EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv1 과정

제 3기

2022. 01. 15

김원석

CONTENTS

EDDI
Electronic Design
Development Institute

- 기계어 분석
  - 핵심 규칙
  - <main> 함수 기계어 분석
    - push
    - mov
    - sub
    - movl
    - callq
  - <test_func> 함수 기계어 분석
    - add
    - pop
    - retq

기계어 분석

EDDI
Electronic Design
Development Institute

핵심 규칙

1. 스택은 거꾸로 자란다.
2. 메모리 단위 연산은 포인터 크기 단위(ALU가 결정)로 이루어진다.
3. ax의 용도 : 함수의 return값 저장

   bp의 용도 : 스택의 기준점

   sp의 용도 : 스택의 최상위

   ip의 용도 : 다음에 실행할 명령어의 주소

기계어 분석

EDDI
Electronic Design
Development Institute

```c
#include <stdio.h>

int test_func(int num)
{
        return num * 2;
}

int main(void)
{
        int num = 3;
        int result = test_func(num);

        printf("result = %d\n", result);

        return 0;
}
~
~
~
~
~
```

위의 c코드를 분석

기계어 분석(main)

EDDI
Electronic Design Development Institute

```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...
(gdb) b main
Breakpoint 1 at 0x115b: file function.c, line 9.
(gdb) r
Starting program: /home/dama/EmbeddedMasterLv1/3기/WSK/c/3/a.out

Breakpoint 1, main () at function.c:9
9       {
(gdb) disas
Dump of assembler code for function main:
=> 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>:    callq  0x555555555149 <test_func>
   0x0000555555555178 <+29>:    mov    %eax,-0x4(%rbp)
   0x000055555555517b <+32>:    mov    -0x4(%rbp),%eax
   0x000055555555517e <+35>:    mov    %eax,%esi
   0x0000555555555180 <+37>:    lea    0xe7d(%rip),%rdi        # 0x555555556004
   0x0000555555555187 <+44>:    mov    $0x0,%eax
   0x000055555555518c <+49>:    callq  0x555555555050 <printf@plt>
   0x0000555555555191 <+54>:    mov    $0x0,%eax
   0x0000555555555196 <+59>:    leaveq
   0x0000555555555197 <+60>:    retq
End of assembler dump.
```

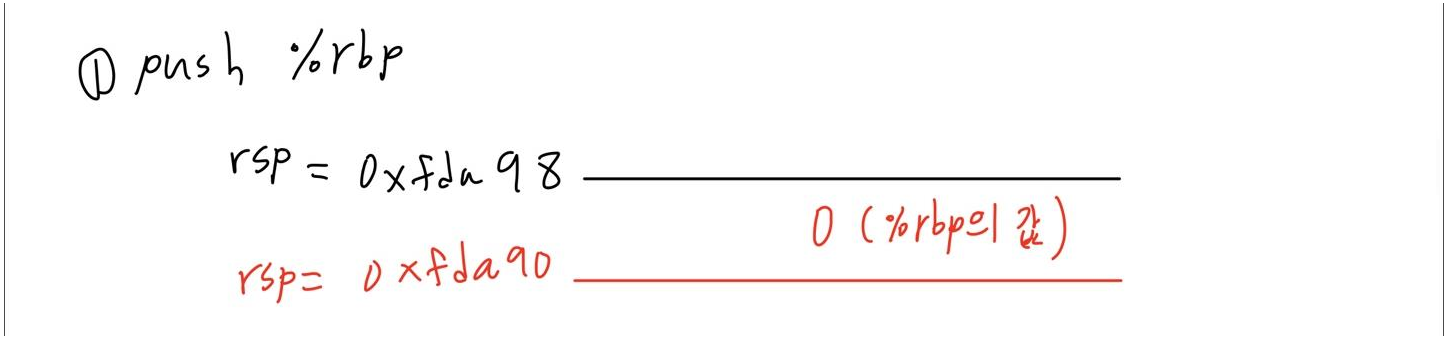
Break Point : main, disas -> 어셈블리어로 표현

기계어 분석(main, push)

EDDI
Electronic Design Development Institute

push

- 현재 스택의 최상위(rsp가 가리키는 메모리 공간)에 뒤쪽의 메모리 값을 넣으시오.



