**T4 Report**

由于两道任务题有非常多的共同之处，因此我没有把代码分开来写。

**功能说明**

从命令行读入两个文件名，如果文件名不存在或打开失败则输出错误信息：”Cannot open the file. “;之后读入第一个文件的内容并统计换行符，并将最后的行数信息输出到第二个文件中；程序内部实现了getline函数，设定的buffer区为30，返回值为读入的字符个数。如果一行内容超出了buffer区的限制，或者读取字符失败，即算作调用失败，返回-1并输出“Stack Overflow or Cannot Read Character.”。程序通过getline的返回值判断是否读到了新的一行，并由此统计行数，这样将两个题目的要求结合起来。

**运行方式**

在服务器命令行依次执行下列指令。

as -o h2.o h2.s -gstabs

ld -o func h2.o -g

./func input.txt output\_line\_num.txt

**源码分析**

**（1）\_start**：准备工作。

**（2）input\_open**：调用SYS\_open尝试打开第一个文件，打开失败则直接跳转fail\_open。

output\_open：调用SYS\_open尝试打开第二个文件，打开失败则跳转fail\_open。

read\_input：调用getline读入，如成功返回则在r9中让行数+1；判断getline返回值是否为0，不为0则继续读入，小于0证明出错跳转fail\_exit，等于0则证明之前已经读完，这一轮没有读到东西。

**（3）getline**：Read函数每次读入一个字符，首先判断返回值，小于0则跳转出错；等于0则已经读完文件需要按读到换行符处理，故直接跳转getline\_success；大于0说明读到东西，判断是否是换行符。如不是，则进入deal函数——判断r10中的文字计数是否超过了buffer大小，如果超过了则跳转错误，未超过则让r10中存储的文字数+1，继续read。如果之前判断到是换行符，则跳转getline\_success，返回r10中存储的字符数目。

**（4）stack\_init & answer\_push & solve\_last**: 将r9中的行数移入%rax并转换为字符，即除以10再加ascii码，然后让字符压栈。

**（5）write\_answer\_to\_output\_file**：写入文件。

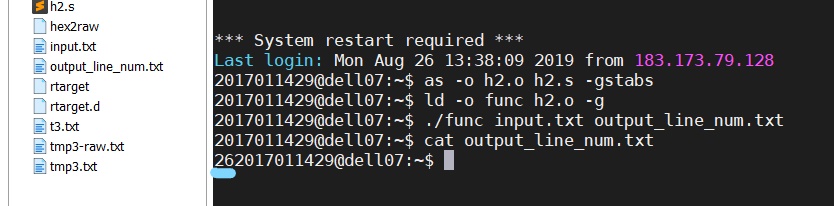
**（6）close\_file:** 关闭文件。

**（7）exit**: 退出程序。

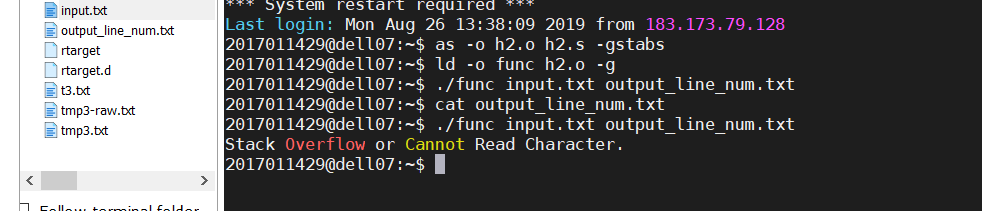
**（8）fail\_exit & fali\_open**: 将错误信息输出。

