EDDI
Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 5기

2023. 05. 19

최성효

확인 사항

EDDI
Electronic Design
Development Institute

cltd, shr, lea 확인 필요

어셈블리어 분석(1)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x0000555555555149 <+0>:     endbr64
0x000055555555514d <+4>:     push   %rbp
0x000055555555514e <+5>:     mov    %rsp,%rbp
0x0000555555555151 <+8>:     sub    $0x10,%rsp
0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

코드

```c
#include <stdio.h>

int main(void)
{
        int  End=5;
        int  sum=0;
        for (int  i=1; i<=End; ++i){
                if(i%2==1){
                        sum+=i;
                }
        }
        printf("%d까지의 홀수의 합은 %d\n",End,sum);
        return 0;
}
```

어셈블리어 분석(2)

EDDI
Electronic Design Development Institute

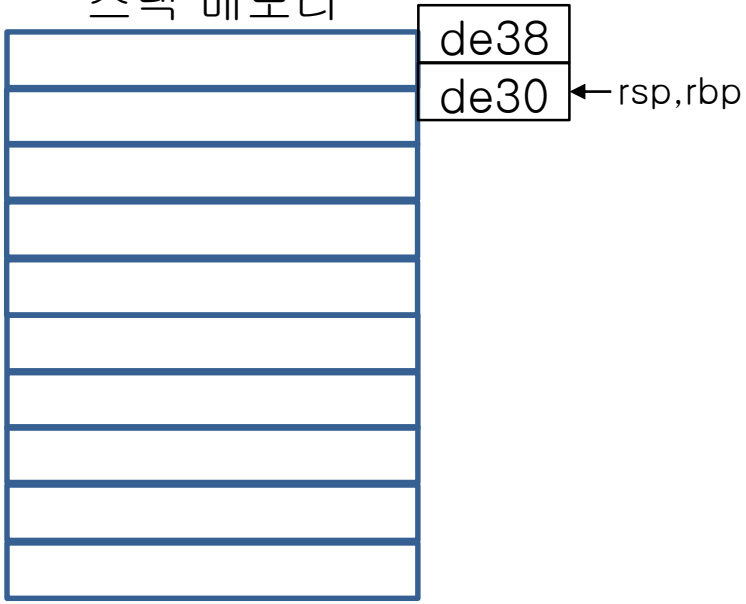
어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
=> 0x0000555555555151 <+8>:     sub    $0x10,%rsp
   0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
   0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
   0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
   0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
   0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
   0x000055555555516f <+38>:    cltd
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
   0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

레지스터

rbp	0x7fffffffde30	0x7fffffffde30
rsp	0x7fffffffde30	0x7fffffffde30

스택 메모리



mov는 rbp에 rsp 주소 대입
rsp=rbp로 같게한다

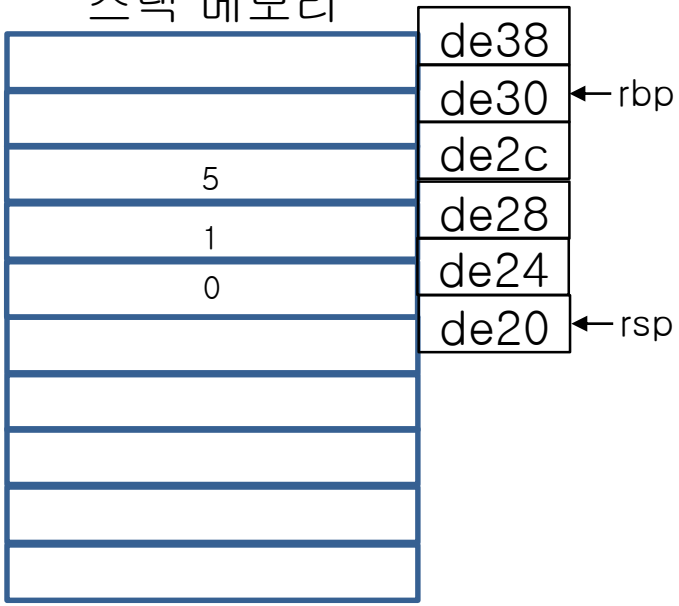
어셈블리어 분석(3)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     sub    $0x10,%rsp
   0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
   0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
=> 0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
   0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
   0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
   0x000055555555516f <+38>:    cltd
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
   0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

