CS4400 *Alex Clemmer, u0458675*

Problem Set 9

1 7.6

Symbol	swap.o .symtab entry?	Symbol type	Module where defined	Section
buf	yes	extern	main.o	.data
bufp0	yes	global	swap.o	.data
bufp1	yes	local	swap.o	.text
swap	yes	global	swap.o	.text
temp	no	-	-	-
incr	yes	local	swap.o	.text
count	yes	local	swap.o	.data

2 7.7

This transformation is trivial. All we need to do is to overwrite the memory locations of x and y again:

```
static double x;

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

3 7.8

1. REF(main.1) --> DEF(main.1) and REF(main.2) --> DEF(main.2). The main in module 2 is static, and thus local, so there will be no conflict in main.1; in main.2, the memory location will be overwritten to accommodate the definition of the symbol locally.

2. UNKNOWN. The reason is because they are both weak, so we can pick either arbitrarily.

3. ERROR. They are both strong, which is illegal.

4 7.9

This happens because `main` is clearly bound as a method in `foo6.c`; this supercedes the weak `char main`, and thus the program prints the odd result.

5 7.10

1.
```
gcc p.o libx.a
```

2.
```
gcc p.o libx.a liby.a libx.a
```

3.
```
gcc p.o libx.a liby.a libz.a libx.a liby.a
```

6 7.11

Interestingly, this indicates that we will be allocating `.bss`, but that it will be left as zeros. The data itself will occupy the first chunk, as noted in the question.