**实现功能截图**

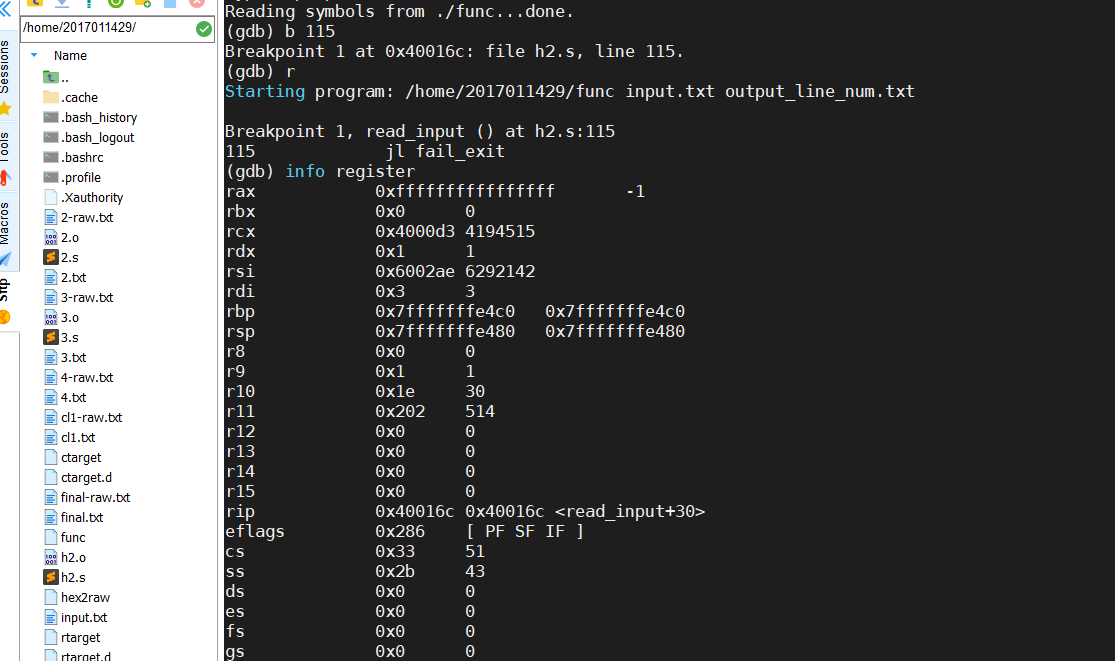
**（1）**input.txt为26个字母，每个一行；output\_line\_num.txt为空文件。输出行数。



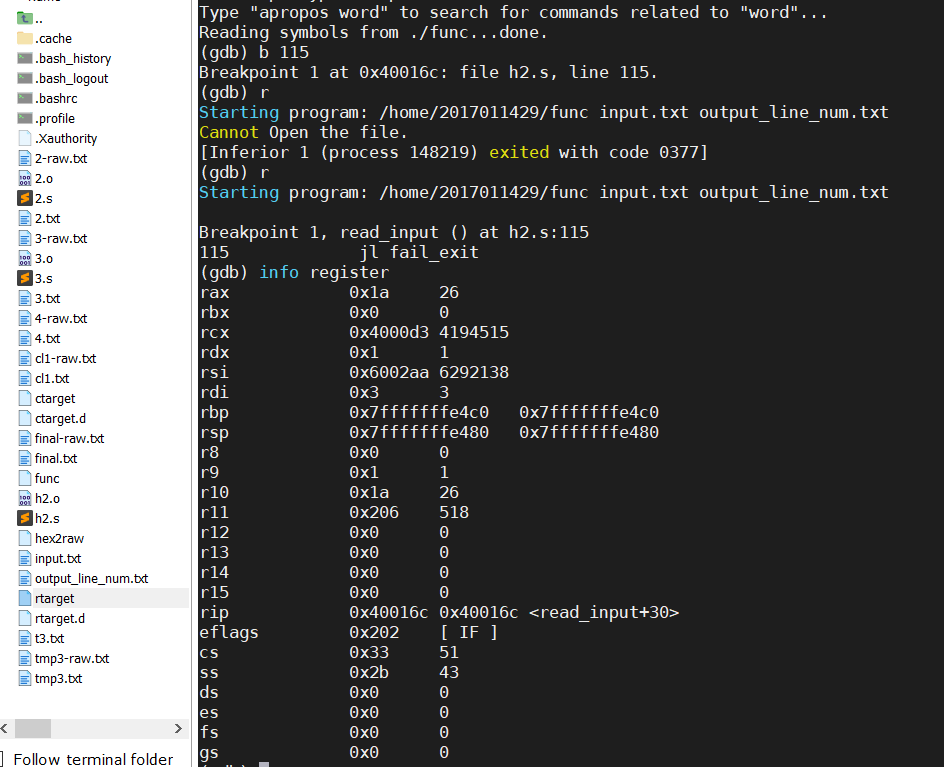
**（2）**input.txt的第一行为26个字母+10个数字，超过buffer大小，提示错误。



**（3）**读入（2）中txt，getline调用失败，返回值-1。



**（4）**将（2）中txt的第一行修改为26个字母，再次运行，getline返回字符数。



**（5）**文件不存在，删除output\_line\_num.txt，返回错误信息。

