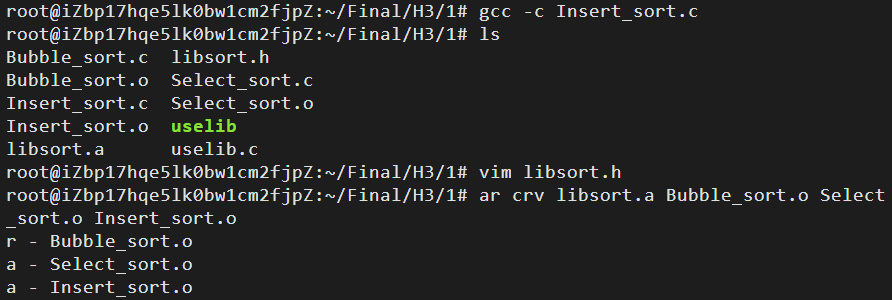
Linux程序设计H3

1191200426万嘉涌

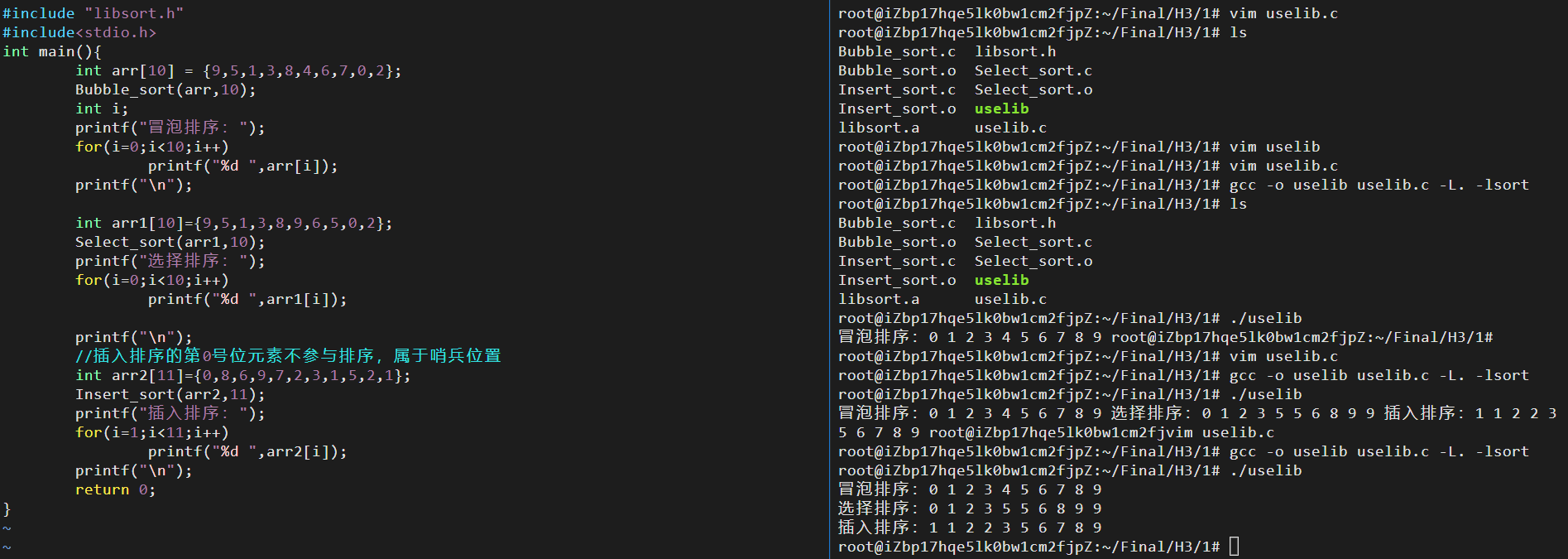
1. 编写程序对一个长为10的数组进行排序，分别使用冒泡排序法、选择排序和插入排序法排序，三种排序方案分别编译成为动态库和静态库文件，在主函数进行调用，编写实验报告，提交源代码。

