리버스 엔지니어링 바이블

ch02. C문법과 디스어셈블링

저자 : 강병탁

김영철

2015. 10. 20.

함수의 기본 구조

```
Dump of assembler code for function add:
   0x08048421 <+0>:     push   %ebp
   0x08048422 <+1>:     mov    %esp,%ebp
   0x08048424 <+3>:     mov    0xc(%ebp),%eax
   0x08048427 <+6>:     mov    0x8(%ebp),%edx
   0x0804842a <+9>:     add    %edx,%eax
   0x0804842c <+11>:    pop    %ebp
   0x0804842d <+12>:    ret
End of assembler dump.
```

```
Dump of assembler code for function sum:
   0x080483ed <+0>:     push   %ebp
   0x080483ee <+1>:     mov    %esp,%ebp
   0x080483f0 <+3>:     sub    $0x10,%esp
   0x080483f3 <+6>:     mov    0xc(%ebp),%eax
   0x080483f6 <+9>:     mov    0x8(%ebp),%edx
   0x080483f9 <+12>:    add    %edx,%eax
   0x080483fb <+14>:    mov    %eax,-0x4(%ebp)
   0x080483fe <+17>:    mov    -0x4(%ebp),%eax
   0x08048401 <+20>:    leave
   0x08048402 <+21>:    ret
End of assembler dump.
```

함수의 호출 규약

1. cdecl 방식
2. stdcall 방식
3. fastcall 방식
4. thiscall 방식

http://tadis.tistory.com/53

함수의 호출 규약

```
#include <stdio.h>

int sum(int a, int b)
{
        int c = a + b;
        return c;
}

int main()
{
        sum(1, 2);
        return 0;
}
```

```
#include <stdio.h>

class Calculator
{
        public:
                int sum(int a, int b);
};

int Calculator::sum(int a, int b)
{
        int c = a + b;
        return c;
}

int main()
{
        Calculator cal;
        cal.sum(1, 2);
        return 0;
}
```

함수의 호출 규약

- cdecl 방식 – 보통의 함수들

```
Dump of assembler code for function main:
   0x08048403 <+0>:     push   ebp
   0x08048404 <+1>:     mov    ebp,esp
   0x08048406 <+3>:     sub    esp,0x8
   0x08048409 <+6>:     mov    DWORD PTR [esp+0x4],0x2
   0x08048411 <+14>:    mov    DWORD PTR [esp],0x1
   0x08048418 <+21>:    call   0x80483ed <sum>
   0x0804841d <+26>:    mov    eax,0x0
   0x08048422 <+31>:    leave
   0x08048423 <+32>:    ret
End of assembler dump.
```

```
Dump of assembler code for function sum:
   0x080483ed <+0>:     push   ebp
   0x080483ee <+1>:     mov    ebp,esp
   0x080483f0 <+3>:     sub    esp,0x10
   0x080483f3 <+6>:     mov    eax,DWORD PTR [ebp+0xc]
   0x080483f6 <+9>:     mov    edx,DWORD PTR [ebp+0x8]
   0x080483f9 <+12>:    add    eax,edx
   0x080483fb <+14>:    mov    DWORD PTR [ebp-0x4],eax
   0x080483fe <+17>:    mov    eax,DWORD PTR [ebp-0x4]
   0x08048401 <+20>:    leave
   0x08048402 <+21>:    ret
End of assembler dump.
```