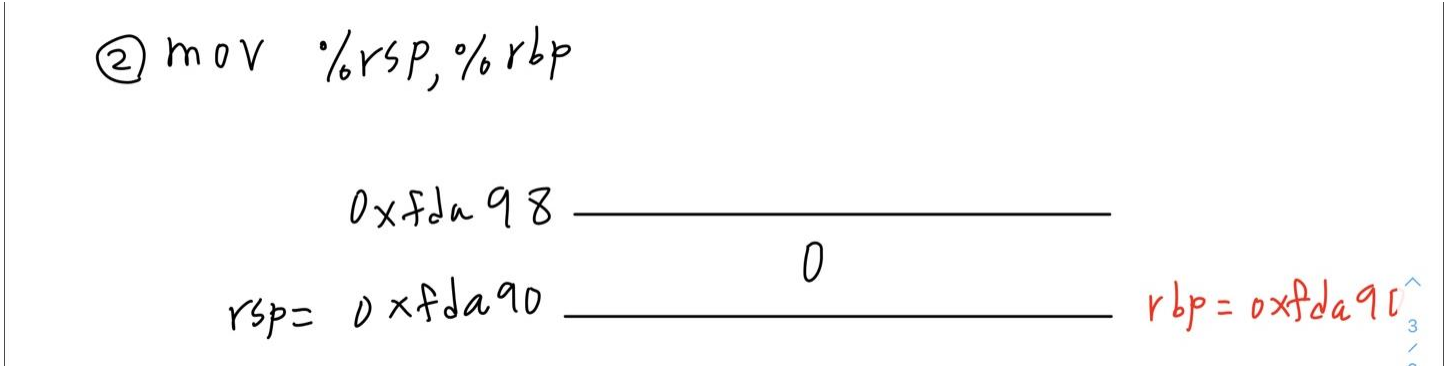
rbp의 메모리 값 0이 스택의 최상위에 들어가고 rsp 이동

기계어 분석(main, mov)

EDDI
Electronic Design Development Institute

mov

- 좌측의 값을 우측에 복사하시오.



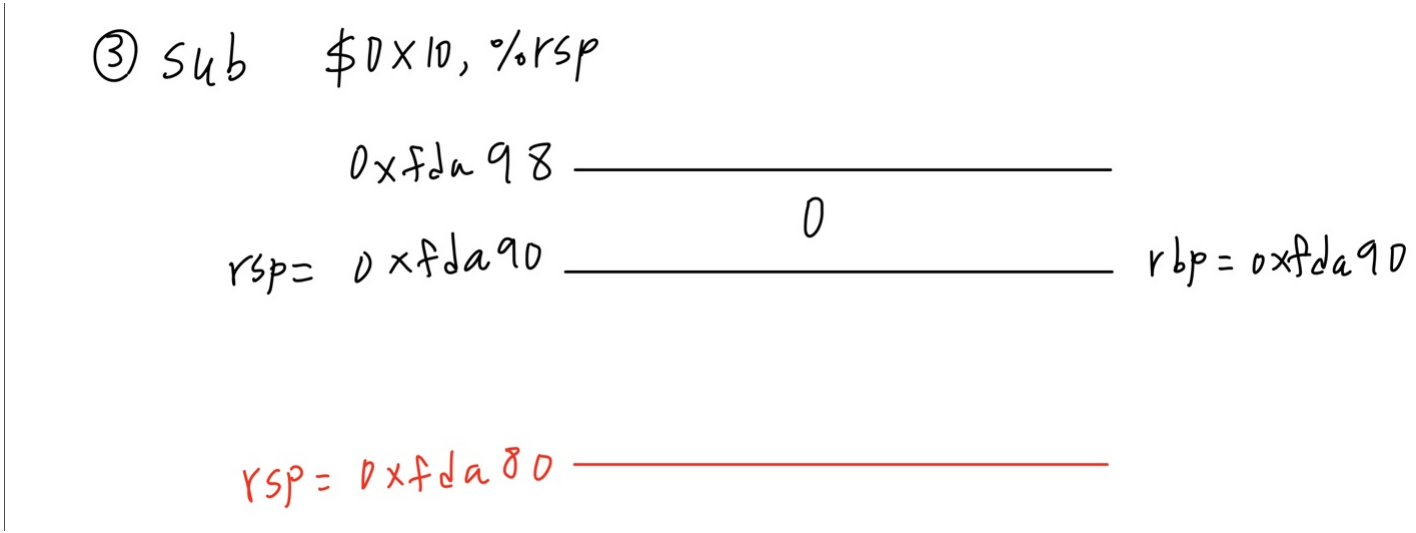
rsp의 값을 rbp에 복사. 1단계와 2단계를 거쳐 <main> 함수의 Stack Frame(c에서 '{' 에 해당) 생성

