**北 京 林 业 大 学**

**2017学年—2018学年第 二 学期 Linux应用实验报告书**

专 业： 计算机科学与技术(创新实验班) 班 级： 计创16

姓 名： 陈楠 学 号： 161002107

实验地点： 计算中心N09 任课教师： 李群

实验题目： Linux下C编程

实验环境： Linux操作系统

实验目的、实现内容、实验结果及结论分析等：

一．实验目的：

1. 掌握vi编辑器的使用；
2. 掌握GCC的编译器的使用；
3. 掌握调试工具GDB的使用；
4. 掌握编写makefile文件的方法。

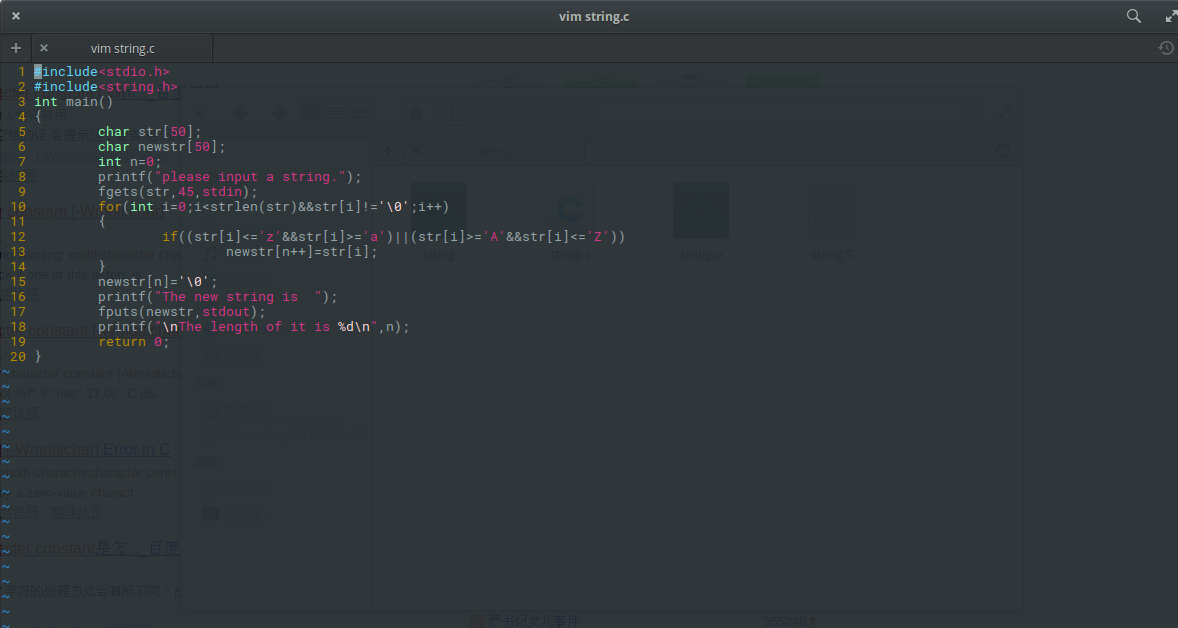
二．实验内容：

1. 用vi编辑器编辑一个应用程序，功能：输入一个字符串，过滤此串，只保留串中的字母字符，并统计新生成串中包含的字母个数。（例如：输入的字符串为ab234$df4，新生成的串为abdf 。）使用gcc进行编译，并分别使用-E，-S，-c，-o，-static，-O2(用time命令)等选项，编译为可执行文件并执行。用抓图的形式保存整个程序的显示、编译、运行过程。（注：Linux下输入字符串函数fgets()、输出字符串函数fputs()。）
2. 编写一个应用程序，功能：从键盘输入一个整数，判断其是否为素数，然后在main函数中输出相应的结论信息。（提示：素数也称质数。一个大于1的自然数，除了1和它自身外，不能被其他自然数整除的数叫做质数，否则称为合数。例如：7是素数，8不是素数。）使用gcc进行编译，并分别使用-E，-S，-c，-o，-static，-O2(用time命令)等选项，编译为可执行文件并执行。用抓图的形式保存整个程序的显示、编译、运行过程。
3. 编写一个应用程序，功能是：实现1～100求和。使用gdb调试，调试中使用到本章节所介绍的GDB的几个主要功能来完成调试过程。用抓图的形式保存整个程序的调试、运行过程。
4. 实现一应用程序，该程序至少包含有两个c文件构成，使用makefile来完成对该程序的编译功能。

