

StackFramePractice

안태진(taejin@codecure.smuc.ac.kr)

상명대학교 보안동아리 CodeCure

목차

- Environment
 - Platform
 - Files
- StackFramePractice1.c
- StackFramePractice2.c
 - Stack Canary
 - Process

Environment

- Platform
 - OS: Ubuntu 18.04 (64bit)
 - Windows Ubuntu (Microsoft Store에서 다운)
- Language: C, Python(2.7.15+)
- Compiler: gcc 7.4.0
- Debugger: gdb-peda
 - gdb version 8.1.0

목차

- Environment
 - Platform
 - Files
- StackFramePractice1.c
- StackFramePractice2.c
 - Stack Canary
 - Process

Environment

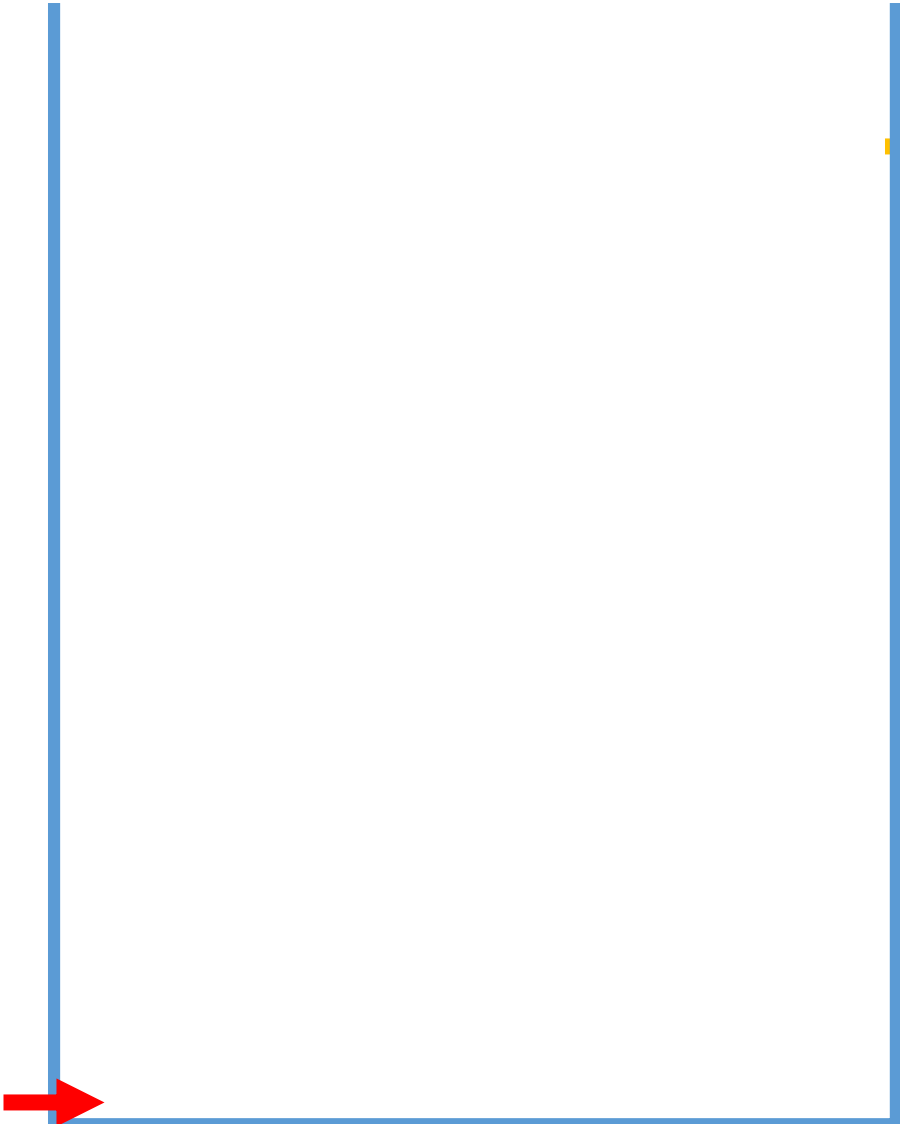
- Files
 - NAS/2019_2ndSemester/bof/Codes/
 - StackFramePractice1.c
 - StackFramePractice2.c

목차

- Environment
 - Platform
 - Files
- StackFramePractice1.c
- StackFramePractice2.c
 - Stack Canary
 - Process

StackFramePractice1.c

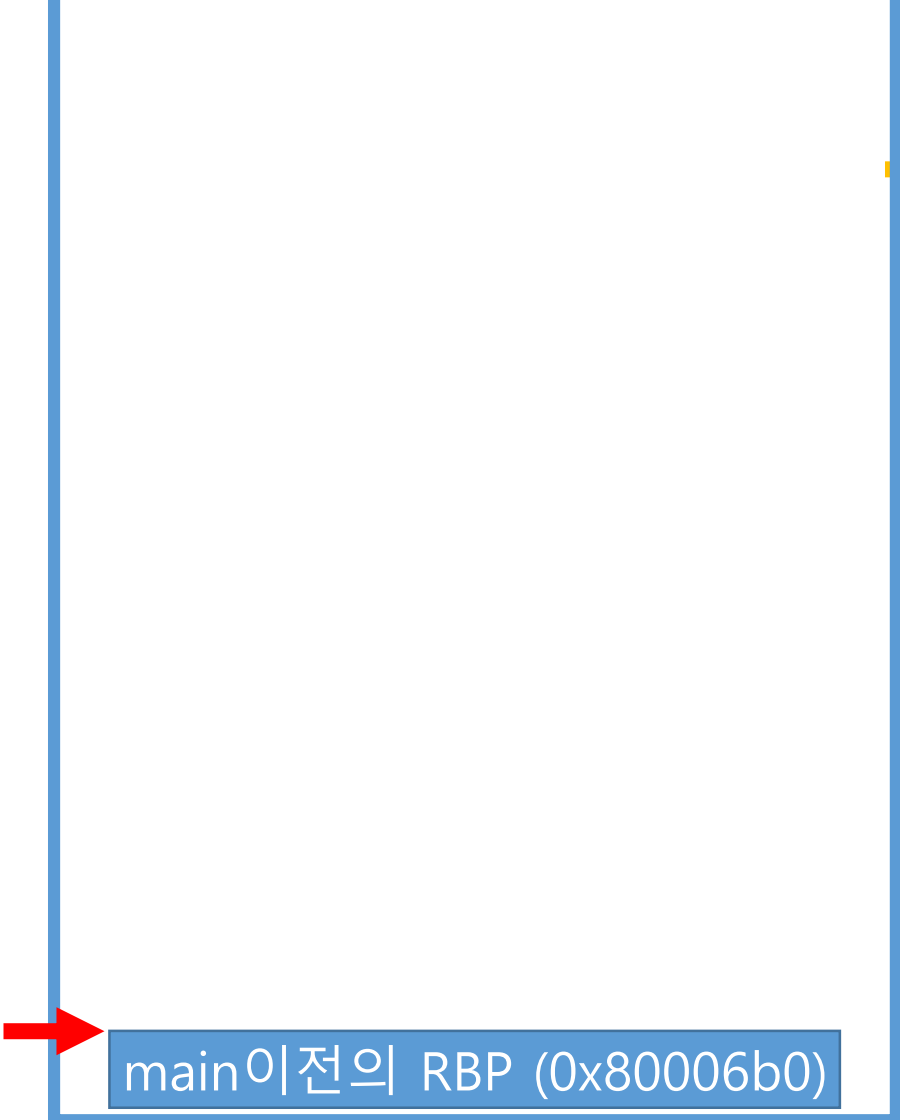
- StackFramePractice1.c (main) (1/17)



```
0x000000000800064a <+0>:  push    rbp
0x000000000800064b <+1>:  mov     rsp,rsp
0x000000000800064e <+4>:  sub     rsp,0x10
0x0000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc],0xa
0x0000000008000659 <+15>: mov     DWORD PTR [rbp-0x8],0x14
0x0000000008000660 <+22>: mov     edx,DWORD PTR [rbp-0x8]
0x0000000008000663 <+25>: mov     eax,DWORD PTR [rbp-0xc]
0x0000000008000666 <+28>: mov     esi,edx
0x0000000008000668 <+30>: mov     edi,eax
0x000000000800066a <+32>:  call    0x800068f <Add>
0x000000000800066f <+37>: mov     DWORD PTR [rbp-0x4],eax
0x0000000008000672 <+40>: mov     eax,DWORD PTR [rbp-0x4]
0x0000000008000675 <+43>: mov     esi,eax
0x0000000008000677 <+45>: lea     rdi,[rip+0xb6]          # 0x8000734
0x000000000800067e <+52>: mov     eax,0x0
0x0000000008000683 <+57>:  call    0x8000520 <printf@plt>
0x0000000008000688 <+62>: mov     eax,0x0
0x000000000800068d <+67>: leave
0x000000000800068e <+68>:  ret
```

StackFramePractice1.c

- StackFramePractice1.c (main) (2/17)