기계어 분석(main, sub)

EDDI
Electronic Design Development Institute

sub

- 뺄셈



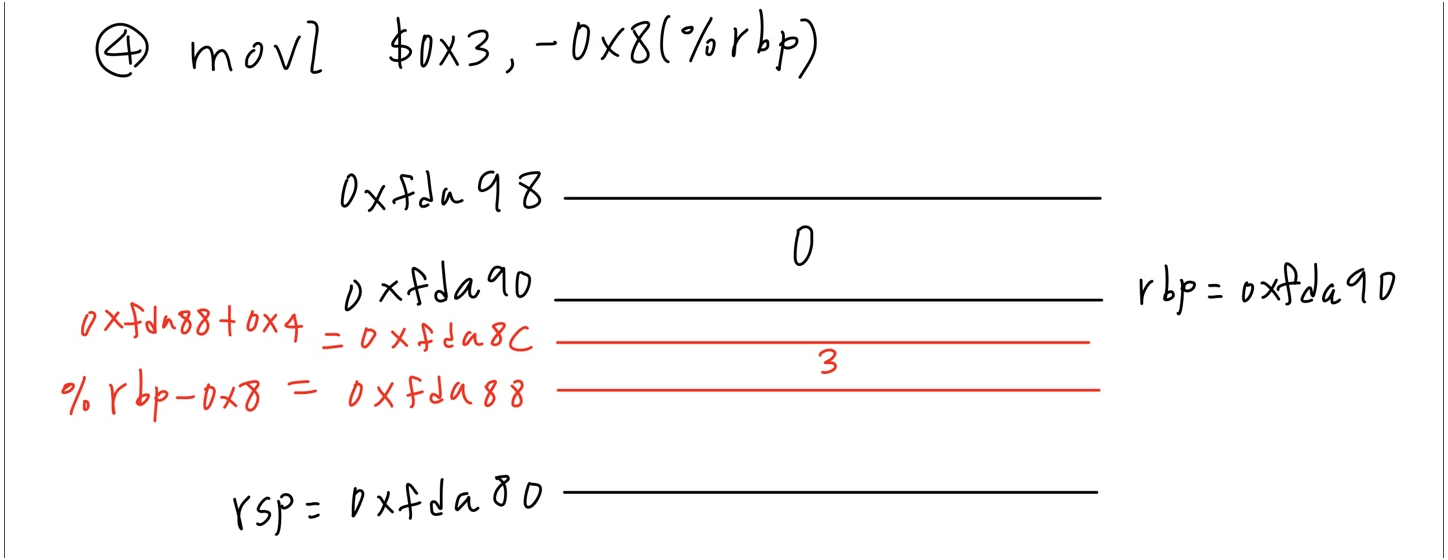
rsp – 0x10을 rsp에 대입

기계어 분석(main, movl)