스택 메모리



End 값, sum값, for문 내 i값 저장

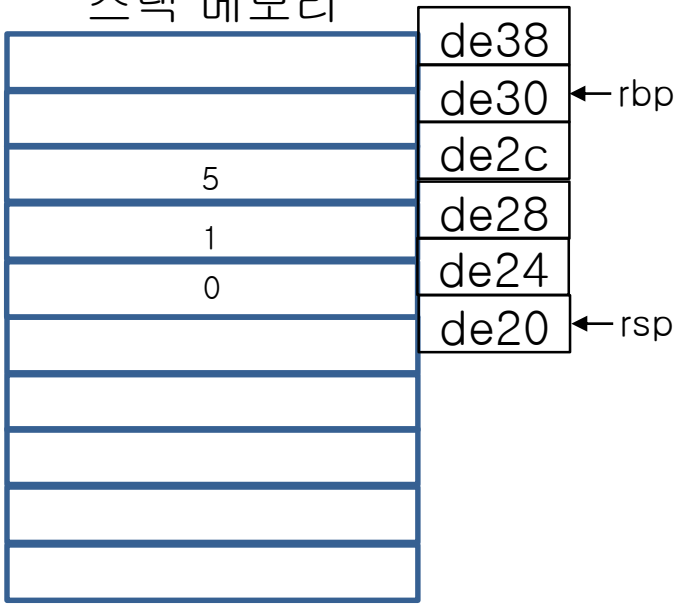
어셈블리어 분석(4)

EDDI
Electronic Design Development Institute

어셈블리어

```
Dump of assembler code for function main:
   0x0000555555555149 <+0>:     endbr64
   0x000055555555514d <+4>:     push   %rbp
   0x000055555555514e <+5>:     mov    %rsp,%rbp
   0x0000555555555151 <+8>:     sub    $0x10,%rsp
   0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
   0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
   0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
   0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
   0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
   0x000055555555516f <+38>:    cltd
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
   0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
   0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
   0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
   0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
   0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
=> 0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
   0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
   0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
   0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
   0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
   0x0000555555555197 <+78>:    mov    %eax,%esi
   0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
   0x00005555555551a0 <+87>:    mov    %rax,%rdi
   0x00005555555551a3 <+90>:    mov    $0x0,%eax
   0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
   0x00005555555551ad <+100>:   mov    $0x0,%eax
   0x00005555555551b2 <+105>:   leave
   0x00005555555551b3 <+106>:   ret
```

스택 메모리



jmp -> 특정 주소로 이동

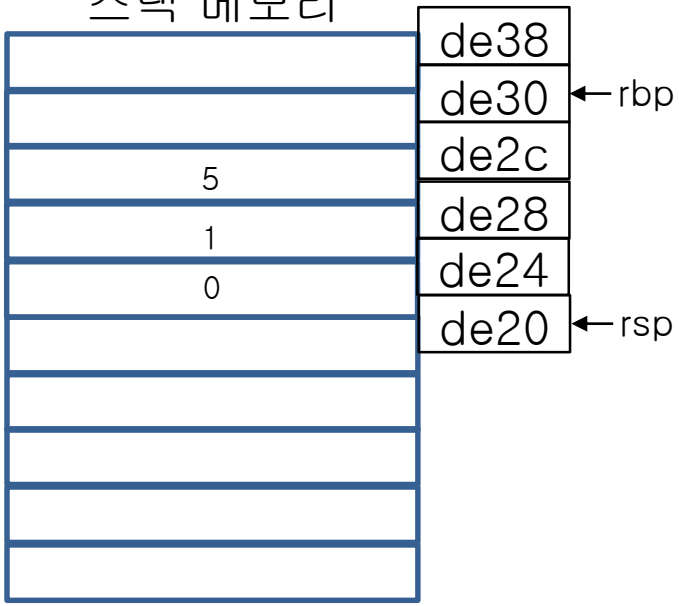
어셈블리어 분석(5)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x0000555555555149 <+0>:     endbr64
0x000055555555514d <+4>:     push   %rbp
0x000055555555514e <+5>:     mov    %rsp,%rbp
0x0000555555555151 <+8>:     sub    $0x10,%rsp
0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



eax에 0x8(%rbp)값(=1) 대입
0x8(%rbp)값(=1)< 0x4(%rbp)값(=5) 비교
-> End값과 i값 비교

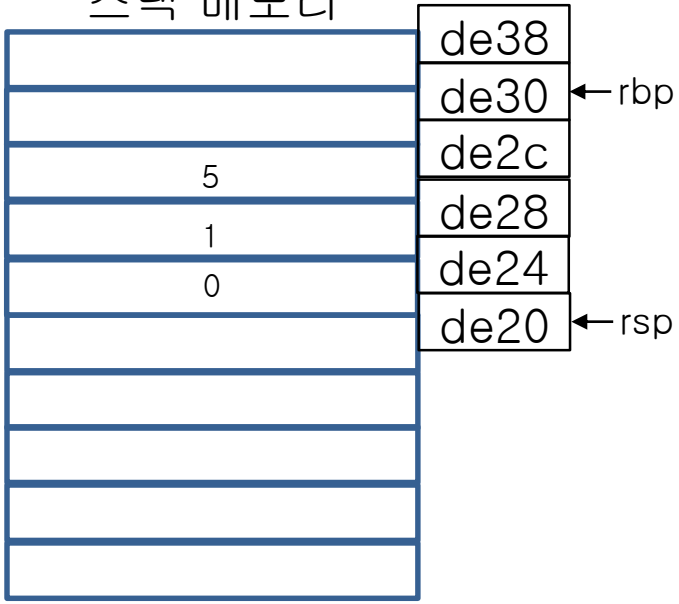
어셈블리어 분석(6)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x0000555555555149 <+0>:     endbr64
0x000055555555514d <+4>:     push   %rbp
0x000055555555514e <+5>:     mov    %rsp,%rbp
0x0000555555555151 <+8>:     sub    $0x10,%rsp
0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