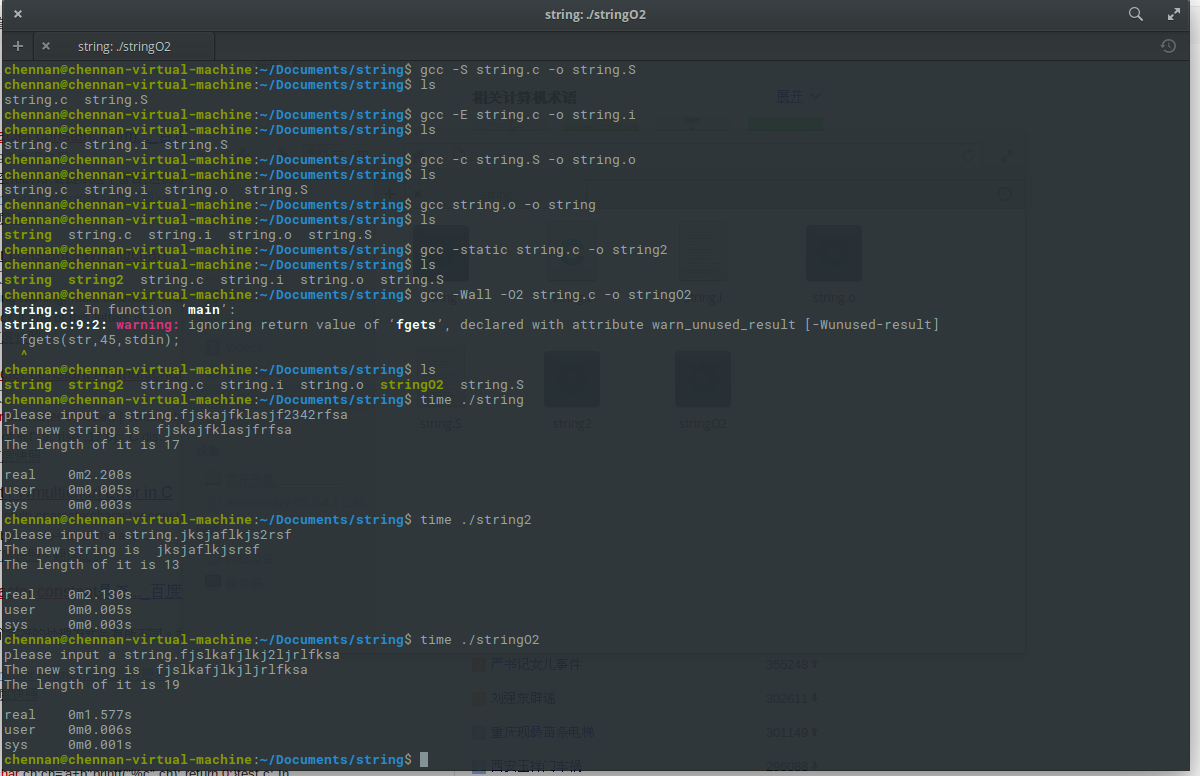
三．实验结果：

1. 用vi编辑器编辑一个应用程序，功能：输入一个字符串，过滤此串，只保留串中的字母字符，并统计新生成串中包含的字母个数。（例如：输入的字符串为ab234$df4，新生成的串为abdf 。）使用gcc进行编译，并分别使用-E，-S，-c，-o，-static，-O2(用time命令)等选项，编译为可执行文件并执行。用抓图的形式保存整个程序的显示、编译、运行过程。（注：Linux下输入字符串函数fgets()、输出字符串函数fputs()。