함수의 호출 규약

- stdcall 방식 - 함수가 직접 스택을 정리

```
Dump of assembler code for function main:
   0x08048405 <+0>:     push   ebp
   0x08048406 <+1>:     mov    ebp,esp
   0x08048408 <+3>:     sub    esp,0x8
   0x0804840b <+6>:     mov    DWORD PTR [esp+0x4],0x2
   0x08048413 <+14>:    mov    DWORD PTR [esp],0x1
   0x0804841a <+21>:    call   0x80483ed <sum>
   0x0804841f <+26>:    sub    esp,0x8
   0x08048422 <+29>:    mov    eax,0x0
   0x08048427 <+34>:    leave
   0x08048428 <+35>:    ret
End of assembler dump.
```

```
Dump of assembler code for function sum:
   0x080483ed <+0>:     push   ebp
   0x080483ee <+1>:     mov    ebp,esp
   0x080483f0 <+3>:     sub    esp,0x10
   0x080483f3 <+6>:     mov    eax,DWORD PTR [ebp+0xc]
   0x080483f6 <+9>:     mov    edx,DWORD PTR [ebp+0x8]
   0x080483f9 <+12>:    add    eax,edx
   0x080483fb <+14>:    mov    DWORD PTR [ebp-0x4],eax
   0x080483fe <+17>:    mov    eax,DWORD PTR [ebp-0x4]
   0x08048401 <+20>:    leave
   0x08048402 <+21>:    ret    0x8
End of assembler dump.
```

함수의 호출 규약

- fastcall 방식 - 인자를 레지스터에 저장

```
Dump of assembler code for function main:
   0x08048409 <+0>:     push   ebp
   0x0804840a <+1>:     mov    ebp,esp
   0x0804840c <+3>:     mov    edx,0x2
   0x08048411 <+8>:     mov    eax,0x1
   0x08048416 <+13>:    call   0x80483ed <sum>
   0x0804841b <+18>:    mov    eax,0x0
   0x08048420 <+23>:    pop    ebp
   0x08048421 <+24>:    ret
End of assembler dump.
```

```
Dump of assembler code for function sum:
   0x080483ed <+0>:     push   ebp
   0x080483ee <+1>:     mov    ebp,esp
   0x080483f0 <+3>:     sub    esp,0x18
   0x080483f3 <+6>:     mov    DWORD PTR [ebp-0x14],eax
   0x080483f6 <+9>:     mov    DWORD PTR [ebp-0x18],edx
   0x080483f9 <+12>:    mov    eax,DWORD PTR [ebp-0x18]
   0x080483fc <+15>:    mov    edx,DWORD PTR [ebp-0x14]
   0x080483ff <+18>:    add    eax,edx
   0x08048401 <+20>:    mov    DWORD PTR [ebp-0x4],eax
   0x08048404 <+23>:    mov    eax,DWORD PTR [ebp-0x4]
   0x08048407 <+26>:    leave
   0x08048408 <+27>:    ret
End of assembler dump.
```

함수의 호출 규약

- fastcall 방식 - 인자의 수가 3개가 초과하는 경우

```
fastcall.c:4:1: warning: argument to ‘ regparm’  attribute larger than 3 [-Wattributes]
```

```
Dump of assembler code for function main:
   0x0804840d <+0>:     push   ebp
   0x0804840e <+1>:     mov    ebp,esp
   0x08048410 <+3>:     sub    esp,0x10
   0x08048413 <+6>:     mov    DWORD PTR [esp+0xc],0x4
   0x0804841b <+14>:    mov    DWORD PTR [esp+0x8],0x3
   0x08048423 <+22>:    mov    DWORD PTR [esp+0x4],0x2
   0x0804842b <+30>:    mov    DWORD PTR [esp],0x1
   0x08048432 <+37>:    call   0x80483ed <sum>
   0x08048437 <+42>:    mov    eax,0x0
   0x0804843c <+47>:    leave
   0x0804843d <+48>:    ret
End of assembler dump.
```

```
Dump of assembler code for function sum:
   0x080483ed <+0>:     push   ebp
   0x080483ee <+1>:     mov    ebp,esp
   0x080483f0 <+3>:     sub    esp,0x10
   0x080483f3 <+6>:     mov    eax,DWORD PTR [ebp+0xc]
   0x080483f6 <+9>:     mov    edx,DWORD PTR [ebp+0x8]
   0x080483f9 <+12>:    add    edx,eax
   0x080483fb <+14>:    mov    eax,DWORD PTR [ebp+0x10]
   0x080483fe <+17>:    add    edx,eax
   0x08048400 <+19>:    mov    eax,DWORD PTR [ebp+0x14]
   0x08048403 <+22>:    add    eax,edx
   0x08048405 <+24>:    mov    DWORD PTR [ebp-0x4],eax
   0x08048408 <+27>:    mov    eax,DWORD PTR [ebp-0x4]
   0x0804840b <+30>:    leave
   0x0804840c <+31>:    ret
End of assembler dump.
```