静态库：

分别写好三个排序方法的c文件（Bubble\_sort.c和Insert\_sort.c和Select\_sort.c）并编译得到.o文件，将三者创建库文件libsort.a

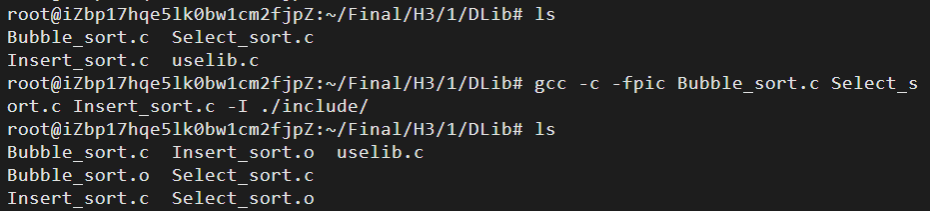


编写uselib.c来测试三个排序方法，编译时应用库文件libsort.a，因此需加上参数 -L. -lsort,结果显示如下

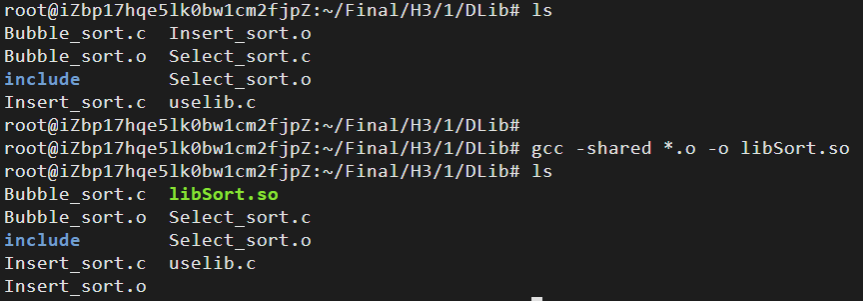


动态库：

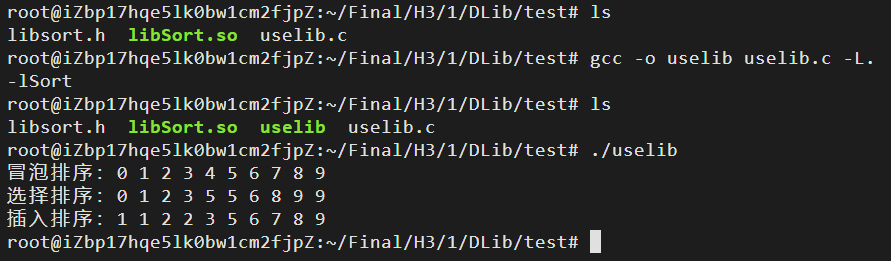
将三个排序c文件和测试文件uselib.c移到新的文件夹Dlib中，并在编译排序文件时添加-fpic



