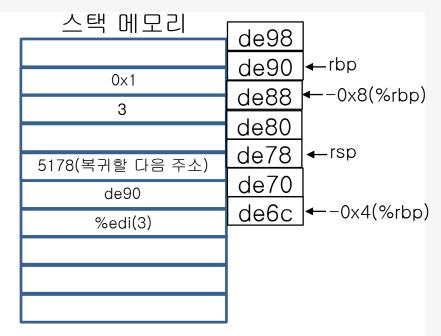
어셈블리어

```
endbr64
   0x00005555555555149 <+0>:
  0x0000555555555514d <+4>:
                               push
                                     %гЬр
  0x0000555555555514e <+5>:
                                     %rsp,%rbp
                               mov
  0x000055555555555151 <+8>:
                                     %edi,-0x4(%rbp)
                               mov
  0x000055555555555554 <+11>:
                                      -0x4(%rbp),%eax
                               mov
  0x000055555555555557 <+14>:
                               add
                                     %eax,%eax
  %гЬр
                               pop
=> 0x00005555555555515a <+17>:
                               ret
```

```
%eax,-0x4(%rbp)
0x0000555555555517b <+32>:
                                      -0x4(%rbp),%eax
0x0000555555555517e <+35>:
                                     %eax,%esi
0x00005555555555180 <+37>:
                                     0xe7d(%rip).%rax
                                                              # 0x5555555600
                              lea
                   <+44>:
                                     %rax,%rdi
0x0000555555555518a <+47>:
                                     $0x0,%eax
0x0000555555555518f <+52>:
0x000055555555555194 <+57>:
                                     $0x0,%eax
0x00005555555555199 <+62>:
                              leave
    00555555555519a <+63>:
```

레지스터

гbр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7fffffffde78	0x7fffffffde78



pop

->rsp값을 rbp에 대입후 rsp를 한칸 올린다.

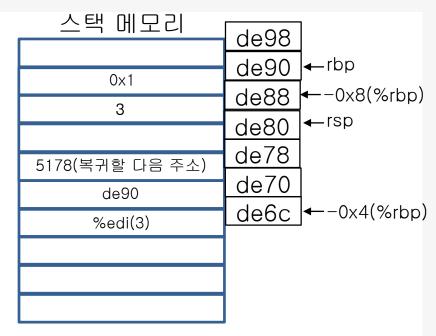
어셈블리어 분석(15)



```
어셈블리어
   0x000055555555514d <+4>:
                                           %гьр
                                   push
   0x0000555555555514e <+5>:
                                           %rsp,%rbp
                                   MOV
                                           %edi,-0x4(%rbp)
    0x0000555555555555151 <+8>:
                                    mov
   0x000055555555555554 <+11>:
                                           -0x4(%rbp),%eax
                                    mov
   0x000055555555555555 <+14>:
                                    add
                                           %eax,%eax
   %rbp
                                    pop
    0x00005555555555515a <+17>:
                                   ret
                         call
     05555555555178 <+29>:
                               %eax,-0x4(%rbp)
                               -0x4(%rbp),%eax
                         MOV
                               %eax.%esi
0x00005555555555180 <+37>:
                               0xe7d(%rip),%rax
                                                    # 0x55555556004
                         lea
                               %rax.%rdi
0x00005555555555187 <+44>:
0x0000555555555518a <+47>:
                               S0x0.%eax
0x00000555555555518f <+52>:
                         call
0x000005555555555194 <+57>:
                               $0x0,%eax
                         mov
0x00005555555555199 <+62>:
                         leave
0x0000555555555519a <+63>:
                         ret
```

레지스터

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7fffffffde80	0x7fffffffde80



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

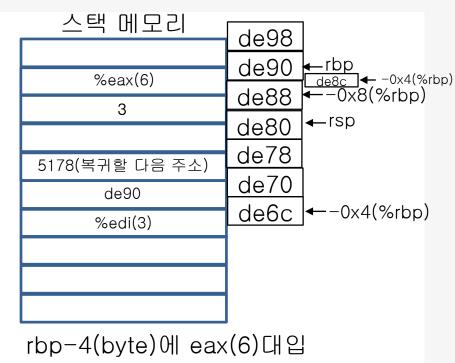
어셈블리어 분석(16)



