

I. Sections & Symbol Table

Assume we compile the foo.c:

```
const int const_ten = 10;
const int const_zero = 0;
int ten = 10;
int zero = 0;
int uninitialized;
static int static_ten = 10;
static int static_uninit;

int f() {
    static f_static_i = 10;
    int f_i = 0;
}

int main() {
}
```

Please fill the following table:

(Note: show the distinction between .bss and COMM)

Symbol	.symtab entry?	Symbol type	Global/Local	Section
const_ten	T	OBJECT	G	.rodata
const_zero	T	OBJECT	G	.rodata
ten	T	OBJECT	G	.data
zero	T	OBJECT	G	.bss
uninitialized	T	OBJECT	G	COMM
static_ten	T	OBJECT	L	.data
static_uninit	T	OBJECT	L	.bss
f	T	FUNC	G	.text
f_static_i	T	OBJECT	L	.data
f_i	F			
main	T	FUNC	G	.text

II.Symbol Resolution

Assume we have a foobar.c:

```
1 #include <stdio.h>
2
3 int x;
4 int y;
5 int z = 10;
6 static int a = 10;
7 static int b = 20;
```

```

8
9 void bar();
10
11 void foo() {
12     static int b = 10;
13     int c = 5;
14
15     printf("%d\n", x);
16     printf("%d\n", b);
17     printf("%d\n", c);
18     bar();
19 }
20
21 __attribute__((weak)) void bar() {
22     static int d = 10;
23     int c = 5;
24
25     printf("%d\n", x);
26     printf("%d\n", b);
27     printf("%d\n", c);
28     printf("%d\n", d);
29 }

```

And a barbaz.c:

```

1 #include <stdio.h>
2
3 int x = 5;
4 int y = 10;
5 extern int z;
6 static int a = 40;
7 static int b = 30;
8
9 void bar();
10
11 void baz() {
12     static int b = 10;
13     int c = 5;
14
15     printf("%d\n", x);
16     printf("%d\n", b);
17     printf("%d\n", c);
18     printf("%d\n", y);
19     bar();
20 }
21
22 void bar() {

```

```

23      static int d = 10;
24      int c = 5;
25
26      printf("%d\n", x);
27      printf("%d\n", b);
28      printf("%d\n", c);
29      printf("%d\n", z);
30  }

```

Please show the actually used variable in each reference:

Reference	Variable
foobar.c:15:x	barbaz.c:3:x
foobar.c:16:b	foobar.c:12:b
foobar.c:17:c	foobar.c:13:c
foobar.c:18:bar	barbaz.c:22:bar
foobar.c:25:x	barbaz.c:3:x
foobar.c:26:b	foobar.c:7:b
foobar.c:27:c	foobar.c:23:c
foobar.c:28:d	foobar.c:22:d
barbaz.c:15:x	barbaz.c:3:x
barbaz.c:16:b	barbaz.c:12:b
barbaz.c:17:c	barbaz.c:13:c
barbaz.c:18:y	barbaz.c:4:y
barbaz.c:19:bar	barbaz.c:22:bar
barbaz.c:26:x	barbaz.c:3:x
barbaz.c:27:b	barbaz.c:7:b
barbaz.c:28:c	barbaz.c:24:c
barbaz.c:29:z	foobar.c:5:z

III. Relocation

1. Please show the relocation entry and process of following instruction:

Disassembly of section .text:

0000000000000000 <main>:

...

2: c7 05 00 00 00 00 ad de 00 00 movl \$0xdead, buf(%rip)

Relocation entry:

offset: 4 type: R_X86_64_PC32 symbol: buf addend: -8

Relocation:

ADDR(.text) = 0x4004d6

refaddr = ADDR(.text) + r.offset = 0x4004da

ADDR(r.symbol) = 0x601030

*refptr = ADDR(r.symbol) + r.addend - refaddr
= 0x601030 - 8 - 0x4004d6
= 0x200b4e