

Homework 10

Problem 1

Let a and b denote object modules or static libraries in the current directory, and let $a \rightarrow b$ denote that a depends on b , in the sense that b defines a symbol that is referenced by a . For each of the following scenarios, show the minimal command line (i.e., one with the least number of object file and library arguments) that will allow the static linker to resolve all symbol references:

- A. $p.o \rightarrow \text{libx.a} \rightarrow p.o$
- B. $p.o \rightarrow \text{libx.a} \rightarrow \text{liby.a}$ and $\text{liby.a} \rightarrow \text{libx.a}$
- C. $p.o \rightarrow \text{libx.a} \rightarrow \text{liby.a} \rightarrow \text{libz.a}$ and $\text{liby.a} \rightarrow \text{libx.a} \rightarrow \text{libz.a}$

Problem 2

Consider the C code and corresponding reloadable object module in following Figure.

- A. Determine which instructions in `.text` will need to be modified by the linker when the module is relocated. For each such instruction, list the information in its relocation entry: section offset, relocation type, and symbol name.
- B. Determine which data objects in `.data` will need to be modified by the linker when the module is relocated. For each such instruction, list the information in its relocation entry: section offset, relocation type, and symbol name.

Feel free to use tools such as `objdump` to help you solve this problem.

(a) C code

```
1  extern int p3(void);
2  int x = 1;
3  int *xp = &x;
4
5  void p2(int y) {
6  }
7
8  void p1() {
9      p2(*xp + p3());
10 }
```

(b) .text section of relocatable object file

```
1 00000000 <p2>:
2 0: 55 push %ebp
3 1: 89 e5 mov %esp,%ebp
4 3: 89 ec mov %ebp,%esp
5 5: 5d pop %ebp
6 6: c3 ret

7 00000008 <p1>:
8 8: 55 push %ebp
9 9: 89 e5 mov %esp,%ebp
10 b: 83 ec 08 sub $0x8,%esp
11 e: 83 c4 f4 add $0xffffffff,%esp
12 11: e8 fc ff ff call 12 <p1+0xa>
13 16: 89 c2 mov %eax,%edx
14 18: a1 00 00 00 00 mov 0x0,%eax
15 1d: 03 10 add (%eax),%edx
16 1f: 52 push %edx
17 20: e8 fc ff ff call 21 <p1+0x19>
18 25: 89 ec mov %ebp,%esp
19 27: 5d pop %ebp
20 28: c3 ret
```

(c) .data section of relocatable object file

```
1 00000000 <x>:
2 0: 01 00 00 00
3 00000004 <xp>:
4 4: 00 00 00 00
```

Problem 3

Consider the C code and corresponding relocatable object module in Figure 7.21.

A. Determine which instructions in .text will need to be modified by the linker when the module is relocated. For each such instruction, list the information in its relocation entry: section offset, relocation type, and symbol name.

B. Determine which data objects in .rodata will need to be modified by the linker when the module is relocated. For each such instruction, list the information in its relocation entry: section offset, relocation type, and symbol name.

Feel free to use tools such as objdump to help you solve this problem.

(a) C code

```
1 int relo3(int val) {
2     switch (val) {
3         case 100:
4             return(val);
5         case 101:
6             return(val+1);
7         case 103: case 104:
8             return(val+3);
9         case 105:
10            return(val+5);
11        default:
12            return(val+6);
13    }
14 }
```

(b) .text section of relocatable object file

```
1 00000000 <relo3>:
2 0: 55 push %ebp
3 1: 89 e5 mov %esp,%ebp
4 3: 8b 45 08 mov 0x8(%ebp),%eax
5 6: 8d 50 9c lea 0xffffffff9c(%eax),%edx
6 9: 83 fa 05 cmp $0x5,%edx
7 c: 77 17 ja 25 <relo3+0x25>
8 e: ff 24 95 00 00 00 00 jmp *0x0(,%edx,4)
9 15: 40 inc %eax
10 16: eb 10 jmp 28 <relo3+0x28>
11 18: 83 c0 03 add $0x3,%eax
12 1b: eb 0b jmp 28 <relo3+0x28>
13 1d: 8d 76 00 lea 0x0(%esi),%esi
14 20: 83 c0 05 add $0x5,%eax
15 23: eb 03 jmp 28 <relo3+0x28>
16 25: 83 c0 06 add $0x6,%eax
17 28: 89 ec mov %ebp,%esp
18 2a: 5d pop %ebp
19 2b: c3 ret
```

(c) .rodata section of relocatable object file

```
This is the jump table for the switch statement
1 0000 28000000 15000000 25000000 18000000 4 words at offsets 0x0,0x4,0x8, and 0xc
2 0010 18000000 20000000 2 words at offsets 0x10 and 0x14
```