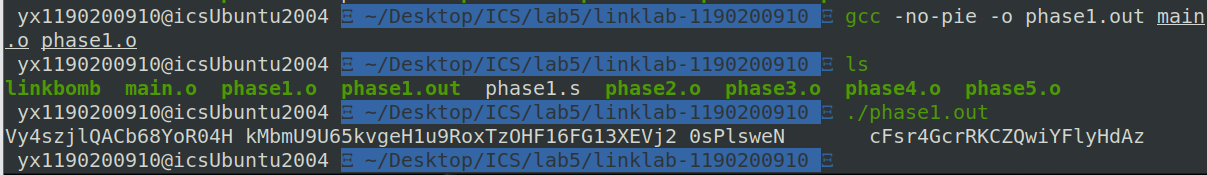
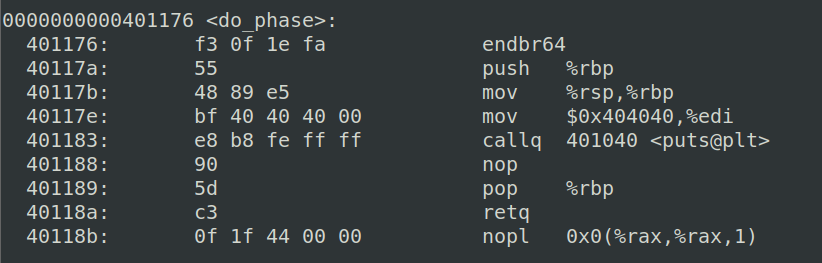
首先直接编译链接main.o和phase1.o，并运行，可以看到输出了乱码。

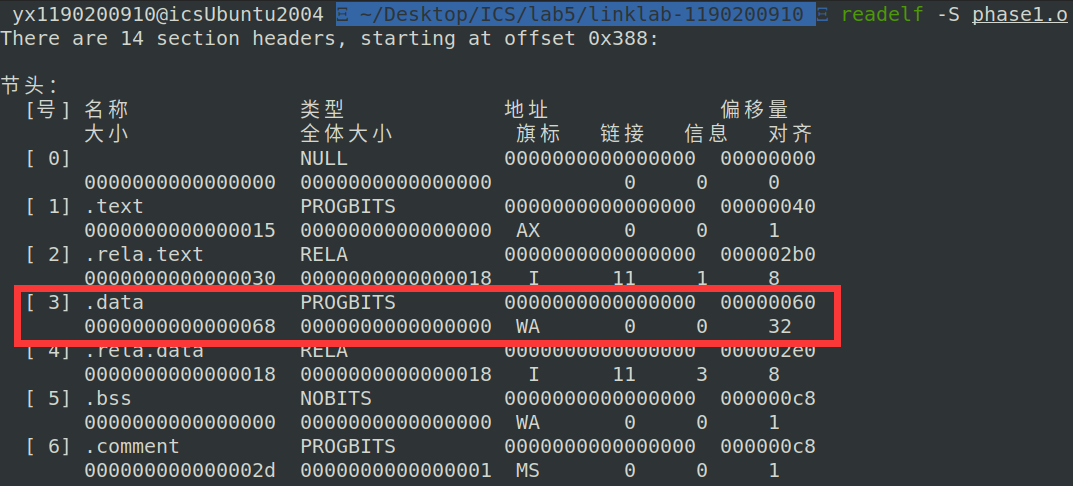


使用objdump看一下链接phase1.o后生成的可执行文件的do\_phase函数:



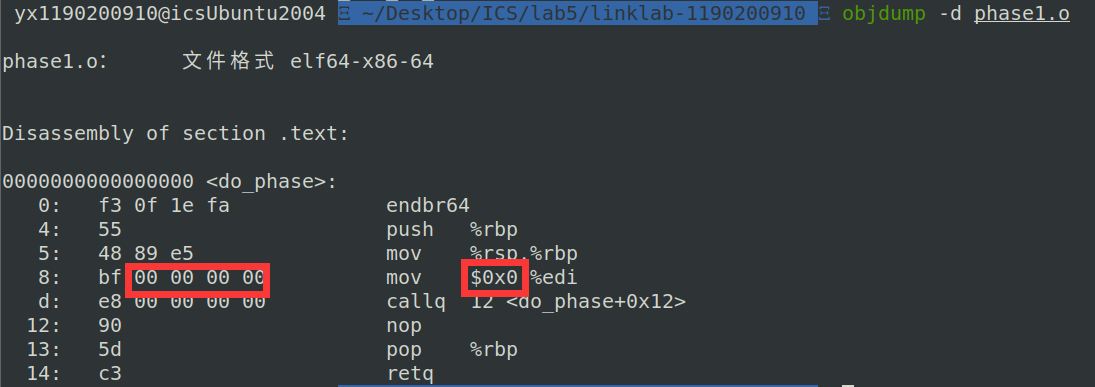
在callq puts函数之前，将0x404040作为puts的第一个参数传到了edi寄存器，说明这个地方应该存着字符串常量的地址，应该在.data节。