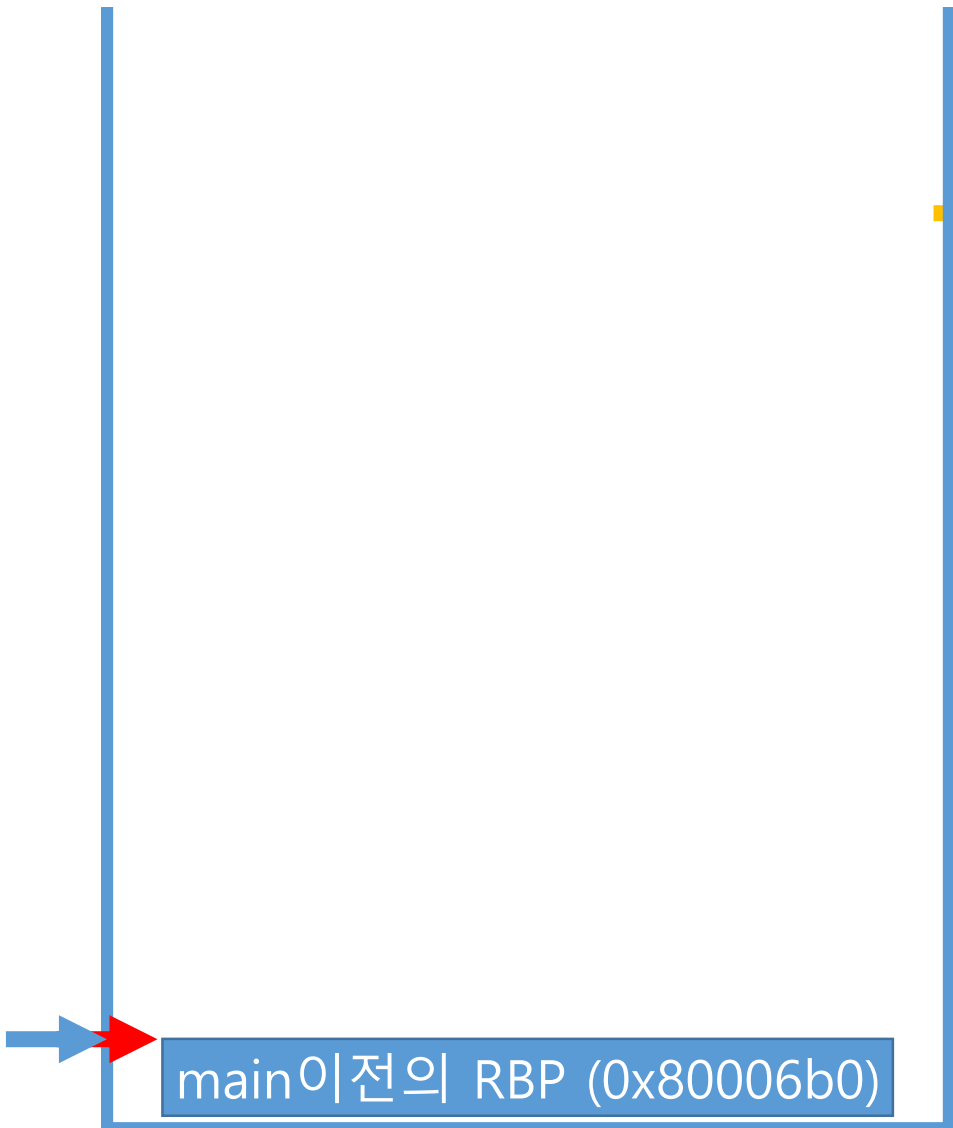


```
0x00000000800064a <+0>: push    rbp
0x00000000800064b <+1>: mov     rbp, rsp
0x00000000800064e <+4>: sub     rsp, 0x10
0x000000008000652 <+8>: mov     DWORD PTR [rbp-0xc], 0xa
0x000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x000000008000666 <+28>: mov     esi, edx
0x000000008000668 <+30>: mov     edi, eax
0x00000000800066a <+32>: call    0x800068f <Add>
0x00000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x000000008000675 <+43>: mov     esi, eax
0x000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x00000000800067e <+52>: mov     eax, 0x0
0x000000008000683 <+57>: call    0x8000520 <printf@plt>
0x000000008000688 <+62>: mov     eax, 0x0
0x00000000800068d <+67>: leave
0x00000000800068e <+68>: ret
```

main이전의 RBP (0x80006b0)

StackFramePractice1.c

- StackFramePractice1.c (main) (3/17)

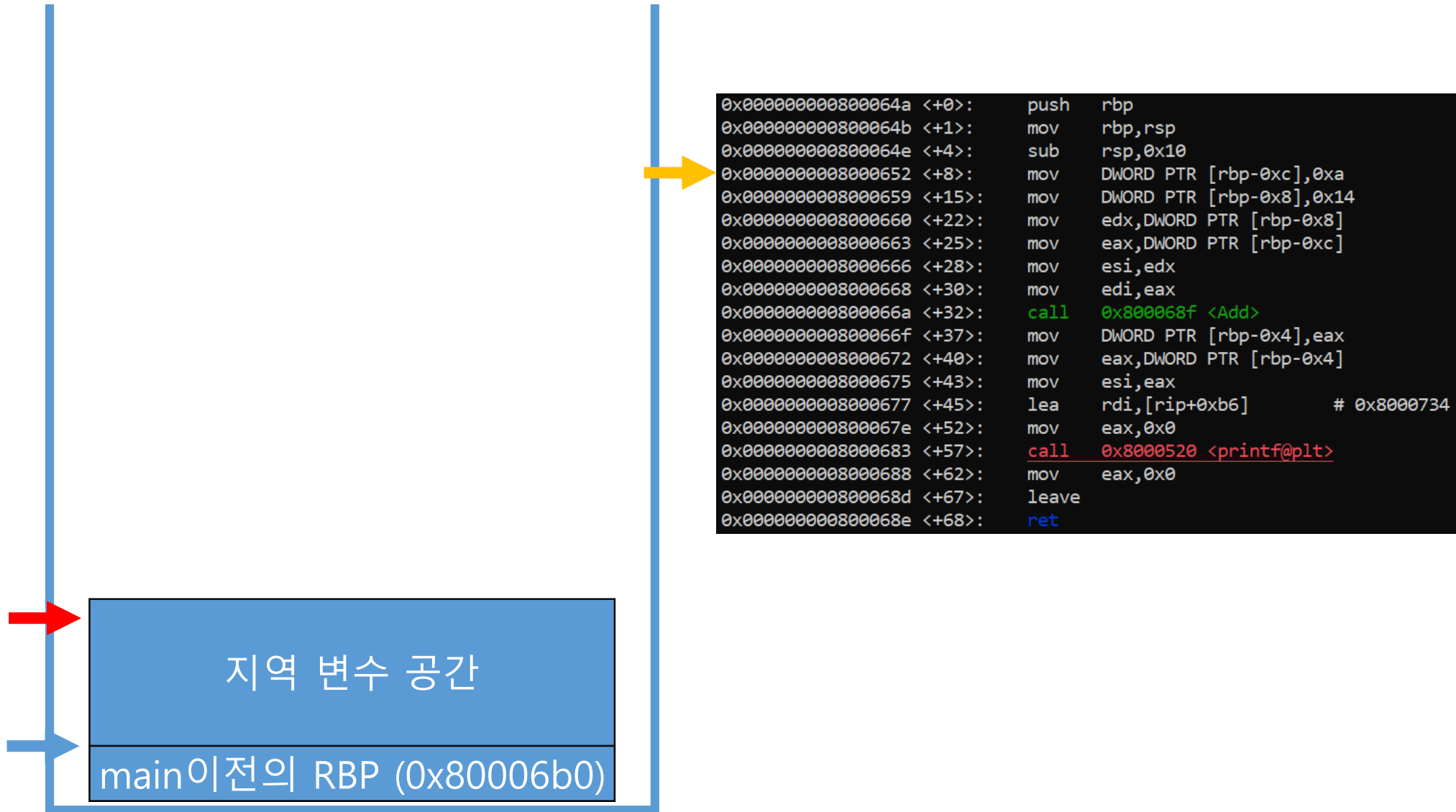


```
0x00000000800064a <+0>:  push    rbp
0x00000000800064b <+1>:  mov     rbp, rsp
0x00000000800064e <+4>:  sub     rsp, 0x10
0x000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x000000008000666 <+28>: mov     esi, edx
0x000000008000668 <+30>: mov     edi, eax
0x00000000800066a <+32>:  call    0x800068f <Add>
0x00000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x000000008000675 <+43>: mov     esi, eax
0x000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x00000000800067e <+52>: mov     eax, 0x0
0x000000008000683 <+57>:  call    0x8000520 <printf@plt>
0x000000008000688 <+62>: mov     eax, 0x0
0x00000000800068d <+67>:  leave
0x00000000800068e <+68>:  ret
```

main이전의 RBP (0x80006b0)

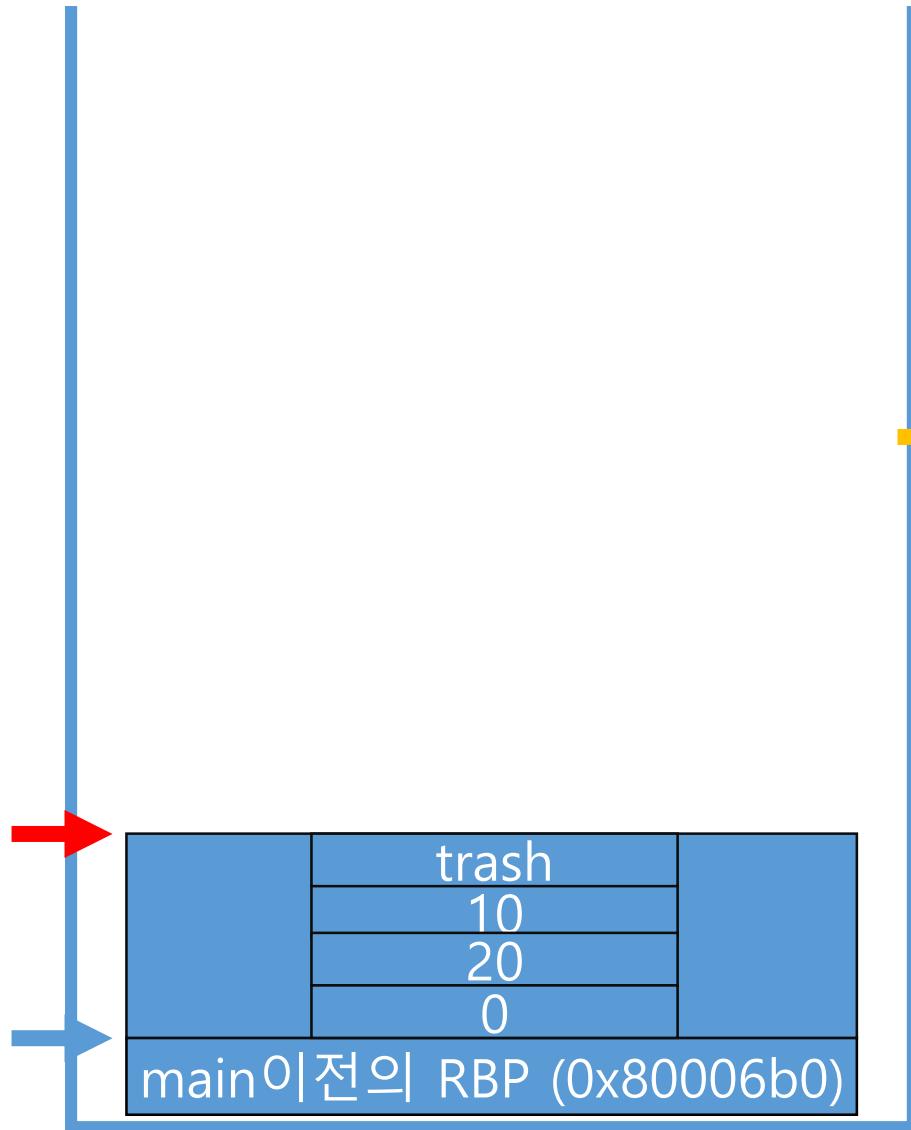
StackFramePractice1.c

- StackFramePractice1.c (main) (4/17)