어셈블리어

```
0x555555555149 <multiply two>
                      <+24>:
                                 call
   0x00005555555555178 <+29>:
                                 MOV
                                        %eax,-0x4(%rbp)
                                        -0x4(%rbp),%eax
=> 0x0000555555555517b <+32>:
                                 mov
  0x0000555555555517e <+35>:
                                        %eax,%esi
                                 mov
                                        0xe7d(%rip),%rax
  0x00005555555555180 <+37>:
                                 lea
                                                                 # 0x55555556004
                                        %rax,%rdi
  0x00005555555555187 <+44>:
                                 mov
                                        $0x0,%eax
  0x0000555555555518a <+47>:
                                 mov
  0x0000555555555518f <+52>:
                                 call
                                        $0x0,%eax
  0x00005555555555194 <+57>:
                                 mov
   0x000055555555555199 <+62>:
                                 leave
  0x0000555555555519a <+63>:
                                 ret
```

гЬр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7ffffffffde80	0x7fffffffde80



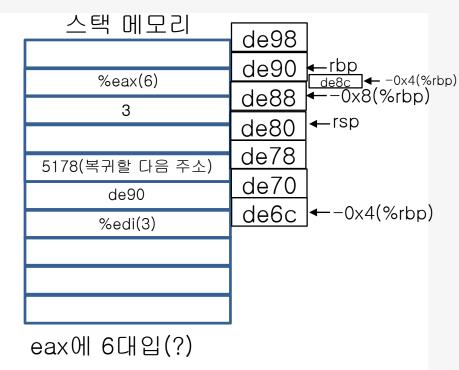
어셈블리어 분석(16)



어셈블리어

```
endbr64
0x00005555555555555 <+4>:
                             push
                                    %гЬр
0x00005555555555160 <+5>:
                             mov
                                    %rsp,%rbp
0x00005555555555163 <+8>:
                             sub
                                    $0x10,%rsp
                                    $0x3,-0x8(%rbp)
                             movl
0x0000555555555516e <+19>:
                             MOV
                                    -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                    %eax.%edi
                             mov
                             call
0x00005555555555178 <+29>:
                             mov
                                    %eax,-0x4(%rbp)
                                    -0x4(%rbp),%eax
     00555555555517b <+32>:
                             MOV
                                    %eax.%esi
0x0000555555555517e <+35>:
                             mov
0x000005555555555180 <+37>:
                             lea
                                    0xe7d(%rip),%rax
                                                            # 0x55555556004
0x00005555555555187 <+44>:
                                    %rax,%rdi
                             mov
0x0000555555555518a <+47>:
                                    $0x0,%eax
                             MOV
0x0000555555555518f <+52>:
                             call
0x00005555555555194 <+57>:
                             MOV
                                    $0x0,%eax
0x00005555555555199 <+62>:
                             leave
                             ret
```

гbр	0x7fffffffde90	0x7fffffffde90
гѕр	0x7ffffffffde80	0x7fffffffde80



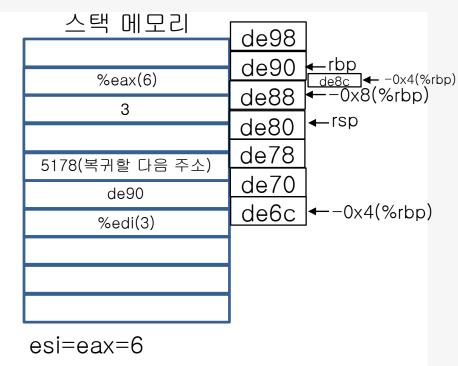
어셈블리어 분석(17)



어셈블리어

```
endbr64
0x00005555555555515f <+4>:
                              push
                                    %гЬр
                                     %rsp,%rbp
0x00005555555555163 <+8>:
                                     $0x10,%rsp
                                     $0x3,-0x8(%rbp)
0x00005555555555167 <+12>:
0x0000555555555516e <+19>:
                                      -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                     %eax,%edi
0x000055555555555173 <+24>:
                                     %eax,-0x4(%rbp)
0x00005555555555178 <+29>:
                                     -0x4(%rbp),%eax
       55555555517b <+32>:
                                     %eax,%esi
                                     0xe7d(%rip),%rax
                                                              # 0x5555555600
0x000055555555555187 <+44>:
                                     %rax.%rdi
0x0000555555555518a <+47>:
                                     $0x0,%eax
0x00005555555555194 <+57>:
                                     $0x0,%eax
                              leave
0x00000555555555519a <+63>:
```





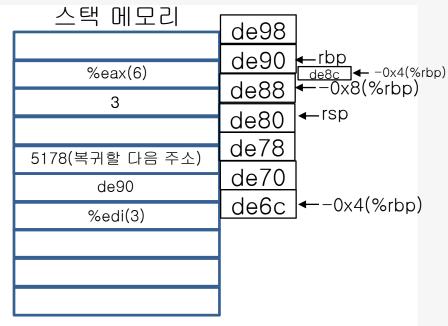
어셈블리어 분석(18)