jle = 왼쪽 인자가 오른쪽 인자보다 작거나 같으면 jmp

※jle= jump if less or eqaul

어셈블리어 분석(7)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
   0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
=> 0x000055555555516f <+38>:    cltd
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
   0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

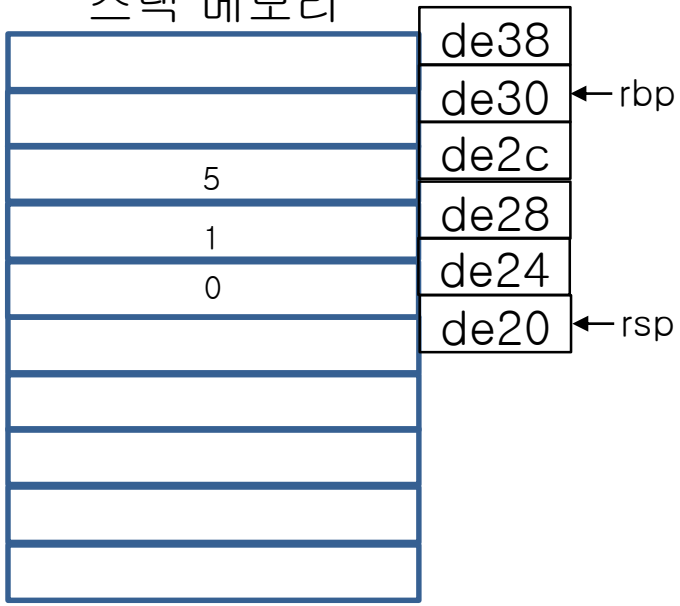
레지스터

```
(gdb) p $edx
$3 = -8360
```

→

```
(gdb) p $edx
$9 = 0
```

스택 메모리



cltd= cdq 명령어 확인 필요

어셈블리어 분석(8)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
=> 0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