StackFramePractice1.c

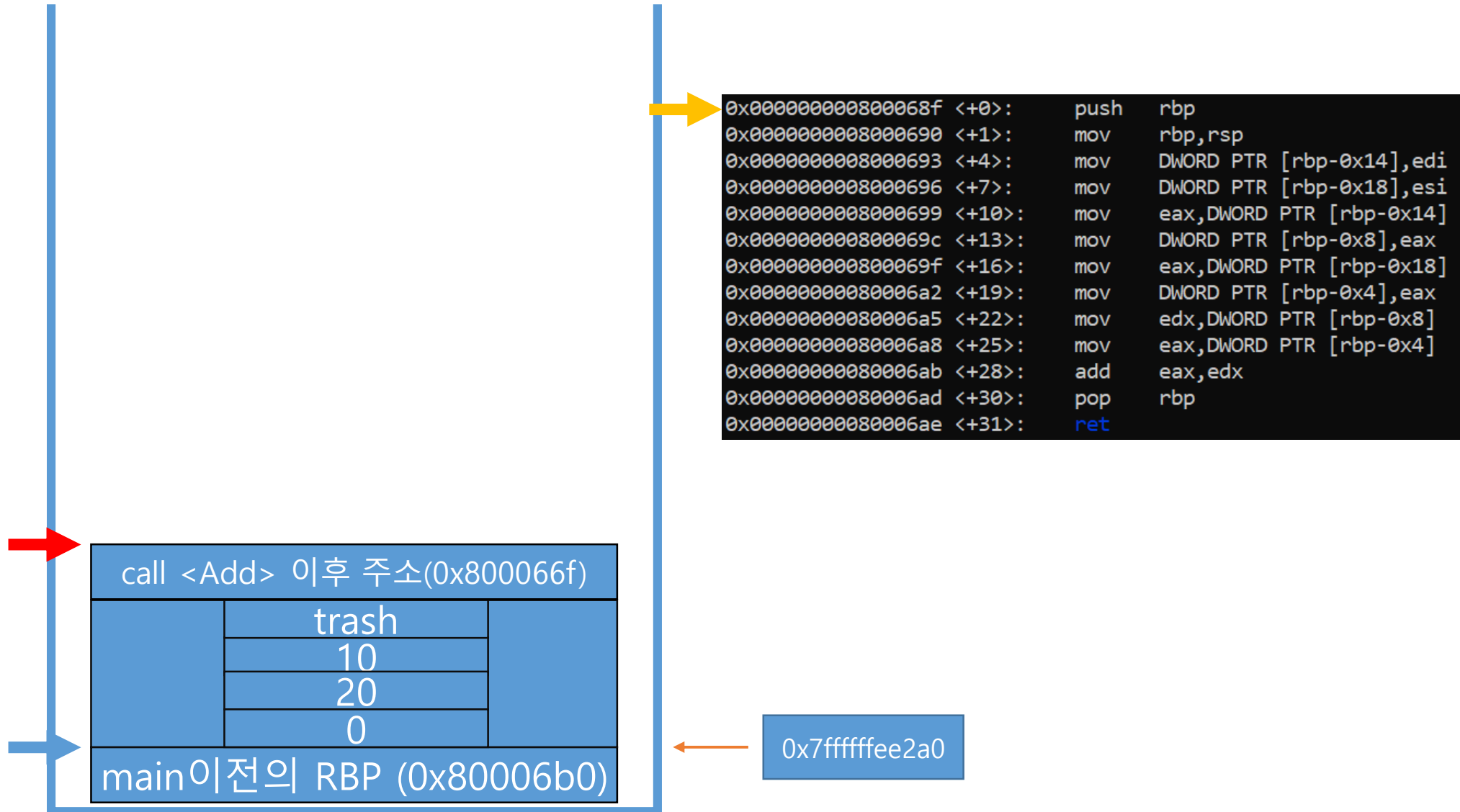
- StackFramePractice1.c (main) (5/17)



```
0x00000000800064a <+0>:  push    rbp
0x00000000800064b <+1>:  mov     rbp, rsp
0x00000000800064e <+4>:  sub     rsp, 0x10
0x000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x000000008000666 <+28>: mov     esi, edx
0x000000008000668 <+30>: mov     edi, eax
0x00000000800066a <+32>: call    0x800068f <Add>
0x00000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x000000008000675 <+43>: mov     esi, eax
0x000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x00000000800067e <+52>: mov     eax, 0x0
0x000000008000683 <+57>: call    0x8000520 <printf@plt>
0x000000008000688 <+62>: mov     eax, 0x0
0x00000000800068d <+67>: leave
0x00000000800068e <+68>: ret
```

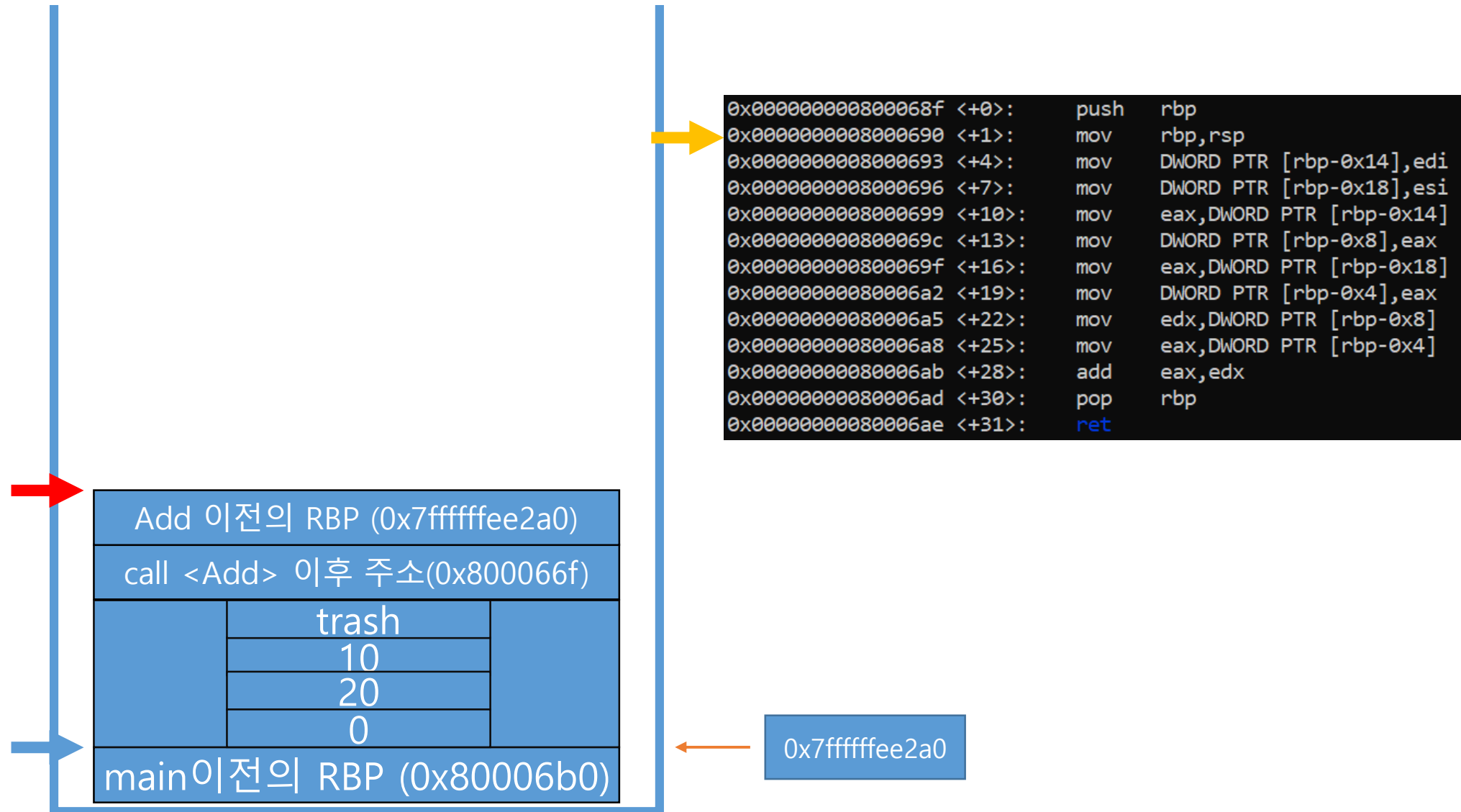
StackFramePractice1.c

- StackFramePractice1.c (Add) (6/17)



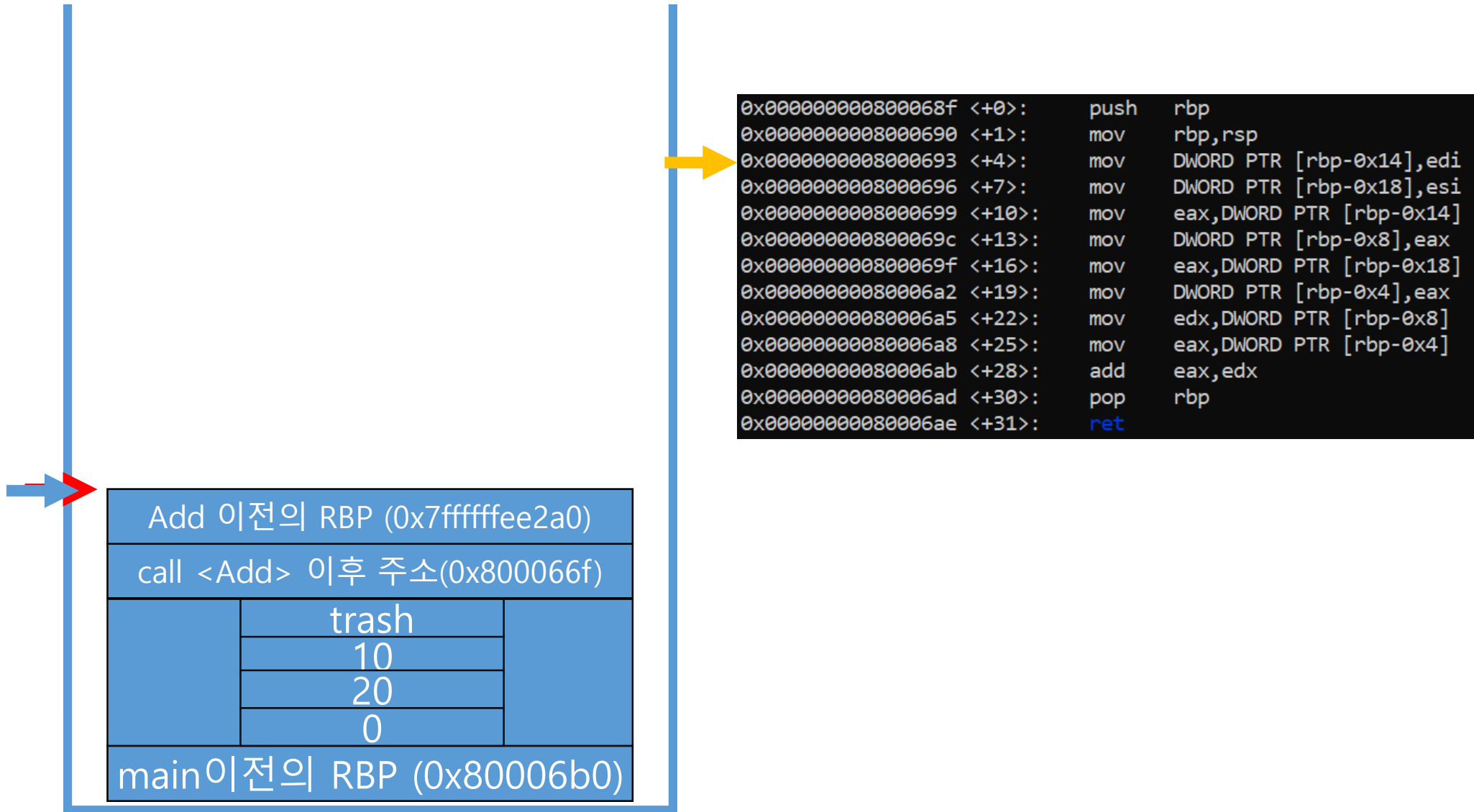
StackFramePractice1.c

- StackFramePractice1.c (Add) (7/17)



