ICS Homework 6

Cao Shengcao

7.6

Symbol	`swap.o` `.symtab` entry?	Symbol type	Module where defined	Section
`buf`	Yes	Extern	`m.o`	`.data`
`bufp0`	Yes	Global	`swap.o`	`.data`
`bufp1`	Yes	Local	`swap.o`	`.bss`
`swap`	Yes	Global	`swap.o`	`.text`
`temp`	No	/	/	/
`incr`	Yes	Local	`swap.o`	`.text`
`count`	Yes	Local	`swap.o`	`.bss`

Recall:

- Local nonstatic variables don't have their corresponding symbols in `.symtab`.
- `.data` section contains symbols for initialized (non-zero) global and static variables.
- `.bss` section contains symbols for uninitialized static variables and global or static variables that are initialized to zero.
- `COMMON` "pseudosection" contains symbols for uninitialized global variables.

7.7

To prevent duplicate symbol names, we can simply make `x` in `bar5.c` a local static variable:

```c
static double x;

void f()
{
    x = -0.0;
}
```

7.12

Apply the algorithm described in 7.7.1.

- A. The value should be `0xa`.

```
ADDR(s) = ADDR(.text)
        = 0x4004e0
ADDR(r.symbol) = ADDR(swap)
               = 0x4004f8
refaddr = ADDR(s) + r.offset
        = 0x4004e0 + 0xa = 0x4004ea
*refptr = ADDR(r.symbol) + r.addend - refaddr
        = 0x4004f8 + (-4) - 0x4004ea = 0xa
```

- B. The value should be `0x22`.

```
ADDR(s) = ADDR(.text)
        = 0x4004d0
ADDR(r.symbol) = ADDR(swap)
               = 0x400500
refaddr = ADDR(s) + r.offset
        = 0x4004d0 + 0xa = 0x4004da
*refptr = ADDR(r.symbol) + r.addend - refaddr
        = 0x400500 + (-4) - 0x4004da = 0x22
```

7.13

- A. First we need to `locate` this two libraries. Type `locate libc.a`, we can find three locations:

```
/usr/lib/x86_64-linux-gnu/libc.a
/usr/lib32/libc.a
/usr/libx32/libc.a
```

  Take the first one as an example. Different versions of the library may include different numbers of `.o` files. Go to this directory and use the AR tool by typing `ar -t libc.a`, and we'll see a really long list. To count exactly how many `.o` files there are, we must use some tricks.

  When I saw this exercise last year, I redirected the output to a file like `ar -t libc.a >; ~/list.txt` and opened the file to see how many lines there were, and I got the following result:

  - 1579 `.o` files in `libc.a`
  - 471 `.o` files in `libm.a`

  A more graceful way to do this is `ar -t libc.a | wc -l`. How does this work? The character `|` builds something called "pipe" between the output of `ar` and the input of `wc`. The former command's output acts as the latter one's input, and `wc -l` counts how many lines there are in the list.

- B. Yes.

  For example, we can write a simple hello-world C program `test.c`:

```c
#include <stdio.h>
int main()
{
    printf("Hello world!\n");
    return 0;
}
```

Then we compile it with command `gcc test.c -o test_1 -Og` and `gcc test.c -o test_2 -Og -g` respectively. The latter one is larger in size (8.6KB and 11.3KB).

Use the READELF tool by typing `readelf -a test_1` and `readelf -a test_2`, and then we can compare them. At least, we can see there are six more segment headers:

```
  [28] .debug_aranges    PROGBITS         0000000000000000  0000106c
       0000000000000030  0000000000000000           0     0     1
  [29] .debug_info       PROGBITS         0000000000000000  0000109c
       0000000000000346  0000000000000000           0     0     1
  [30] .debug_abbrev     PROGBITS         0000000000000000  000013e2
       000000000000013c  0000000000000000           0     0     1
  [31] .debug_line       PROGBITS         0000000000000000  0000151e
       00000000000000e3  0000000000000000           0     0     1
  [32] .debug_str        PROGBITS         0000000000000000  00001601
       0000000000000266  0000000000000001  MS       0     0     1
  [33] .debug_loc        PROGBITS         0000000000000000  00001867
       0000000000000028  0000000000000000           0     0     1
```

Obviously, `-g` option provides us with much more debug information.

- C. Use the LDD tool by typing `ldd /usr/bin/gcc`, and we can see the shared libraries that the GCC driver use:
  - `linux-vdso.so.1 =>;  (0x00007ffe1af96000)`
  - `libc.so.6 =>; /lib/x86_64-linux-gnu/libc.so.6 (0x00007ff0f985f000)`
  - `/lib64/ld-linux-x86-64.so.2 (0x000055c72ec81000)`