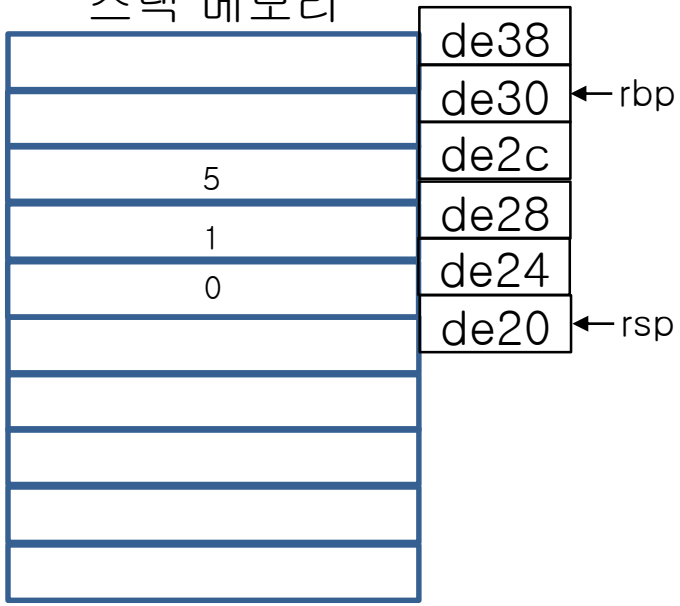
레지스터

```
(gdb) p $edx
$9 = 0
```

→

```
(gdb) p $edx
$10 = 0
```

스택 메모리



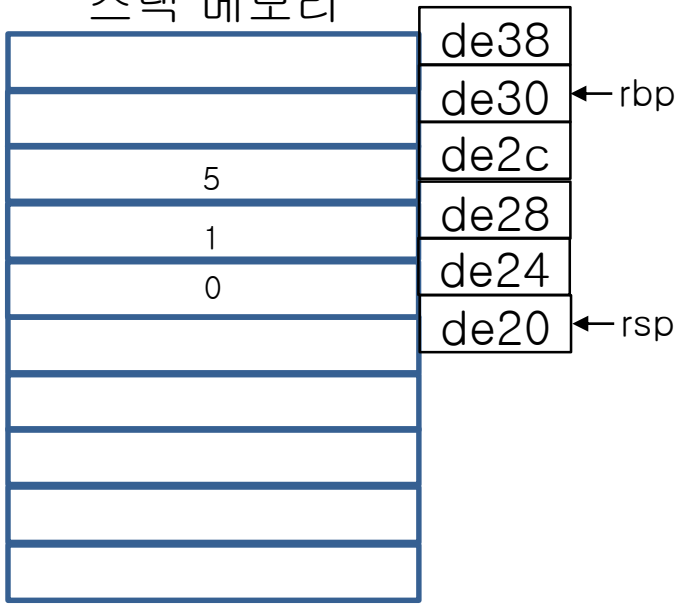
shr $0x1f,%edx
->shr은 오른쪽으로 $0x1f 만큼 비트를 민다
->$0x1f=31, -> 사인 부호만 남는다
-> 따라서 edx는 00이 된다.

어셈블리어 분석(9)

어셈블리어

```
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
=> 0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

스택 메모리



eax += edx
eax=1, edx=0;
edx가 위 +39줄에 의해서 거의 0이 되므로
eax값과 같음

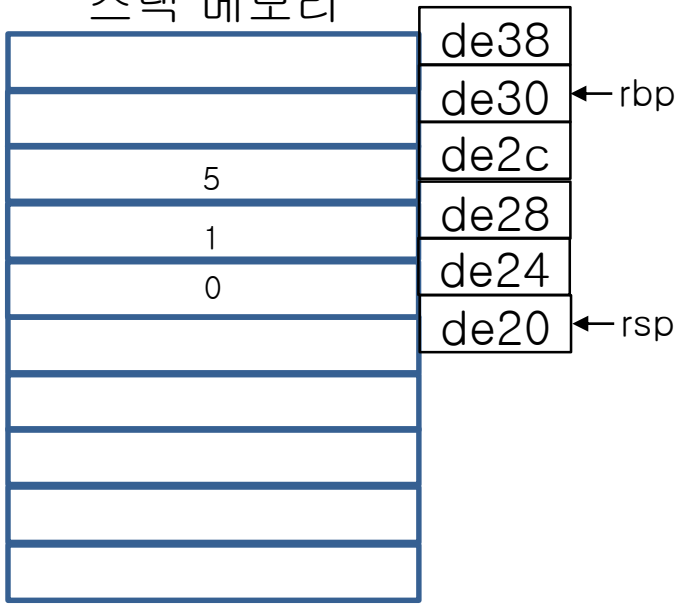
어셈블리어 분석(10)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
=> 0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

스택 메모리



1 & eax -> mod(i,2)를 의미
짝수일 경우 eax=0
홀수일 경우 eax=1

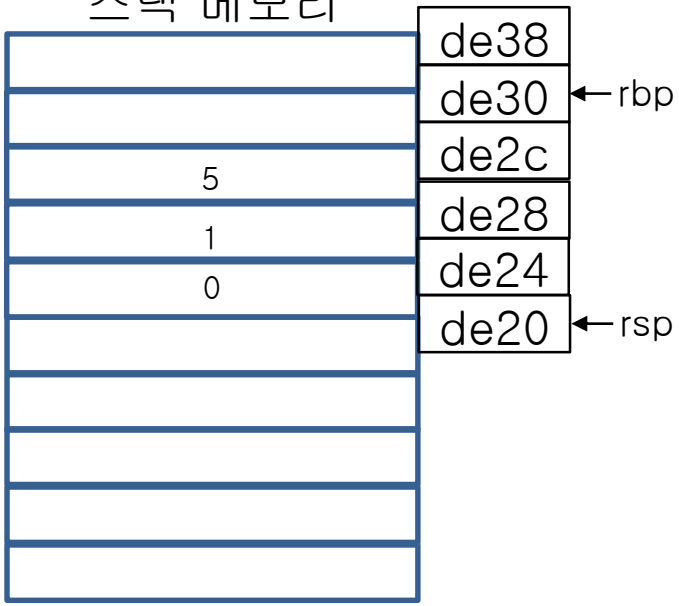
어셈블리어 분석(11)

EDDI
Electronic Design Development Institute

어셈블리어