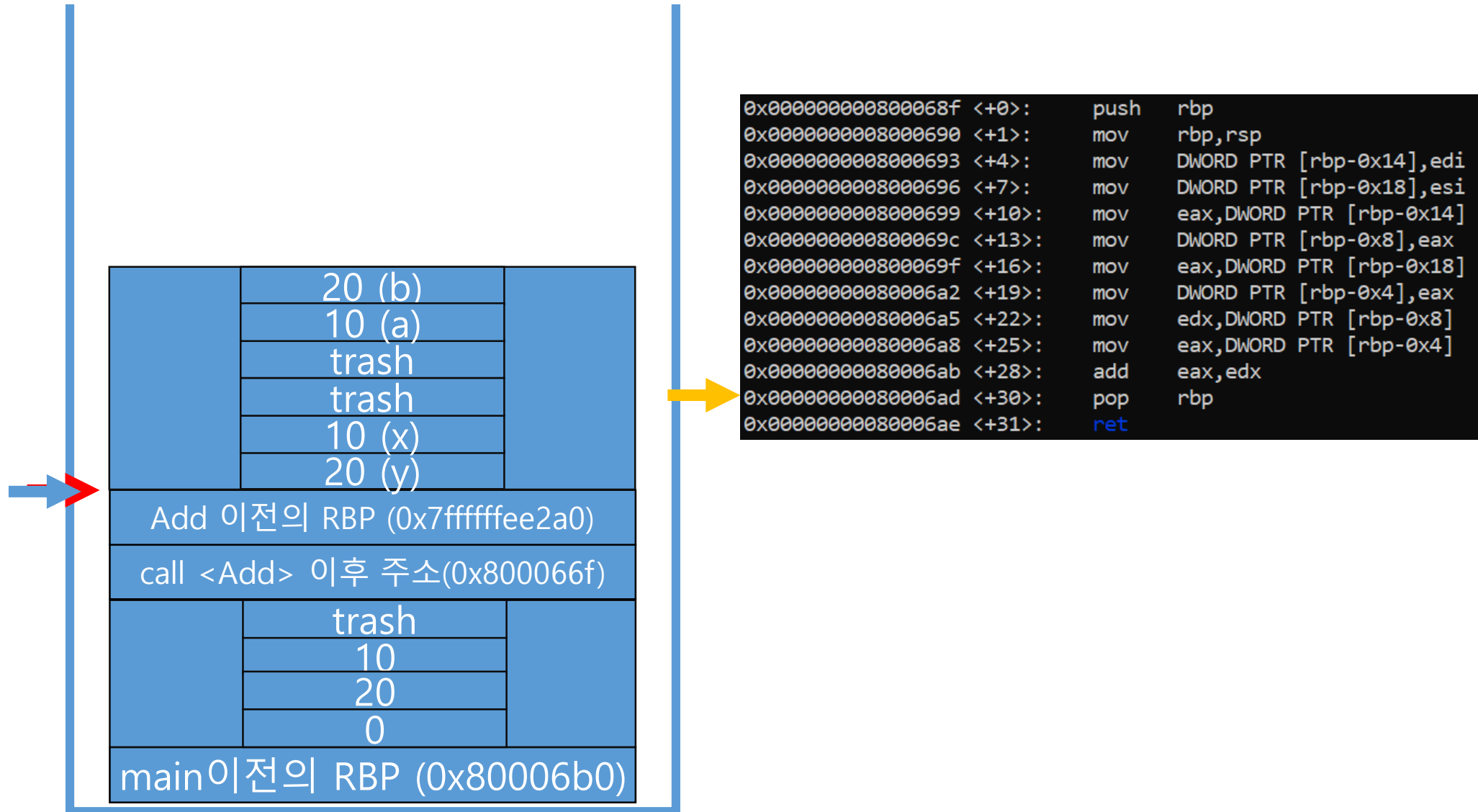
StackFramePractice1.c

- StackFramePractice1.c (Add) (8/17)



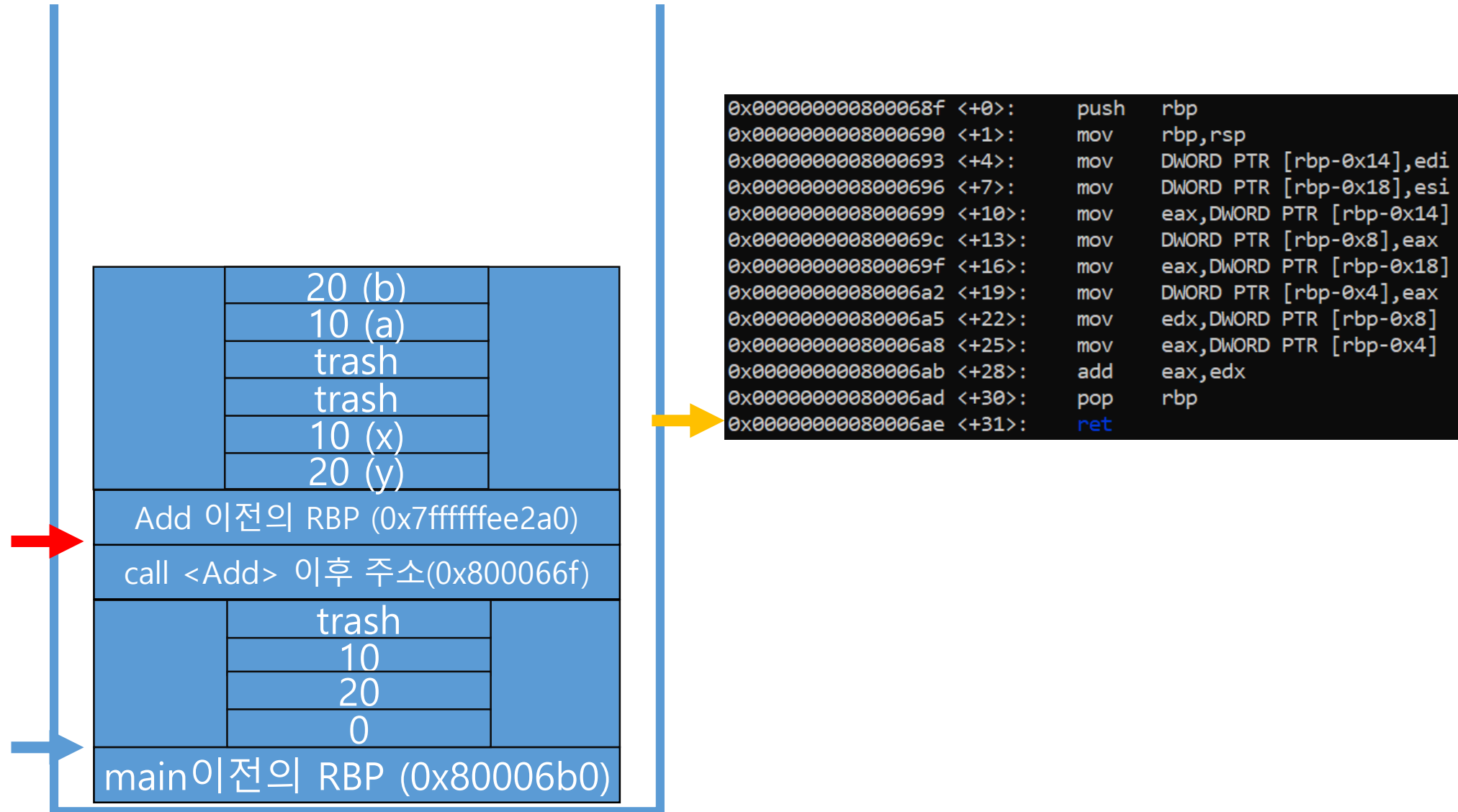
StackFramePractice1.c

- StackFramePractice1.c (Add) (9/17)