EDDI
Electronic Design Development Institute

movl

- mov와 같은데 l이 붙어 4byte 크기로 복사



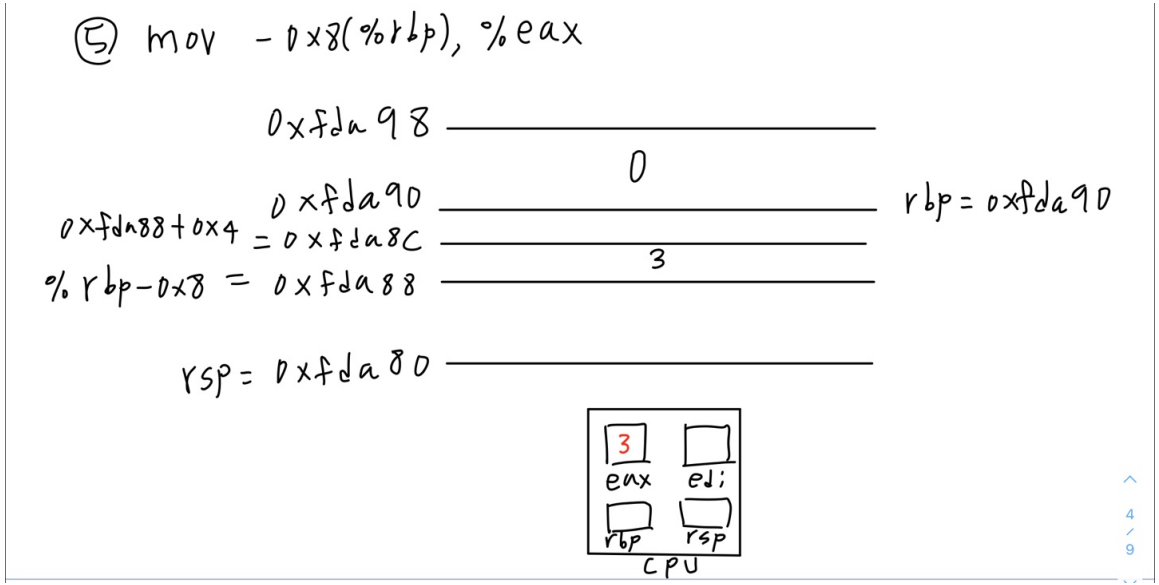
rsp – 0x10을 rsp에 대입

기계어 분석(main, mov)

EDDI
Electronic Design Development Institute

mov

- 좌측의 값을 우측에 복사하시오.



rbp – 0x8의 메모리값을 eax에 복사

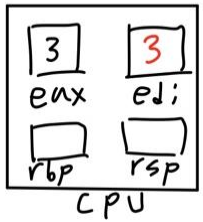
기계어 분석(main, mov)

EDDI
Electronic Design Development Institute

mov

- 좌측의 값을 우측에 복사하시오.

⑥ mov %eax, %edi



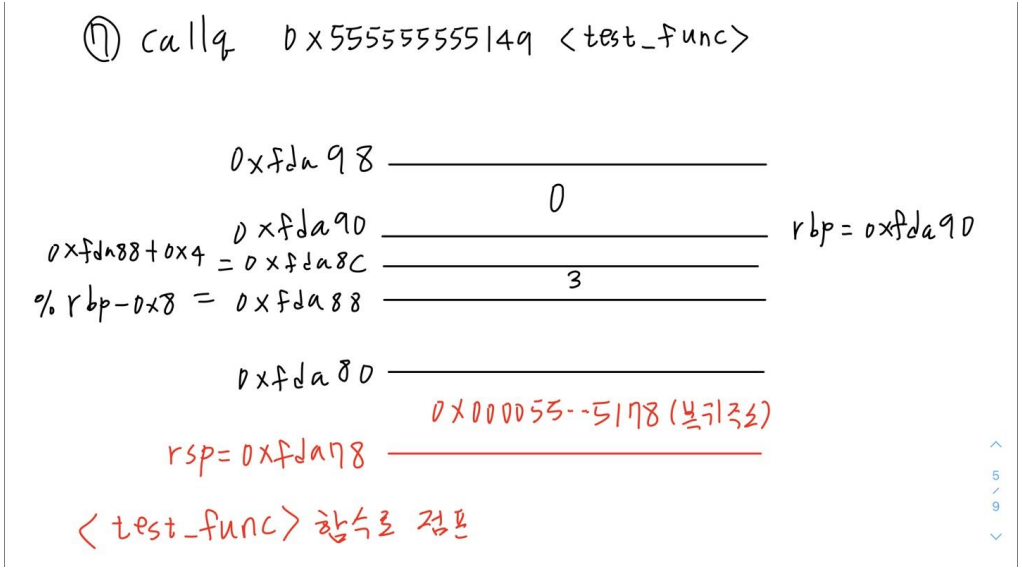
eax의 값을 edi에 복사

기계어 분석(main, callq)

EDDI
Electronic Design Development Institute

callq

- 함수 호출 시 동작. 복귀주소 push 후 함수로 jump. q가 붙어 8byte 크기로 복사



복귀주소 push 후 <test_func>함수로 jump

기계어 분석(test_func)

