实验四 GCC 和 GDB 的使用

小组成员： 李文浩 指导教师：周庆国

一、实验目的：

1. 掌握Linux操作系统下最常用的C语言编译器gcc的使用
2. 掌握Linux操作系统下最常用的代码调试器gdb的使用
3. 掌握调试代码的基本方法，如观察变量、设置断点等。

二、实验内容：

阅读在线帮助命令man gcc，以及man gdb的内容,了解gcc、gdb的基本使用。使用gdb观察变量、设置断点等。

三、实验步骤：

Helloworld.c程序(vim编译器编辑程序)：

#include <stdio.h>

int main(int argc, char\* argv[])

{

printf(" hello,this is our embedded world!\n ");

return 0;

}

编译： gcc helloworld.c –o helloworld

运行： ./helloworld

文件名：test.c

#include<stdio.h>

main(){

int i=0;

int h=19;

int m=0;

for(i=0;i<5;i++){

m=m+h+i;

}

}

gcc –g test.c //-g表示编译过程中生成可调式程序，默认生成a.out

gdb //运行gdb

(gdb) file a.out //打开a.out文件

(gdb) list //显示代码

(gdb) break 5 //设置断点在第5行

(gdb) run //运行程序

(gdb) print h //打印变量h值

(gdb) continue //继续执行

(gdb) quit //退出gdb

(gdb) info break //查看断点信息

(gdb) next（或n） //单条语句执行，next命令简写

(gdb) backtrace（或bt） //查看各级函数调用及参数

(gdb) step（s）//执行下一行语句，有函数调用则进入到函数中

(gdb) set var //修改变量的值

(gdb) finish //退出函数

四、收获总结。

虽然已经学过了，但是这次学习不仅加深了印象，所谓温故而知新，还学到了很多新的知识，深入了解了计算机内部实现的细节，对其原理有了更深刻的见解。