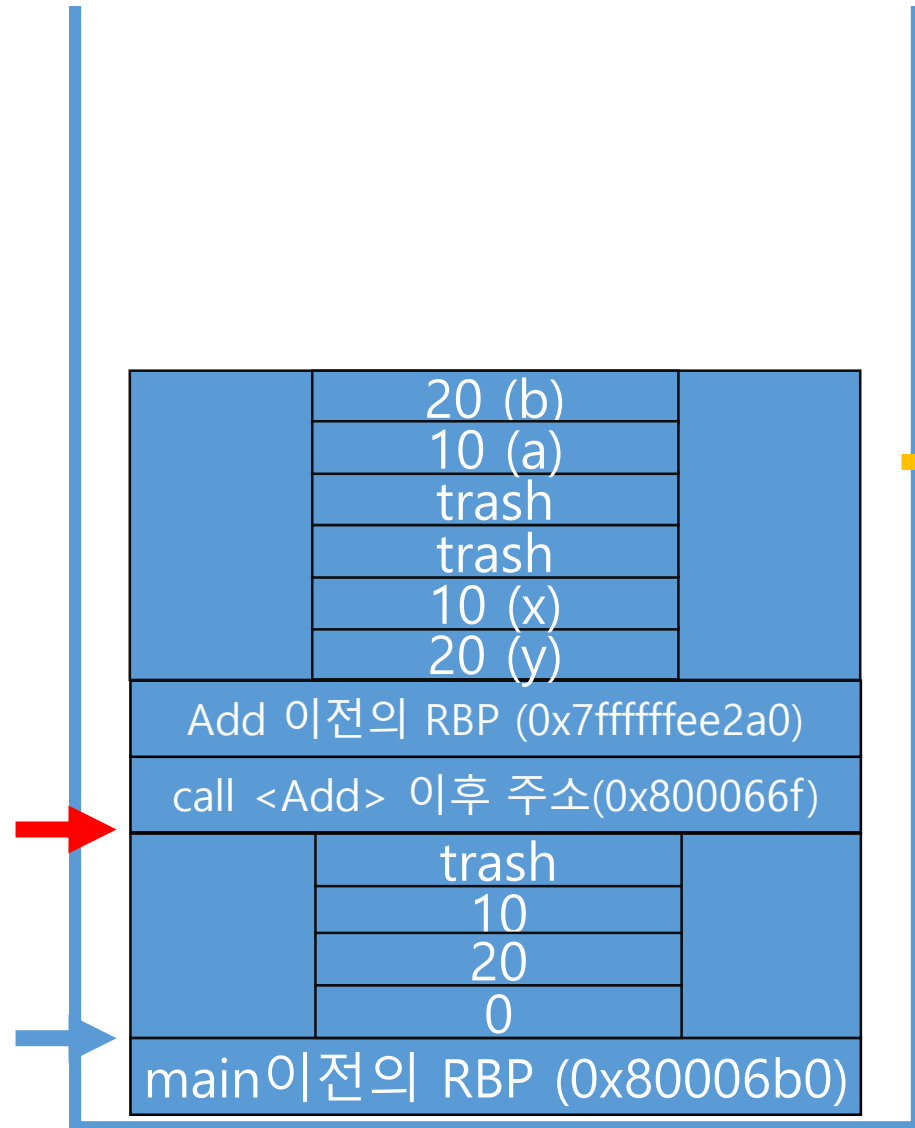
StackFramePractice1.c

- StackFramePractice1.c (Add) (10/17)



StackFramePractice1.c

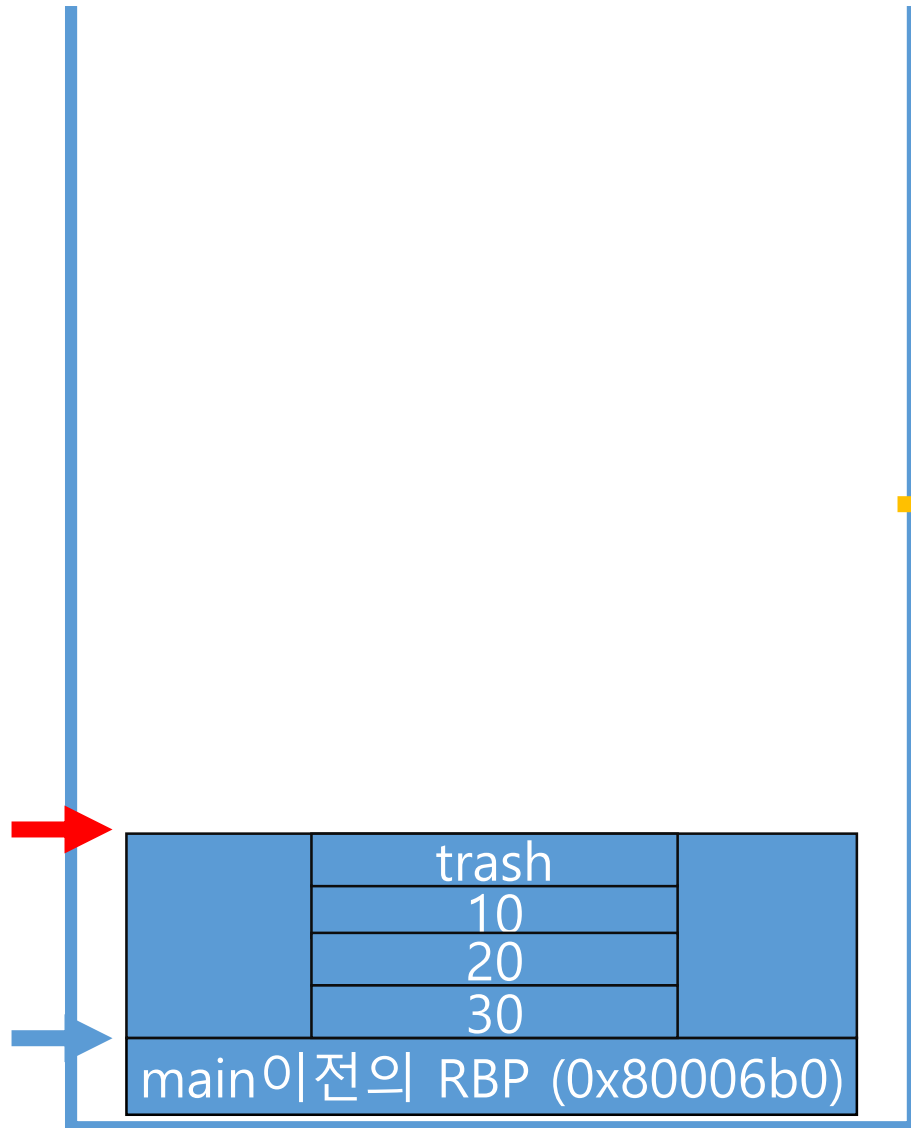
- StackFramePractice1.c (main) (11/17)



```
0x000000000800064a <+0>:  push    rbp
0x000000000800064b <+1>:  mov     rbp, rsp
0x000000000800064e <+4>:  sub     rsp, 0x10
0x0000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x0000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x0000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x0000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x0000000008000666 <+28>: mov     esi, edx
0x0000000008000668 <+30>: mov     edi, eax
0x000000000800066a <+32>:  call    0x800068f <Add>
0x000000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x0000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x0000000008000675 <+43>: mov     esi, eax
0x0000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x000000000800067e <+52>: mov     eax, 0x0
0x0000000008000683 <+57>:  call    0x8000520 <printf@plt>
0x0000000008000688 <+62>: mov     eax, 0x0
0x000000000800068d <+67>: leave
0x000000000800068e <+68>:  ret
```

StackFramePractice1.c

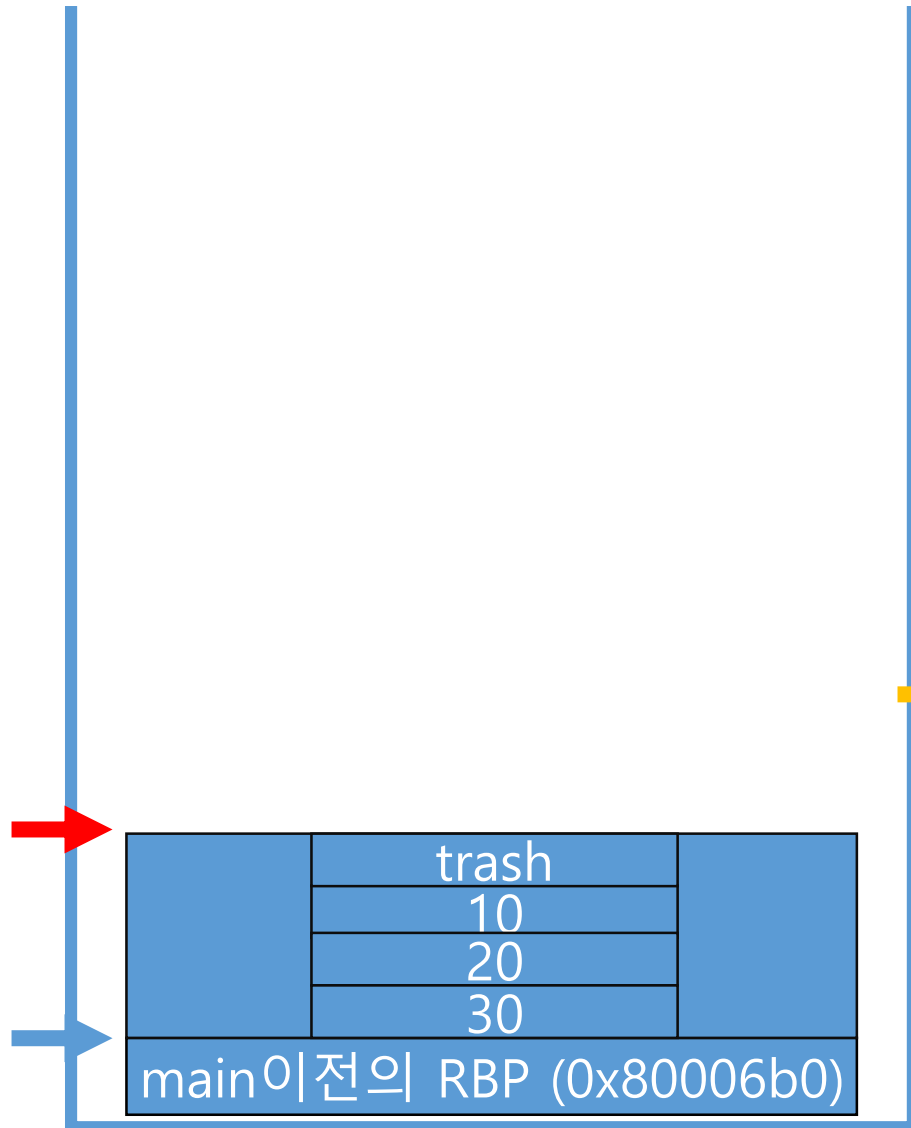
- StackFramePractice1.c (main) (12/17)



```
0x00000000800064a <+0>:  push    rbp
0x00000000800064b <+1>:  mov     rbp, rsp
0x00000000800064e <+4>:  sub     rsp, 0x10
0x000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x000000008000666 <+28>: mov     esi, edx
0x000000008000668 <+30>: mov     edi, eax
0x00000000800066a <+32>: call    0x800068f <Add>
0x00000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x000000008000675 <+43>: mov     esi, eax
0x000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x00000000800067e <+52>: mov     eax, 0x0
0x000000008000683 <+57>: call    0x8000520 <printf@plt>
0x000000008000688 <+62>: mov     eax, 0x0
0x00000000800068d <+67>: leave
0x00000000800068e <+68>: ret
```

StackFramePractice1.c

- StackFramePractice1.c (main) (13/17)

