实验五 GDB深入学习、Makefile

小组成员： 陆烽 指导教师：周庆国

一、实验目的：

1.深入学习GDB编译环境

2.学习Makefile的基本功能和基本规则和工作过程

二、实验内容：

GDB举例、GDB深入、makefile介绍、示例、内容详解

三、实验步骤：

1. GCC回顾

2. GDB举例

* #include<stdio.h>
* main(){

int i=0;

int h=19;

int m=0;

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

m=m+h+i;

printf("m=%d\n",m);

}

}

3. GDB深入

1.断点

(1)Break + 函数名

(2)Break + 行号

(3)Break + 行号 +条件

(4)Info break 查看所有设置的断点

(5)Delete +断点编号 删除断点

2.单步执行

(1)Next 不进入子函数

(2)Step 进入子函数

3.打印值：

(1)Print +变量名 打印变量值

(2)Watch + 变量名 对指定变量进行监控

4. Makefile功能介绍

* **定义整个工程的编译规则—依赖关系**
* **自动化编译**

5. Makefile内容详解

* **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

Makefile中变量的使用:

objects = main.o kbd.o command.o display.o \

insert.o search.o files.o utils.o

edit : $(objects)

cc -o edit $(objects)

用包含“+=”的一行可以为变量附加文本，如：

objects = main.o kbd.o command.o display.o

objects += insert.o search.o files.o utils.o

等同于：

objects = main.o kbd.o command.o display.o

objects = $(objects) insert.o search.o files.o utils.o

如果变量此前没有定义，“+=”与普通的“=”相同。但在此前有定义的情况下，“+=”按照变量初始定义的类型扩展。

6. 自己编写Makefile

**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。

深入学习了GDB编译的环境，初步了解了makefile的作用以及具体的工作流程，手动尝试编写了一个makefile的文件。