

7.6

```
extern int buf[]; // buf 是一个在其他文件中定义的外部整型数组
int* bufp0 = &buf[0]; // 声明 int 类型指针，指向 buf 数组首元素地址。
static int* bufp1; // 静态全局变量声明

static void incr( ) { // 静态函数的声明和定义
    static int count=0;
    count++;
}

void swap( ) {
    int temp;
    incr( );

    bufp1 = &buf[1];
    temp = *bufp0; // 将 bufp0 指向的值赋给 temp。
    *bufp0 = *bufp1; // 将 bufp1 指向的值赋给 bufp0 指向的位置。
    *bufp1 = temp; // 将 temp 的值赋给 bufp1 指向的位置，完成了两个元素的交换操作。
}
```

根据上述分析，结合模块、符号、节等的知识内容，可填充下列表格：

符号	swap.o.symtab条目?	符号类型	定义符号的模块	节
buf	是	外部	m.o	.data
bufp0	是	全局	swap.o	.data
bufp1	是	局部	swap.o	.bss
swap	是	全局	swap.o	.text
temp	否			
incr	是	局部	swap.o	.text
count	是	局部	swap.o	.bss

7.10

与 Text 中的题目类似，要注意，要适配库相互依赖的需求，必须要使得命令行中处于后面的库是被前面的库引用的。

- A. p.o -> libx.a -> p.o
- B. p.o -> libx.a -> liby.a 和 liby.a -> libx.a
- C. p.o -> libx.a -> liby.a -> libz.a 和 liby.a -> libx.a -> libz.a

A. gcc p.o libx.a

B. gcc p.o libx.a liby.a libx.a

C. gcc p.o libx.a liby.a libx.a libz.a

7.12

7.12 考虑目标文件 m.o 中对函数 swap 的调用(作业题 7.6)。

```
9:  e8 00 00 00 00      callq  e <main+0xe>      swap()
```

具有如下重定位条目：

```
r.offset = 0xa
r.symbol = swap
r.type   = R_X86_64_PC32
r.addend = -4
```

A. 假设链接器将 m.o 中的 .text 重定位到地址 0x4004e0，把 swap 重定位到地址 0x4004f8。那么 callq 指令中对 swap 的重定位引用的值应该是什么？

B. 假设链接器将 m.o 中的 .text 重定位到地址 0x4004d0，把 swap 重定位到地址 0x400500。那么 callq 指令中对 swap 的重定位引用的值应该是什么？

A、根据重定位的知识，由题可知：

$$\text{ADDR}(s) = \text{ADDR}(.text) = 0x4004e0$$
$$\text{ADDR}(r.symbol) = \text{ADDR}(swap) = 0x4004f8$$

计算引用的运行时地址：

$$\text{refaddr} = \text{ADDR}(s) + r.offset = 0x4004ea$$

再修改此引用：

$$\begin{aligned} *refptr &= (\text{unsigned})(\text{ADDR}(r.symbol) + r.addend - \text{refaddr}) \\ &= 0x4004f8 - 0x4 - 0x4004ea = 0xa \end{aligned}$$

所以在可执行目标文件中对 swap 的 PC 相对引用的值为 0x10.

对应： 400e9: e8 0a 00 00 00 callq 4004f8 <swap>

B、由题可知：

$$\text{ADDR}(s) = \text{ADDR}(.text) = 0x4004d0$$
$$\text{ADDR}(r.symbol) = \text{ADDR}(swap) = 0x400500$$

计算引用的运行时地址：

$$\text{refaddr} = \text{ADDR}(s) + r.offset = 0x4004da$$

修改此引用：

$$\begin{aligned} *refptr &= (\text{unsigned})(\text{ADDR}(r.symbol) + r.addend - \text{refaddr}) \\ &= 0x400500 - 0x4 - 0x4004da = 0x22 \end{aligned}$$

所以在可执行目标文件中对 swap 的 PC 相对引用的值为 0x22.

对应： 400d9: e8 22 00 00 00 callq 4004f8 <swap>