#### 实验1（暂定，可能会有更新）

注：gcc、gdb、objdump使用介绍见课本3.2.2 & 3.11

gcc编译时都使用1级优化

1. 熟悉Linux常用命令，如cd、mkdir、rmdir、rm、cat等。

创建自己的工作目录。

1. 修改第1章中 hello.c，将字符串中的 “world”替换为自己的姓名拼音或英文名
2. 用gcc –O1 -S命令生成.s文件，查看其中的汇编代码
3. 用gcc –O1 –c命令生成.o文件
4. 用objdump对.o文件反汇编，参数-d只查看函数代码，-D查看更多
5. 用gcc –O1命令生成可执行目标文件a.out（或自定义文件名）
6. 输入 ./a.out 运行程序
7. 用objdump对a.out反汇编，查看字符串存储地址
8. 根据该地址查看字符串，有两种方法：

* Objdump –D 后找到该地址，查看之后的各字节

注：这里应为字符串的ASCII码。反汇编后看起来像指令，实际是只读数据段里的数据，可忽略右侧汇编代码，只查看十六进制数据。

* 使用gdb工具的x命令查看指定地址处开始的n字节内容

1. 编程实现第2章图2.4、图2.5代码，查看自己学号的int、float、指针的字节表示，分析float如何编码。
2. 编程实现练习题2.12，练习掩码的使用，掌握清零、取补、置1的位级运算。

取w=8, 16, 32的不同字长运行程序

1. 编程实现练习题2.14，了解位级运算和逻辑运算的区别
2. 编程实现移位运算，将自己学号的负值左移和右移不同位数k，测试k大于31时的运行结果
3. A = 0xA1A0，B = 0xB1A0，A1A0为学号末两位数字，B1为A1最高位取反。

如，学号为15211001，则A = 0x01，B = 0x81

编程输出A、B分别作为char、unsigned char类型数据时的十进制数值，分析有无符号数之间的数值大小关系

1. A、B取值同上，编程将A、B转换为unsigned short和short，查看转换后的十六进制表示和十进制数值
2. 编程实现练习题2.43，查看.s文件，掌握常数乘除的汇编代码实现。
3. 查看以上各程序的汇编代码。