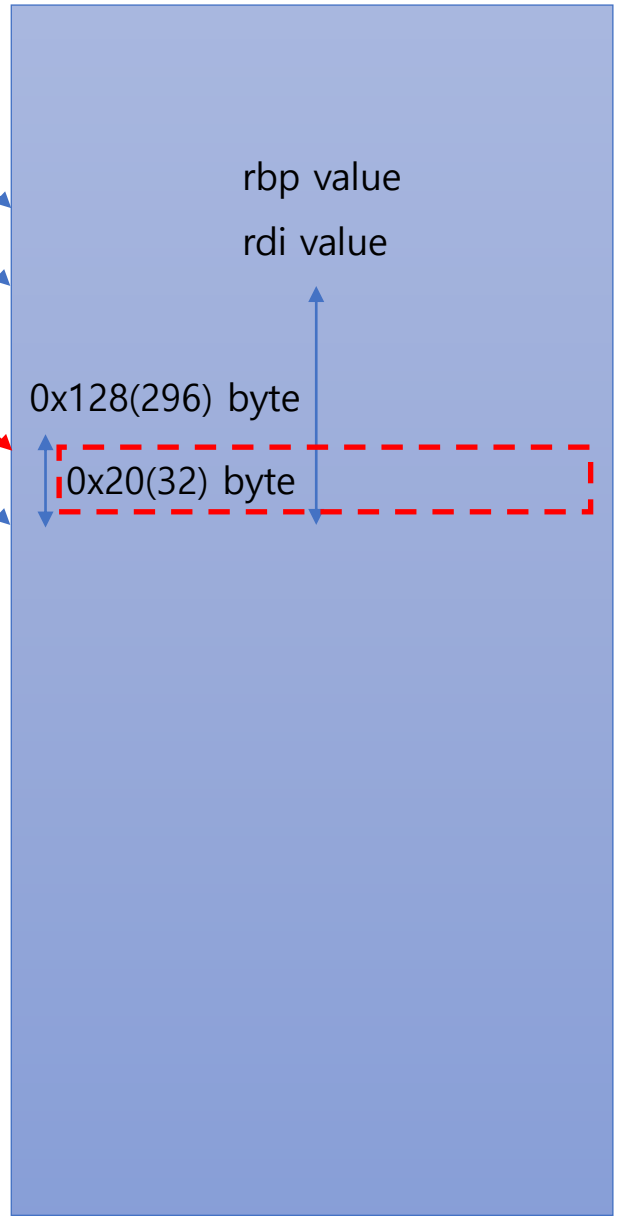
—: SP
—: BP

현재 SP

빨간 점선은 Stack 공간
해당 공간 내부에
return_value 변수가 있음

for_assembly_function_test의
number가
main의 SP와 동일했음



Virtual Memory (가상 메모리)

