实验一

实验报告

计算机科学与技术系

191220008 陈南曈

**一、实验目的**

1. 学会自己安装 Linux 系统

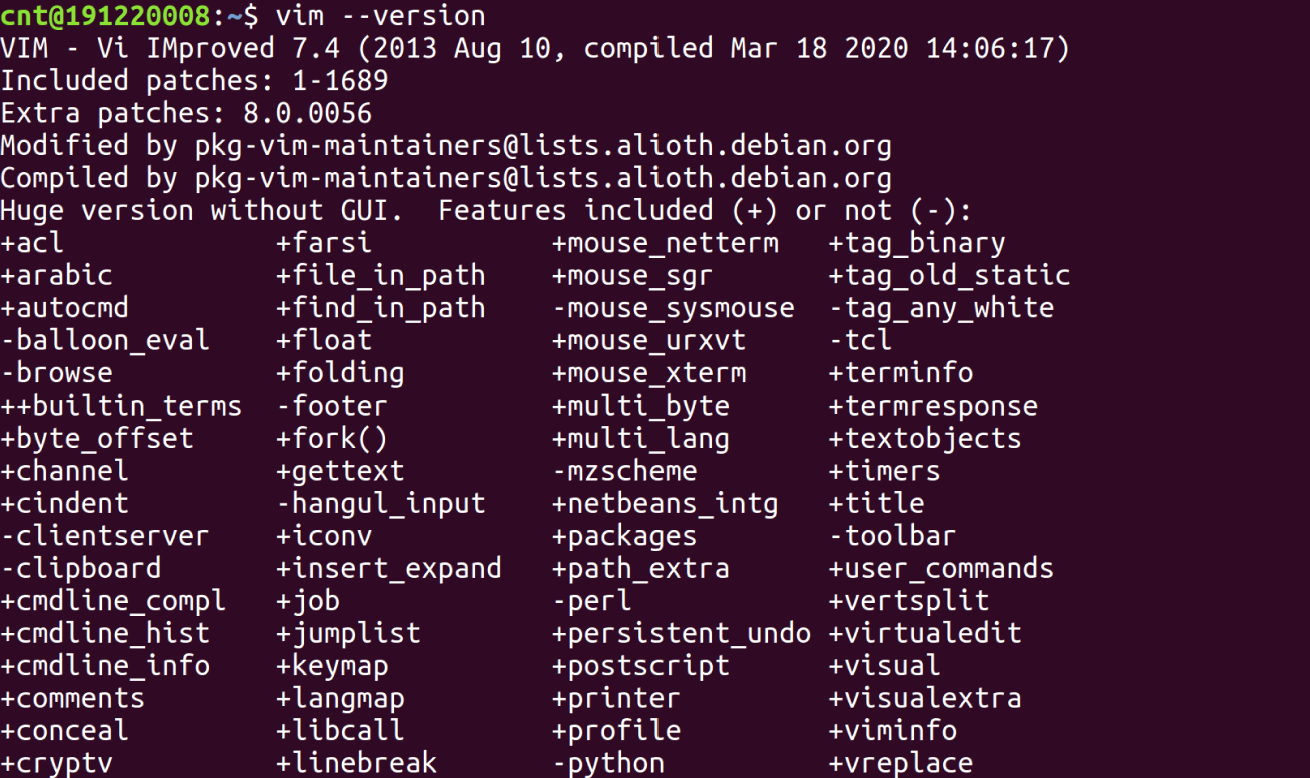
2. 学会配置简单的 Linux 开发环境

3. 在 Linux 下完成简单编程练习并熟悉各种命令行工具的使用方法

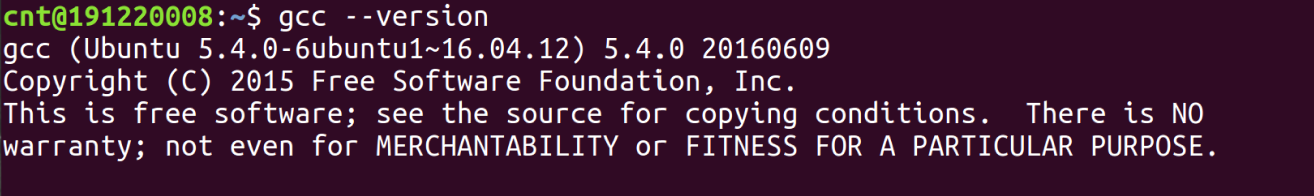
**二、实验内容**

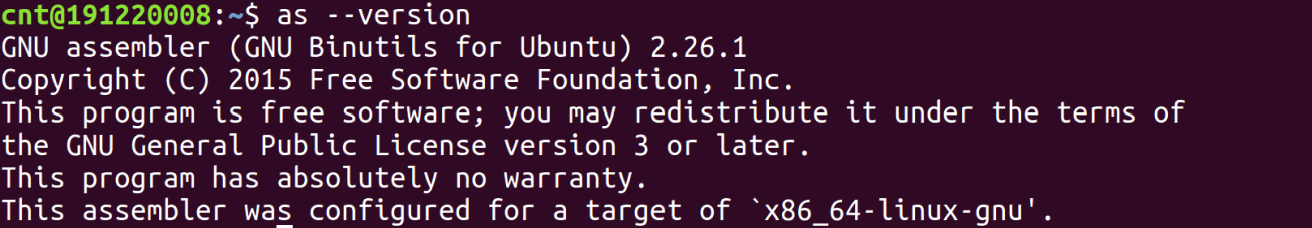
1、Linux的安装和配置

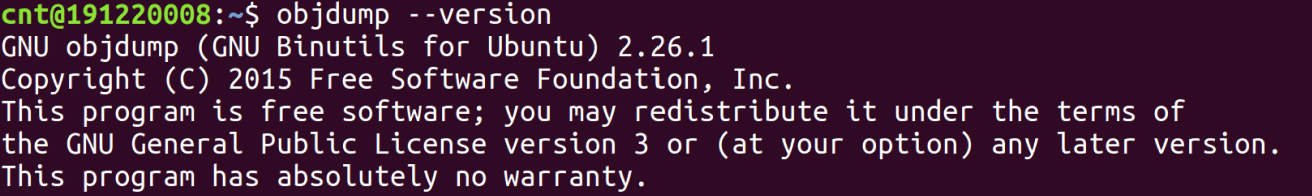
ii. 使用 man 查询 vim/git/gcc/as/objdump/gdb 版本的命令，然后使用查到的命令打印出对应版本