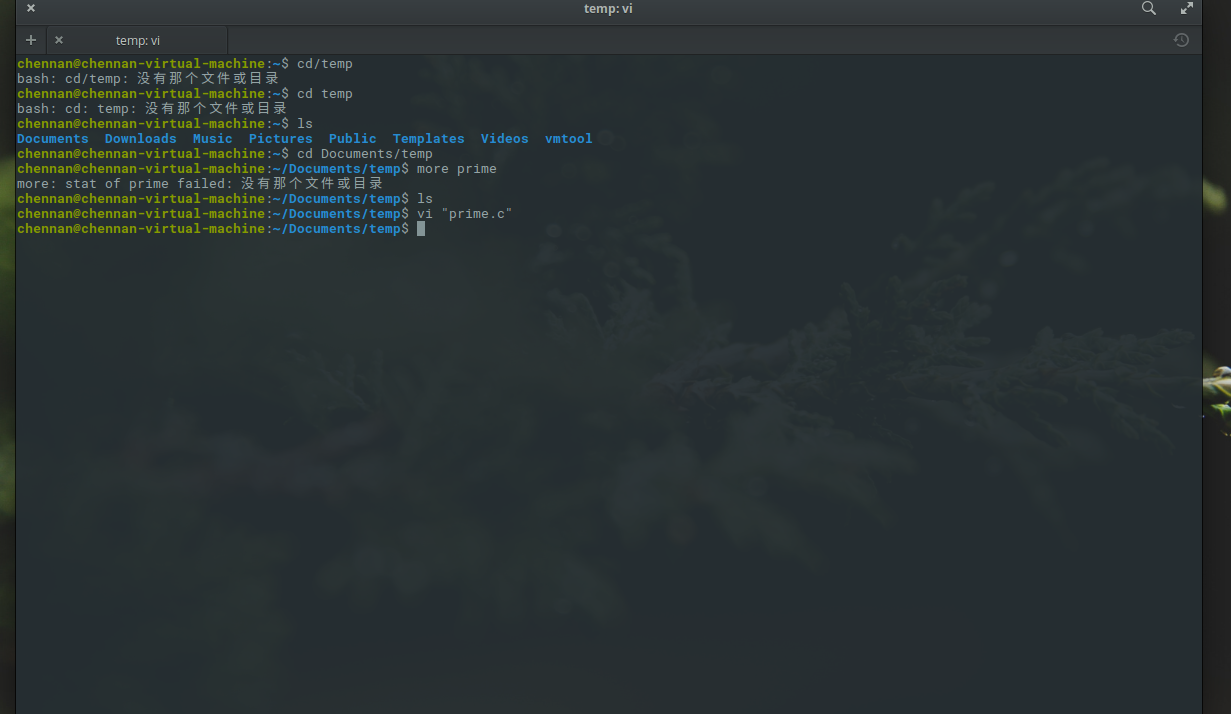
建立string.c文件



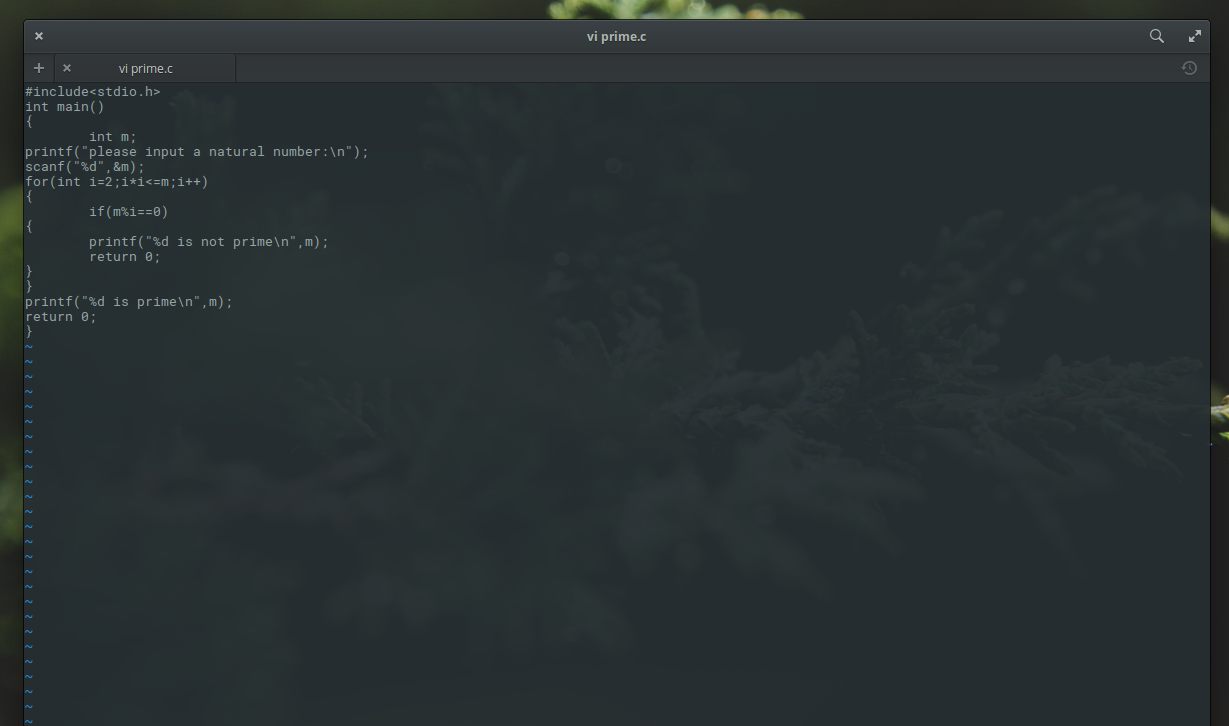
编译运行string.c文件



1. 编写一个应用程序，功能：从键盘输入一个整数，判断其是否为素数，然后在main函数中输出相应的结论信息。（提示：素数也称质数。一个大于1的自然数，除了1和它自身外，不能被其他自然数整除的数叫做质数，否则称为合数。例如：7是素数，8不是素数。）使用gcc进行编译，并分别使用-E，-S，-c，-o，-static，-O2(用time命令)等选项，编译为可执行文件并执行。用抓图的形式保存整个程序的显示、编译、运行过程。

