

实验报告

叶增渝 519030910168

练习 1 gcc:

只需将宏定义修改为:

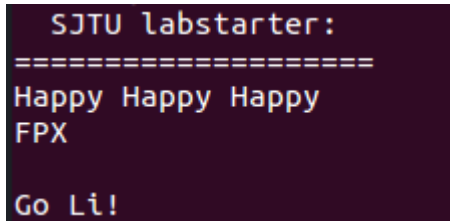
```
#define V0 3
```

```
#define V1 3
```

```
#define V2 3 (非 0 正数均可)
```

```
#define V3 3
```

如此修改即可获得如图输出



```
SJTU labstarter:
=====
Happy Happy Happy
FPX
Go Li!
```

练习 2 GDB

- 1.使用 run 命令行参数 1 命令行参数 2 …… 或者 set args 命令行参数 1 命令行参数 2 ……;
- 2.可以在 break 后加上相应条件, 如: break 28(行数) if 为真条件;
- 3.使用 next 进行单步执行, 执行下一行代码;
- 4.使用 step 进入函数后继续用 next 单步执行, 或者在想要检查的函数处设置断点;
- 5.使用 continue 来将函数继续执行至断点或程序结束;
- 6.使用 print 变量名 来单次查看某变量值, 或者使用 display 变量名 来持续显示变量值;
- 7.使用 display 变量名 在每次执行后持续显示变量值;
- 8.使用 info args 来查看当前函数中的所有局部变量及其值;
- 9.使用 quit 命令可以退出 GDB, 如果还有尚在执行的程序, 可以先使用 kill 命令异常结束程序。

练习 3 调试

错误原因: 在 while 判断时仅判断 a 是否为空, 导致第二次调用函数时 b 先变 NULL 然后在 b->val 这里产生错误。

解决办法: 将 while 的判断条件改为 a != NULL && b != NULL

函数实现:

```
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;
    }
    /* lists are equal if a and b are both null */
    return a == b;
}
```

练习 4 Make 初步

回答：\$./wc wc.c 执行的是 lab0 文件下 wc.c 所编译的可执行文件；\$ wc wc.c 执行的是 Ubuntu 中的 wc 命令，计算 wc.c 文件的字符数、行数、单词数。

修改后的 wc 函数：

```
void wc(FILE *ofile, FILE *infile, char *inname)
{
    // your code here:
    char ch;
    int character = 0, word = 0, line = 0;
    int isAWord = 0;
    char str[1024] = {};
    if(infile == NULL)
    {
        while(fgets(str, sizeof(str), stdin))
        {
            ++line;
            for(int i=0; str[i] != '\0'; ++i)
            {
                ++character;
                ch = str[i];
                if(ch == ' ' || ch == '\n')
                {
                    if(isAWord)
                    {
                        word++;
                        isAWord=0;
                    }
                }
                else isAWord = 1;
            }
            --character;
        }
    }
    else
    {
        while(fgets(str, sizeof(str), infile))
        {
            ++line;
            for(int i=0; str[i] != '\0'; ++i)
            {
                ++character;
                ch = str[i];
                if(ch == ' ' || ch == '\n')
                {
```

```

        if(isAWord)
        {
            word++;
            isAWord=0;
        }
    }
    else isAWord = 1;
}
--character;
}
fclose(infile);
}

if(ofile == NULL)
{
    printf("%s", "lines:");
    printf("%d", line);
    printf("%s", " words:");
    printf("%d", word);
    printf("%s", " characters:");
    printf("%d", character);
}
else{
    fprintf(ofile, "%s", "lines:");
    fprintf(ofile, "%d", line);
    fprintf(ofile, "%s", " words:");
    fprintf(ofile, "%d", word);
    fprintf(ofile, "%s", " characters:");
    fprintf(ofile, "%d", character);
    fclose(ofile);
}
}

```