함수의 호출 규약

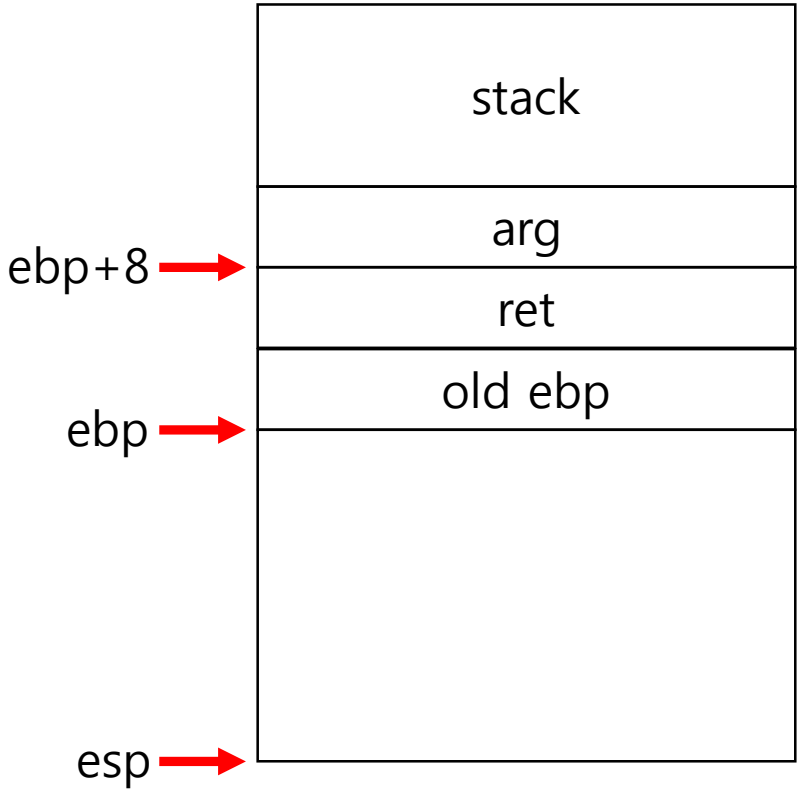
- thiscall 방식 - 객체 저장

```
Dump of assembler code for function main:
   0x08048404 <+0>:     push   ebp
   0x08048405 <+1>:     mov    ebp,esp
   0x08048407 <+3>:     sub    esp,0x1c
   0x0804840a <+6>:     mov    DWORD PTR [esp+0x8],0x2
   0x08048412 <+14>:    mov    DWORD PTR [esp+0x4],0x1
   0x0804841a <+22>:    lea    eax,[ebp-0x1]
   0x0804841d <+25>:    mov    DWORD PTR [esp],eax
   0x08048420 <+28>:    call   0x80483ee <_ZN10Calculator3sumEii>
   0x08048425 <+33>:    mov    eax,0x0
   0x0804842a <+38>:    leave
   0x0804842b <+39>:    ret
End of assembler dump.
```

```
Dump of assembler code for function _ZN10Calculator3sumEii:
   0x080483ee <+0>:     push   ebp
   0x080483ef <+1>:     mov    ebp,esp
   0x080483f1 <+3>:     sub    esp,0x10
   0x080483f4 <+6>:     mov    eax,DWORD PTR [ebp+0x10]
   0x080483f7 <+9>:     mov    edx,DWORD PTR [ebp+0xc]
   0x080483fa <+12>:    add    eax,edx
   0x080483fc <+14>:    mov    DWORD PTR [ebp-0x4],eax
   0x080483ff <+17>:    mov    eax,DWORD PTR [ebp-0x4]
   0x08048402 <+20>:    leave
   0x08048403 <+21>:    ret
End of assembler dump.
```

