Homework

1. 本题是关于下图的 m.o 模块和下面的 swap.c 函数的。

```
/* m.c */
1 void swap();
2
3 int buf[2] = {1, 2};
4
5 int main()
6 {
7       swap();
8       return 0;
9 }
```

```
/* swap.c */
1 extern int buf[];
2
3 int *bufp0 = &buf[0];
4 static int *bufp1;
5
6 static void incr()
7 {
8         static int count = 0;
9
10        count++;
11 }
12
13 void swap()
14 {
15    int temp;
16
17    incr();
18    bufp1 = &buf[1];
19    temp = *bufp0;
20    *bufp0 = *bufp1;
21    *bufp1 = temp
22 }
```

对于每个 swap.o 中定义和引用的符号，请指出它是否在模块 swap.o 的.symtab 节中有符号表条目。如果是这样，请指出定义该符号的模块(swap.o 或 m.o)、符号类型(局部、全局或外部)以及它在模块中所处的节(.text、.data 或.bss)。

符号	.symtab 条目？	符号类型	定义符号的模块	节
buf	Yes	extern	m	.data
bufp0	Yes	global	swap	.data
bufp1	Yes	local	swap	.bss
swap	Yes	global	swap	.text
temp	No	—	—	—
incr	Yes	local	swap	.text
count	Yes	local	swap	.bss

2. 在此题中，REF(x,i)->DEF(x,k)表示链接器将任意对模块 i 中符号 x 的引用与模块 k 中符号 x 的定义相关联。在下面每个例子中，用这种符号来说明链接器是如何解析在每个模块中有多重定义的引用的。如果出现链接时错误(规则 1)，写“错误”。如果链接器从定义中任意选择一个(规则 3)，那么写“未知”。

A.

```
/* Module 1 */
int main()
{
}
```

```
/* Module 2 */
static int main = 1;
int p2()
{
}
```

(a) REF(main.1) -> DEF(main.1)
(b) REF(main.2) -> DEF(main.2)

B.

```
/* Module 1 */
int x;
void main()
{
}
```

```
/* Module 2 */
double x;
int p2()
{
}
```

(a) REF(x.1) -> DEF(unknown)
(b) REF(x.2) -> DEF(unknown)

C.

```
/* Module 1 */
int x = 1;
void main()
{
}
```

```
/* Module 2 */
double x = 1.0;
int p2()
{
}
```

(a) REF(x.1) -> DEF(error)
(b) REF(x.2) -> DEF(error)

3. 考虑目标文件 m.o 中对函数 swap 的调用（作业题 1 中的程序）

```
 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？

Solution:
A.
ADDR(s) = ADDR(.text) = 0x4004e0
ADDR(r.symbol) = ADDR(swap) = 0x4004f8
refaddr = ADDR(s) + r.offset = 0x4004ea
*refptr = (unsigned) (ADDR(r.symbol) + r.addend - refaddr) = 0xa
B.
ADDR(s) = ADDR(.text) = 0x4004d0
ADDR(r.symbol) = ADDR(swap) = 0x400500
refaddr = ADDR(s) + r.offset = 0x4004da
*refptr = (unsigned) (ADDR(r.symbol) + r.addend - refaddr) = 0x22