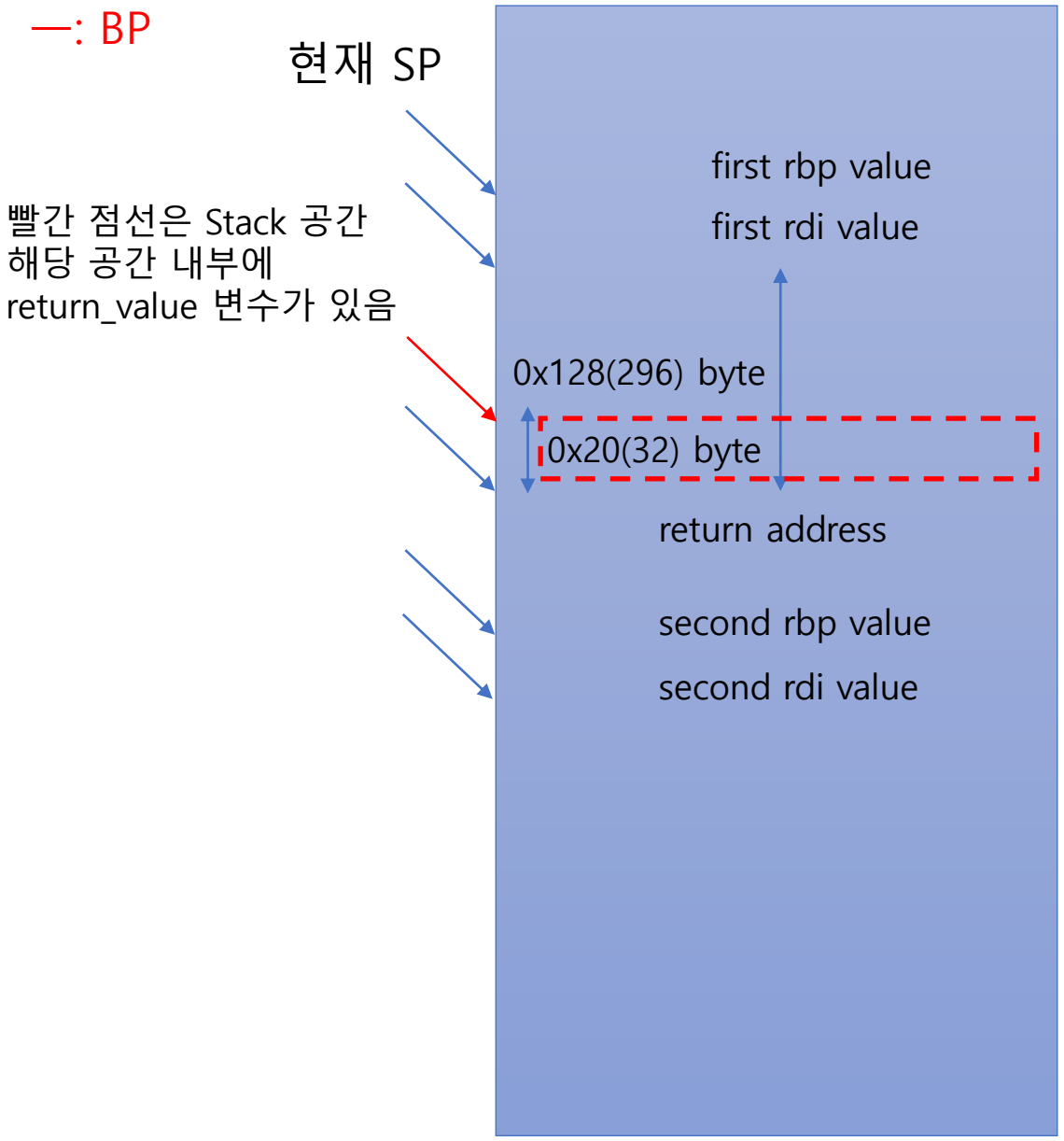
main

```
00007FF607F318C0  push        rbp
00007FF607F318C2  push        rdi
00007FF607F318C3  sub         rsp,128h
00007FF607F318CA  lea         rbp,[rsp+20h]
00007FF607F318CF  lea         rcx,[__E5398147_main@c (07FF607F41008h)]
00007FF607F318D6  call        __CheckForDebuggerJustMyCode (07FF607F31370h)
00007FF607F318DB  mov         dword ptr [input_parameter],3
00007FF607F318E2  mov         ecx,dword ptr [input_parameter]
00007FF607F318E5  call        for_assembly_function_test (07FF607F311CCh)
00007FF607F318EA  mov         dword ptr [return_value],eax
```

for_assembly_function_test

```
00007FF607F31880  mov         dword ptr [rsp+8],ecx
00007FF607F31884  push        rbp
00007FF607F31885  push        rdi
00007FF607F31886  sub         rsp,0E8h
00007FF607F3188D  lea         rbp,[rsp+20h]
00007FF607F31892  lea         rcx,[__E5398147_main@c (07FF607F41008h)]
00007FF607F31899  call        __CheckForDebuggerJustMyCode (07FF607F31370h)
00007FF607F3189E  mov         eax,dword ptr [number]
00007FF607F318A4  shl         eax,1
00007FF607F318A6  lea         rsp,[rbp+0C8h]
00007FF607F318AD  pop         rdi
00007FF607F318AE  pop         rbp
00007FF607F318AF  ret
```