然后使用readelf -S看一眼phase1.o文件的节头信息：



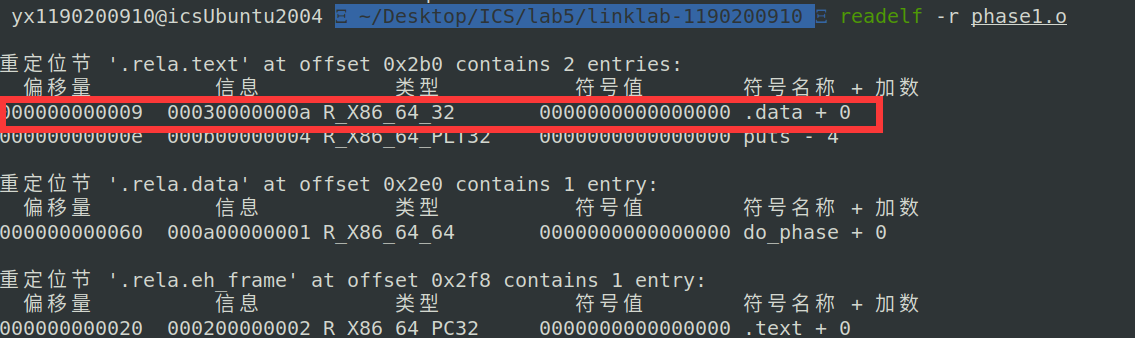
其中我们需要重点关注.data节在文件中的偏移，这里是偏移0x60个字节。因此马上修改ELF文件中.data节的内容时，就从文件的第0x60个字节开始修改。