어셈블리어

```
endbr64
0x00005555555555515f <+4>:
                              push
                                    %гЬр
0x00005555555555160 <+5>:
                                     %rsp,%rbp
0x00005555555555163 <+8>:
                                     $0x10,%rsp
                                     $0x3,-0x8(%rbp)
0x00005555555555167 <+12>:
0x0000555555555516e <+19>:
                                     -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                     %eax.%edi
0x000055555555555173 <+24>:
0x00005555555555178 <+29>:
                                     %eax,-0x4(%rbp)
0x0000555555555517b <+32>:
                                     -0x4(%rbp),%eax
                                     %eax,%esi
                                                             # 0x55555556004
                                     0xe7d(%rip),%rax
0x000055555555555187 <+44>:
                                     %rax.%rdi
                                     $0x0,%eax
0x00005555555555194 <+57>:
                                     $0x0,%eax
                              leave
0x0000555555555519a <+63>:
```

레지스터



lea는 주소를 대입하기 위해 사용 rax에 6004대입

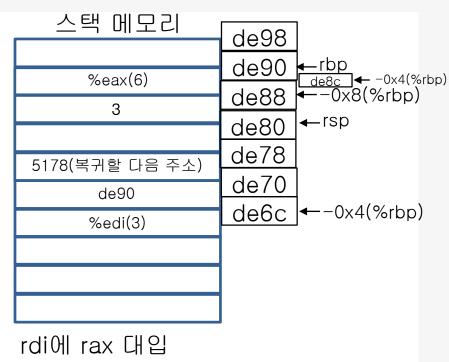
어셈블리어 분석(19)



어셈블리어

```
0x00005555555555555 <+0>:
                              endbr64
0x000005555555555555 <+4>:
                              push
                                     %гьр
0x00005555555555160 <+5>:
                                     %rsp,%rbp
0x000005555555555163 <+8>:
                              sub
                                     $0x10,%rsp
0x000055555555555167 <+12>:
                                     $0x3,-0x8(%rbp)
0x00005555555555516e <+19>:
                                     -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                     %eax,%edi
                              mov
0x000055555555555173 <+24>:
0x000055555555555178 <+29>:
                              mov
                                     %eax,-0x4(%rbp)
                                     -0x4(%rbp),%eax
0x0000555555555517b <+32>:
0x00005555555555517e <+35>:
                              mov
                                     %eax.%esi
0x00005555555555180 <+37>:
                                     0xe7d(%rip),%rax
                                                              # 0x55555556004
                                     %rax,%rdi
0x0000555555555518a <+47>:
                                     $0x0,%eax
0x0000555555555518f <+52>:
                              call
0x000055555555555194 <+57>:
                                     $0x0,%eax
0x000055555555555199 <+62>:
                              leave
0x0000555555555519a <+63>:
```

гах	0x55555556004	93824992239620
гЬх	0x0	0
гсх	0x55555557dc0	93824992247232
rdx	0x7fffffffdfb8	140737488347064
rsi	0x6	6
rdi	0x55555556004	93824992239620

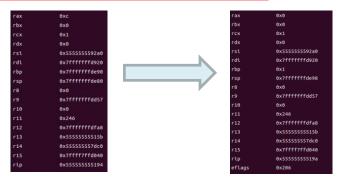


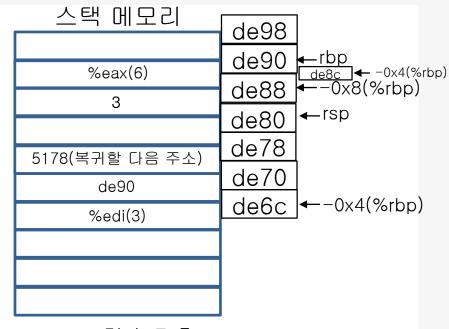
어셈블리어 분석(20)



어셈블리어

```
x000005555555555515b <+0>:
                              endbr64
0x00005555555555515f <+4>:
                              push
                                     %rbp
                                     %rsp,%rbp
                                     $0x10.%rsp
0x00005555555555167 <+12>:
                                     $0x3,-0x8(%rbp)
0x0000555555555516e <+19>:
                                     -0x8(%rbp),%eax
0x00005555555555171 <+22>:
                                     %eax,%edi
0x00005555555555173 <+24>:
0x00005555555555178 <+29>:
                                     %eax,-0x4(%rbp)
0x0000555555555517b <+32>:
                                     -0x4(%rbp),%eax
                                     %eax,%esi
                                     0xe7d(%rip),%rax
                                                              # 0x55555556004
0x00005555555555180 <+37>:
                              lea
                                     %rax,%rdi
                                     $0x0,%eax
                                     0x5555555555050 <printf@plt>
                                     $0x0,%eax
0x00005555555555199 <+62>:
                              leave
       55555555519a <+63>:
```





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