—: SP
—: BP



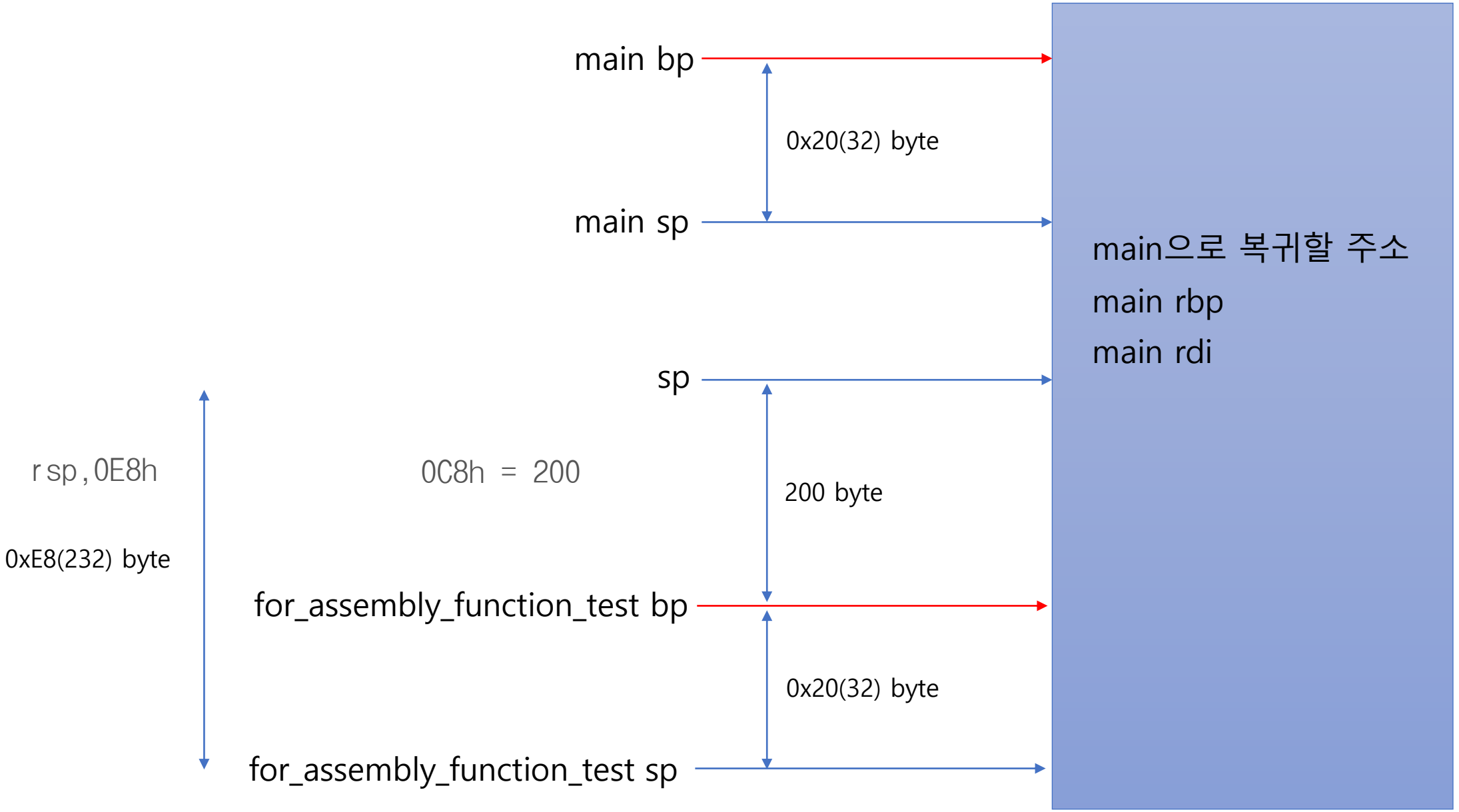
for_assembly_function_test

```
00007FF607F31880  mov         dword ptr [rsp+8],ecx
00007FF607F31884  push        rbp
00007FF607F31885  push        rdi
00007FF607F31886  sub         rsp,0E8h        # 232
00007FF607F3188D  lea         rbp,[rsp+20h]
00007FF607F31892  lea         rcx,[__E5398147_main@c (07FF607F41008h)]
00007FF607F31899  call        __CheckForDebuggerJustMyCode (07FF607F31370h)
00007FF607F3189E  mov         eax,dword ptr [number]
00007FF607F318A4  shl         eax,1
00007FF607F318A6  lea         rsp,[rbp+0C8h]
00007FF607F318AD  pop         rdi
00007FF607F318AE  pop         rbp
00007FF607F318AF  ret
```