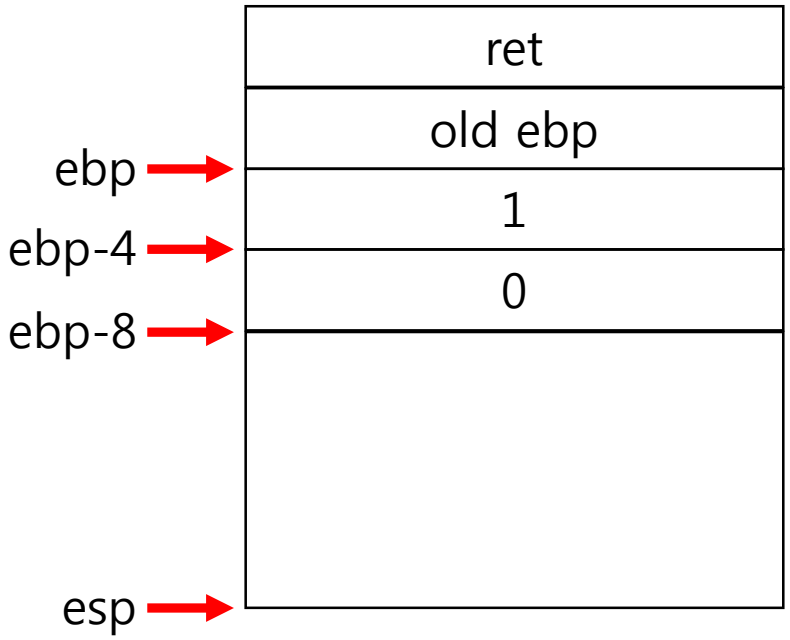
if문

```
Dump of assembler code for function checkA:
   0x0804841d <+0>:     push   ebp
   0x0804841e <+1>:     mov    ebp,esp
   0x08048420 <+3>:     sub    esp,0x18
   0x08048423 <+6>:     cmp    DWORD PTR [ebp+0x8],0x1
   0x08048427 <+10>:    jne    0x8048437 <checkA+26>
   0x08048429 <+12>:    mov    DWORD PTR [esp],0x8048500
   0x08048430 <+19>:    call   0x80482f0 <puts@plt>
   0x08048435 <+24>:    jmp    0x8048443 <checkA+38>
   0x08048437 <+26>:    mov    DWORD PTR [esp],0x8048507
   0x0804843e <+33>:    call   0x80482f0 <puts@plt>
   0x08048443 <+38>:    leave
   0x08048444 <+39>:    ret
End of assembler dump.
```