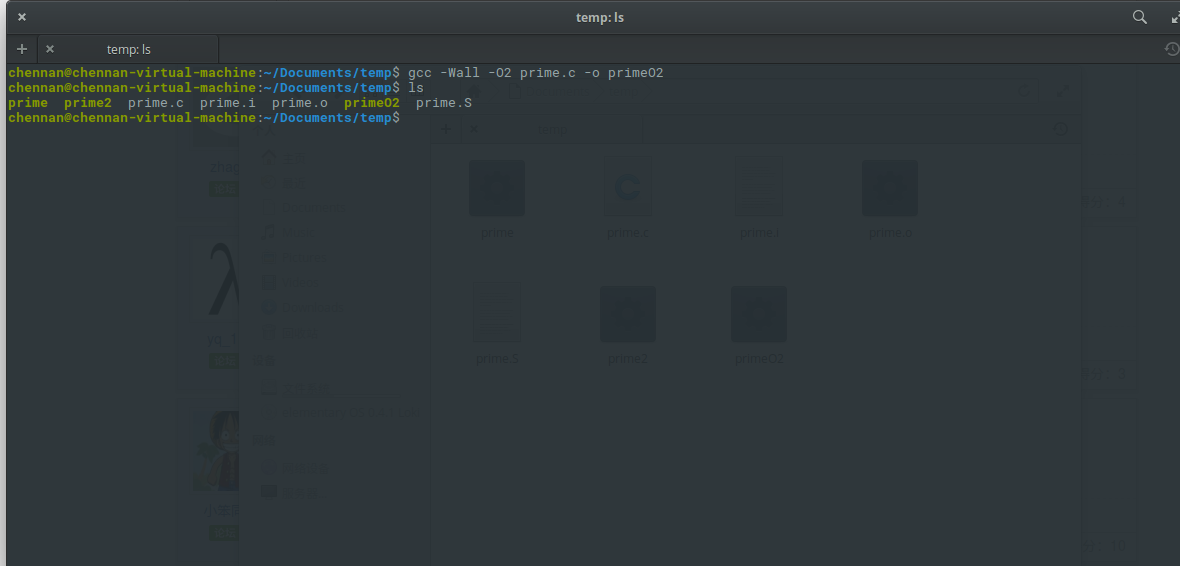


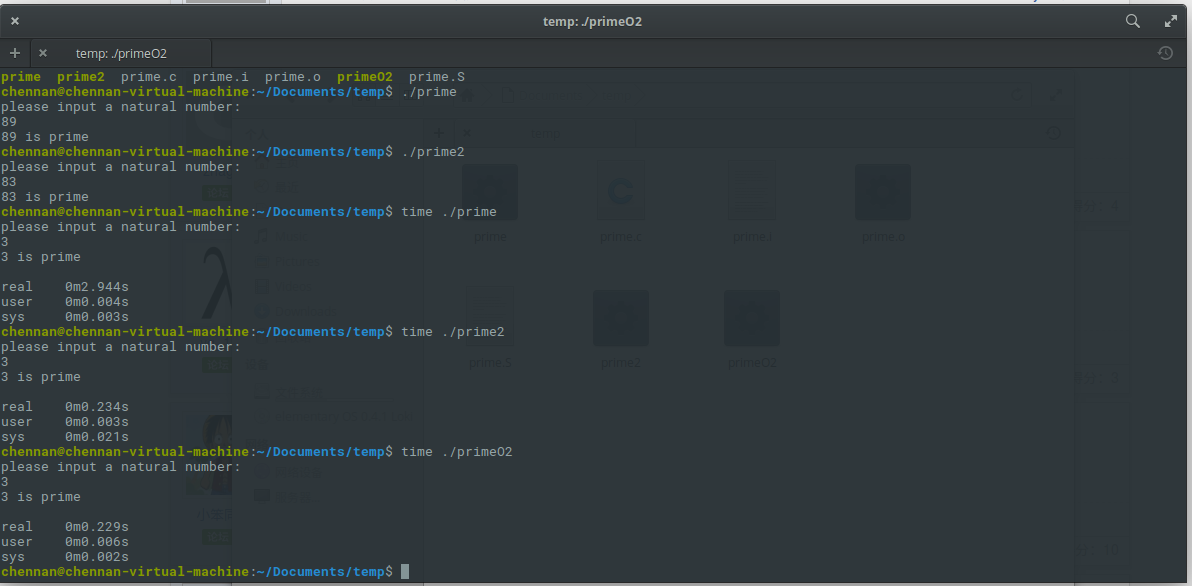
建立并编写prime.c文件



编译并运行prime.c文件

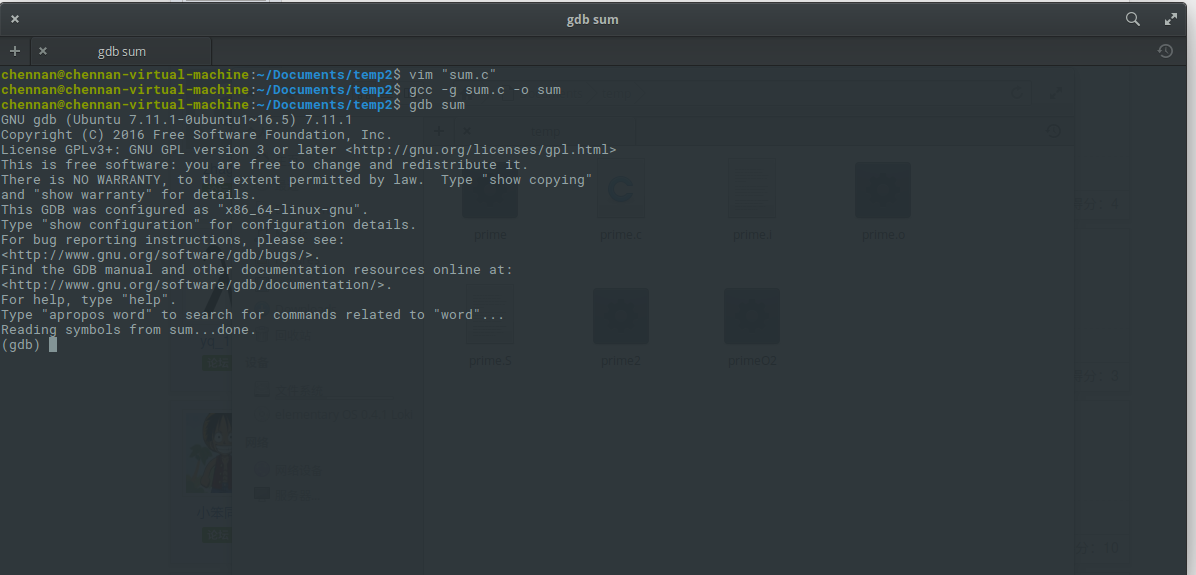




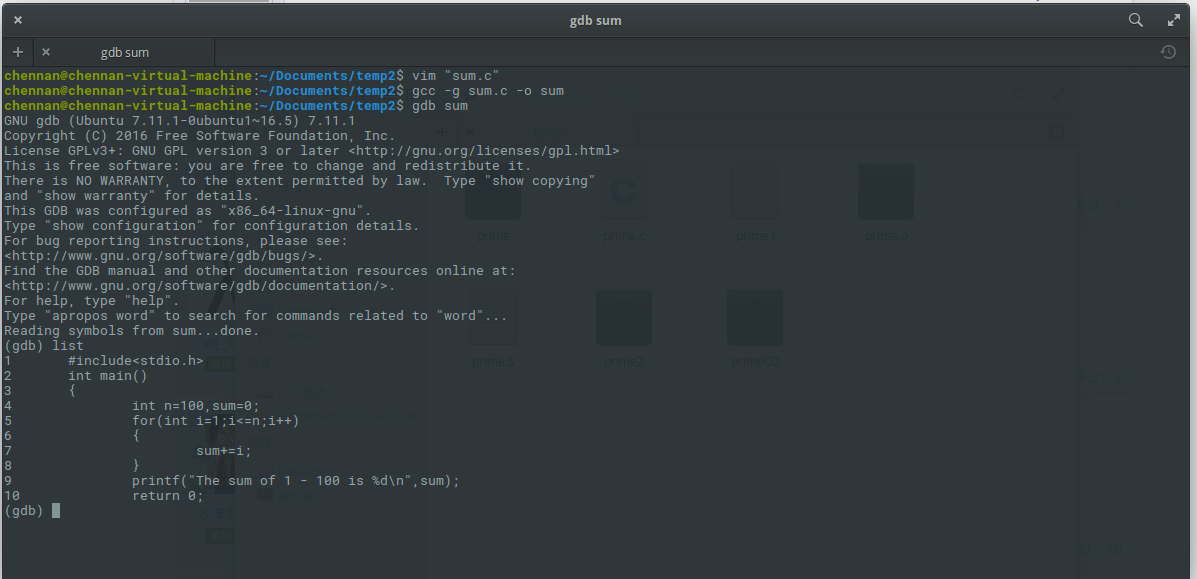


1. 编写一个应用程序，功能是：实现1～100求和。使用gdb调试，调试中使用到本章节所介绍的GDB的几个主要功能来完成调试过程。用抓图的形式保存整个程序的调试、运行过程。