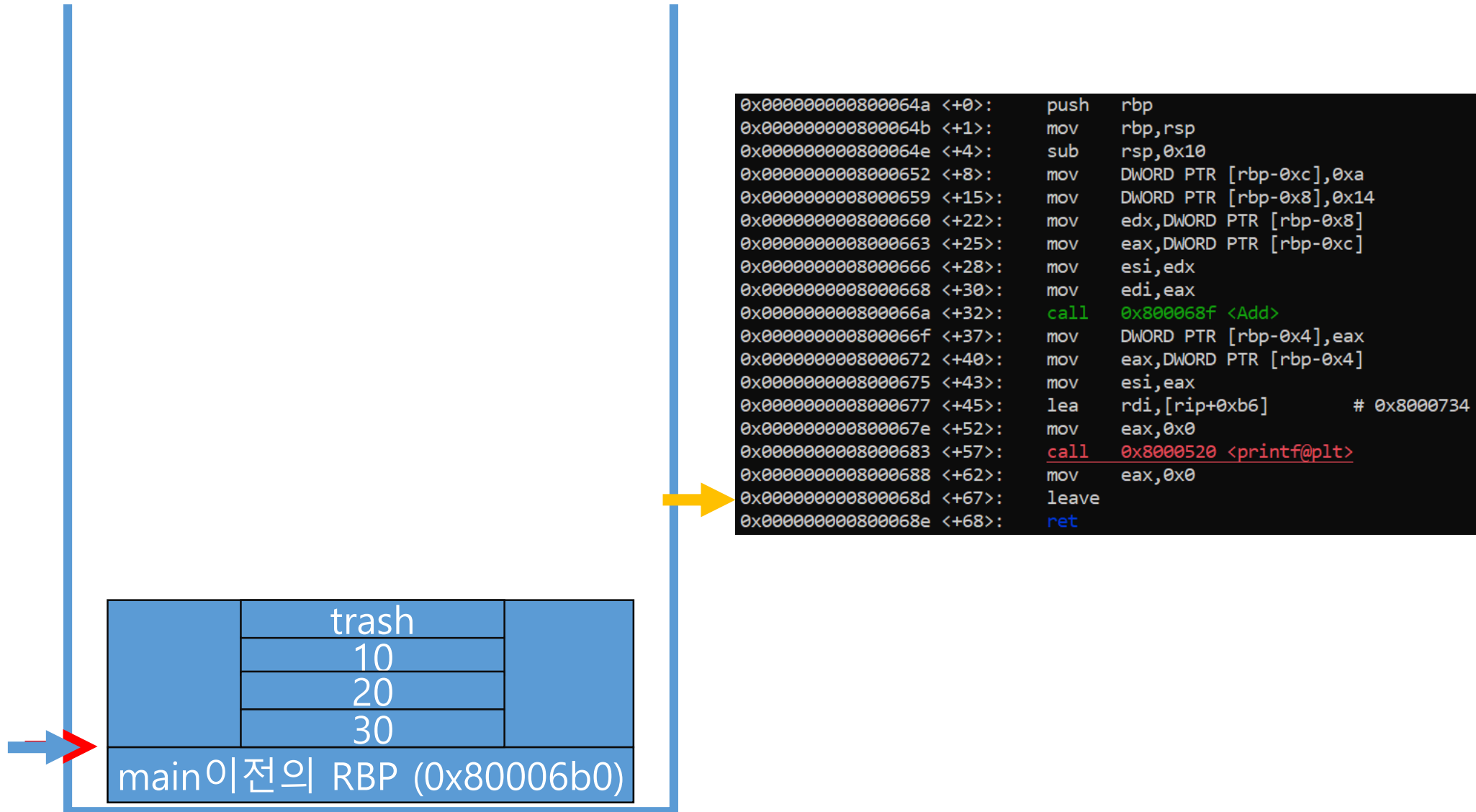


```
0x000000000800064a <+0>:  push    rbp
0x000000000800064b <+1>:  mov     rbp, rsp
0x000000000800064e <+4>:  sub     rsp, 0x10
0x0000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x0000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x0000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x0000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x0000000008000666 <+28>: mov     esi, edx
0x0000000008000668 <+30>: mov     edi, eax
0x000000000800066a <+32>: call    0x800068f <Add>
0x000000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x0000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x0000000008000675 <+43>: mov     esi, eax
0x0000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x000000000800067e <+52>: mov     eax, 0x0
0x0000000008000683 <+57>: call    0x8000520 <printf@plt>
0x0000000008000688 <+62>: mov     eax, 0x0
0x000000000800068d <+67>: leave
0x000000000800068e <+68>: ret
```



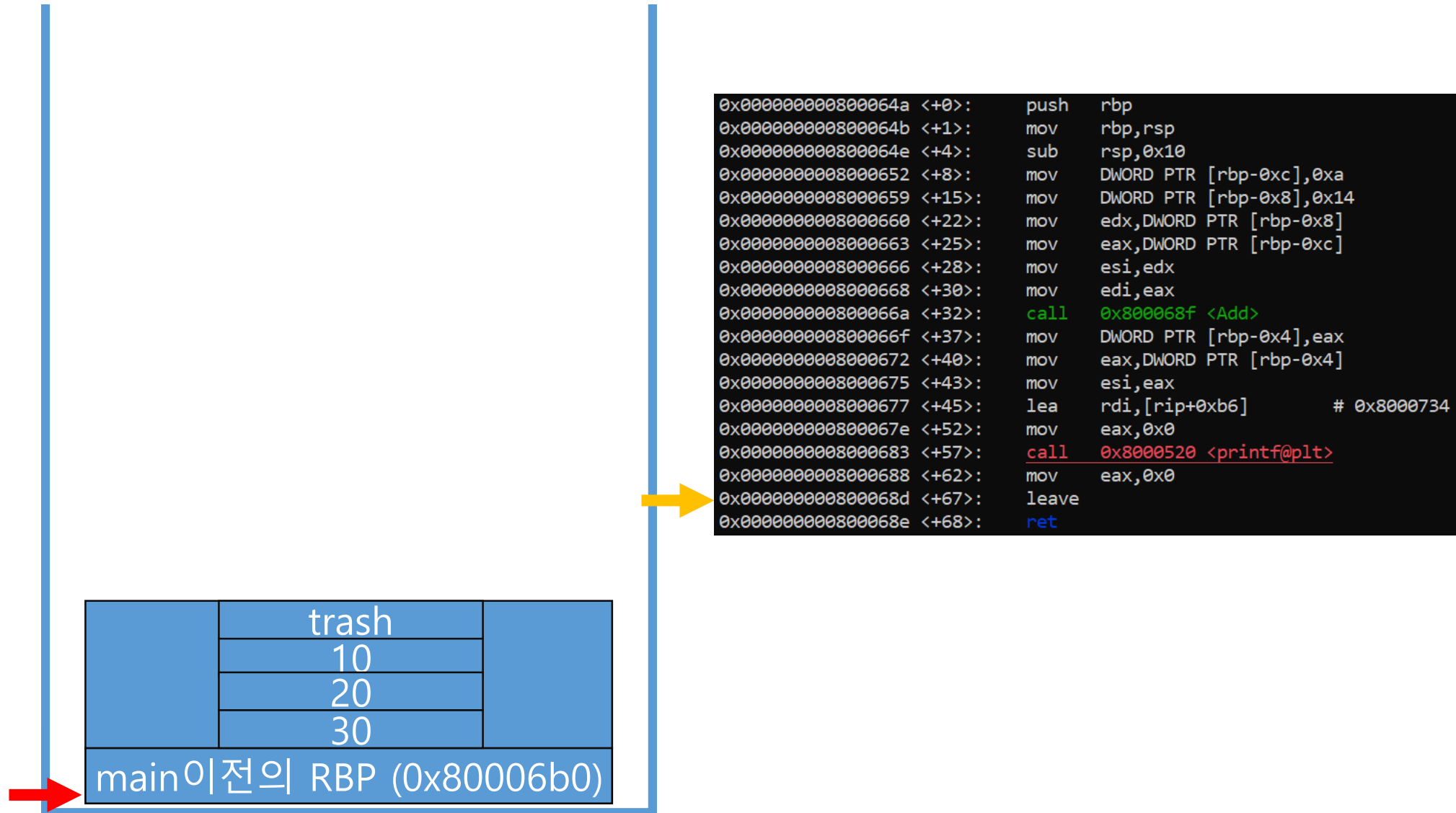
StackFramePractice1.c

- StackFramePractice1.c (main) (14/17)



StackFramePractice1.c

- StackFramePractice1.c (main) (15/17)



StackFramePractice1.c

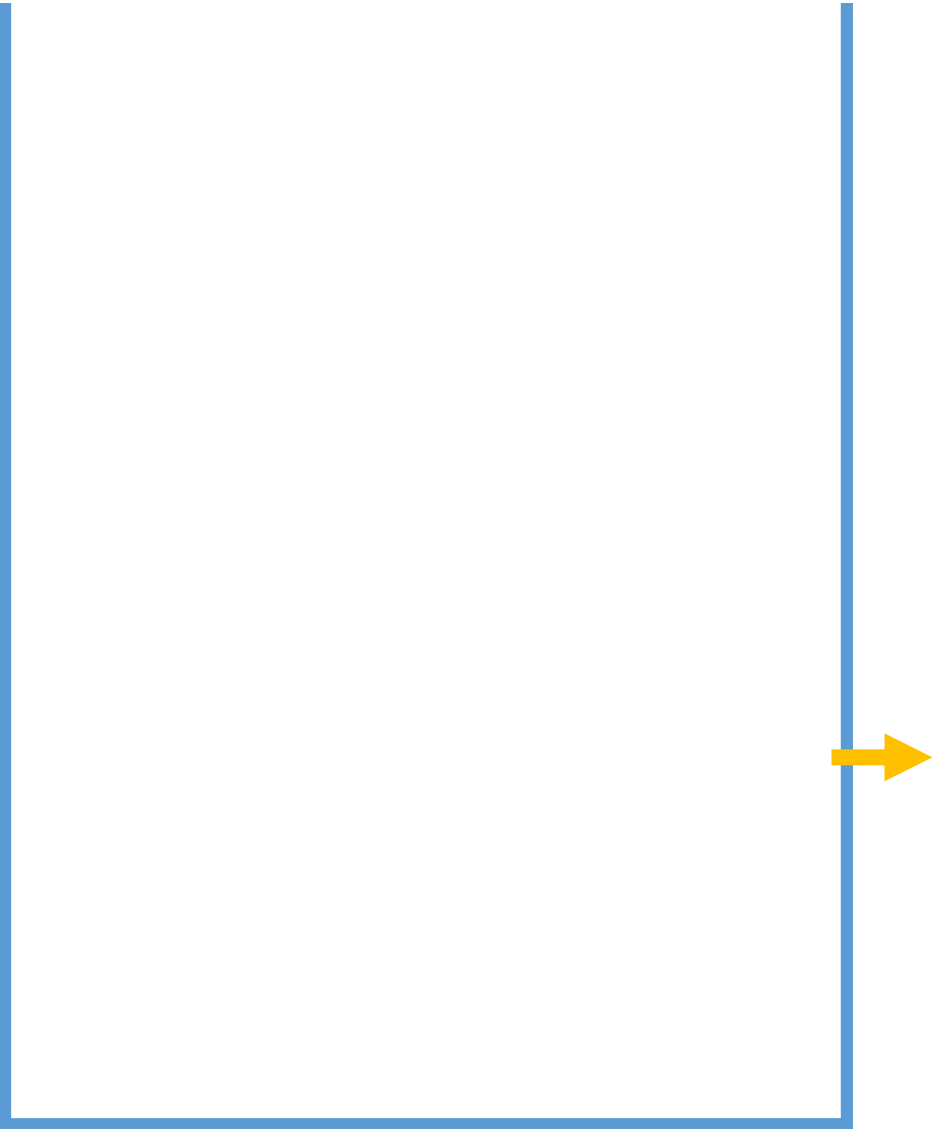
- StackFramePractice1.c (main) (16/17)

```
0x000000000800064a <+0>:  push    rbp
0x000000000800064b <+1>:  mov     rbp, rsp
0x000000000800064e <+4>:  sub     rsp, 0x10
0x0000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x0000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x0000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x0000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x0000000008000666 <+28>: mov     esi, edx
0x0000000008000668 <+30>: mov     edi, eax
0x000000000800066a <+32>: call    0x800068f <Add>
0x000000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x0000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x0000000008000675 <+43>: mov     esi, eax
0x0000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x000000000800067e <+52>: mov     eax, 0x0
0x0000000008000683 <+57>: call    0x8000520 <printf@plt>
0x0000000008000688 <+62>: mov     eax, 0x0
0x000000000800068d <+67>: leave
0x000000000800068e <+68>: ret
```