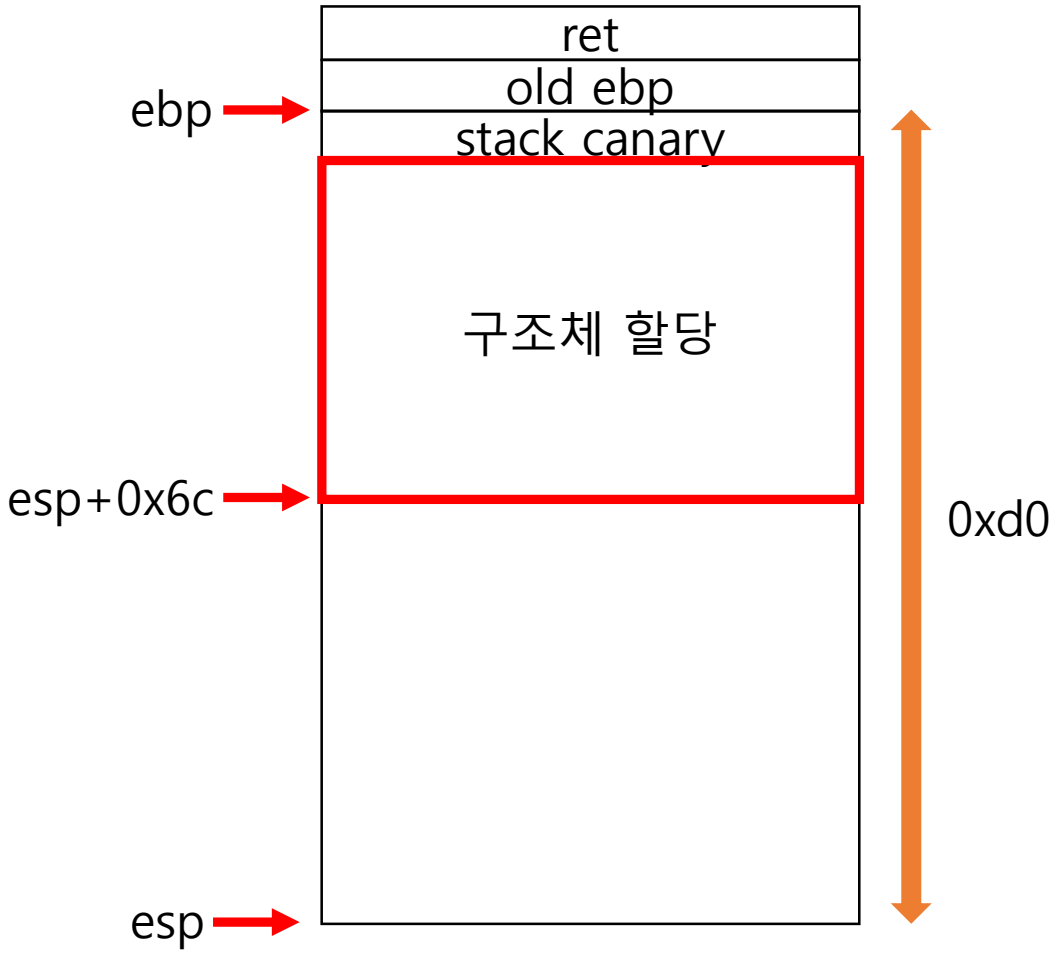
반복문

```
Dump of assembler code for function main:
   0x080483ed <+0>:     push   ebp
   0x080483ee <+1>:     mov    ebp,esp
   0x080483f0 <+3>:     sub    esp,0x10
   0x080483f3 <+6>:     mov    DWORD PTR [ebp-0x8],0x0
   0x080483fa <+13>:    mov    DWORD PTR [ebp-0x4],0x1
   0x08048401 <+20>:    jmp    0x804840d <main+32>
   0x08048403 <+22>:    mov    eax,DWORD PTR [ebp-0x4]
   0x08048406 <+25>:    add    DWORD PTR [ebp-0x8],eax
   0x08048409 <+28>:    add    DWORD PTR [ebp-0x4],0x1
   0x0804840d <+32>:    cmp    DWORD PTR [ebp-0x4],0x3e7
   0x08048414 <+39>:    jle    0x8048403 <main+22>
   0x08048416 <+41>:    leave
   0x08048417 <+42>:    ret
End of assembler dump.
```