EDDI
Electronic Design Development Institute

```
(gdb) disas
Dump of assembler code for function test_func:
=> 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>:    retq
```

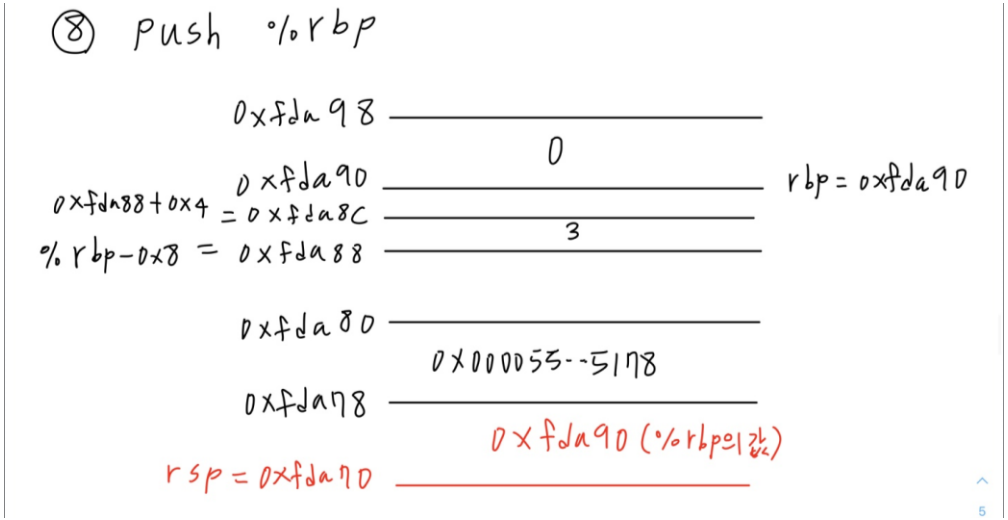
main함수 내의 <test_func>함수의 어셈블리어

기계어 분석(test_func, push)

EDDI
Electronic Design Development Institute

push

- 현재 스택의 최상위에 뒤쪽의 메모리 값을 넣으시오.

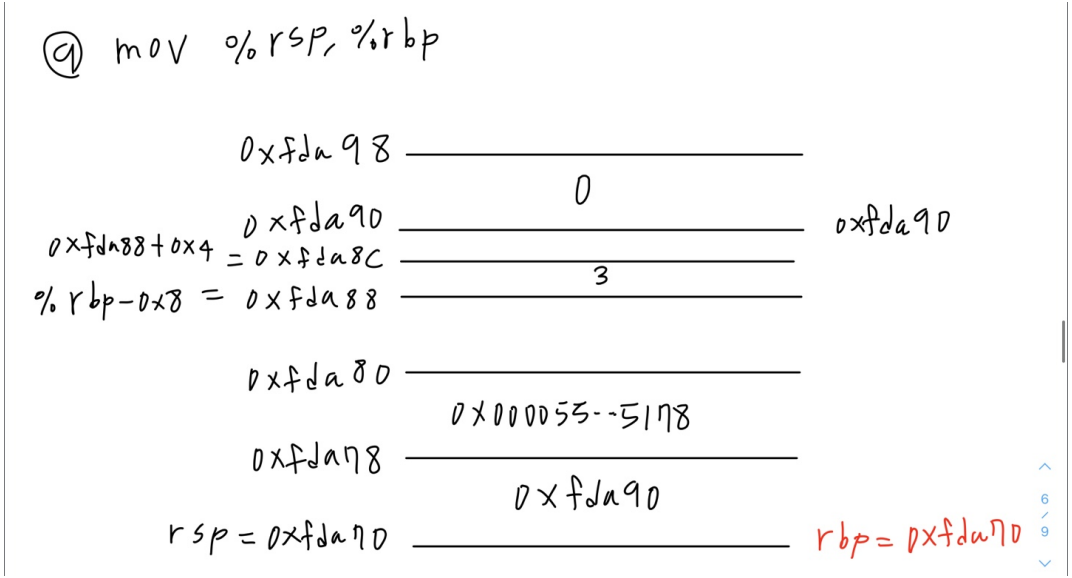


스택의 최상위에 rbp의 메모리값을 넣고 rsp 이동

기계어 분석(test_func, mov)

mov

- 좌측의 값을 우측에 복사하시오.



