实验 1 实验报告

余北辰 519030910245

练习 1 gcc

将V0-V3的值全部设置为3即可得到所要求的输出。

练习 2 GDB

在实验报告中回答以下问题:

---

1. How do you pass command line arguments to a program when using gdb?

有两种方法。第一种，在运行时直接在后面跟上命令行参数，如：

```
(gdb) r argv1 argv2 argv3
```

第二种，先用set args命令设置好参数的值：

```
(gdb) set args argv1 argv2 argv3
```

再正常运行即可。

---

2. How do you set a breakpoint which only occurs when a set of conditions is true (e.g. when certain variables are a certain value)?

在设断点时后边加上一条条件语句即可，如：

```
(gdb) b 12 if a==0
```

表示的是在第12行处设置一个a的值为0时触发的断点。

---

3. How do you execute the next line of C code in the program after stopping at a breakpoint?

使用next命令：

```
(gdb) n
```

---

4. If the next line of code is a function call, you'll execute the whole function call at once if you use your answer to #3. How do you tell GDB that you want to debug the code inside the function instead?

使用step命令：

```
(gdb) s
```

---

5. How do you resume the program after stopping at a breakpoint?

使用continue命令：

```
(gdb) c
```

---

6. How can you see the value of a variable (or even an expression like 1+2) in gdb?

使用print命令：

```
(gdb) p variable(or expression)
```

GDB就会给出所选择的变量或表达式的值。

---

7. How do you configure gdb so it prints the value of a variable after every step?

使用display命令:

```
(gdb) display variable
```

这样单步调试时，每一次步进时都会把所选参数的值显示出来。

---

8. How do you print a list of all variables and their values in the current function?

使用":"符号：

```
(gdb) p function::variable
```

GDB会给出在所选的函数中的变量的值。

---

9. How do you exit out of gdb?

使用quit命令：

```
(gdb) q
```

练习 3 调试

正确的函数实现如下：

```
int ll_equal(const node* a, const node* b){
while (a != NULL && b != NULL) {
    if (a->val != b->val)
        return 0;
    a = a->next;
    b = b->next;
}
if(a == NULL && b == NULL) return a == b;
else return 0;
}
```

练习 4 Make 初步

我的wc.c具体实现如下：

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
void wc(FILE *ofile, FILE *infile, char *inname)
{
    char str[999];
    int len;
    int i;
    int flag;
    int word_num, line_num, char_num;
    char ch;
    word_num = line_num = char_num = 0;
    flag = 1;
    infile = fopen(inname,"rb");
    if(infile)
    {
        while(fgets(str,999,infile))
        {
            len = strlen(str);
            for(i = 0;i < len;++i)
            {
                ch = str[i];
                if(ch == ' '||ch == '\t')
                {
                    char_num++;
                    if(!flag) word_num++;
                    flag = 1;
                }
                else if(ch != '\n' && ch != '\r')
                {
                    char_num++;
                    flag = 0;
                }
            }
            if(!flag) word_num++;
            flag = 1;
            char_num++;
            line_num++;
        }
    }
    fclose(infile);
    printf(" %d %d %d %s\n", line_num, word_num, char_num, inname);
}
int main (int argc, char *argv[]) {
    FILE* infile,* ofile;
    infile = NULL;
    ofile = NULL;
```

```
    wc(infile, ofile, argv[1]);
    return 0;
}
```