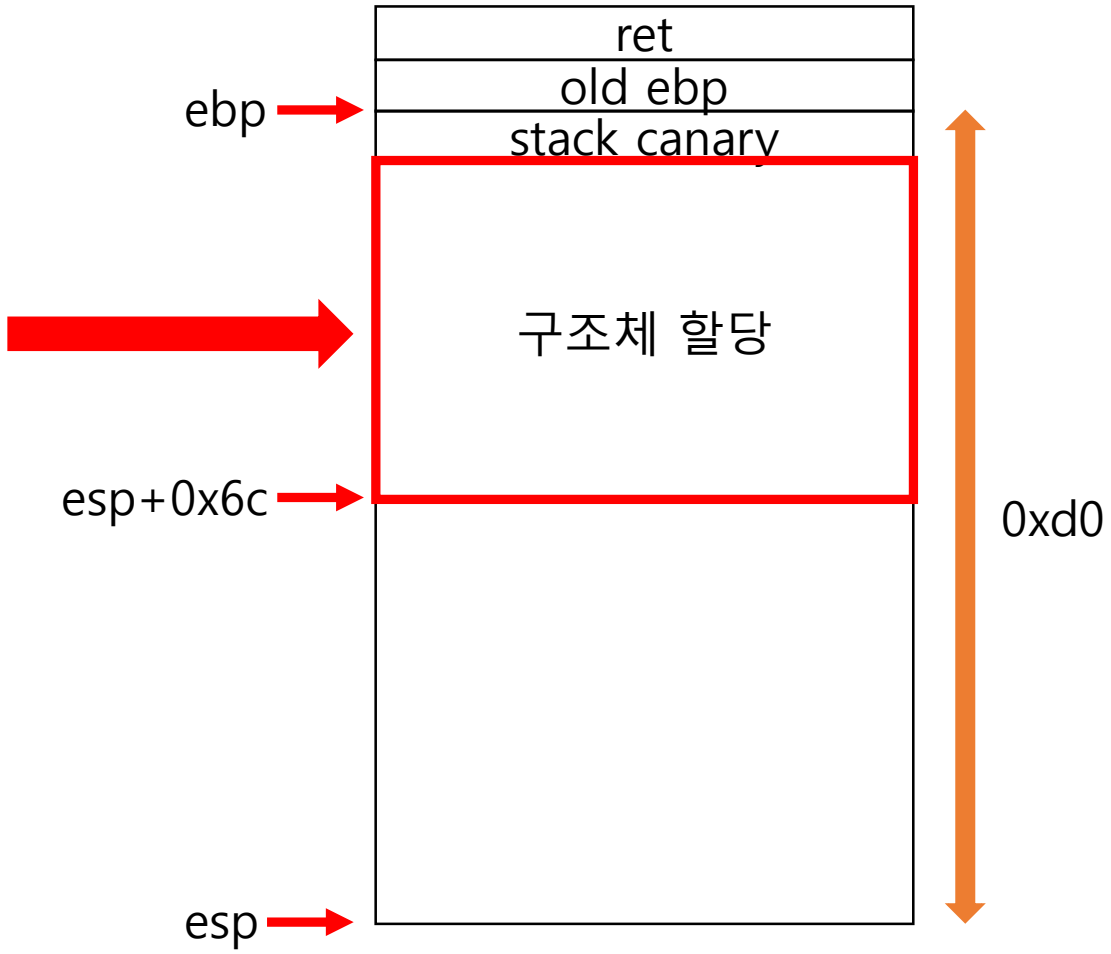
구조체

```
Dump of assembler code for function main:
   0x080484ab <+0>:     push   ebp
   0x080484ac <+1>:     mov    ebp,esp
   0x080484ae <+3>:     push   edi
   0x080484af <+4>:     push   esi
   0x080484b0 <+5>:     push   ebx
   0x080484b1 <+6>:     and    esp,0xfffffff0
   0x080484b4 <+9>:     sub    esp,0xd0
   0x080484ba <+15>:    mov    eax,gs:0x14
   0x080484c0 <+21>:    mov    DWORD PTR [esp+0xcc],eax
   0x080484c7 <+28>:    xor    eax,eax
   0x080484c9 <+30>:    lea    edx,[esp+0x6c]
   0x080484cd <+34>:    mov    ebx,0x8048640
   0x080484d2 <+39>:    mov    eax,0x18
   0x080484d7 <+44>:    mov    edi,edx
   0x080484d9 <+46>:    mov    esi,ebx
   0x080484db <+48>:    mov    ecx,eax
   0x080484dd <+50>:    rep movs DWORD PTR es:[edi],DWORD PTR ds:[esi]
   0x080484df <+52>:    mov    edx,esp
   0x080484e1 <+54>:    lea    ebx,[esp+0x6c]
   0x080484e5 <+58>:    mov    eax,0x18
   0x080484ea <+63>:    mov    edi,edx
   0x080484ec <+65>:    mov    esi,ebx
   0x080484ee <+67>:    mov    ecx,eax
   0x080484f0 <+69>:    rep movs DWORD PTR es:[edi],DWORD PTR ds:[esi]
   0x080484f2 <+71>:    call   0x804846d <printPerson>
   0x080484f7 <+76>:    mov    edx,DWORD PTR [esp+0xcc]
   0x080484fe <+83>:    xor    edx,DWORD PTR gs:0x14
   0x08048505 <+90>:    je     0x804850c <main+97>
   0x08048507 <+92>:    call   0x8048340 <__stack_chk_fail@plt>
   0x0804850c <+97>:    lea    esp,[ebp-0xc]
   0x0804850f <+100>:   pop    ebx
   0x08048510 <+101>:   pop    esi
```