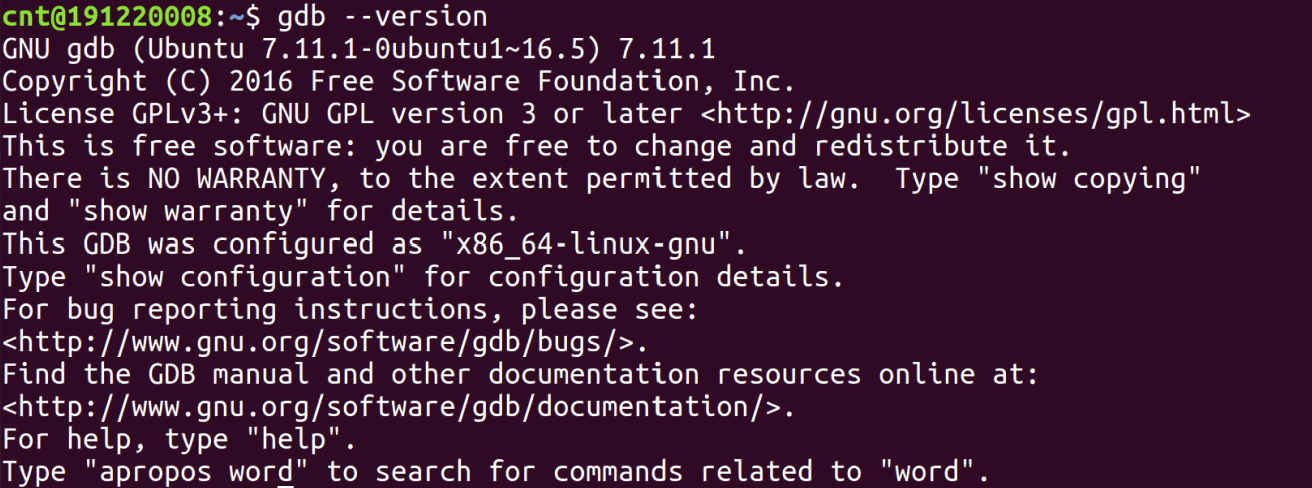












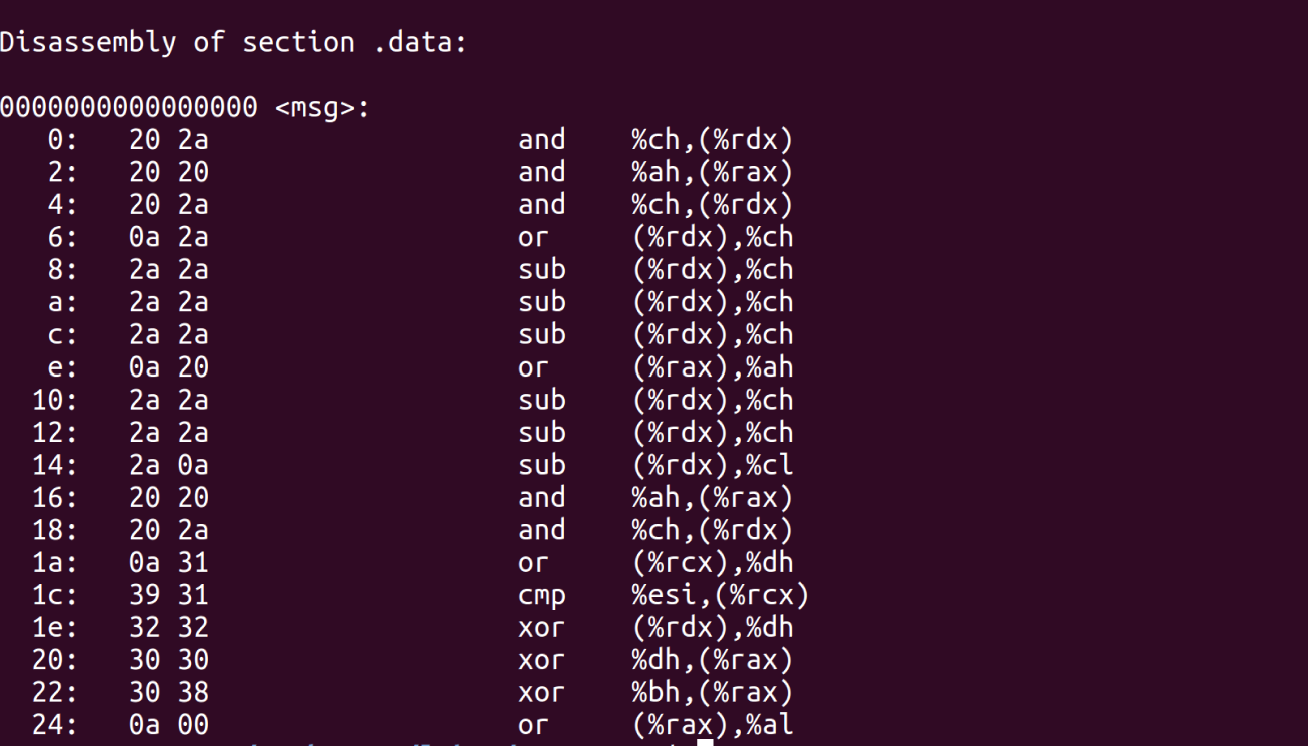
（gdb的版本查看方式未在man中找到，猜测可以用--version查看，事实也的确如此）

iii. 写下你在安装过程中遇到的问题，并说明你是如何解决的

（无）之前已经安装完毕

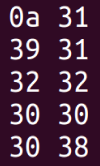
3、熟悉工具

i. 使用 objdump 的-D 选项反汇编 heart.o 文件，找到你学号的位置



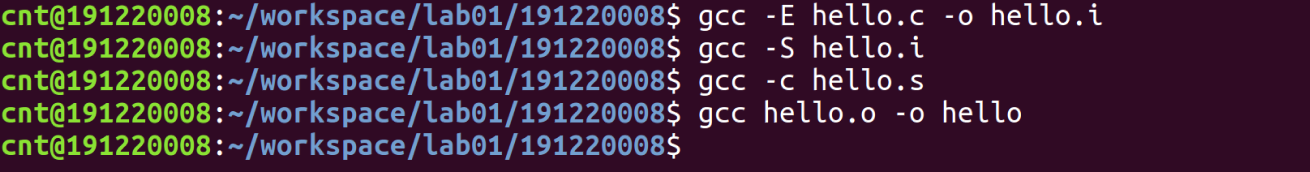
真实学号：191220008

学号如下（除去0a）



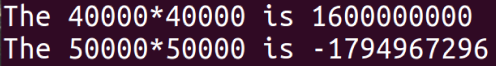
即：31 39 31 32 32 30 30 30 38

ii. 编写简单的 C 语言源程序 hello.c，通过预处理、编译、汇编、链接四个步骤将 C 语言源程序转换为可执行文件，即 hello.c -> hello.i -> hello.s -> hello.o -> hello



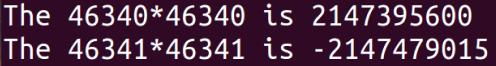
4、数据的表示范围及不同类型的数据长度实验

i. 将输出结果导出，说明发生这种现象的原因？



原因： int类型整数对应的二进制位数是有上限的，当该整数超出2147483647时会发生溢出，导致结果错误

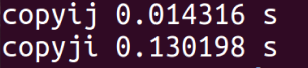
ii. 寻找在该程序中保证结果正确的最大整数值？



答案：46340

5、矩阵运算执行时间比较

i. 比较两个矩阵复制函数的执行时间。



ii. 说明为什么会出现这个差别。

原因：第一种复制函数按行进行读取（或者读取该行下一个），是连续访问内存；第二种复制函数按列进行读取，是跳跃访问内存。在进行数据缓存的时候是将二维数组的整行进行存储（其实不一定是整行，严格地说是存储一个区域的元素，只是在本题中2048×2048的数组中，因为元素较多，只能缓存部分元素，具体能存储多少元素，取决于数组大小、缓存大小、页面大小），如果按列读取则会去内存中读取。直接从缓存中读取数据的速度比从内存中读取更快。