```
   0x0000555555555170 <+39>:    shr    $0x1f,%edx
=> 0x0000555555555173 <+42>:    add    %edx,%eax
   0x0000555555555175 <+44>:    and    $0x1,%eax
   0x0000555555555178 <+47>:    sub    %edx,%eax
   0x000055555555517a <+49>:    cmp    $0x1,%eax
```

스택 메모리



sub %edx,%eax
eax-=edx
eax= 1 or 0,
edx는 0
연산 결과 eax=1 or 0

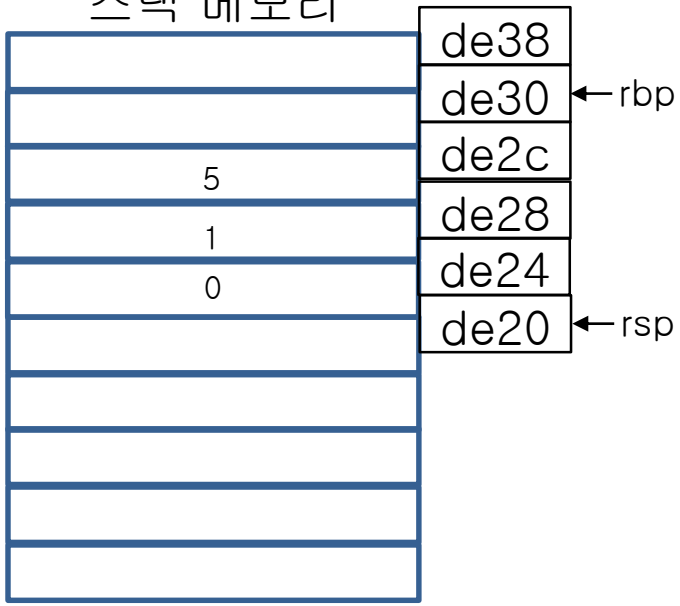
어셈블리어 분석(12)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



jne- eax와 같지 않을 경우 60번째 라인으로 점프
->if(i%2)==1 조건문 부분
※jne= jump if not equal
eax=1이므로 54번째 라인 실행

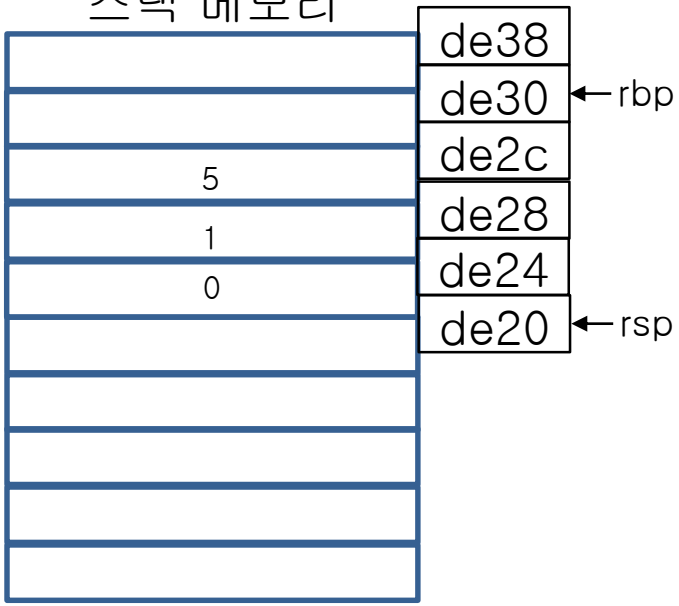
어셈블리어 분석(13)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