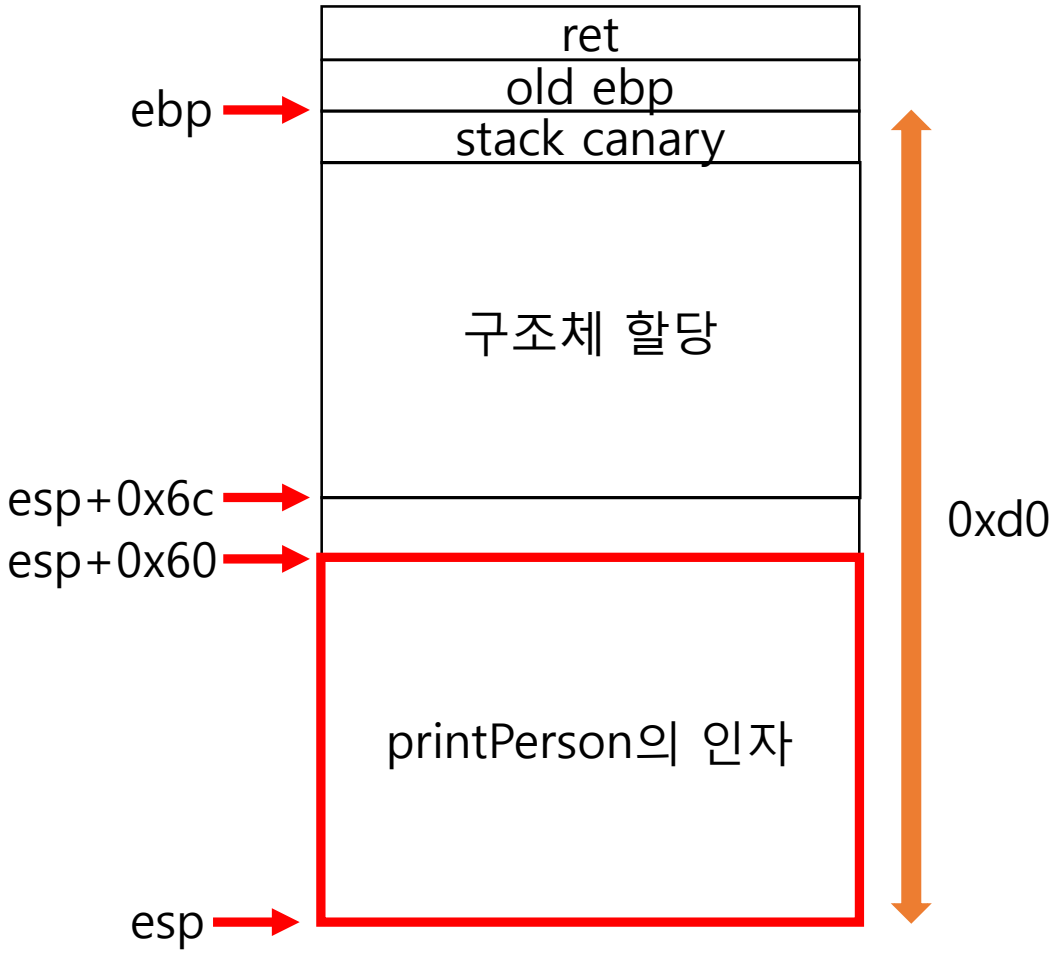
구조체

```
(gdb) x/24x $esi
0x8048640:      0xec80b9ea      0xb2ec8198      0x000000a0      0x00000018
0x8048650:      0xedb4bbec      0x84eda893      0xb5b3eab0      0xea9995ed
0x8048660:      0x0000bcb3      0x00000004      0xec9c84ec      0x0000b89a
0x8048670:      0x00000000      0x00000000      0x00000000      0x00000000
0x8048680:      0x00000000      0x00000000      0x00000000      0x00000000
0x8048690:      0x2d303130      0x32323939      0x3537302d      0x00000038
```



