

ICS homework3

张运吉 211300063

16/11/2022

P3

符号	是否在 test.o 的符号表中	定义模块	符号类型	节
a	是	main.o	外部	.data
val	是	test.o	全局	.data
sum	是	test.o	全局	.text
i	不是			

P4

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	外部	.data
bufp0	是	swap.o	全局	.data
bufp1	是	swap.o	本地	.bss
incr	是	swap.o	本地	.text
count	是	swap.o	本地	.data
swap	是	swap.o	全局	.text
temp	不是			

P5

(1)

main.c 中的强符号有 x、z、main，弱符号有 y、proc1；
proc1.c 中的强符号有 proc1，弱符号有 x。

(2)

x 被分配在 main.o 的.data 节中，占 4 个字节，其后是 z，占 2 个字节，
z 后的 2 字节空余。所以，执行第 7 行前 x 和 z 中存放的内容如下：

	0	1	2	3
&z	02	00
&x	01	01	00	00

根据符号解析的规则，在执行 proc1 时，x 绑定的是 main.o 中定义的 x，所以 $x = -1.5$ 实际上做的是把 -1.5 的机器数放到 x 开始的 8 个字节，执行第 7 行前 x 和 z 中存放的内容如下：

	0	1	2	3
&z	00	00	F8	BF
&x	00	00	00	00

所以最后打印的结果是“x=0,z=0”。

若第 3 行改为“short y=1, z=2;” 执行第 7 行前.data 中存放的内容如下：

	0	1	2	3
&y	01	00	02	00
&x	01	01	00	00

执行第 7 行后.data 中存放的内容如下：

	0	1	2	3
&y	00	00	F8	BF
&x	00	00	00	00

所以最好打印的结果是”x=0,z=-16392”.

(3)

将模块 proc1.c 中第 1 行修改为”static double x;” 就会使 x 被定义为本地符号，从而在调用 proc1 的时候 x 被重定位到 proc1.c 中定义的 x，就不会修改 main.c 中定义的 x 的值，从而得到期望的结果。

P6(3)

- 1.REF(m1.main)→ 在 m1 中没有对 main 的引用
- 2.REF(m2.main)→DEF(m1.main)
- 3.REF(m1.p1)→DEF(m2.p1)
- 4.REF(m1.x)→ 这里的 x 是局部变量
- 5.REF(m2.x)→DEF(m2.x)

P7

m1.c 中定义的 main 是强符号，p1 是弱符号，m2.c 中定义的 main 是弱符号，p1 是强符号。执行 main 函数时，调用的 p1 是在 m2.c 中定义的 p1，而 m2.c 中的 p1 函数中打印的 main 是 m1.c 中的 main，即输出 main 函数起始地址的两个字节内容，因为 main 函数的前两条机器指令是”55”和”89 e5”，所以最后会打印出字符串”0x5589”.

P8

可读写数据段由所有可重定位目标文件中的.data 节和.bss 节合并形成，.data 节中存的是初始化的全局变量，必须在可执行文件中记录它们的值，

而.bss 节存的是未初始化的全局变量，因此可执行文件中没有记录它们的值。

由图可知，只需将可执行文件中偏移地址 0x448 开始的 0xe8 个字节存放到对应的虚拟地址空间中，所以 0xe8 是可执行文件中.data 节的大小，后面的 28 字节是未初始化的全局变量所在的区域。

P9

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

P10

需重定位的符号名: swap

相对.text 节起始位置的偏移:0x7

所在指令行号: 6

重定位类型: R_386_PC32

重定位前的内容: $0xffffffffc = -4$

重定位后的内容: 07000000

计算过程:main 函数最后一条指令地址: $0x8048386+0x12 = 0x8048398$,swap 代码紧跟其后，按 4 字节边界对齐，所以 swap 的首地址为 0x8048398, 所以偏移量 $= 0x8048398 - (0x8048386 + 0x7) - (-4) = 7$