StackFramePractice1.c

- StackFramePractice1.c (main) (17/17)



```
0x000000000800064a <+0>:  push    rbp
0x000000000800064b <+1>:  mov     rbp, rsp
0x000000000800064e <+4>:  sub     rsp, 0x10
0x0000000008000652 <+8>:  mov     DWORD PTR [rbp-0xc], 0xa
0x0000000008000659 <+15>: mov     DWORD PTR [rbp-0x8], 0x14
0x0000000008000660 <+22>: mov     edx, DWORD PTR [rbp-0x8]
0x0000000008000663 <+25>: mov     eax, DWORD PTR [rbp-0xc]
0x0000000008000666 <+28>: mov     esi, edx
0x0000000008000668 <+30>: mov     edi, eax
0x000000000800066a <+32>:  call    0x800068f <Add>
0x000000000800066f <+37>: mov     DWORD PTR [rbp-0x4], eax
0x0000000008000672 <+40>: mov     eax, DWORD PTR [rbp-0x4]
0x0000000008000675 <+43>: mov     esi, eax
0x0000000008000677 <+45>: lea     rdi, [rip+0xb6]          # 0x8000734
0x000000000800067e <+52>: mov     eax, 0x0
0x0000000008000683 <+57>:  call    0x8000520 <printf@plt>
0x0000000008000688 <+62>: mov     eax, 0x0
0x000000000800068d <+67>:  leave
0x000000000800068e <+68>:  ret
```

목차

- Environment
 - Platform
 - Files
- StackFramePractice1.c
- StackFramePractice2.c
 - Stack Canary
 - Process

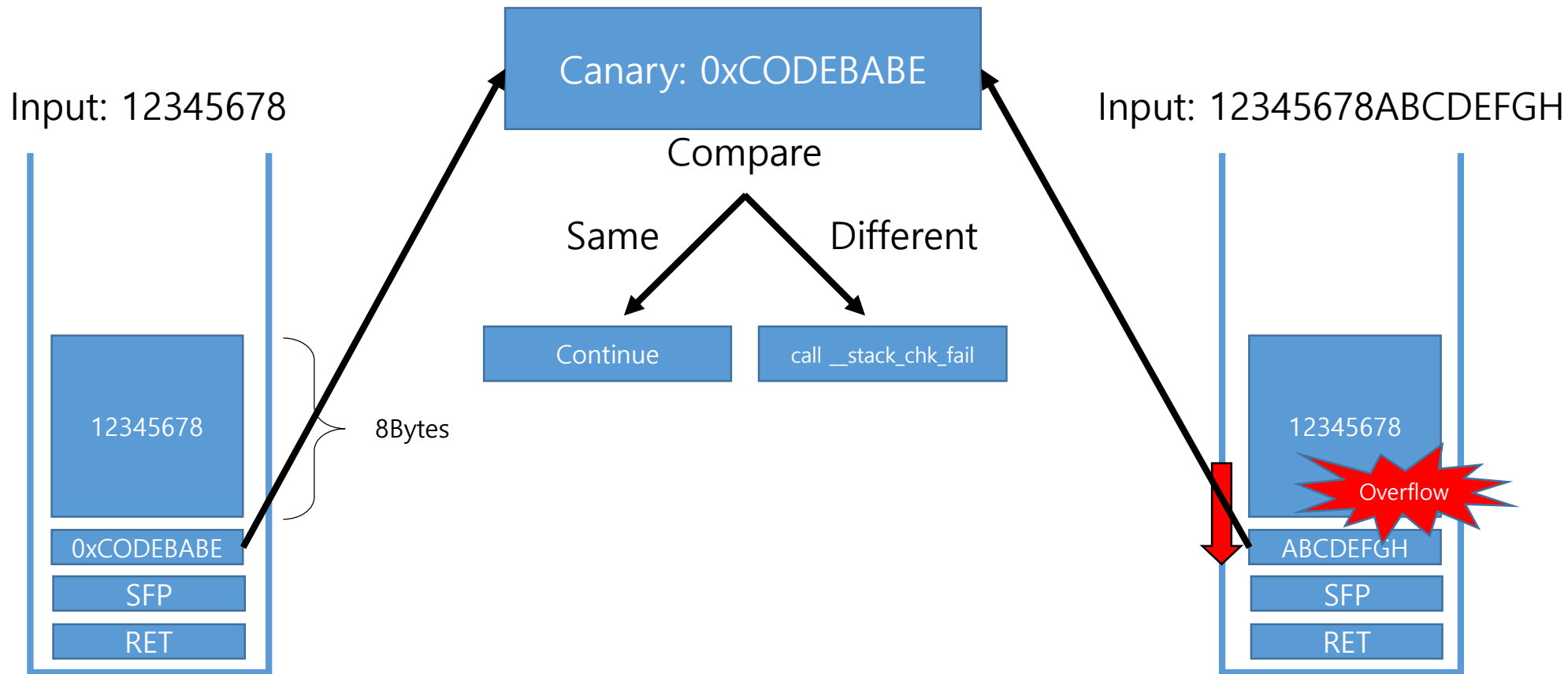
StackFramePractice2.c

- Stack Canary (1/2)
 - BOF 공격 방어 기법
 - Buffer를 사용하는 프로그램에 존재
- SFP 위의 자리에 Canary 저장, 프롤로그 때 그 Canary를 특정 위치에 저장
- 에필로그 때 특정 위치에 저장한 Canary 값과 현재 Canary 값을 비교
- 둘이 다르다면 BOF가 발생하였다고 판단 함수를 종료

StackFramePractice2.c

- Stack Canary (2/2)

- 기존의 Canary 값을 0xCODEBABE라고 할 때



목차

- Environment
 - Platform
 - Files
- StackFramePractice1.c
- StackFramePractice2.c
 - Stack Canary
 - Process

StackFramePractice2.c

- StackFramePractice2.c (1/8)

- Canary 저장

```
0x00000000800076a <+0>: push rbp
0x00000000800076b <+1>: mov rbp, rsp
0x00000000800076e <+4>: sub rsp, 0x20
0x000000008000772 <+8>: mov rax, QWORD PTR fs:0x28
0x00000000800077b <+17>: mov QWORD PTR [rbp-0x8], rax
0x00000000800077f <+21>: xor eax, eax
0x000000008000781 <+23>: movabs rax, 0x6572754365646f43
0x00000000800078b <+33>: mov edx, 0x0
0x000000008000790 <+38>: mov QWORD PTR [rbp-0x20], rax
0x000000008000794 <+42>: mov QWORD PTR [rbp-0x18], rdx
0x000000008000798 <+46>: lea rax, [rbp-0x20]
0x00000000800079c <+50>: mov rsi, rax
0x00000000800079f <+53>: lea rdi, [rip+0x14e] # 0x80008f4
0x0000000080007a6 <+60>: mov eax, 0x0
0x0000000080007ab <+65>: call 0x8000630 <printf@plt>
0x0000000080007b0 <+70>: lea rax, [rbp-0x20]
0x0000000080007b4 <+74>: mov rdi, rax
0x0000000080007b7 <+77>: call 0x800081f <IncrementString>
0x0000000080007bc <+82>: lea rax, [rbp-0x20]
0x0000000080007c0 <+86>: mov rsi, rax
0x0000000080007c3 <+89>: lea rdi, [rip+0x12a] # 0x80008f4
0x0000000080007ca <+96>: mov eax, 0x0
0x0000000080007cf <+101>: call 0x8000630 <printf@plt>
0x0000000080007d4 <+106>: lea rax, [rbp-0x20]
0x0000000080007d8 <+110>: mov rsi, rax
0x0000000080007db <+113>: lea rdi, [rip+0x11c] # 0x80008fe
0x0000000080007e2 <+120>: mov eax, 0x0
0x0000000080007e7 <+125>: call 0x8000640 <_isoc99_scanf@plt>
0x0000000080007ec <+130>: lea rax, [rbp-0x20]
0x0000000080007f0 <+134>: mov rsi, rax
0x0000000080007f3 <+137>: lea rdi, [rip+0x107] # 0x8000901
0x0000000080007fa <+144>: mov eax, 0x0
0x0000000080007ff <+149>: call 0x8000630 <printf@plt>
0x000000008000804 <+154>: mov eax, 0x0
0x000000008000809 <+159>: mov rcx, QWORD PTR [rbp-0x8]
0x00000000800080d <+163>: xor rcx, QWORD PTR fs:0x28
0x000000008000816 <+172>: je 0x800081d <main+179>
0x000000008000818 <+174>: call 0x8000620 <__stack_chk_fail@plt>
0x00000000800081d <+179>: leave
0x00000000800081e <+180>: ret
```

StackFramePractice2.c

- StackFramePractice2.c (2/8)