구조체

```
Dump of assembler code for function main:
   0x080484ab <+0>:     push   ebp
   0x080484ac <+1>:     mov    ebp,esp
   0x080484ae <+3>:     push   edi
   0x080484af <+4>:     push   esi
   0x080484b0 <+5>:     push   ebx
   0x080484b1 <+6>:     and    esp,0xfffffff0
   0x080484b4 <+9>:     sub    esp,0xd0
   0x080484ba <+15>:    mov    eax,gs:0x14
   0x080484c0 <+21>:    mov    DWORD PTR [esp+0xcc],eax
   0x080484c7 <+28>:    xor    eax,eax
   0x080484c9 <+30>:    lea    edx,[esp+0x6c]
   0x080484cd <+34>:    mov    ebx,0x8048640
   0x080484d2 <+39>:    mov    eax,0x18
   0x080484d7 <+44>:    mov    edi,edx
   0x080484d9 <+46>:    mov    esi,ebx
   0x080484db <+48>:    mov    ecx,eax
   0x080484dd <+50>:    rep movs DWORD PTR es:[edi],DWORD PTR ds:[esi]
   0x080484df <+52>:    mov    edx,esp
   0x080484e1 <+54>:    lea    ebx,[esp+0x6c]
   0x080484e5 <+58>:    mov    eax,0x18
   0x080484ea <+63>:    mov    edi,edx
   0x080484ec <+65>:    mov    esi,ebx
   0x080484ee <+67>:    mov    ecx,eax
   0x080484f0 <+69>:    rep movs DWORD PTR es:[edi],DWORD PTR ds:[esi]
   0x080484f2 <+71>:    call   0x804846d <printPerson>
   0x080484f7 <+76>:    mov    edx,DWORD PTR [esp+0xcc]
   0x080484fe <+83>:    xor    edx,DWORD PTR gs:0x14
   0x08048505 <+90>:    je     0x804850c <main+97>
   0x08048507 <+92>:    call   0x8048340 <__stack_chk_fail@plt>
   0x0804850c <+97>:    lea    esp,[ebp-0xc]
   0x0804850f <+100>:   pop    ebx
   0x08048510 <+101>:   pop    esi
```



구조체

```
Dump of assembler code for function printPerson:
   0x0804846d <+0>:     push   ebp
   0x0804846e <+1>:     mov    ebp,esp
   0x08048470 <+3>:     sub    esp,0x28
   0x08048473 <+6>:     mov    edx,DWORD PTR [ebp+0x2c]
   0x08048476 <+9>:     mov    eax,DWORD PTR [ebp+0x14]
   0x08048479 <+12>:    lea    ecx,[ebp+0x58]
   0x0804847c <+15>:    mov    DWORD PTR [esp+0x18],ecx
   0x08048480 <+19>:    lea    ecx,[ebp+0x30]
   0x08048483 <+22>:    mov    DWORD PTR [esp+0x14],ecx
   0x08048487 <+26>:    mov    DWORD PTR [esp+0x10],edx
   0x0804848b <+30>:    lea    edx,[ebp+0x18]
   0x0804848e <+33>:    mov    DWORD PTR [esp+0xc],edx
   0x08048492 <+37>:    mov    DWORD PTR [esp+0x8],eax
   0x08048496 <+41>:    lea    eax,[ebp+0x8]
   0x08048499 <+44>:    mov    DWORD PTR [esp+0x4],eax
   0x0804849d <+48>:    mov    DWORD PTR [esp],0x80485e0
   0x080484a4 <+55>:    call   0x8048330 <printf@plt>
   0x080484a9 <+60>:    leave
   0x080484aa <+61>:    ret
End of assembler dump.
```

```
(gdb) x/s 0x80485e0
0x80485e0:      "name : %s\nage : %d\nmajor : %s\ngrade : %d\naddress : %s\nphone : %s\n"
```