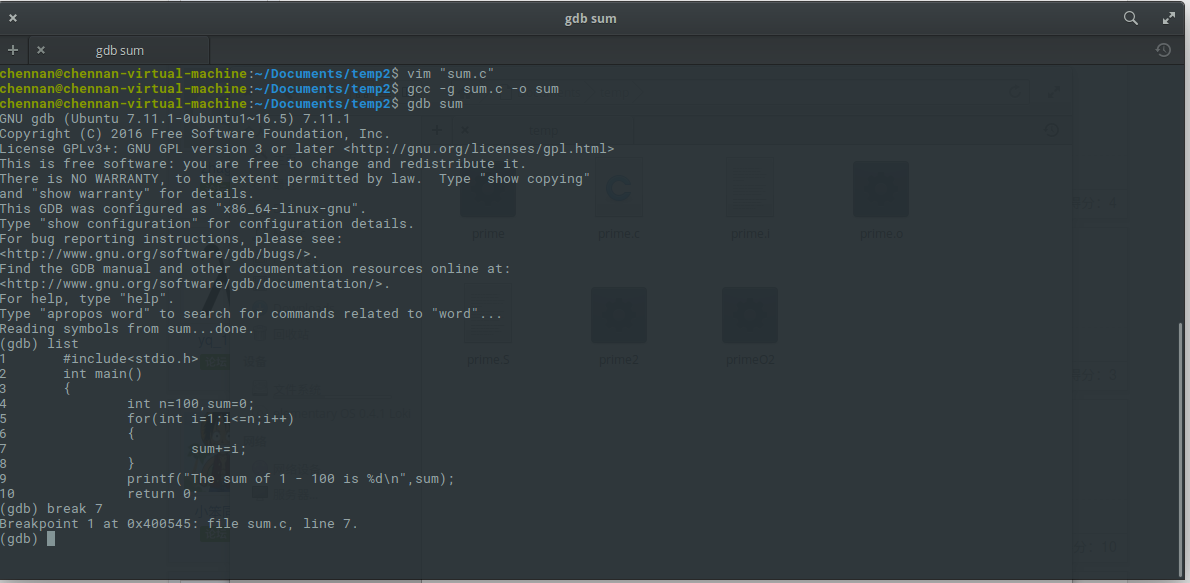
进入gdb



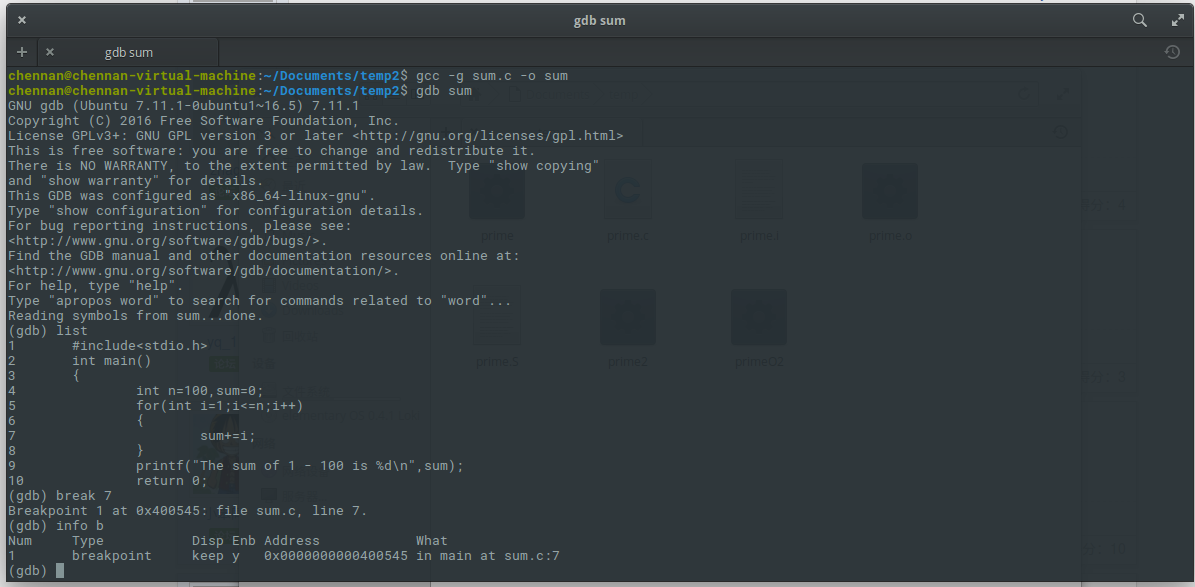
显示程序



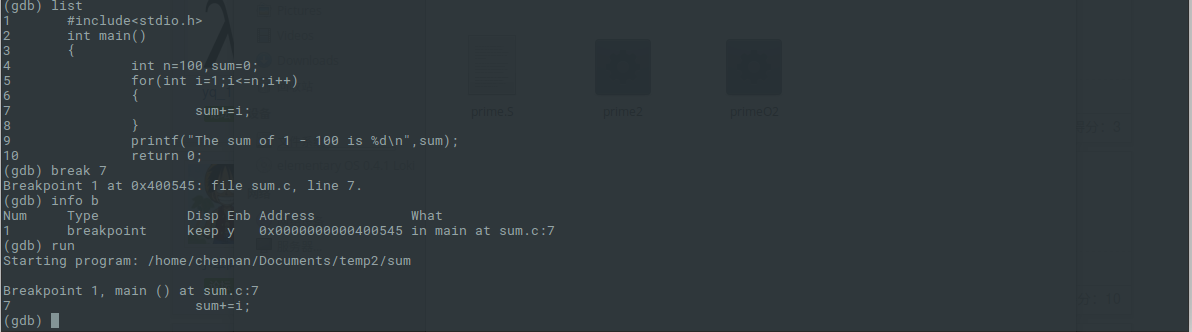
设置断点



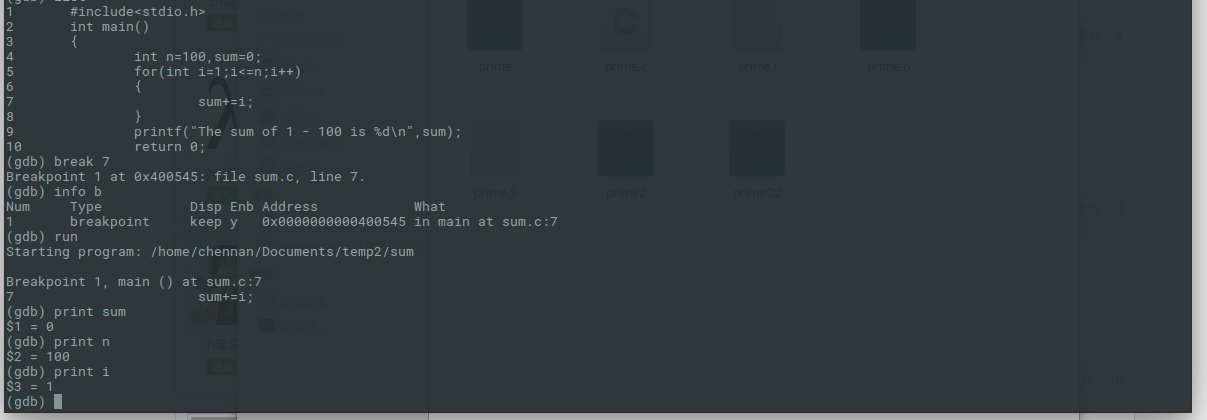
显示断点信息

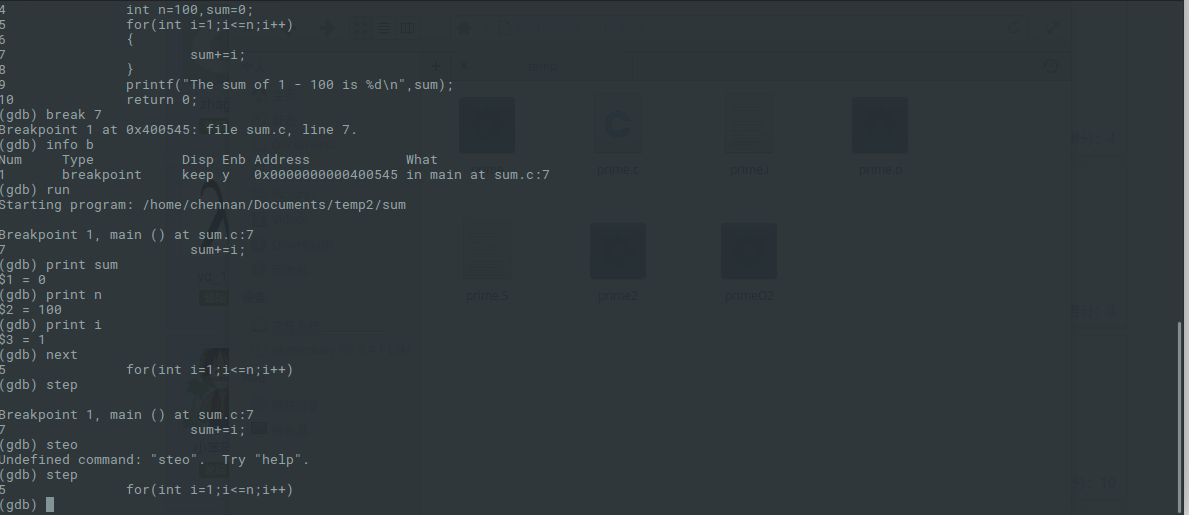


执行程序

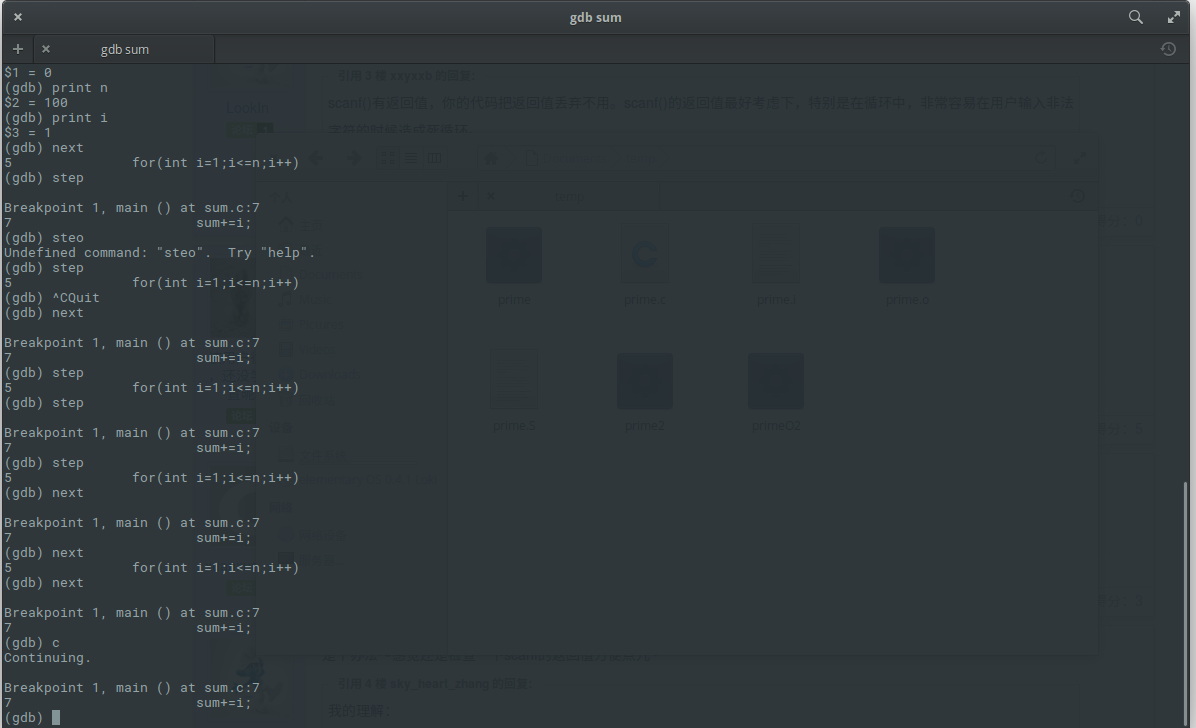


打印变量值

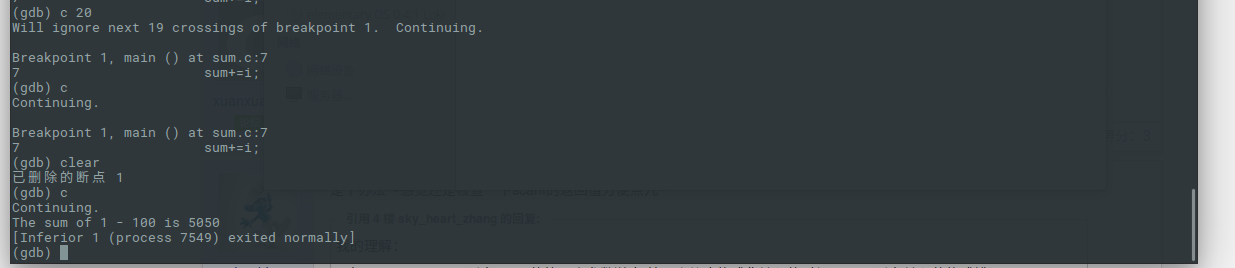