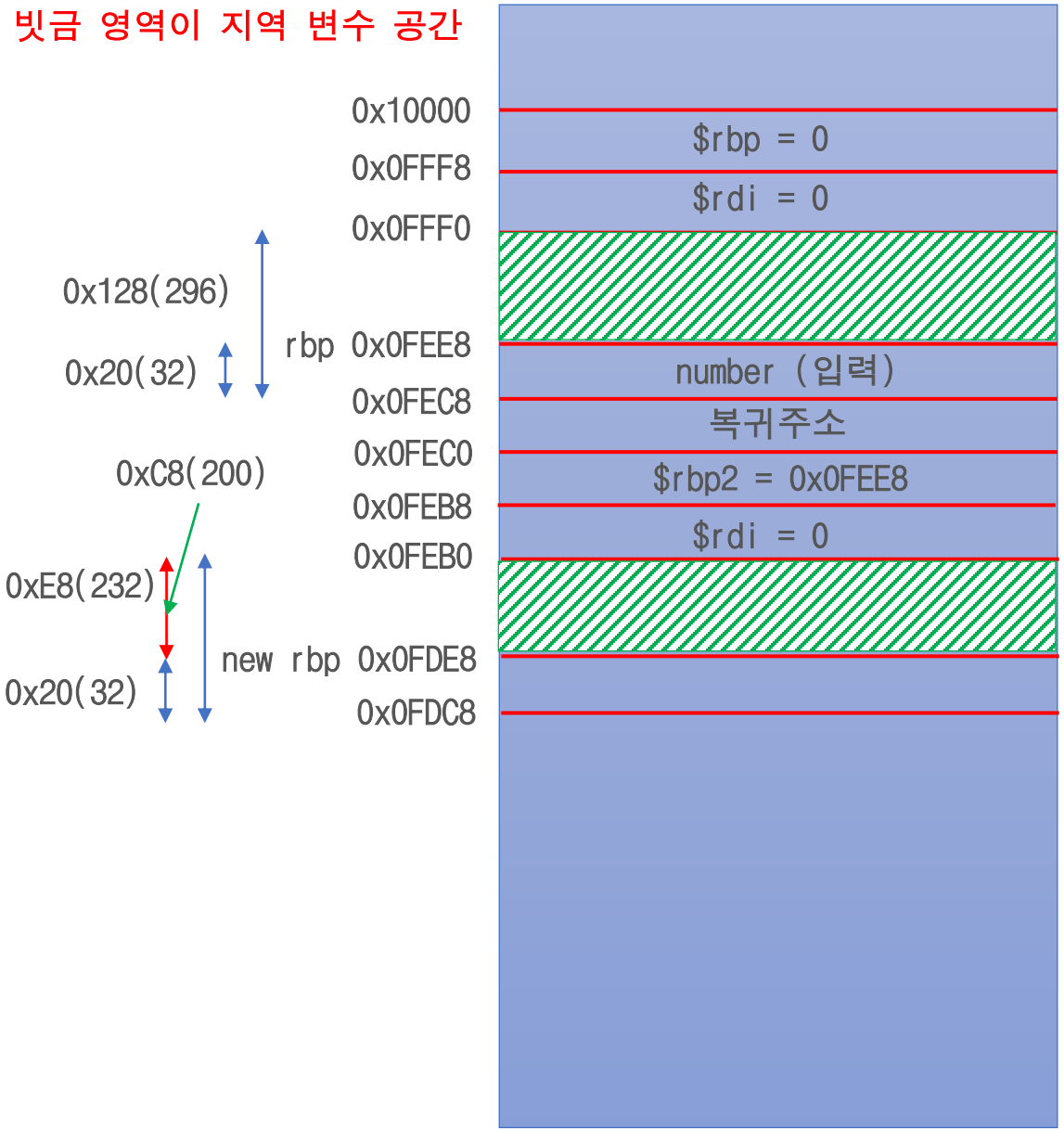
pop 명령어는 현재 Stack의 최상위 (sp)에서 내용을 빼와서
뒤에 있는 레지스터나 메모리에 배치합니다.

pop A의 경우 sp에서 빼온 값을 A에 배치합니다.
pop을 하면 값을 빼는 것이기 때문에 SP + 8이 진행됩니다

ret는 pop ip와 동의어입니다.
그러므로 복귀 주소가 ip에 들어가면서
다음에 실행해야 하는 위치로 복원이 됩니다.


main bp
0x20(32) byte
main sp
main으로 복귀할 주소
main rbp
main rdi
sp
rsp,0E8h
0C8h = 200
200 byte
0xE8(232) byte
for_assembly_function_test bp
0x20(32) byte
for_assembly_function_test sp

빗금 영역이 지역 변수 공간



Virtual Memory (가상 메모리)

main

```
00007FF607F318C0  push        rbp
00007FF607F318C2  push        rdi
00007FF607F318C3  sub         rsp,128h
00007FF607F318CA  lea         rbp,[rsp+20h]
00007FF607F318CF  lea         rcx,[__E5398147_main@c (07FF607F41008h)]
00007FF607F318D6  call        __CheckForDebuggerJustMyCode (07FF607F31370h)
00007FF607F318DB  mov         dword ptr [input_parameter],3
00007FF607F318E2  mov         ecx,dword ptr [input_parameter]
00007FF607F318E5  call        for_assembly_function_test (07FF607F311CCh)
00007FF607F318EA  mov         dword ptr [return_value],eax
```

결론적으로 Window의 sub로 생성되는
영역 전체가 지역 변수로 활용된다 보는 것이 적합합니다.

for_assembly_function_test

```
00007FF607F31880  mov         dword ptr [rsp+8],ecx
00007FF607F31884  push        rbp
00007FF607F31885  push        rdi
00007FF607F31886  sub         rsp,0E8h
00007FF607F3188D  lea         rbp,[rsp+20h]
00007FF607F31892  lea         rcx,[__E5398147_main@c (07FF607F41008h)]
00007FF607F31899  call        __CheckForDebuggerJustMyCode (07FF607F31370h)
00007FF607F3189E  mov         eax,dword ptr [number]
00007FF607F318A4  shl         eax,1
00007FF607F318A6  lea         rsp,[rbp+0C8h]
00007FF607F318AD  pop         rdi
00007FF607F318AE  pop         rbp
00007FF607F318AF  ret
```