

4. 符号	是否在 swap.o 的符号表中	定义模块	符号类型	节
buf	是	main.o	extern	.data
bufp0	是	swap.o	global	.data
bufp1	是	swap.o	local	.bss
incr	是	swap.o	local	.text
count	是	swap.o	local	.data
swap	是	swap.o	global	.text
temp	否	swap.o	——	——

5. (1) main.c 中 x, z, main 是强符号, y 和 proc1 为弱符号.

proc1.c 中 proc1 是强符号, x 是弱符号.

(2) 打印的结果是 $x=0, z=0$.

proc1 函数执行前:

proc1 函数执行后:

&x 01 01 00 00

&x 00 00 00 00

&z 02 00

&z 00 00 F8 BF

0 1 2 3

0 1 2 3

若第3行改为 "printf y, z;" 打印结果是 $x=0, z=-16392$.

proc1 函数执行前:

proc1 函数执行后:

&x 01 01 00 00

&x 00 00 00 00

&y 01 00 02 00

&y 00 00 F8 BF

0 1 2 3

0 1 2 3

B) 将第1行修改为 "static double x;" 即可.

7. 全局符号 main 在 M1 中是强符号, 在 M2 中是弱符号, 则以 M1 中的定义为准.

M1 中 main 的定义位于 .text 节中, 函数对应的机器码前两个字节为 55H 和 89H.

在 M2 中以单字节 (char) 格式访问, 十六进制格式打印, 结果为 5589.

10. main.o 的 .text 节中两重定位符号名为 swap。

相对于 .text 节起始位置偏移为 7。

所在指令行号为 6。

重定位类型为按照 PC 相对地址方式 (R-386-PC32) 重定位。

重定位前的机器数为 $0xffffffff$ ，值为 -1。

重定位后的内容为 $addr(swap) - (addr(.text) + r_offset - inst)$

即 $(addr(swap) - addr(.text)) - r_offset + inst = 0xb - 0x7 - 0x4 = 7$ 。

机器数为 $0x00000007$ ，第 6 行内容为 `07 00 00 00`。

序号	符号	偏移	指令所在行号	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	0x00000000	0x8049620
2	buf(.data)	0xC	6~7	R-386-32	0x00000004	0x80495cc
3	bufp0(.data)	0x11	10	R-386-32	0x00000000	0x80495d0
4	bufp0(.data)	0x15	14	R-386-32	0x00000000	0x80495d0
5	bufp1(.bss)	0x21	17	R-386-32	0x00000000	0x8049620
6	bufp1(.bss)	0x2a	21	R-386-32	0x00000000	0x8049620