按步执行

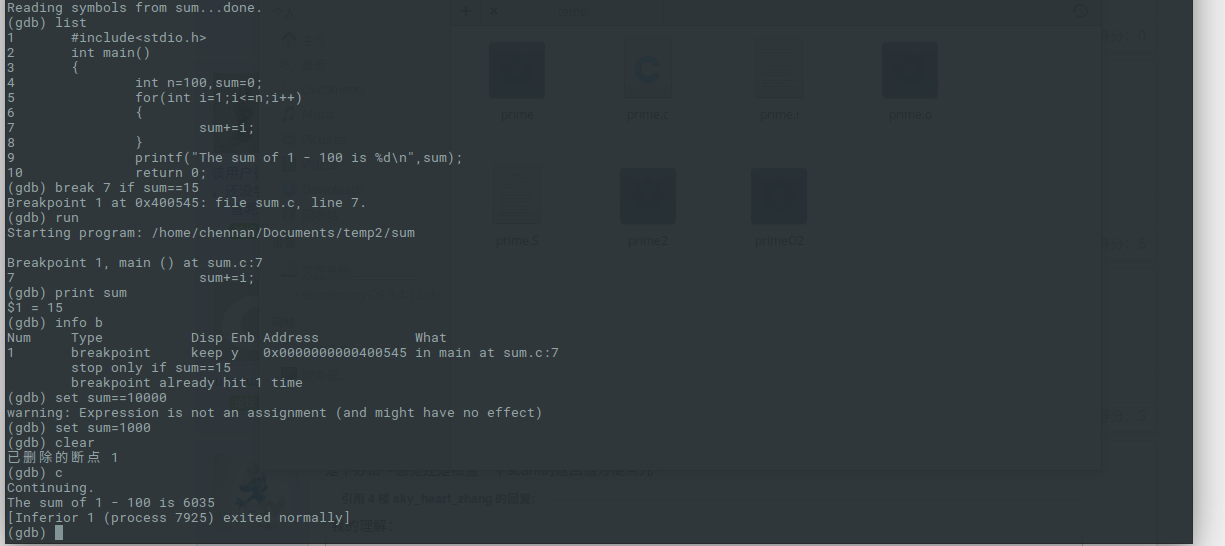


删除断点，运行到程序结束



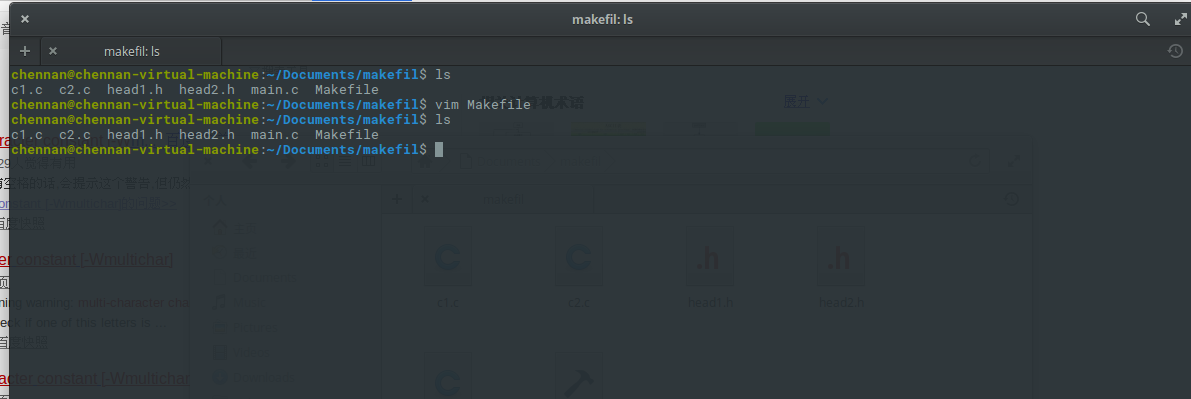
设置断点中变量到具体值时停止。

并给变量重新赋值。

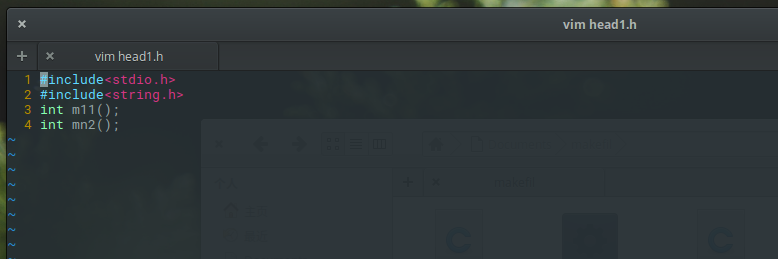


1. 实现一应用程序，该程序至少包含有两个c文件构成，使用makefile来完成对该程序的编译功能。

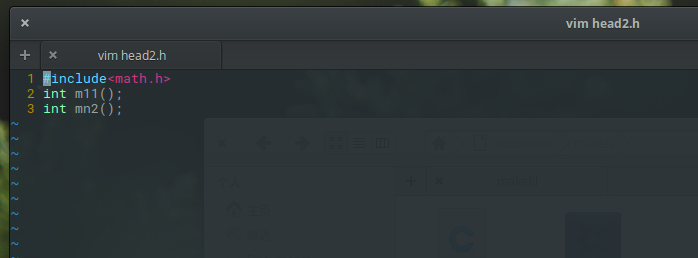
建立head1.h head2.h c1.c c2.c main.c 文件。



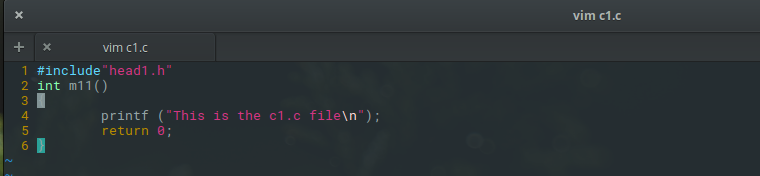
