

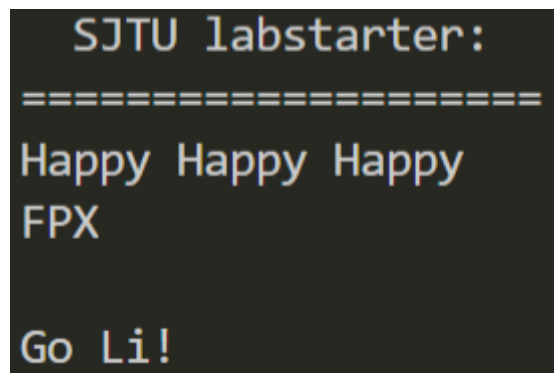
实践I 实验报告

陈文迪 519021910071

练习1 GCC

```
#define V0 3
#define V1 3
#define V2 1
#define V3 3
```

阅读源码之后，我们可以像上面的这样设置四个宏的值来得到合适的输出，如下图。



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

练习2 GDB

实验过程：

1. 执行 `break main`。
2. 执行 `run`。
3. 使用 `step` 或 `next` 单步调试。

问题回答：

1. 我们可以利用 `run` 命令来开始执行程序，同时也可以在其后面附加参数来将参数传入程序。例如 `run arg1 "arg2" ...`。
2. 我们可以利用 `break if` 语句来设置条件断点。语法规则为 `break line-or-function if expr`，在本例中我们可以设置 `break 12 if ch=='o'`。
3. 我们可以使用 `next` 和 `step` 命令来逐行调试。需要注意对于一个函数调用语句，`next` 会直接越过这条语句，而 `step` 则会进入该被调用的函数。
4. 我们可以用 `step` 命令。
5. 我们可以用 `continue` 命令。
6. 我们可以使用 `print` 命令。例如 `print ch` 或者 `print 1 + 2`。
7. 我们可以使用 `display` 命令。例如 `display i`。
8. 我们可以使用 `info locals` 命令。
9. 我们可以使用 `quit` 命令。

练习3 调试

在 `!!equal()` 内设置断点并单步调试，发现错误发生在对 `b` 空指针的访问，因此我们需要添加对 `b` 是否为空的判断。

```
11             if (a->val != b->val)
1: a = (const node *) 0x7fffffffed7d0
2: b = (const node *) 0x0
(gdb) next
```

Program terminated with signal SIGSEGV, Segmentation fault.
The program no longer exists.

修改后的代码如下：

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

练习4 Make 初步

输出不同的原因是执行 `wc wc.c` 时，调用的是Ubuntu自带的 `wc` 程序 `/usr/bin/wc`，而 `./wc wc.c` 执行的是实验目录中由 `wc.c` 编译产生的 `wc` 程序。

在编写 `wc` 程序时，需要注意行数、单词数统计的规则以及边界条件处理。我实现的代码如下。

```
void wc(FILE *ofile, FILE *infile, char *iname)
{
    char c;
    if (iname)
        c = fgetc(infile);
    else
        c = getchar();
    int line_count = 0, word_count = 0, char_count = 0, blank_buffer = 1;
    while (1)
    {
        if (c == '\n' || c == EOF)
        {
            if (!blank_buffer)
                ++word_count;
            if (c == EOF)
                break;
            ++char_count, ++line_count;
            blank_buffer = 1;
        }
        else
        {
            if (isblank(c) || c == '\r')
            {
                if (!blank_buffer)
                    ++word_count;
                blank_buffer = 1;
            }
        }
    }
}
```

```

        else
            blank_buffer = 0;
            ++char_count;
        }
        if (inname)
            c = fgetc(infile);
        else
            c = getchar();
    }
    if (inname)
    {
        if (ofile)
            fprintf(ofile, "%d %d %d %s\n", line_count, word_count, char_count,
inname);
        else
            printf("%d %d %d %s\n", line_count, word_count, char_count, inname);
    }
    else
    {
        printf("%d %d %d\n", line_count, word_count, char_count);
    }
}

```

经过测试，上述模块运行良好。

```

onlooker@MSI:~/Architecture/lab0$ ./wc wc.c
59 157 1475 wc.c
onlooker@MSI:~/Architecture/lab0$ wc wc.c
 59  157 1475 wc.c
onlooker@MSI:~/Architecture/lab0$ ./wc test.c
742 1280 23481 test.c
onlooker@MSI:~/Architecture/lab0$ wc test.c
742  1280 23481 test.c

```