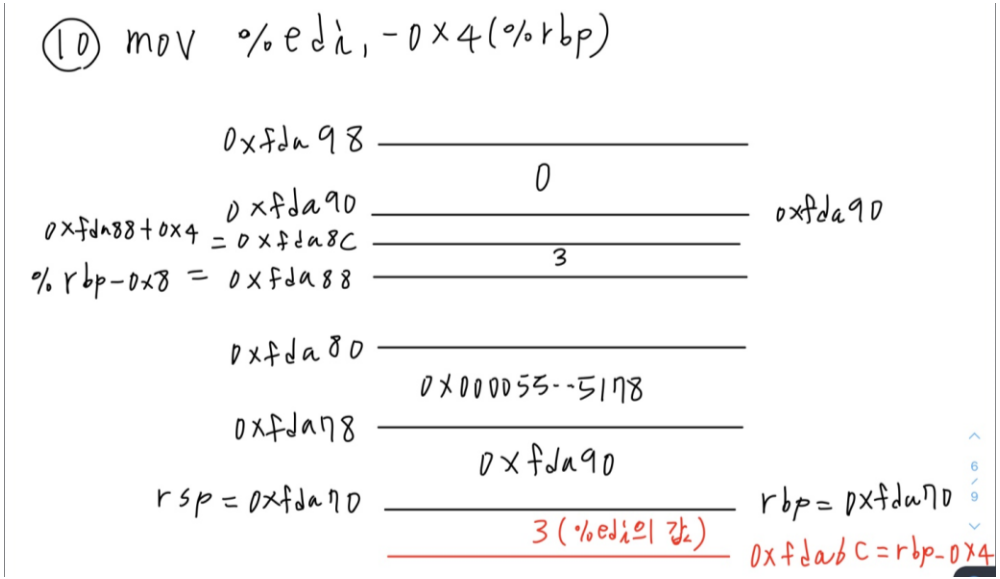
rsp의 값을 rbp에 복사. 8단계와 9단계를 거쳐 <test_func> 함수의 Stack Frame생성

기계어 분석(test_func, mov)

EDDI
Electronic Design Development Institute

mov

- 좌측의 값을 우측에 복사하시오.



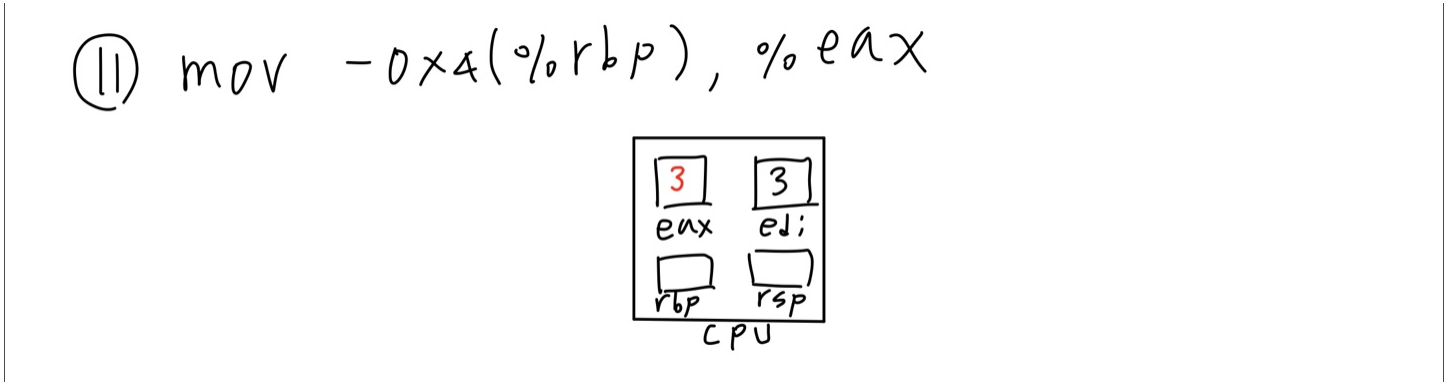
edi값을 rbp – 0x4의 메모리에 복사

기계어 분석(test_func, mov)