eax에 0x8(%rbp)=1 대입

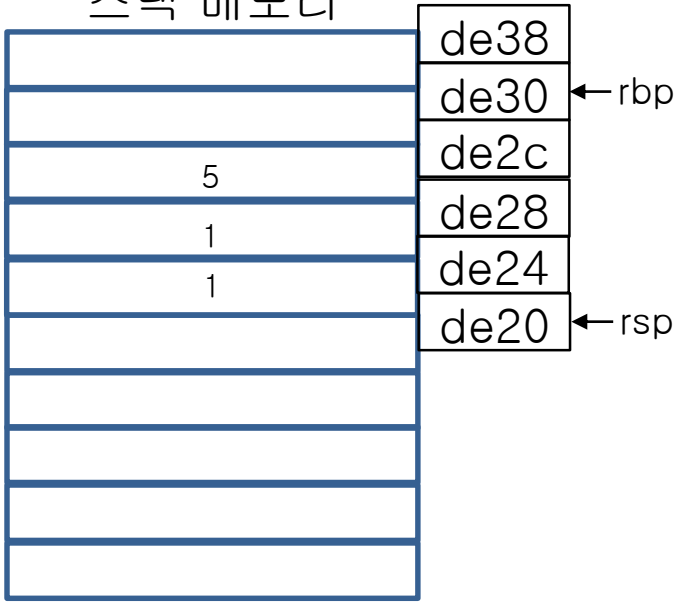
어셈블리어 분석(14)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



eax + 0xc(%rbp)=0
sum+=i 부분
sum=10이 됨

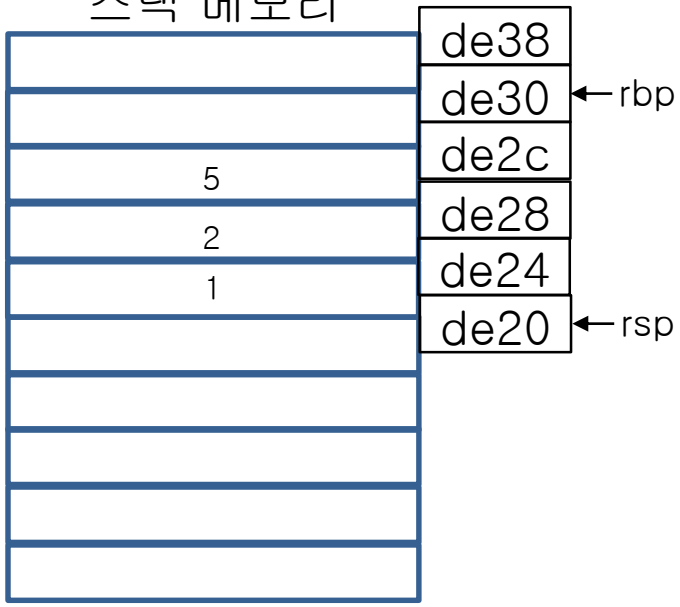
어셈블리어 분석(15)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



i+1이 됨

i++부분

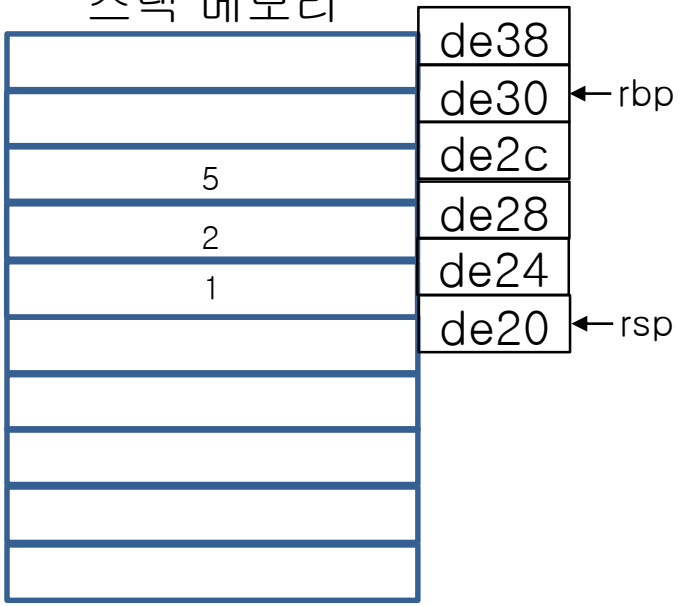
어셈블리어 분석(16)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



eax에 i값(=2) 대입

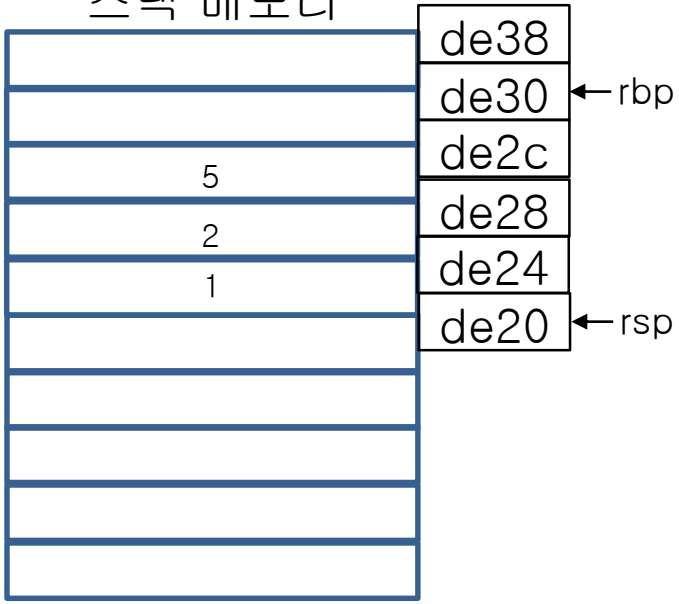
어셈블리어 분석(17)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



end값과 i값 비교
0x4(%rbp)=End
eax=i값(현재 i=2)

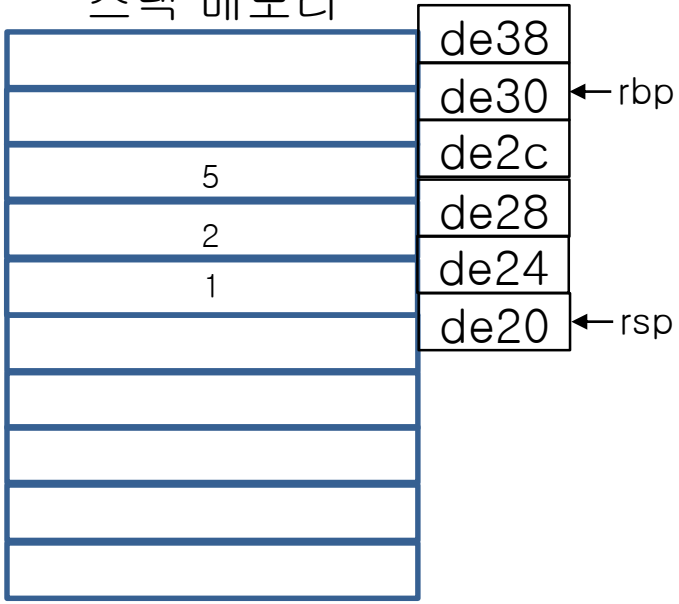
eax값이 작으므로 35번째 라인으로 점프

어셈블리어 분석(18)

어셈블리어