然后我们用objdump -d看一眼phase1.o的反汇编代码：



上图中的用红框框起来的部分就是需要重定位所在的地址。

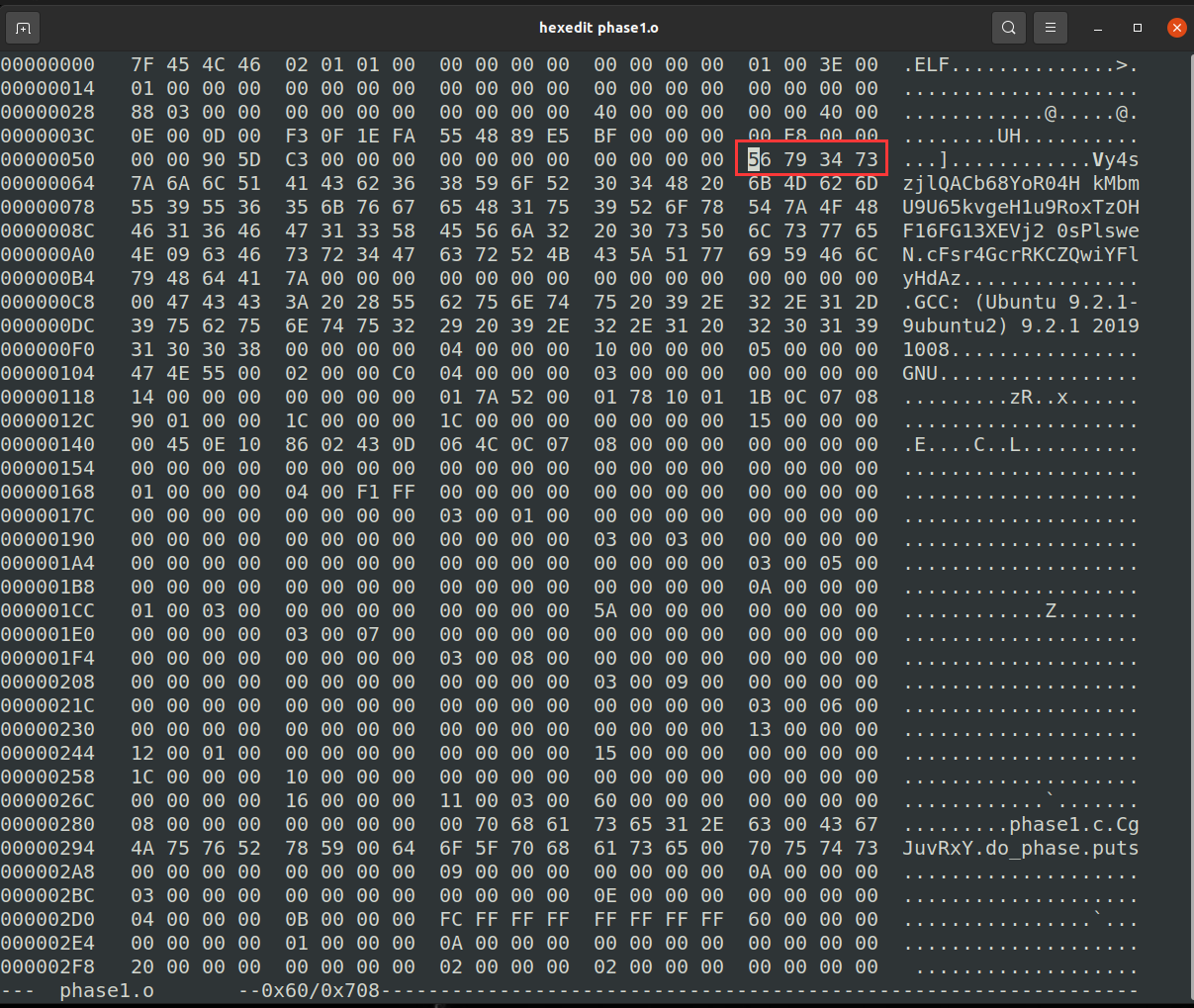
然后再用readelf -r看一眼phase1.o的重定位信息：



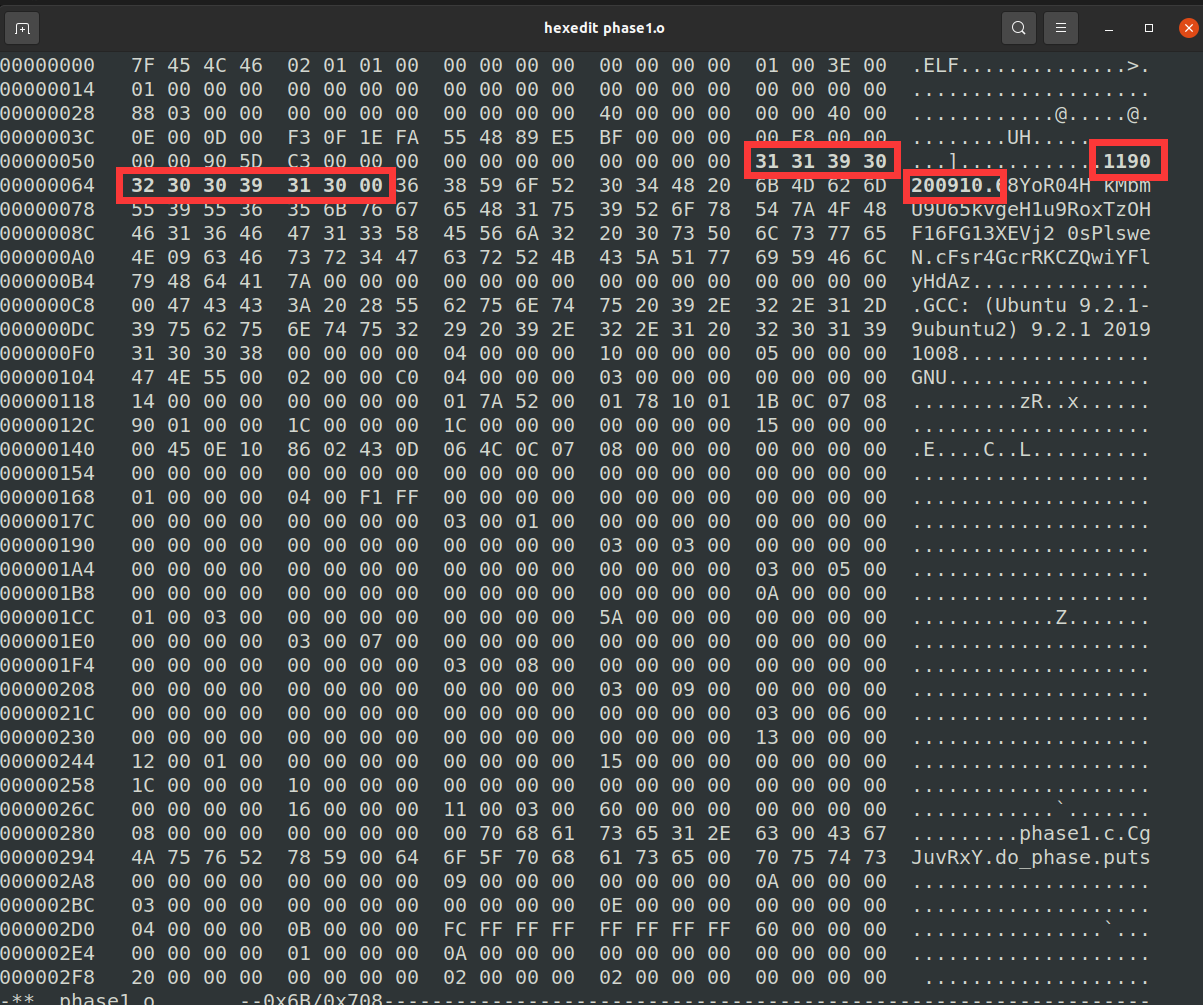
用红框框起来的这一条重定位条目，应该就是puts输出的字符串的所在地址，它位于.data节偏移量0的位置，也就是我们需要从.data节偏移量0的地方修改。

综合上面的分析，.data节从phase1.o文件的第0x60个字节开始，我们需要修改.data节偏移量为0的字符串。

然后我们开始使用hexedit：



框起来的部分是phase1.o文件的第0x60个字节，也就是.data节开始的地方，我们需要修改这里保存的字符串，将它改为“1190200910”，用ASCII码写入。注意字符串要以0结尾。



写入完成后按下Ctrl+W保存，按Ctrl+X退出hexedit。

然后再重新链接并运行程序，可以看到已经输出我的学号。

