# 第四章作业

**191220008 陈南疃**

**4、**

| 符号 | 是否在swap.o的符号表中 | 定义模块 | 符号类型 | 节 |
| --- | --- | --- | --- | --- |
| buf | 在 | main.o | 外部符号 | .data |
| bufp0 | 在 | swap.o | 全局符号 | .data |
| bufp1 | 在 | swap.o | 本地符号 | .bss |
| incr | 在 | swap.o | 本地符号 | .text |
| count | 在 | swap.o | 本地符号 | .data |
| swap | 在 | swap.o | 全局符号 | .text |
| temp | 不在 | 无 | 无 | 无 |

**5、**

（1）

main.c：

强符号：x、z、main

弱符号：y、proc1

proc1.c：

强符号：proc1

弱符号：x

（2）

程序执行后打印的结果是：x=0，z=0

proc1调用前：&x=0x00000101，&z=0x0002

proc1调用后：&x=0x00000000，&z=0x0000

将第3行改为”short y=1，z=2“后，打印的结果是：x=0，z=-16392

（3）

将proc.c中 “double x” 改为 “static double x”

**7、**

由于全局符号main在m1中是强符号，在m2中是弱符号

当main在m2中被调用时，以m1中的定义为准

通常情况下，main函数最开始的两条指令是：“push %ebp” 和 “mov %ebp %esp”，

其所对应的指令码分别为 0x55 和 0x89 0xe5

当输出（”0x%x%x\n“，main[0]，main[1]）时，main[0]为0x55，main[1]为0x89，因而输出得到“0x5589\n”

**10、**

main.o的.text节中需要重定位的是在main.c中被引用的全局符号swap

相对于.text节起始位置的位移为0x7

所在指令行号为6

重定位类型为R_386_32

重定位前的内容为0xfffffffc，值为-4

重定位后的内容为0x00000007，值为7

计算过程：

main函数占有0x12字节的空间，其起始地址ADDR(.text)为0x8048386

因此与main紧跟着的swap函数起始地址ADDR(.swap)为0x8048386+0x12=0x8048398，按4字节对齐后仍是0x8048398

故重定位值为ADDR(.swap)-((ADDR(.text)+r_offset)-init)=0x8048398-((0x8048386+7)-(-4))=7

**11、**

| 序号 | 符号 | 位移 | 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) | 0x1b | 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 |