实验2.1 Linux下C编程

一、 实验目的

1、练习并掌握Linux提供的vi编辑器来编译C程序。

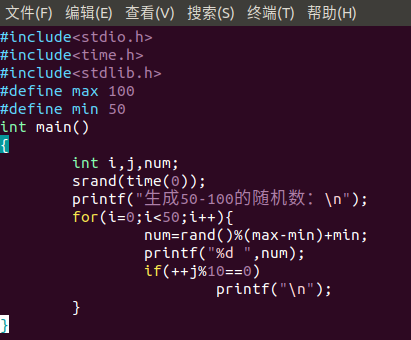
2、了解Linux C库函数的使用方法，掌握常用函数来解决实际问题。

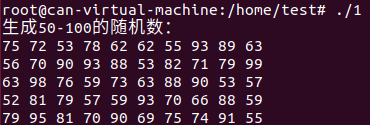
3、学会利用gcc、gdb编译、调试C程序。

4、了解静态链接、动态链接的过程及原理。

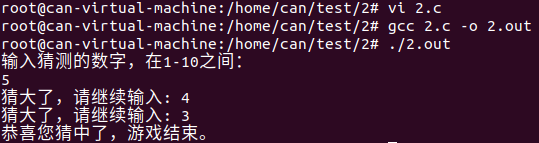
二、实验内容

1、 设计一个程序，生成50个100-1000之间的随机数。

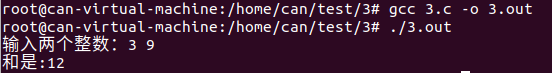




2、 设计一个猜数游戏的程序，先产生一个随机数，要求被试输入一个数，计

算机会提示猜大了，猜小了或恭喜您猜中了，直到猜中，退出程序。 

3、 设计一个程序，要求输入两个整数，求和输出。通过使用gcc的参数，控

制gcc的编译过程，了解gcc的编译过程，进一步认识gcc的灵活性。 

4、 设计一个程序，要求把输入的字符串原样输出，程序中的头文件自己定义，

源程序文件为“1-4.c”，自定义的头文件为“my.h”。

5、 设计一个程序，要求把输入的数字作为X轴坐标，算出它的sin值。

6、 编写程序main.c，实现格式为a+b=c格式的输出。

（1） 使用printf函数实现输出，如printf（“%d+%d=%d”，a，b，a+b）。

（2） 实现main命令行参数的获得，其中第一个参数是a，第二个参数是b；定

义add函数int add（int p1，int p2）。

（3） 实现add函数，并于printf中采用printf

（“%d+%d=%d”，a，b，add(a+b)）。

基于以上代码，分别进行全静态和全动态的编译。

（4） 执行objdump -x命令，查看包含NEEDED的字段，分析生成的两种执行

文件的依赖库的不同。

（5） 执行file和ld命令，查看两个执行文件的不同。

7、 分别编写add.h、add.c、main.c文件，其中add.h中包含加法函数的定义

int add（int p1，int p2），add.c文件中包含加法函数的实现。

（1） 对add.c进行编译，生成目标文件add.o。

（2） 执行ar命令，生成libadd.a。

（3） 编写main.c文件，通过include add.h头文件和gcc编译，实现对libadd.a

中加法函数的调用。

8、 分别编写add.h、add.c、main.c文件，其中add.h中包含加法函数的定义

int add（int p1，int p2），add.c文件中包含加法函数的实现，对add.c进

行编译生成共享库libadd.so。

编写main.c文件，通过include add.h头文件和gcc编译，实现对libadd.so

中加法函数的调用。

9、 针对题8的共享库，以至少三种方式实现共享库的加载运行，针对每种方

式编写一个run.sh来实现运行。

10、 基于题8的共享库重新编写add的实现，完成针对“按位与”操作，运行题

8生成的可执行文件main，使其加载修改成的libadd.so，显示“按位与”的

结果，如1&0=0，1&4=0。