head1.h文件



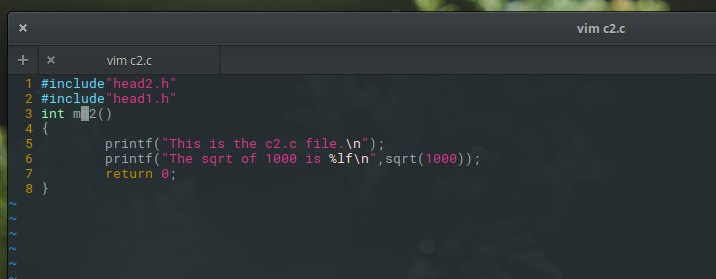
head2.h文件



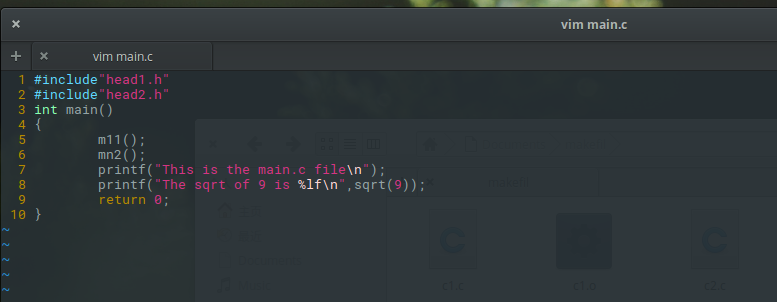
c1.c文件



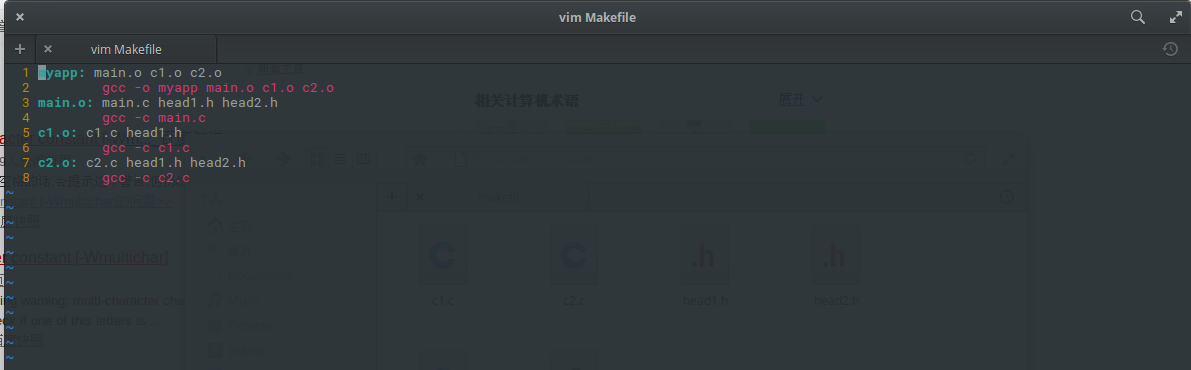
c2.c文件



main.c文件

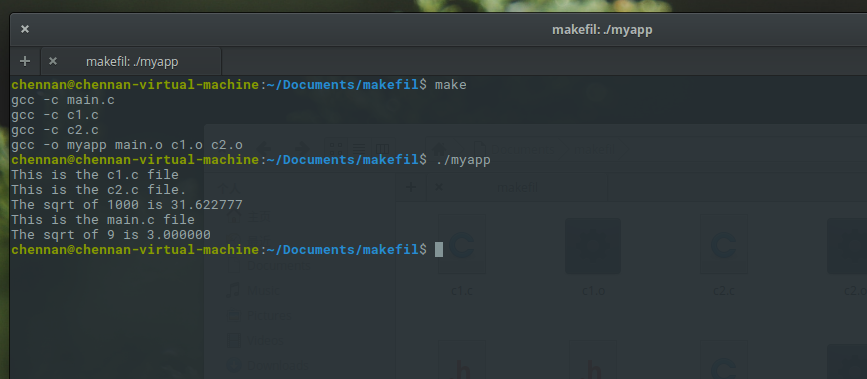


依赖关系如下makefile文件所示：



执行makefile文件，编译myapp程序。

执行myapp程序，运行正常，输出正确结果。



四．结论分析：

通过本次Linux下C编程实验，我对Linux下的C编程有了更加深入地理解。在此次实验中，我首先使用了vi编辑器，后来换成了功能更加强大的vim编辑器，通过在其中编写程序，我对vi/vim编辑器的操作有了一定的了解。写完程序后由自己而不是IDE来编译代码，更令我对C程序的编译过程有了深入的理解，编译中不同的命令会产生不同的结果，而且他们之间还有严格的先后顺序，这要求我们必须要真正理解C程序的编译原理，才能不出错的完成这一步。写代码出bug是再正常不过的一件事，通过GDB强大的调试功能，我们可以逐步，仔细的查找程序中的问题，改正错误。越大的工程代码量越大，当文件变多后，手动编译程序变得越来越麻烦，这时候善于利用makefile，可以节约很多工作。而且编写简单的makefile文件，也让我对makefile的原理与使用方法有了更深入的理解。