```
0x0000555555555149 <+0>:     endbr64
0x000055555555514d <+4>:     push   %rbp
0x000055555555514e <+5>:     mov    %rsp,%rbp
0x0000555555555151 <+8>:     sub    $0x10,%rsp
0x0000555555555155 <+12>:    movl   $0x5,-0x4(%rbp)
0x000055555555515c <+19>:    movl   $0x0,-0xc(%rbp)
0x0000555555555163 <+26>:    movl   $0x1,-0x8(%rbp)
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
```

스택 메모리



if 조건문 실행부분
eax는 i(=2)가 되며
edx=0
(+44)라인 실행후 eax=0
eax와 1이 같지 않음으로 60번째 라인으로 점프

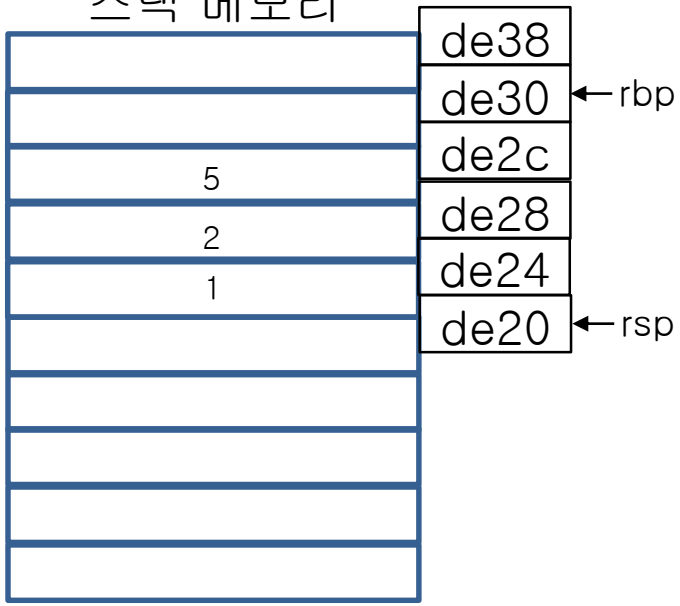
어셈블리어 분석(19)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
```