生成动态库libSort.so

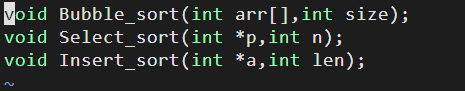


将测试动态链接库所需要的文件移到新建目录test下，编译uselib.c获得可执行文件，执行成功！

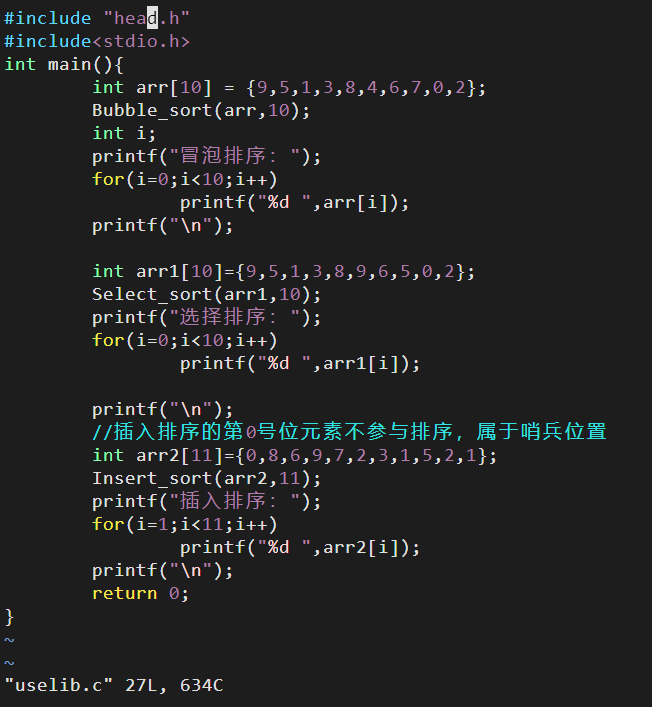


代码如下:

libsort.h

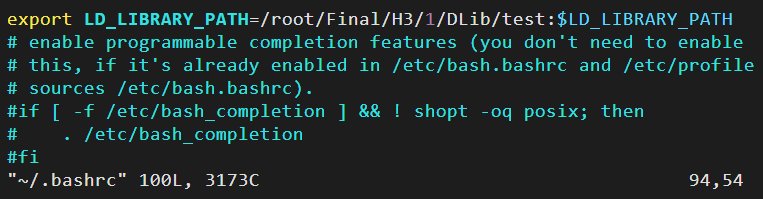


uselib.c



将测试动态链接库所需要的文件移到新建目录test下，并在~/.bashrc中添加环境变量export LD\_LIBRARY\_PATH=……$LD\_LIBRARY\_PATH

，编译uselib.c获得可执行文件，执行成功！



1. 编写程序对每种矩阵的加法和乘法以及行列式进行计算，使用随机赋值方式，程序计算写成单独的函数文件和头文件，由主函数调用。要求编写makefile文件对程序进行编译。提交所有源码，实验报告和makefile文件。

显示当前目录下编写好的c文件和makefile



MatrixAdd.c(矩阵加法，用两个3\*3的矩阵进行测试)

