



ICS 第四章

4. 填表如下:

符号	是否在 swap.o 符号表中	定义模块	符号类型	节
buf	✓	main.o	extern	.data
bufp0	✓	swap.o	global	.data
bufp1	✓	swap.o	local	.bss
incr	✓	swap.o	local	.text
count	✓	swap.o	local	.data
swap	✓	swap.o	global	.text
temp	x	\	\	\

5. 11) 强符号有: main.c 中的 x, z, main; proc1.c 中的 proc1;

弱符号有: main.c 中的 y 和 proc1; proc1.c 中的 x;

12) proc1() 函数调用前:

proc1() 函数调用后:

0	1	2	3	0	1	2	3
8x	01	01	00	00	8x	00	00
8y	02	00	8z	00	00

其中 x 机器数 BFF80000 0000 0000H 的高 16 位放在了 z 后面, 打印输出 x=0, z=0
改后打印 x=0, z=-16392.

13) 将 proc1.c 的第 1 行改为 "static double x".

7. 因为 main 函数对应的机器码开始两个字节为 55H, 89H, 如下:

00000000 <main>:

```
0: 55          push %ebp
1: 89 e5       mov %esp, %ebp
```

因此调用数组元素 main[0] 和 main[1] 输出 55H 和 89H.





9. (1) gcc -static -o p p.o libx.a liby.a p.o
 (2) gcc -static -o p p.o libx.a liby.a libx.a
 (3) gcc -static -o p p.o libx.a liby.a libx.a libz.a.

11. 填表内容:

序号	符号	位移	指令行号	重定位类型	重定位前内容	重定位后内容
1	bufp1(.bss)	0x8	6-7	R_386_32	0x0	0x8049620
2	buf(.data)	0xc	6-7		0x4	0x80495cc
3	bufp0(.data)	0x11	10		0x0	0x80495d0
4	bufp0(.data)	0x1b	14		0x0	0x80495d0
5	bufp1(.bss)	0x21	17		0x0	0x8049620
6	bufp1(.bss)	0x20	21		0x0	0x8049620