- "CodeCure" 저장

```
0x00000000800076a <+0>: push rbp
0x00000000800076b <+1>: mov rbp, rsp
0x00000000800076e <+4>: sub rsp, 0x20
0x000000008000772 <+8>: mov rax, QWORD PTR fs:0x28
0x00000000800077b <+17>: mov QWORD PTR [rbp-0x8], rax
0x00000000800077f <+21>: xor eax, eax
0x000000008000781 <+23>: movabs rax, 0x6572754365646f43
0x00000000800078b <+33>: mov edx, 0x0
0x000000008000790 <+38>: mov QWORD PTR [rbp-0x20], rax
0x000000008000794 <+42>: mov QWORD PTR [rbp-0x18], rdx
0x000000008000798 <+46>: lea rax, [rbp-0x20]
0x00000000800079c <+50>: mov rsi, rax
0x00000000800079f <+53>: lea rdi, [rip+0x14e] # 0x80008f4
0x0000000080007a6 <+60>: mov eax, 0x0
0x0000000080007ab <+65>: call 0x8000630 <printf@plt>
0x0000000080007b0 <+70>: lea rax, [rbp-0x20]
0x0000000080007b4 <+74>: mov rdi, rax
0x0000000080007b7 <+77>: call 0x800081f <IncrementString>
0x0000000080007bc <+82>: lea rax, [rbp-0x20]
0x0000000080007c0 <+86>: mov rsi, rax
0x0000000080007c3 <+89>: lea rdi, [rip+0x12a] # 0x80008f4
0x0000000080007ca <+96>: mov eax, 0x0
0x0000000080007cf <+101>: call 0x8000630 <printf@plt>
0x0000000080007d4 <+106>: lea rax, [rbp-0x20]
0x0000000080007d8 <+110>: mov rsi, rax
0x0000000080007db <+113>: lea rdi, [rip+0x11c] # 0x80008fe
0x0000000080007e2 <+120>: mov eax, 0x0
0x0000000080007e7 <+125>: call 0x8000640 <_isoc99_scanf@plt>
0x0000000080007ec <+130>: lea rax, [rbp-0x20]
0x0000000080007f0 <+134>: mov rsi, rax
0x0000000080007f3 <+137>: lea rdi, [rip+0x107] # 0x8000901
0x0000000080007fa <+144>: mov eax, 0x0
0x0000000080007ff <+149>: call 0x8000630 <printf@plt>
0x000000008000804 <+154>: mov eax, 0x0
0x000000008000809 <+159>: mov rcx, QWORD PTR [rbp-0x8]
0x00000000800080d <+163>: xor rcx, QWORD PTR fs:0x28
0x000000008000816 <+172>: je 0x800081d <main+179>
0x000000008000818 <+174>: call 0x8000620 <__stack_chk_fail@plt>
0x00000000800081d <+179>: leave
0x00000000800081e <+180>: ret
```

StackFramePractice2.c

- StackFramePractice2.c (3/8)

- str1 에 입력

```
0x00000000800076a <+0>:  push    rbp
0x00000000800076b <+1>:  mov     rbp, rsp
0x00000000800076e <+4>:  sub     rsp, 0x20
0x000000008000772 <+8>:  mov     rax, QWORD PTR fs:0x28
0x00000000800077b <+17>: mov     QWORD PTR [rbp-0x8], rax
0x00000000800077f <+21>: xor     eax, eax
0x000000008000781 <+23>: movabs  rax, 0x6572754365646f43
0x00000000800078b <+33>: mov     edx, 0x0
0x000000008000790 <+38>: mov     QWORD PTR [rbp-0x20], rax
0x000000008000794 <+42>: mov     QWORD PTR [rbp-0x18], rdx
0x000000008000798 <+46>: lea     rax, [rbp-0x20]
0x00000000800079c <+50>: mov     rsi, rax
0x00000000800079f <+53>: lea     rdi, [rip+0x14e]      # 0x80008f4
0x0000000080007a6 <+60>: mov     eax, 0x0
0x0000000080007ab <+65>: call    0x8000630 <printf@plt>
0x0000000080007b0 <+70>: lea     rax, [rbp-0x20]
0x0000000080007b4 <+74>: mov     rdi, rax
0x0000000080007b7 <+77>: call    0x800081f <IncrementString>
0x0000000080007bc <+82>: lea     rax, [rbp-0x20]
0x0000000080007c0 <+86>: mov     rsi, rax
0x0000000080007c3 <+89>: lea     rdi, [rip+0x12a]      # 0x80008f4
0x0000000080007ca <+96>: mov     eax, 0x0
0x0000000080007cf <+101>: call    0x8000630 <printf@plt>
0x0000000080007d4 <+106>: lea     rax, [rbp-0x20]
0x0000000080007d8 <+110>: mov     rsi, rax
0x0000000080007db <+113>: lea     rdi, [rip+0x11c]      # 0x80008fe
0x0000000080007e2 <+120>: mov     eax, 0x0
0x0000000080007e7 <+125>: call    0x8000640 <_isoc99_scanf@plt>
0x0000000080007ec <+130>: lea     rax, [rbp-0x20]
0x0000000080007f0 <+134>: mov     rsi, rax
0x0000000080007f3 <+137>: lea     rdi, [rip+0x107]      # 0x8000901
0x0000000080007fa <+144>: mov     eax, 0x0
0x0000000080007ff <+149>: call    0x8000630 <printf@plt>
0x000000008000804 <+154>: mov     eax, 0x0
0x000000008000809 <+159>: mov     rcx, QWORD PTR [rbp-0x8]
0x00000000800080d <+163>: xor     rcx, QWORD PTR fs:0x28
0x000000008000816 <+172>: je      0x800081d <main+179>
0x000000008000818 <+174>: call    0x8000620 <__stack_chk_fail@plt>
0x00000000800081d <+179>: leave
0x00000000800081e <+180>: ret
```

StackFramePractice2.c

- StackFramePractice2.c (4/8)

- Canary 비교

```
0x00000000800076a <+0>: push rbp
0x00000000800076b <+1>: mov rbp, rsp
0x00000000800076e <+4>: sub rsp, 0x20
0x000000008000772 <+8>: mov rax, QWORD PTR fs:0x28
0x00000000800077b <+17>: mov QWORD PTR [rbp-0x8], rax
0x00000000800077f <+21>: xor eax, eax
0x000000008000781 <+23>: movabs rax, 0x6572754365646f43
0x00000000800078b <+33>: mov edx, 0x0
0x000000008000790 <+38>: mov QWORD PTR [rbp-0x20], rax
0x000000008000794 <+42>: mov QWORD PTR [rbp-0x18], rdx
0x000000008000798 <+46>: lea rax, [rbp-0x20]
0x00000000800079c <+50>: mov rsi, rax
0x00000000800079f <+53>: lea rdi, [rip+0x14e] # 0x80008f4
0x0000000080007a6 <+60>: mov eax, 0x0
0x0000000080007ab <+65>: call 0x8000630 <printf@plt>
0x0000000080007b0 <+70>: lea rax, [rbp-0x20]
0x0000000080007b4 <+74>: mov rdi, rax
0x0000000080007b7 <+77>: call 0x800081f <IncrementString>
0x0000000080007bc <+82>: lea rax, [rbp-0x20]
0x0000000080007c0 <+86>: mov rsi, rax
0x0000000080007c3 <+89>: lea rdi, [rip+0x12a] # 0x80008f4
0x0000000080007ca <+96>: mov eax, 0x0
0x0000000080007cf <+101>: call 0x8000630 <printf@plt>
0x0000000080007d4 <+106>: lea rax, [rbp-0x20]
0x0000000080007d8 <+110>: mov rsi, rax
0x0000000080007db <+113>: lea rdi, [rip+0x11c] # 0x80008fe
0x0000000080007e2 <+120>: mov eax, 0x0
0x0000000080007e7 <+125>: call 0x8000640 <_isoc99_scanf@plt>
0x0000000080007ec <+130>: lea rax, [rbp-0x20]
0x0000000080007f0 <+134>: mov rsi, rax
0x0000000080007f3 <+137>: lea rdi, [rip+0x107] # 0x8000901
0x0000000080007fa <+144>: mov eax, 0x0
0x0000000080007ff <+149>: call 0x8000630 <printf@plt>
0x000000008000804 <+154>: mov eax, 0x0
0x000000008000809 <+159>: mov rcx, QWORD PTR [rbp-0x8]
0x00000000800080d <+163>: xor rcx, QWORD PTR fs:0x28
0x000000008000816 <+172>: je 0x800081d <main+179>
0x000000008000818 <+174>: call 0x8000620 <__stack_chk_fail@plt>
0x00000000800081d <+179>: leave
0x00000000800081e <+180>: ret
```

StackFramePractice2.c

- StackFramePractice2.c (5/8)
 - 입력 값에 따른 Canary 보호 기법 확인
 - break 포인트 설정

```
gdb-peda$ b *main+130
Breakpoint 1 at 0x7ec
gdb-peda$ b *main+163
Breakpoint 2 at 0x80d
```

- 올바른 Input

```
gdb-peda$ r
Starting program: /home/jin-desk/CodeCure/sfp2
str1: CodeCure
str1: DpefDvsf
RightInput
```


StackFramePractice2.c

- StackFramePractice2.c (6/8)

- 입력 값에 따른 Canary 보호 기법 확인

- Continue, next

```
gdb-peda$ c
Continuing.
str2: RightInput
```

```
gdb-peda$ n
```

- RCX 값

```
RCX: 0x0
```

```
0x0000000008000809 <+159>:  mov    rcx,QWORD PTR [rbp-0x8]
0x000000000800080d <+163>:  xor     rcx,QWORD PTR fs:0x28
0x0000000008000816 <+172>:  je      0x800081d <main+179>
0x0000000008000818 <+174>:  call   0x8000620 <__stack_chk_fail@plt>
0x000000000800081d <+179>:  leave
```

- 이후 올바른 종료

```
gdb-peda$ c
Continuing.
[Inferior 1 (process 157) exited normally]
Warning: not running
```

StackFramePractice2.c

- StackFramePractice2.c (7/8)
 - 입력 값에 따른 Canary 보호 기법 확인
 - break 포인트 설정

```
gdb-peda$ b *main+130
Breakpoint 1 at 0x7ec
gdb-peda$ b *main+163
Breakpoint 2 at 0x80d
```

- 올바른지 않은 input

```
gdb-peda$ r
Starting program: /home/jin-desk/CodeCure/sfp2
str1: CodeCure
str1: DpefDvsf
BadInputHAHAHAHAHAHAHAHAHAHAHAHAHAHAHAHA
```

StackFramePractice2.c

- StackFramePractice2.c (8/8)
 - 입력 값에 따른 Canary 보호 기법 확인
 - Continue, next

```
gdb-peda$ c
Continuing.
str2: BadInputHAHAHAHAHAHAHAHAHAHAHAHAHAHAHAHAHA
```

gdb-peda\$ n

- RCX 값

```
0x0000000008000809 <+159>: mov     rcx,QWORD PTR [rbp-0x8]
0x000000000800080d <+163>: xor     rcx,QWORD PTR fs:0x28
0x0000000008000816 <+172>: je      0x800081d <main+179>
0x0000000008000818 <+174>: call    0x8000620 <__stack_chk_fail@plt>
0x000000000800081d <+179>: leave
```

RCX: 0x32f52b2c88cd1a48

- 이후 오류 발생

```
gdb-peda$ c
Continuing.
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGABRT, Aborted.
```

감사합니다!