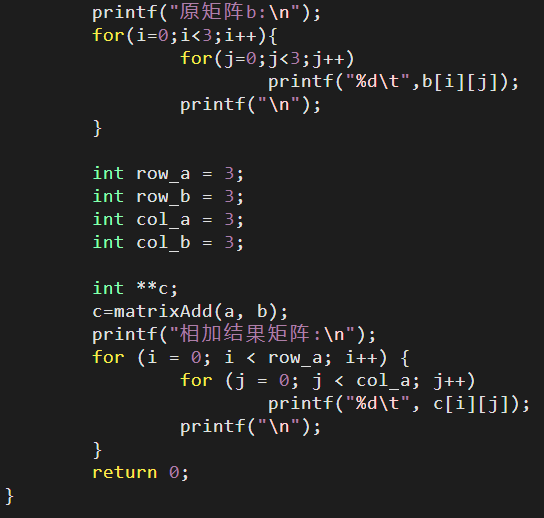


头文件matrix.h

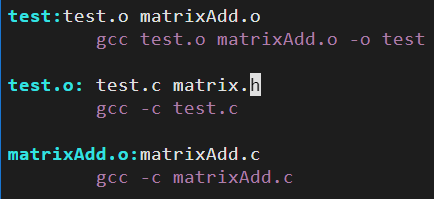


test.c文件内含主函数，用于测试

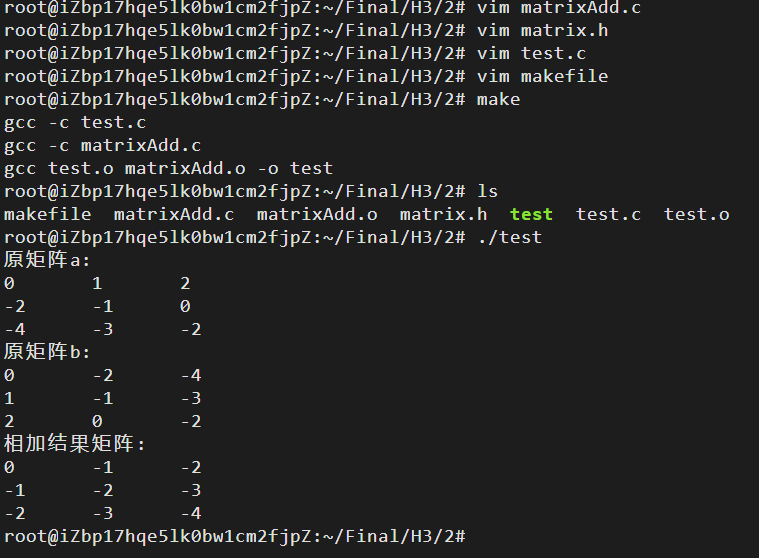




makefile文件如下

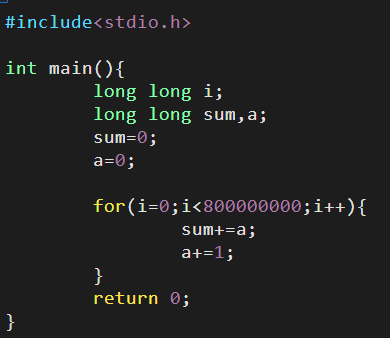


使用make直接调用gcc编译，实现矩阵加法

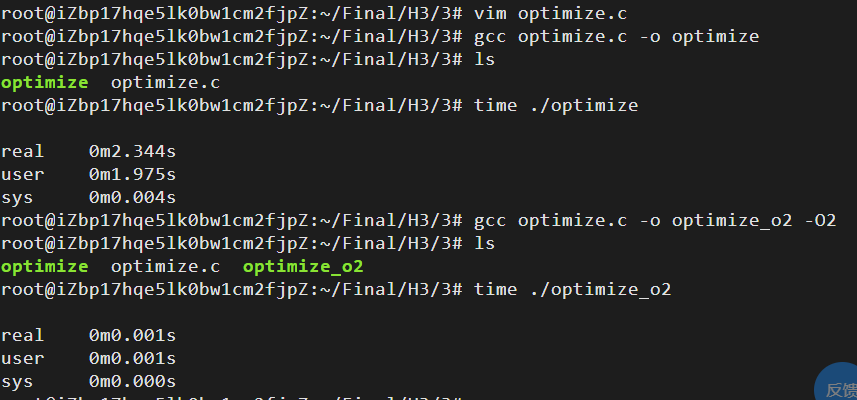


1. 练习代码中对程序进行优化的代码，将实验结果，写入实验报告。

编写计算程序optimize.c执行计算大量任务



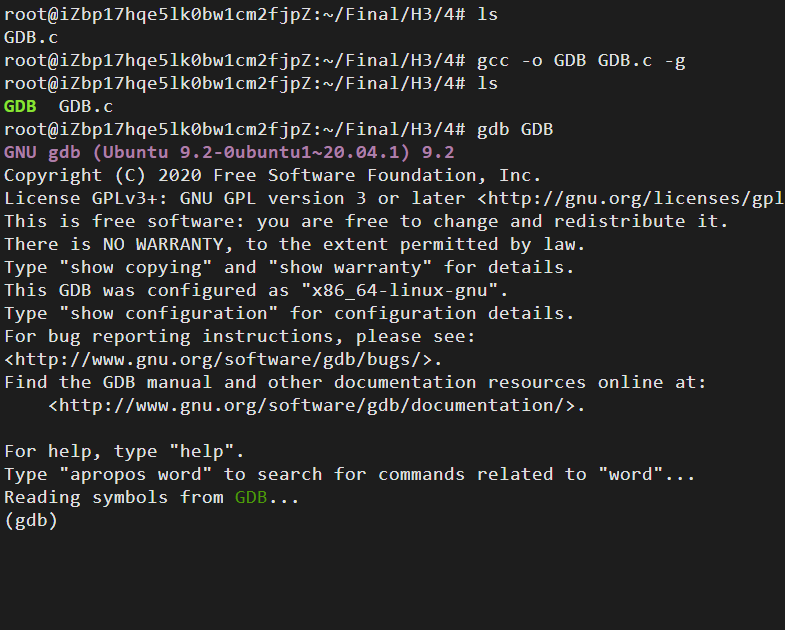
分别正常编译获得未优化可执行文件optimize和优化编译可执行文件optimize\_o2，比较优化前的运行时间和优化后的运行时间，我们可以发现优化后运行时间显著减少。