EDDI
Electronic Design Development Institute

mov

- 좌측의 값을 우측에 복사하시오.



rbp – 0x4의 메모리 값을 eax에 복사

기계어 분석(test_func, add)

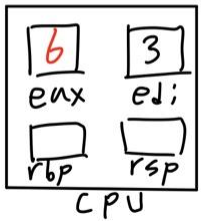
EDDI
Electronic Design Development Institute

add

- 덧셈

⑫ add %eax, %eax

3 + 3 = 6



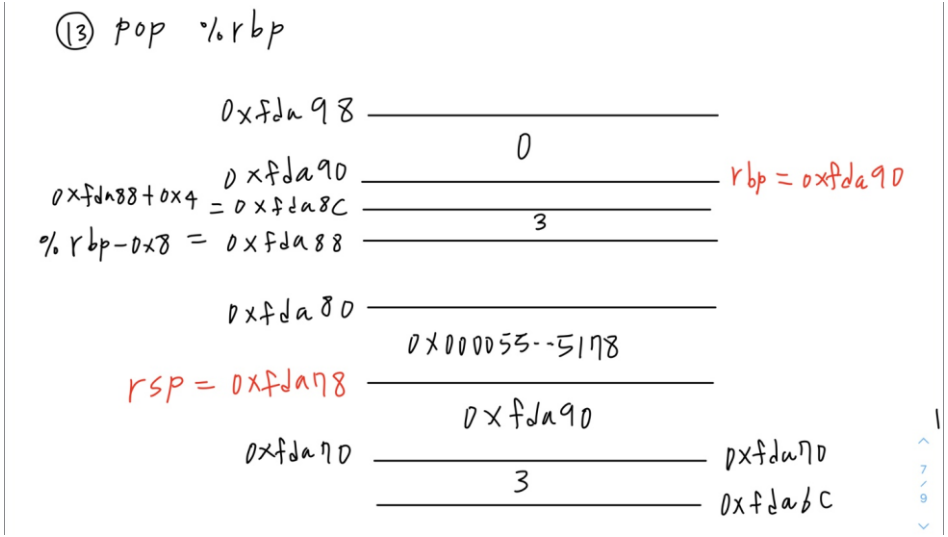
eax값과 eax값을 더하고 eax에 복사

기계어 분석(test_func, pop)

EDDI
Electronic Design Development Institute

pop

- 현재 스택의 최상위(rsp가 가리키는 메모리 공간)에서 값을 꺼내 뒤에 넣으시오.

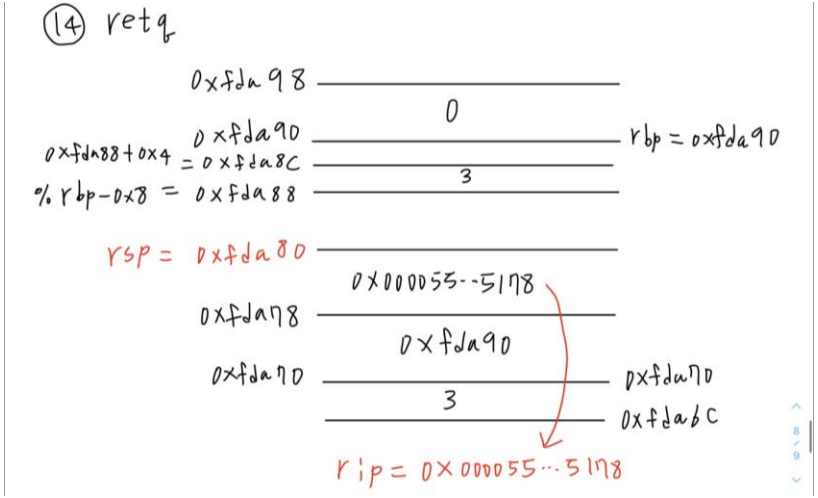


스택의 최상위에서 값(0xfda90)을 꺼내 rbp에 복사하고 rsp 이동

기계어 분석(test_func, retq)

retq

- pop $rip 와 같은 말



스택의 최상위에서 값을 꺼내 rip에 복사하고 rsp 이동

<test_func>의 Stack Frame 해제하고 다시 <main> 함수로 이동