스택 메모리



for문내 for(i++,i<=end)
부분 실행

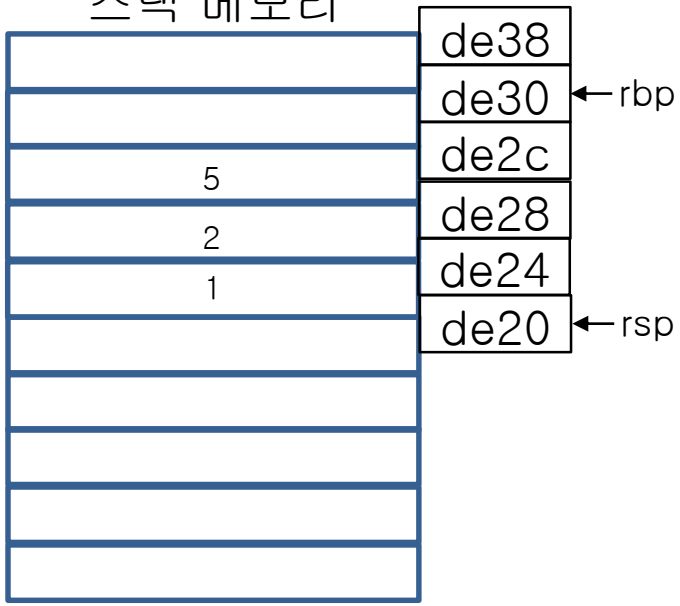
어셈블리어 분석(20)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555516a <+33>:    jmp    0x555555555189 <main+64>
0x000055555555516c <+35>:    mov    -0x8(%rbp),%eax
0x000055555555516f <+38>:    cltd
0x0000555555555170 <+39>:    shr    $0x1f,%edx
0x0000555555555173 <+42>:    add    %edx,%eax
0x0000555555555175 <+44>:    and    $0x1,%eax
0x0000555555555178 <+47>:    sub    %edx,%eax
0x000055555555517a <+49>:    cmp    $0x1,%eax
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
```

스택 메모리



for문내 for(i++,i<=end)
부분 실행

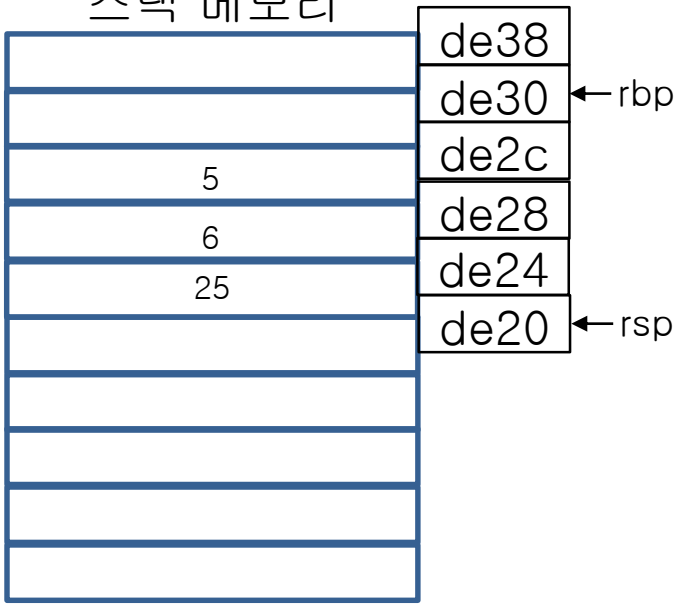
어셈블리어 분석(21,중간 과정 생략)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



i=6일 때
eax(=6)이 0x4(%rbp)(=end,5) 값 보다 큼으로
72번째 라인 실행

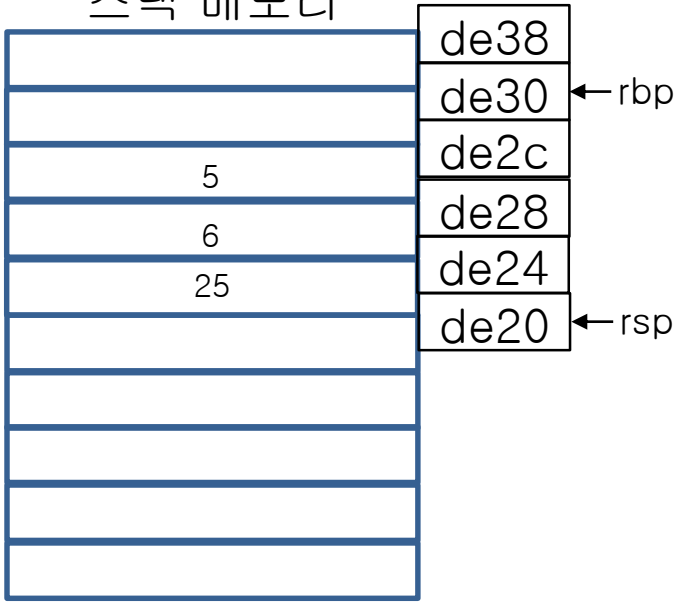
어셈블리어 분석(22,중간 과정 생략)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x000055555555517d <+52>:    jne    0x555555555185 <main+60>
0x000055555555517f <+54>:    mov    -0x8(%rbp),%eax
0x0000555555555182 <+57>:    add    %eax,-0xc(%rbp)
0x0000555555555185 <+60>:    addl   $0x1,-0x8(%rbp)
0x0000555555555189 <+64>:    mov    -0x8(%rbp),%eax
0x000055555555518c <+67>:    cmp    -0x4(%rbp),%eax
0x000055555555518f <+70>:    jle    0x55555555516c <main+35>
0x0000555555555191 <+72>:    mov    -0xc(%rbp),%edx
0x0000555555555194 <+75>:    mov    -0x4(%rbp),%eax
0x0000555555555197 <+78>:    mov    %eax,%esi
0x0000555555555199 <+80>:    lea    0xe68(%rip),%rax        # 0x555555556008
0x00005555555551a0 <+87>:    mov    %rax,%rdi
0x00005555555551a3 <+90>:    mov    $0x0,%eax
0x00005555555551a8 <+95>:    call   0x555555555050 <printf@plt>
0x00005555555551ad <+100>:   mov    $0x0,%eax
0x00005555555551b2 <+105>:   leave
0x00005555555551b3 <+106>:   ret
```

스택 메모리



edx에 sum값 대입(=25)
eax에 end값 대입(=5)
esi에 end값 대입(=5)
print 후 끝