1. 设计一个程序，在屏幕上随机显示一个个位数，并开始等待用户的键盘输入字符，计算从字符显示到用户输入所用的时间，多次重复此过程，最后输出用户的平均反映时间（输入错误的不计入）以及正确率。

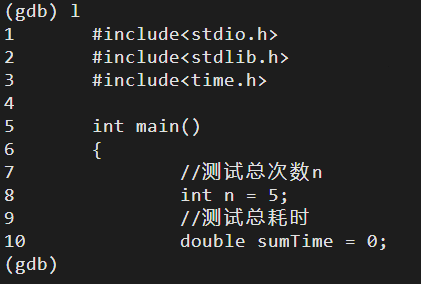
要求：使用gdb调试，将调试的几种命令实验结果及截图写入实验报告

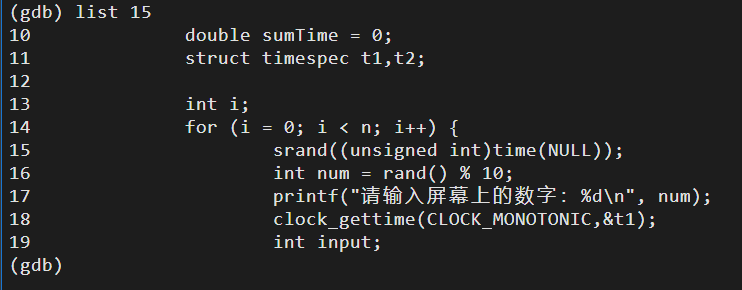
编译GDB.c获得可执行文件GDB,进入gdb调试



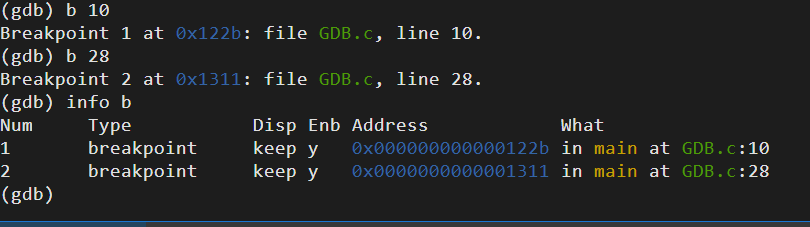
具体调试操作如下：

gdb的list命令

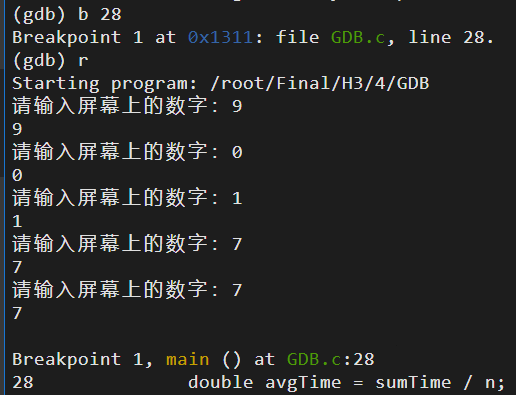




断点设置与查看断点信息



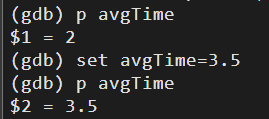
运行程序到断点处停止



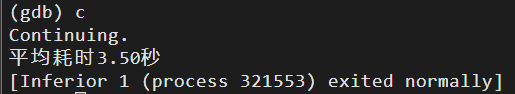
单步运行



查看变量与修改，这里将平均反应时间修改为3.5



继续运行直到结束



退出gdb

