实验五 gdb和makefile文件

小组成员： 陆峰 王建强 李雪 闫英男 指导教师：周庆国

一、实验目的：

1. 深入了解Linux操作系统下最常用的调试工具GDB
2. 掌握Makefile文件的基本操作，可以进行简单的c语言文件编译

二、实验内容：

学习使用gdb单步调试程序;掌握Makefile文件的基本语法，并可以写出简单的makefile文件

三、实验步骤：

深化GDB：

1、单步执行

（1）next 不进入子函数

（2）step 进入子函数

2、打印值

（1）print + 变量名 打印变量值

（2）watch + 变量名 对制定变量进行监控

makefile文件

步骤一 掌握makefile文件规则：

简单的构造文件包含如下形式的规则：

TARGET ... : DEPENDENCIES ...

COMMAND

...

...

TARGET 一般是程序生成的文件名，如可执行文件或目标文件。它也可以是某个要执行的操作的名称，比如“clean”。目标

DEPENDENCIES 是生成目标所用的输入文件。一个目标可以依靠多个文件。依赖

COMMAND 是make要执行的操作。一条规则可以包含多个命令，一个命令占一行。注意：在每行命令前需要有一个tab字符。

步骤二 makefile的书写规则

规则举例

foo.o : foo.c defs.h

gcc -c -g foo.c

文件的依赖关系：foo.o依赖于foo.c和defs.h的文件，如果foo.c和defs.h的文件日期要比foo.o文件日期要新，或是foo.o不存在，那么依赖关系发生。

生成目标的方法：gcc命令，说明如何生成foo.o这个文件。

通配符的使用

foo.o : \*.c \*.h

步骤三 makefile实例：

1.hello.c

#include <stdio.h>

int main()

{

printf("Hello World!\n");

return 0;

}

2.Makefile

hello : hello.o    gcc -o hello hello.o

hello.o : hello.c   gcc -c hello.c

clean :     rm hello.o

四、收获总结。

通过本节课程，进一步深化了解了linux下的调试工具GDB，知道了step和next的区别。

掌握了makefile文件的书写规则，可以写出简单的makefile文件来编译c文件。